

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



GobHash: Plataforma de análisis sobre las adquisiciones del
Estado de Guatemala.

Trabajo de graduación en la modalidad de Megaproyecto presentado por
Hugo Rafael Berganza Vásquez
para optar al grado académico de Licenciado en Ingeniería Industrial.

Javier Josué Fong Guzmán

Roberto Chiroy García

William Alberto Fuentes Ramírez

Juan Carlos Canteo Boix

Pablo Danilo Díaz Márquez

Gustavo Adolfo Morales Martínez

para optar al grado académico de Licenciados en Ingeniería en Ciencias
de la Computación y Tecnologías de la Información.

Guatemala,

2017

GobHash: Plataforma de análisis sobre las adquisiciones del
Estado de Guatemala.

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



GobHash: Plataforma de análisis sobre las adquisiciones del
Estado de Guatemala.

Trabajo de graduación en la modalidad de Megaproyecto presentado por
Hugo Rafael Berganza Vásquez
para optar al grado académico de Licenciado en Ingeniería Industrial.

Javier Josué Fong Guzmán

Roberto Chiroy García

William Alberto Fuentes Ramírez

Juan Carlos Canteo Boix

Pablo Danilo Díaz Márquez

Gustavo Adolfo Morales Martínez

para optar al grado académico de Licenciados en Ingeniería en Ciencias de
la Computación y Tecnologías de la Información.

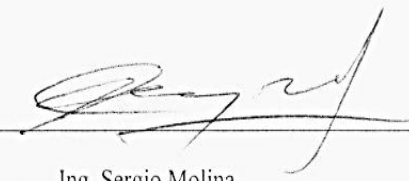
Guatemala,

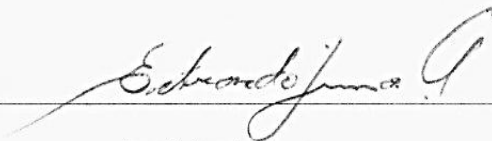
2017


Vo. Bo. :

(f) 
Ing. Sergio Molina

Tribunal Examinador:

(f) 
Ing. Sergio Molina

(f) 
Ing. Estuardo Sierra

(f) 
MSc. Douglas Barrios

Fecha de aprobación: Guatemala, 6 de diciembre de 2017

PREFACIO

El siguiente megaproyecto nació de un trabajo de graduación de la Maestría en Business Intelligence de la Universidad del Valle de Guatemala en la cual, la estudiante Alejandra Wyss, propuso la creación de una plataforma de análisis de los datos de las adquisiciones del Estado de Guatemala, actualmente mostrados en Guatecompras, que permitiera a la ciudadanía informarse de forma más descriptiva sobre la ejecución del presupuesto de la nación. Posteriormente, la propuesta de este proyecto fue llevada a nosotros como estudiantes de Ingeniería en Ciencias de la Computación y Tecnologías de la Información por Sergio Molina, nuestro coordinador del proyecto.

De esta manera se formó un equipo interdisciplinario conformado por estudiantes de la Universidad del Valle de Guatemala, de las carreras de Ingeniería Industrial e Ingeniería en Ciencias de la Computación y Tecnologías de la Información, para llevar a cabo el proyecto.

Agradecemos a Stuardo Rodríguez por su contribución a la extracción de datos, agradecemos a los asesores del proyecto: Sergio Izquierdo, Melinton Navas, Héctor Hurtarte, Bidkar Pojoy, Lynette García, Samuel Chávez y Rodrigo Gil por su contribución y asesoramiento en temas específicos. Agradecemos a nuestros padres de familia por el apoyo a lo largo de la carrera.

ÍNDICE

PREFACIO	X
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABLAS.....	XVII
RESUMEN	XVIII
ABSTRACT.....	XVIII
I. INTRODUCCIÓN.....	1
II. OBJETIVOS	3
III. JUSTIFICACIÓN.....	5
IV. MARCO TEÓRICO	7
V. DISEÑO E IMPLEMENTACIÓN DE MODELO DE DATOS DE ANÁLISIS DESCRIPTIVO	41
VI. DISEÑO E IMPLEMENTACIÓN DE INDICADORES ESTADÍSTICOS REPRESENTADOS A TRAVÉS DE GRÁFICOS CON BASE EN LA INFORMACIÓN EXTRAÍDA DEL PORTAL DE GUATECOMPRAS	61
VII. EXTRACCIÓN DE DATOS DE ADJUDICACIONES, PROVEEDORES Y COMPRADORES DEL SISTEMA GUATECOMPRAS	83
VIII. DISEÑO Y DESARROLLO DE ESTUDIO DE USABILIDAD PARA LA INTERACCIÓN HUMANO-COMPUTADOR.....	103
IX. DISEÑO E IMPLEMENTACIÓN DEL BACKEND.....	149
X. ANÁLISIS, DISEÑO Y DESARROLLO DE FRONTEND.....	173
XI. CONCLUSIONES.....	215
XII. RECOMENDACIONES	219
XI. BIBLIOGRAFIA	223
XII. ANEXOS.....	225
XIII. GLOSARIO.....	247

ÍNDICE DE FIGURAS

Figura 1. Ciclo metodología CRISP-DM	10
Figura 2 Capas de una aplicación web	12
Figura 3 Interacción entre capas.....	13
Figura 4 Estructura de frameworks basados en el patrón MVC	14
Figura 5 Frameworks más usados durante el 2017.....	16
Figura 6 Velocidad de desarrollo de cada framework basado en Go	23
Figura 7 Flujo del protocolo HTTP	25
Figura 8 Ejemplo de un GET visto desde el navegador	26
Figura 9 respuesta de una solicitud enviada mediante el protocolo http	26
Figura 10. Diagrama entidad-relación plataforma GobHash.....	44
Figura 11. Primera versión modelo de datos plataforma GobHash	45
Figura 12. Versión final modelo de datos plataforma GobHash	45
Figura 13. Flujo de formateo de datos.....	53
Figura 14. Flujo de integración de datos	54
Figura 15. Modelo dimensional de adjudicaciones del estado de Guatemala	56
Figura 16 Modelo entidad relación de la tabla “udvcomgt_gobhash”	65
Figura 17 Primera vista del editor para crear una gráfica.....	66
Figura 18 Segunda vista del editor para crear una gráfica seleccionado entidades.	67
Figura 19 Segunda vista del editor para crear una gráfica, campos disponibles por entidad.....	67
Figura 20 Segunda vista del editor para crear una gráfica, seleccionar una columna por entidad.....	68
Figura 21 Segunda vista del editor para crear una gráfica, todos los campos completados	68
Figura 22 Vista para agregar filtros normales y filtros por fecha	69
Figura 23 Vista con algunos filtros por fecha agregados	69
Figura 24 “definición” de un gráfico en formato Json	70
Figura 25 Ejemplo del texto generado por Sqel.JS.....	73
Figura 26 Lista de datos que retorna el “endpoint” luego construir y ejecutar la consulta.....	74
Figura 27 Ejemplo de previsualización de un gráfico	75
Figura 28 Vista para crear una publicación	76
Figura 29. Detalle de la opción 5 para búsquedas	84
Figura 30. Resultados de la búsqueda con opción 5.....	85
Figura 31. Método de procesamiento de solicitudes HTML	87
Figura 32. Detalle de proveedor, persona individual.....	88
Figura 33. Detalle de adjudicación, información general.....	89
Figura 34. Detalle de proveedor, persona jurídica.....	90
Figura 35. Duración de extracción datos de 2016 por mes, PC vs Server	92
Figura 36. Adjudicaciones obtenidas por semana	93
Figura 37. Duración de extracción datos de 2016 por semana, PC vs Server	93
Figura 38. Duración de extracción de datos marzo 2016, PC vs Server.....	94
Figura 39. Duración de extracción de datos abril 2016, PC vs Server	94
Figura 40. Duración de extracción de datos septiembre 2016, PC vs Server	95
Figura 41. Comparación de algoritmos para extracción de datos, ejecutados en Server	95
Figura 42. Comparación de algoritmos para extracción de datos, ejecutados en Server	95
Figura 43. Consumo de recursos en Server	97
Figura 44 Encabezado de la plataforma del FMI.....	107
Figura 45 Categorías de datos.	107
Figura 46 Vista de datos por país.	107

Figura 47 Editor de gráficas	108
Figura 48 Encabezado de plataforma de datos abiertos de EEUU	109
Figura 49 Categorías de sets de datos.....	109
Figura 50 Página de datos financieros.	109
Figura 514 Detalle de set de datos.....	110
Figura 525 Competencia de área financiera.	111
Figura 53 Encabezado de la plataforma de datos abiertos del Reino Unido.....	112
Figura 54 Categorización de datos.	112
Figura 558 Búsqueda de datos.....	112
Figura 56 Búsqueda de zona específica.....	113
Figura 57 Detalle de set de datos.....	113
Figura 58 Formatos de set de datos.	114
Figura 59 Información adicional de set de datos.	114
Figura 60 Vista de aplicaciones.....	115
Figura 61 Vista detallada de App.	115
Figura 62 Diagrama de flujo de aplicación.	118
Figura 63 Pantalla de inicio – Prototipo dibujado.	119
Figura 64 Pantalla principal – Prototipo dibujado.....	120
Figura 65 Pantalla mis hash – Prototipo dibujado.....	121
Figura 66 Pantalla de perfil de usuario – Prototipo dibujado.	122
Figura 67 Pantalla de registro de usuario – Prototipo dibujado.....	123
Figura 68 Vista de detalle de hash – Prototipo dibujado.	124
Figura 69 Logo de GobHash con la paleta de colores.	124
Figura 70 Pantalla de inicio – Prototipo diseñado.	125
Figura 71 Pantalla principal – Prototipo diseñado.....	126
Figura 72 Pantalla de mis Hash – Prototipo diseñado.	127
Figura 73 Pantalla de perfil de usuario – Prototipo diseñado.	128
Figura 74 Pantalla de registro de usuario – Prototipo diseñado.	129
Figura 75 Pantalla de vista detallada de hash – Prototipo diseñado.	130
Figura 76 Página de inicio – Prototipo interactivo	131
Figura 77 Página principal – Prototipo interactivo.....	131
Figura 78 Pantalla de registro de usuario – Prototipo interactivo.....	132
Figura 79 Pantalla de perfil de usuario – Prototipo interactivo.	132
Figura 80 Pantalla de creación de hash – Prototipo interactivo.....	133
Figura 81 Pantalla de detalle de hash – Prototipo interactivo.....	133
Figura 82 Esquema de pruebas de usabilidad.....	135
Figura 83 Relación de éxito tarea 1.	137
Figura 84 Tiempo promedio de finalización tarea 1.	137
Figura 85 Promedio medidas subjetivas tarea 1.	138
Figura 86 Relación de éxito tarea 2.	138
Figura 87 Tiempo promedio de finalización tarea 2.	138
Figura 88 Promedio medidas Subjetivas tarea 2.....	139
Figura 89 Relación de éxito en tarea 3.	139
Figura 90 Tiempo promedio de finalización tarea 3.	140
Figura 91 Promedio medidas subjetivas tarea 3.....	140
Figura 92 Relación de éxito en tarea 4	140
Figura 93 Tiempo promedio de finalización tarea 4.....	141
. Figura 94 Promedio de medidas subjetivas tarea 4.	141
Figura 95 Relación de éxito en tarea 5.	141

Figura 96 Tiempo promedio de finalización tarea 5	142
Figura 97 Promedio medidas subjetivas tarea 5	142
Figura 98 Relación de éxito todas las tareas.....	143
Figura 99 Promedio de medidas subjetivas todas las tareas.	143
Figura 100 Relación de error total.	144
Figura 101 Relación de error crítico.....	144
Figura 102 Relación de error no crítico.....	145
Figura 103. Documentación principal del API.....	157
Figura 104. Documentación secundaria del API.....	157
Figura 105. Inicio de sesión a la documentación del API.....	158
Figura 106. Creación de usuario para acceder a la documentación del API.....	158
Figura 107. Organización GobHash con los repositorios del proyecto.....	159
Figura 108. Compra del dominio en godaddy.com	159
Figura 109. Servicio de envío de correos Mailgun.....	160
Figura 110. Base de datos MongoDB almacenada en mLab.....	160
Figura 111. Panel de ambientes del API en Heroku.....	161
Figura 112. Servicio de pruebas automatizadas Travis CI.....	161
Figura 113. Resumen de pruebas unitarias ejecutadas.....	162
Figura 114. Estado base de la aplicación en reposo en Keymetrics	162
Figura 115. Resultado prueba de estrés feed.....	163
Figura 116. Resultado prueba de estrés feed	163
Figura 117. Resumen resultado de prueba de estrés feed.....	164
Figura 118. Resultado prueba de estrés endpoint gráficos	164
Figura 119. Resumen prueba de estrés endpoint gráficos.....	165
Figura 120. Prueba de estrés del protocolo de tiempo real con Artillery	165
Figura 121. Resultado de prueba de estrés del protocolo de tiempo real con Artillery.....	166
Figura 122. Diagrama de flujo: Inicio de sesión.....	175
Figura 123. Diagrama de flujo: Registro de usuario.....	176
Figura 124. Diagrama de flujo: Perfil del usuario	177
Figura 125. Diagrama de flujo: Solicitar cambio de contraseña.....	178
Figura 126. Diagrama de flujo: Recuperación de contraseña por medio de token	179
Figura 127. Diagrama de flujo: Crear una publicación “Post”	180
Figura 128. Diagrama de flujo: Feed Inicial	181
Figura 129. Diagrama de flujo: Publicaciones en tiempo real.....	182
Figura 130. Diagrama de estados sobre la aplicación de GobHash.....	187
Figura 131. Modelo de datos definido para Frontend.....	187
Figura 132. Plantilla con estructura base utilizando Metronic	190
Figura 133. Plantilla con estructura base para un post utilizando Metronic	191
Figura 134. Configuración inicial del proyecto de Frontend.....	193
Figura 135. Configuración de <i>ExpressJS</i> y <i>NodeJs</i>	193
Figura 136. Dependencias y versiones del proyecto de Frontend	194
Figura 137. <i>Filesystem</i> del proyecto GobHash utilizado.....	197
Figura 138. Repositorio para control de versiones de prototipo den Github	198
Figura 139. Repositorio para control de versiones de GobHash en Github.....	198
Figura 140- Servidor automático para producción de GobHash	199
Figura 141. Servidor automático para desarrollo de GobHash.....	199
Figura 142. Prototipo: Página inicial	199
Figura 143. Prototipo: Inicio de sesión.....	200
Figura 144. Prototipo: Registro de usuario.....	200

Figura 145. Prototipo: Solicitar cambio de contraseña.....	200
Figura 146. Prototipo: Página inicial luego de haber iniciado sesión.....	201
Figura 147. Prototipo: Perfil del usuario.....	201
Figura 148. Prototipo: Ver posts y agregar comentarios.....	202
Figura 149. GobHash: Página inicial.....	202
Figura 150. GobHash: Inicio de sesión.....	203
Figura 151. GobHash: Registro de usuario.....	203
Figura 152. GobHash: Solicitud de cambio de contraseña.....	204
Figura 153. GobHash: Recuperación de contraseña mediante token.....	204
Figura 154. GobHash: Feed inicial.....	205
Figura 155. GobHash: Perfil del usuario.....	205
Figura 156. GobHash: Cambio de foto de perfil.....	206
Figura 157. GobHash: Cambio de contraseña mediante plataforma.....	206
Figura 158. GobHash: Estadísticas generales del usuario.....	206
Figura 159. GobHash: Vista de las publicaciones creadas por el usuario actual.....	207
Figura 160. GobHash: Vista individual de una publicación.....	208
Figura 161. GobHash: Comentarios sobre una publicación.....	208
Figura 162. GobHash: Agregar una nueva publicación.....	209
Figura 163. Integración con el módulo desarrollo e implementación de indicadores estadísticos.....	210
Figura 164. Integración de gráficos por el módulo desarrollo e implementación de indicadores.....	210
Figura 165. Página principal www.guatecompras.gt	225
Figura 166. Sección de búsqueda de adjudicaciones de www.guatecompras.gt	226
Figura 167. Selección de adjudicación por NOG.....	226
Figura 168. Despliegue de datos del concurso con NOG 6912710 en www.guatecompras.com	227
Figura 169. Visualización de tablas de tipos de anexos, tipo de Producto, proveedor adjudicado.....	228
Figura 170. Visualización de tabla: Historial de Acciones.....	228
Figura 171. Despliegue de datos de la unidad compradora Hospital Dr. Juan José Arévalo.....	229
Figura 172. Despliegue de datos de la entidad Instituto Guatemalteco de Seguridad.....	229
Figura 173. Visualización de la identificación en la página de registro de proveedores.....	230
Figura 174. Visualización de la identificación en la página registro de proveedores.....	231
Figura 175. Estructura JSON schemas.json.....	236
Figura 176. Estructura JSON files.json.....	237
Figura 177. Página inicial de búsquedas en Guatecompras.....	239
Figura 178. Solicitud de parámetros de búsqueda por parte de Guatecompras.....	239
Figura 179. Ejemplo de parámetros de búsqueda.....	240
Figura 180. Resultado de la búsqueda con los parámetros de la Figura 179.....	240
Figura 181. Información general de una adjudicación hecha en Guatecompras.....	241
Figura 182. Información general de una adjudicación.....	241
Figura 183. Criterio de calificación de una adjudicación.....	242
Figura 184. Tipos de producto de una adjudicación.....	242
Figura 185. Detalle de un producto.....	242
Figura 186. Integrantes de la junta de calificación de una adjudicación.....	242
Figura 187. Listado de oferentes de una adjudicación.....	243
Figura 188. Proveedores adjudicados de una adjudicación.....	243
Figura 189. Historial de acciones de una adjudicación.....	243
Figura 190. Página con los detalles de la entidad compradora.....	244
Figura 191. Página con los detalles de la unidad compradora.....	244
Figura 192. Información de proveedor (empresa).....	245
Figura 193. Información de proveedor (persona individual).....	246

ÍNDICE DE TABLAS

Tabla 1 Características de un framework	15
Tabla 2 Objetivo de cada tipo de gráfico.....	27
Tabla 3. Primera propuesta de campos a extraer	51
Tabla 4. Campos extraídos por el módulo de extracción de datos	51
Tabla 5. Resumen reporte de validación inicial	57
Tabla 6. Validación final modelo de datos	57
Tabla 7 Comparación entre frameworks para desarrollo de backend.....	77
Tabla 8 Evaluación de frameworks según indicadores.....	79
Tabla 9 Comparación entre librerías para el manejo de gráficas en AngularJS	80
Tabla 10. Cantidad de adjudicaciones obtenidas por mes en 2016.....	91
Tabla 11. Duración promedio de obtención por adjudicación.....	92
Tabla 12 Comparación de plataformas evaluadas en análisis de competencia.....	117
Tabla 13 Validación de prototipos de diseño.	134
Tabla 14 Métricas en tarea 1.	137
Tabla 15 Métricas en tarea 2.	138
Tabla 16 Métricas en tarea 3.	139
Tabla 17 Métricas en tarea 4.	140
Tabla 18 Métricas en tarea 5	141
Tabla 19 Métricas en todas las tareas.	142
Tabla 20 Relación de error total en pruebas.	143
Tabla 21 Relación de error critico en pruebas.	144
Tabla 22 Relación de error no critico en pruebas	144
Tabla 23. Listado de endpoints disponibles en el API.....	155
Tabla 24. Primera comparación sobre tecnologías para desarrollo de Frontend	184
Tabla 25. Segunda comparación sobre tecnologías para desarrollo de Frontend	185
Tabla 26. Comparación sobre plantillas para desarrollo de vistas.....	189
Tabla 27. Ajustes para validación de datos	232
Tabla 28. Validación por modalidad de compra de enero 2016	233
Tabla 29. Validación por modalidad de compra de febrero 2016.....	233
Tabla 30. Validación por modalidad de compra de marzo 2016.....	233
Tabla 31. Validación por modalidad de compra de abril 2016.....	233
Tabla 32. Validación por modalidad de compra de mayo 2016	234
Tabla 33. Validación por modalidad de compra de junio 2016.....	234
Tabla 34. Validación por modalidad de compra de julio 2016.....	234
Tabla 35. Validación por modalidad de compra de agosto 2016.....	234
Tabla 36. Validación por modalidad de compra de septiembre 2016.....	235
Tabla 37. Validación por modalidad de compra de octubre 2016	235
Tabla 38. Validación por modalidad de compra de noviembre 2016.....	235
Tabla 39. Validación por modalidad de compra de diciembre 2016.....	235

RESUMEN

Gobhash es una plataforma de tipo red social para realizar análisis descriptivo sobre las adquisiciones del Estado Guatemala, que permite a los usuarios sin necesidad de conocimientos o habilidades específicas en computación y estadística, crear, compartir, modificar y discutir indicadores descriptivos creados a partir de datos extraídos del portal de Guatecompras.

Para poder crear la plataforma se crearon siete módulos, de los cuales seis estaban a cargo de estudiantes de ingeniería en computación y el restante a cargo de un estudiante de ingeniería industrial. Dado que se realizó un trabajo grupal se utilizó la metodología *scrum* para poder darle seguimiento al proyecto en cuanto a control de tiempos de entrega y secuencia de entrega de módulos con base en la dependencia entre cada uno de ellos.

La secuencia de los módulos fue la siguiente, se desarrolló un extractor de datos del portal de Guatecompras que luego lo exporta a un archivo de texto. La información extraída pasa a una base de datos, en donde es esquematizada de forma coherente. Luego se generó un protocolo para consultar la información de la base de datos y paralelo a esto un generador de gráficos descriptivos. El siguiente módulo se encargó de generar una interfaz intuitiva y finalmente se desarrolló la plataforma de tipo red social.

Como resultado se obtuvo una plataforma intuitiva, capaz de generar análisis descriptivo sobre información de Guatecompras y soportar una discusión sobre los hallazgos encontrados. Cabe mencionar que la plataforma se publicó en el internet bajo el dominio www.gobhash.com.

ABSTRACT

This module developed the data model and evaluation procedures of the data of the GobHash platform. As result of this project, two data models were obtained. The first, a model based on documents that describes the structure and functionalities of the GobHash platform. Second, a dimensional data model which stores relevant information about the Guatemalan state purchases, based on information extracted from Guatecompras. For future data loads, it is advice to use the data preparation tools produced in this module.

I. INTRODUCCIÓN

GobHash es una plataforma que permite a las personas llevar a cabo distintos análisis con los datos que se presentan en el portal de Guatecompras, todo en un ambiente de red social que es parte esencial de la plataforma. Para lograr esto se obtuvo la información presentada en Guatecompras y se almacenó en archivos separados por comas. Que después de pasar por un proceso de transformación fueron cargados en una base de datos que es consultada por la herramienta encargada de crear distintos tipos de gráficas. Además se realizó un estudio de usabilidad para validar la interacción humano-computador.

Para la extracción de los datos de Guatecompras se utilizó un enfoque iterativo que permitió asegurar la extracción de los datos de manera íntegra. Una metodología similar se aplicó en el módulo encargado del estudio de usabilidad. También se usó este acercamiento por parte del módulo encargado del diseño de la base de datos. Otro enfoque fue el utilizado por el módulo que se encargó del diseño de la página web se basó en un acercamiento directo para la resolución del objetivo. En el caso del módulo encargado del backend se aplicó una forma de trabajo similar al anterior. Los resultados que se obtuvieron del módulo de extracción de datos fueron satisfactorios en cuanto a la confiabilidad de los datos. Queda por describir algunos de los hallazgos más importantes hechos durante la elaboración del presente trabajo. La obtención de datos de manera diaria es el enfoque que presenta la menor carga para el servidor como para el equipo donde se ejecute el programa. El modelo de datos se adapta a las necesidades de la plataforma de manera adecuada. A nivel de usabilidad, el sitio web tiene un mejor índice de aprobación, que sitios similares, entre los usuarios.

II. OBJETIVOS

A. Objetivo general

Desarrollar una plataforma de análisis descriptivo sobre las adquisiciones del estado Guatemala, construida como una red social, que permite a los usuarios, sin necesidad de conocimientos o habilidades específicas crear, compartir, modificar y discutir indicadores descriptivos creados a partir de conjuntos de datos extraídos del portal de Guatecompras.

B. Objetivos específicos

1. Diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico
 - a. Diseñar e implementar un modelo de datos que describa la información de adquisiciones del estado.
 - b. Crear un protocolo para la validación de los datos extraídos de Guatecompras por el módulo de extracción de datos.
 - c. Diseñar un modelo de datos que soporte las características del modelo público de la red social GobHash
2. Desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de Guatecompras
 - a. Crear una interfaz dentro de la plataforma GobHash que les permita a los usuarios generar indicadores estadísticos, a través de gráficas, de forma sencilla e intuitiva.
 - b. Permitir al usuario poder generar por lo menos tres tipos diferentes de gráficos.
 - c. Integrar completamente el módulo de “Diseño, Análisis y Desarrollo de Frontend” con el módulo de desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de Guatecompras.
3. Extracción de datos de adjudicaciones, proveedores y compradores del sistema Guatecompras, para su utilización en la plataforma GobHash.
 - a. Obtener los datos correspondientes a las adjudicaciones hechas durante el año 2016 en Guatecompras.gt
 - b. Desarrollar una herramienta que sea capaz de obtener las adjudicaciones de un año en específico en la plataforma Guatecompras.gt, de manera automatizada.
4. Diseño y desarrollo de estudio de usabilidad para la interacción humano-computador.
 - a. Desarrollo y análisis de un estudio de Estándar y usabilidad.
 - b. Desarrollo y análisis de prototipos en papel.

- c. Desarrollo y análisis de prototipos diseñados con herramienta de manejo de imágenes.
- d. Desarrollo y análisis de prototipos interactivos diseñados con herramienta de fabricación de prototipos.
- e. Diseño, desarrollo y análisis de pruebas de Usabilidad.
- f. Cumplir con los requisitos de mejora definidos en Estudio de estándar y usabilidad, basado en resultados de pruebas de usabilidad.

III. JUSTIFICACIÓN

Guatemala cuenta con la Ley de Acceso a la Información Pública según el decreto 57-2008. Decreto en donde se define que se debe garantizar a toda persona interesada, sin discriminación alguna, el derecho a solicitar y a tener acceso a la información pública en posesión de las autoridades y sujetos obligados. Sin embargo, dicha ley no se cumple en cabalidad debido a las limitaciones de las herramientas que el estado ha puesto a disposición de la sociedad para acceder a esta información. Entre estas herramientas se encuentra el portal Guatecompras, medio por el cual las entidades públicas o privadas que administran, manejan o ejecutan fondos del estado pueden adquirir y contratar bienes y servicios. En él se encuentra información de las solicitudes y adquisiciones que ha realizado el estado de Guatemala desde el año 2009.

Por dicha situación nace la necesidad de una plataforma, sistema o vía por la cual nosotros como guatemaltecos podamos estar conscientes de lo que está pasando. El presente megaproyecto propone una plataforma dedicada para auditar el manejo de las finanzas del gobierno, poder compartir los análisis hechos a todas las personas dentro de la plataforma, poder comentar y participar en otras publicaciones, y de esa manera poder actuar a fin de mejorar nuestro país.

GobHash será una plataforma que mejore la manera en la que la información de adquisiciones del estado se entregue al ciudadano. Con GobHash se planea cumplir a cabalidad con las normas actuales de acceso a la información, y, además, fomentar el acceso a la misma, con el objetivo de generar discusión. De esta manera el ciudadano guatemalteco, independientemente del sector o formación académica podrá comprender la ejecución de las adquisiciones del estado y de esta manera, mejorar su criterio respecto de temas de importancia nacional.

IV. MARCO TEÓRICO

A. Bases de datos

Una base de datos es una colección de datos. Un manejador de base datos (DBMS por sus siglas en inglés) es un software diseñado para crear, utilizar y dar mantenimiento a grandes cantidades de información (Ramakrishnan, 2002).

Algunas de las ventajas de utilizar un DBMS:

1. Independencia de la información.

Los DBMS proveen una abstracción de la información que la aísla, de manera que las aplicaciones que la utilizan no deben preocuparse de la manera en que esta está almacenada.

2. Acceso eficiente.

Los DBMS utilizan técnicas sofisticadas para el eficiente almacenamiento y recuperación de los datos.

3. Integridad y seguridad de la información.

Si la información siempre se accede por medio del DBMS, este puede asegurarse que los cambios y transacciones que se hagan cumplan con cierta lógica que refuerce la integridad de la información, sin mencionar que el DBMS es capaz de regular el acceso que diferentes tipos de usuarios tienen sobre la misma información.

4. Acceso concurrente y recuperación de fallas.

El DBMS organiza los accesos concurrentes a la información de manera que para los usuarios pareciera que esta está siendo accedida solamente por un usuario. El software también protege a los usuarios de los efectos de fallas del sistema.

5. Reducción en el tiempo de desarrollo de aplicaciones.

El DBMS se encarga de algunas de las funciones más comunes e importantes que las aplicaciones que utilizan la información necesitan, facilitando el ágil desarrollo de estas.

B. Modelo Relacional

Propuesto por Edgar Codd en 1970, el modelo Relacional revolucionó la manera en la que se manejaban las bases de datos hasta ese momento. Este modelo se basa en que una base de datos es una colección de una o más Relaciones, donde cada relación es una tabla con columnas y filas. El mayor avance de este modelo comparado con los modelos usados anteriormente es la simplicidad con la que los datos son representados y la facilidad con la que consultas complejas pueden ser expresadas. (Gehani, 2006).

A los DBMS que utilizan el modelo Relacional se les conoce como RDBMS por sus siglas en ingles. (Gehani, 2006) Estos utilizan alguna versión del lenguaje SQL (Structured Querying Language) para hacer consultas de la información.

C. Modelo NoSQL

El modelo NoSQL se basa en estructuras diferentes a tablas, filas y columnas para tener modelos de datos más flexibles. NoSQL puede significar que no se utiliza el lenguaje SQL para consultar la información o que no solamente se utiliza el lenguaje SQL para hacer consultas. Este modelo es particularmente útil para almacenar información no estructurada. Algunos tipos de DBMS basados en este modelo incluye al modelo Llave-valor, almacenamiento basado en documentos, bases de datos columnares y bases de datos basadas en grafos. (Gupta, 2015)

D. Bases de datos orientadas a documentos

Subconjunto de las bases de datos NoSQL. Un manejador basado en documentos es usado para almacenar, recuperar y administrar información semiestructurada. A diferencia de las tablas del modelo Relacional la información en este tipo de bases de datos no se rige por un solo esquema, lo que provee una mayor flexibilidad para el modelado de datos. (Gupta, 2015)

El término documento se puede referir a un archivo Microsoft Word o un PDF, pero lo más común es que este se represente utilizando los formatos XML o JSON. En lugar de columnas con nombres y tipos de datos, cada documento contiene una descripción del tipo de dato que almacena y un valor para esa descripción. Cada documento puede tener estructura diferente.

E. Modelación de datos

Proceso por el cual se define la estructura y el significado de la información de un sistema. De este proceso se obtiene un modelo abstracto que organiza la información. Este estandariza y describe las maneras en las que el sistema se Relaciona con el mundo real. (Ross & Kimball, 2011)

F. Modelo de datos dimensional

El modelo dimensional, o esquema estrella, proporciona un método para simplificar y facilitar la comprensión de la base de datos. Una base de datos dimensional se puede concebir como un cubo de tres

o cuatro dimensiones en el que los usuarios pueden acceder a una porción de la información a lo largo de sus dimensiones. (Ross & Kimball, 2011)

Este modelo generalmente consta de una tabla central, a la que le llama tabla de hechos, alrededor de la cual se muestran un conjunto de otras tablas a las que les denominan tablas de dimensiones. La tabla central es la única tabla del esquema con varias uniones hacia las dimensiones. Las tablas de dimensiones solamente tienen una conexión hacia la tabla de hechos.

G. CSV

Comma-separated values o CSV son un tipo de documento de formato abierto sencillo que se utiliza para representar datos en forma de tabla, en las que los valores de las diferentes columnas se separan por comas (o un delimitador predeterminado) y las filas se separan por el salto de línea. (Shafranovich, 2017)

H. Medio social

Plataformas de comunicación en línea donde el contenido es creado por los propios usuarios mediante el uso de tecnologías web o móviles, que facilitan la edición, publicación y el intercambio de información. Estos se basan en la influencia de individuos y la interacción entre pares. Los medios sociales más utilizados son las redes sociales, los blogs, microblogs, y las comunidades dedicadas a la publicación de contenido multimedia.

I. Minería de datos

Campo de la estadística y ciencias de la computación referido al proceso de analizar grandes volúmenes de datos. El objetivo principal de la minería de datos es extraer información de un conjunto de datos y transformarla en una estructura que pueda ser utilizada más adelante. (Nisbet, Elder, & Miner, 2009)

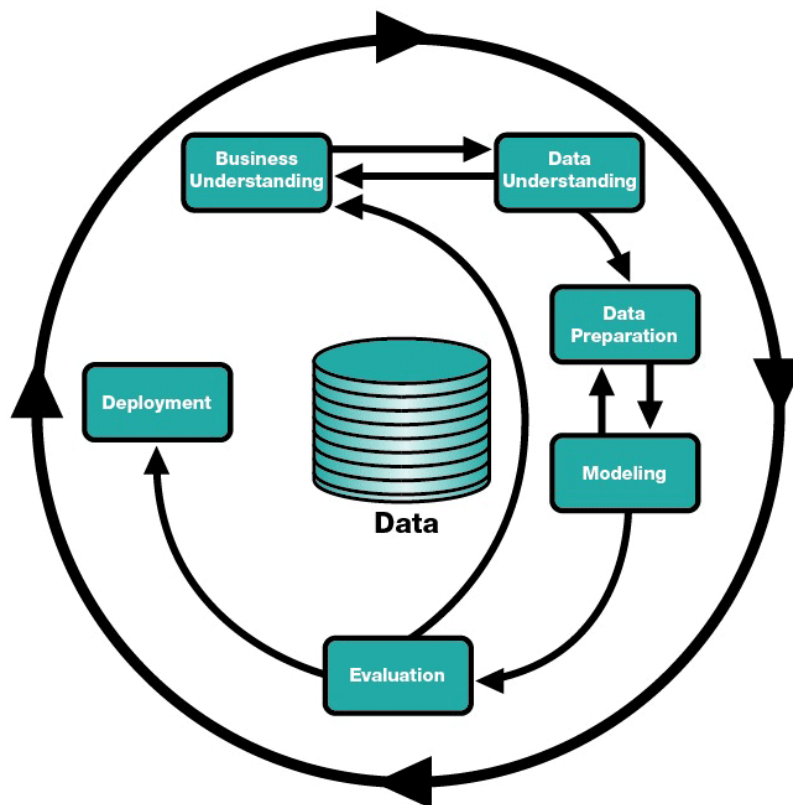
J. Metodología CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Practice for Data Mining) está basada en estándares internacionales con el objetivo de describir, documentar y mejorar procesos de minería de datos. (Smart Vision Europe Ltd., 2015). Este marco de trabajo consta de seis fases:

1. Comprensión del negocio. Esta fase se centra en la comprensión de los objetivos y requisitos del proyecto.

2. Comprensión de los datos. Esta fase comienza con una carga inicial de datos, con el fin de familiarizarse con los datos y determinar la calidad inicial de los mismos.
3. Preparación de los datos. Se lleva a cabo un proceso de construcción de los datos. Se realizan tareas de selección, transformación y limpieza de los datos.
4. Modelación de los datos. Se selecciona una o varias técnicas de modelación de datos y se calibran parámetros para obtener resultados óptimos, según los requerimientos del proyecto.
5. Evaluación de los datos. Se evalúa la precisión, integridad y la consistencia de los datos, entre otras cualidades para medir el grado de calidad de la información.
6. Desarrollo. En esta fase se llevan a cabo los procesos necesarios sobre el modelo final para alcanzar los objetivos del proyecto.

Figura 1. Ciclo metodología CRISP-DM



Fuente: (Smart Vision Europe Ltd., 2015)

K. . Distancia de Levenshtein

La distancia de Levenshtein es el número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Se entiende por operación las acciones de inserción, eliminación o sustitución de un carácter. (Müller, 2009)

L. Aplicación Web

Una aplicación web o Web App es un programa informático que los usuarios pueden utilizar accediendo a un Servidor Web a través de internet mediante un navegador, esto implica que no es una aplicación convencional que se instala en un ordenador y es ejecutada por un sistema operativo, el acceso es independiente al mismo. Estas aplicaciones por lo general generan dinámicamente una serie de páginas, las cuales están soportadas por un navegador web común mediante HTML. (Equipo Vértice, 2009)

Las aplicaciones web tienen varias ventajas en comparación a las aplicaciones de escritorio y entre ellas están las siguientes: en primer lugar se tiene lo práctico del navegador web como un cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y dar mantenimiento a las aplicaciones web sin caer en la obligación de instalar software en nuestro equipo, además la portabilidad de la aplicación es un factor a resaltar ya que se puede ejecutar en cualquier equipo media vez posean un navegador web, incluyendo teléfonos y tablets. (Equipo Vértice, 2009)

Este tipo de aplicaciones tiene una estructura definida que se divide en tres capas, las cuales son:

- Capa de navegador:

Esta capa está formada únicamente por el cliente que en este caso es el navegador. Mediante el uso de lenguajes o arquitecturas como HTML o XML se sirven los datos en el formato adecuado para que el usuario los pueda interpretar, y sean adecuados a sus necesidades.

- Capa de servidor:

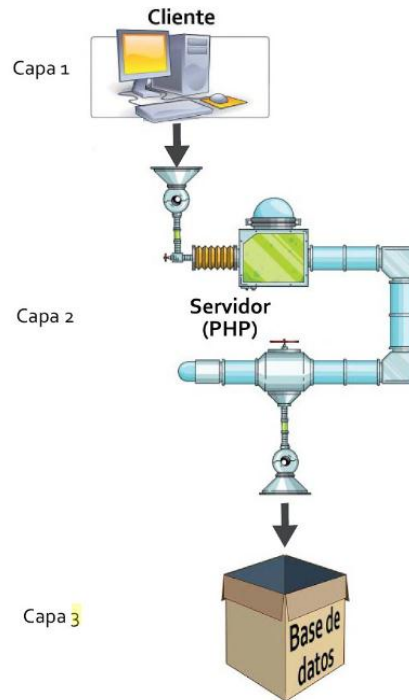
Esta capa también es conocida como capa media y es donde normalmente se encuentra situado el servidor sobre el cual se llevan a cabo la gran mayoría de los procesos, utilizando para ello alguna tecnología web dinámica como: PHP, ASP, JSP, etc.

- Capa de persistencia:

En esta última capa normalmente se encuentra el sistema de administración de la base de datos, es decir el manejador de la base en el cual se almacenan todos los datos que son necesarios en la capa de servidor para poder ser consultados, actualizados, eliminados o agregar cualquier tipo de información conforme sea necesario.

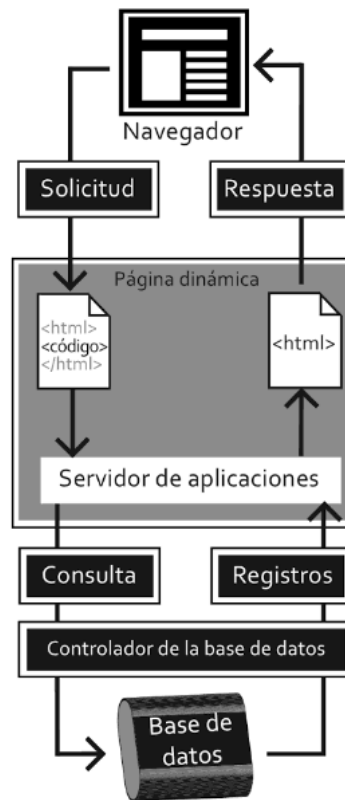
A continuación, se presenta un diagrama que ejemplifica el flujo de las tres capas descritas:

Figura 2 Capas de una aplicación web



La forma en la que interactúan las tres capas es la siguiente: el navegador web se encarga de realizar una petición a la capa intermedia mediante el uso de HTML y algún lenguaje del lado del cliente como JavaScript, la capa del servidor traduce dichas peticiones a consultas SQL que son transmitidas a la base de datos, la cual se encuentra en la capa de persistencia, la ejecuta y devuelve a la capa media generando una página HTML con dichos resultados. (Equipo Vértice, 2009) A continuación se presenta un diagrama que ejemplifica la interacción entre las capas.

Figura 3 Interacción entre capas



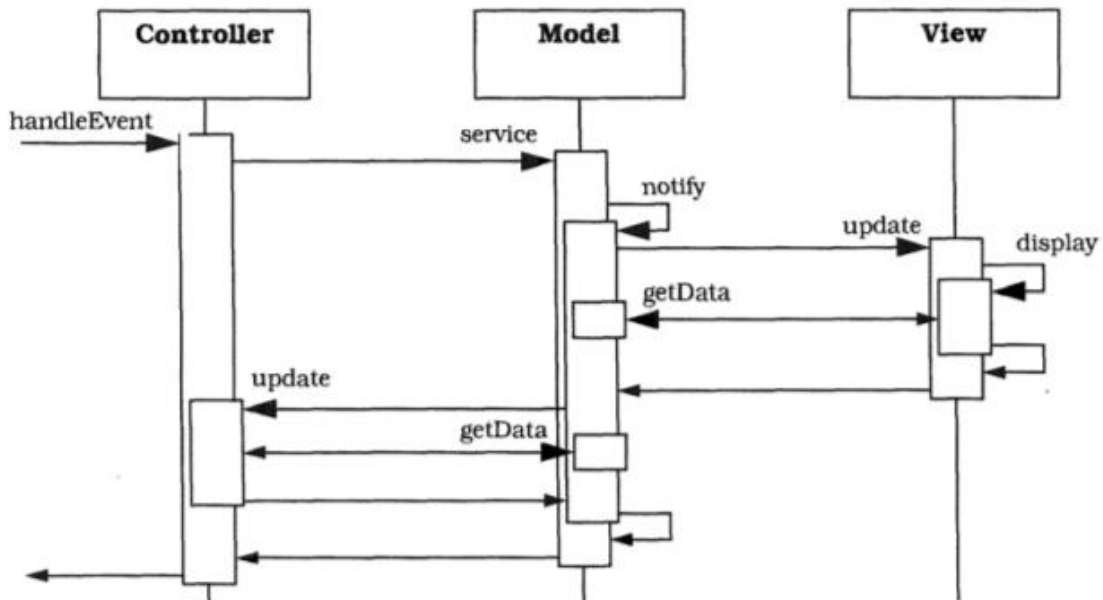
M. Framework

En general, el término *framework*, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Los objetivos por lo cuales se llega a considerar el uso de un *framework* como una herramienta muy útil, es que ayuda a acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un *framework* Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (Javier J. Gutiérrez, 2002)

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Para comprender cómo trabajan los *frameworks* Web existentes es imprescindible conocer el patrón MVC. (Javier J. Gutiérrez, 2002)

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (Javier J. Gutiérrez, 2002).

Figura 4 Estructura de frameworks basados en el patrón MVC



Existen varios tipos de *frameworks* Web, orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como *Tapestry*. La mayoría de *frameworks* Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de *Servlets* y *JSP*, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación. (Javier J. Gutiérrez, 2002)

Existe un conjunto de características que describe por completo a la mayoría de los *frameworks* web la cuales son:

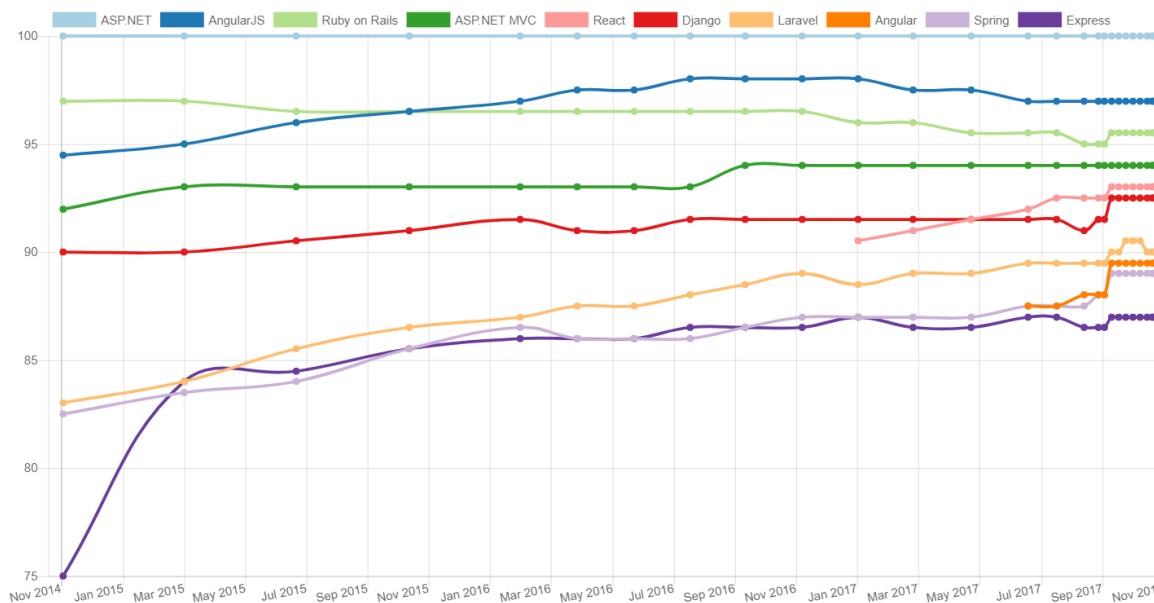
Tabla 1 Características de un framework

Abstracción de <i>URLs</i> y sesiones.	No es necesario manipular directamente las <i>URLs</i> ni las sesiones, el <i>framework</i> ya se encarga de hacerlo.
Acceso a datos.	Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.
Controladores	La mayoría de <i>frameworks</i> implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
Autenticación y control de acceso	Incluyen mecanismos para la identificación de usuarios mediante <i>login</i> y contraseña y permiten restringir el acceso a determinadas páginas a determinados usuarios.
Internacionalización	
Separación entre diseño y contenido	

En el caso de *frameworks* web existen para cubrir dos necesidades diferentes, existen *frameworks* para *backend* y otros para *frontend* entre los más usados se encuentran:

- ASP.NET
- AngularJS
- Ruby and Rails
- ASP.NET MVC
- React
- Django
- Laravel
- Spring
- Express

Figura 5 Frameworks más usados durante el 2017



Esta categorización está basada en el número de repositorios creados en *GitHub* y en datos recopilados del portal “<https://stackoverflow.com/>”

N. JavaScript

JavaScript fue creado por Brendan Eich y vio la luz en el año 1995 con el nombre de LiveScript, que luego fue nombrado JavaScript. Nace como un lenguaje sencillo destinado a añadir algunas características interactivas a las páginas web. Hoy en día se utiliza principalmente para crear páginas web dinámicas, una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (Alberto Ayoze, 2017)

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Alguna de las características principales de este lenguaje son: (Alberto Ayoze, 2017)

- Es liviano.
- Multiplataforma, ya que se puede utilizar en Windows, Linux o Mac o en el navegador de tu preferencia.
- Es imperativo y estructurado, mediante un conjunto de instrucciones indica al computador qué tarea debe realizar.
- Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
- Orientado a objetos y eventos.

- Es Interpretado, no se compila para poder ejecutarse. (Alberto Ayoze, 2017)

O. AngularJS

AngularJS es el nombre que recibe el *framework* de JavaScript creado por empleados de Google, lanzado en octubre de 2010. Se basa en el patrón MVC (Modelo, Vista, Controlador), con el objetivo de separar las capas de presentación, lógica y componentes de una aplicación. (Brad Green, 2013) Este *framework* fue creado con el fin de:

- Desacoplar la manipulación del DOM de la lógica de aplicación
- Desacoplar el lado cliente del lado servidor en una aplicación
- Proveer estructura para el desarrollo de una aplicación

Su principal uso es la creación de *SPA's (Single Page Apps)*, aplicaciones pensadas con el objetivo de lograr la mayor fluidez posible en el diseño de experiencia de usuario, esto se logra haciendo que la comunicación entre cliente y servidor se realice de forma transparente al usuario, por lo que se logra que este tenga la sensación de no abandonar nunca la página principal de la aplicación. (Brad Green, 2013)

Algunas de las ventajas que AngularJS aporta a los desarrolladores de *frontend* son las siguientes: (Brad Green, 2013)

- Enlace de datos bidireccional: A diferencia de la mayoría de sistemas de *'templates'* y/o *frameworks*, AngularJS usa un sistema en el que vista y modelo están en relación constante, se considera el modelo como *'Single-Source-of-Truth'*. Gracias a esto, se logra que todo cambio visual, se actualice a tiempo real en el modelo y viceversa, evitando que sea el desarrollador el encargado de lograr la sincronía entre modelo y vista, como es el caso de otros frameworks. (Brad Green, 2013)
- Directivas: Consisten en marcadores en un elemento de DOM que indican al navegador donde se esté ejecutando AngularJS que dicho elemento tiene un comportamiento específico, gracias a esto, se puede trabajar fácilmente a nivel de componentes, siendo estos, componentes reutilizables en toda la aplicación. (Brad Green, 2013)
- La comunidad de desarrolladores ha dado gran soporte a este *"framework"* (una búsqueda por AngularJS en GitHub, da 117428 resultados), por lo que hay gran cantidad de módulos ya creados. Esto permite facilitar y agilizar el desarrollo de aplicaciones, pues el desarrollador se puede enfocar en las partes más complejas del desarrollo. (Brad Green, 2013)

Pero como la mayoría de *frameworks* también posee ciertas desventajas como las siguientes:(Brad Green, 2013)

- Inyección de dependencias: El sistema de inyección de dependencias de AngularJS carga todos los módulos necesarios al cargar la aplicación, esto supone que incluso si el usuario no accede a ciertas partes de esta, se cargan los módulos necesarios para que dichas partes funcionen. Esto implica un volumen de tráfico mayor del necesario. (Brad Green, 2013)
- Estados compartidos: AngularJS usa los llamados *scopes* como el modelo de la aplicación. Estos se organizan en un árbol con el denominado *root scope* en la raíz que permite que por defecto cada *scope* hijo herede las propiedades de su *scope* padre. Además todos los *scopes* pueden acceder directamente a *root scope*, lo que implica que por defecto AngularJS se encuentra bajo estado compartido, eso no es bueno desde el momento en que se pretendan reusar componentes en la aplicación, pues el estado de esta podría influenciar en la creación o aparición de dichos componentes, por eso interesa que la aplicación sea *stateless* con esto nos referimos a que sea una aplicación que no guarda los datos del cliente generados en una sesión para utilizarlos en la siguiente sesión del mismo cliente.(Brad Green, 2013)

P. EmberJS

EmberJS es un *framework* bajo el patrón MVC (Modelo-Vista-Controlador) para JavaScript. Con él se pueden desarrollar aplicaciones por el lado del cliente, apoyado en varias bibliotecas JavaScript como jQuery, una librería muy conocida en el mundo del desarrollo, o *Handlebars*, una alternativa a jQuery que crea plantillas HTML con datos en objetos en formato JSON. (Kuhn, Zach, 2017)

Lo lógico antes de comenzar a usar EmberJS es saber cuáles son los conceptos que afectan al uso de este marco para la creación de aplicaciones web:

Plantillas: una plantilla, escrita con *Handlebars*, describe la interfaz de usuario de la aplicación. Cada plantilla está respaldada por un Modelo, y si el desarrollador actualiza ese Modelo, la plantilla se modifica de forma automática. El HTML plano de cada una de esas plantillas contiene:(Kuhn, Zach, 2017)

- Expresiones: comandos que activan funcionalidades, por ejemplo, para definir vistas parciales e incorporarlas a otras plantillas, procesos de renderización de vistas y plantilla. (Kuhn, Zach, 2017)
- Outlets: marcadores que conectan unas plantillas con otras. (Kuhn, Zach,2017)
- Componentes: elementos HTML personalizados que se usan para limpiar las plantillas repetitivas o crear controles reutilizables. (Kuhn, Zach, 2017)

- *Router*: se encarga de definir rutas anidadas, cada una de ellas respaldada por un modelo de datos. Las plantillas y los modelos cambian con la navegación del usuario, pero el comportamiento por defecto asociado a esas rutas anidadas se mantiene inalterado. (Kuhn, Zach, 2017)

EmberJS dispone de algunas características que pueden resultar una gran ventaja para un desarrollador de aplicaciones web. Algunas de ellas son:

- Es un *framework* preparado para programar de forma sencilla y elegante, lo que sus creadores definen como *The Ember Way*. Esto permite que se puedan hacer grandes desarrollos con pocas líneas de código, siempre y cuando nos adaptemos a las limitaciones del marco. (Kuhn, Zach, 2017)
- Es un marco de desarrollo que se basa en el paradigma de programación de software Convención sobre Configuración (CoC), que disminuye el número de decisiones que debe tomar el desarrollador gracias a que el *framework* las adopta de forma automática. Con ello se gana en simplicidad, pero se pierde en flexibilidad. El programador solo tiene que especificar los aspectos no convencionales del código de una aplicación porque EmberJS genera todo lo demás. (Kuhn, Zach, 2017)
- El *Router* de EmberJS simplifica mucho el trabajo con su sistema de rutas anidadas: plantillas que comparten *Router*, Controlador y Vista y que replican desde la memoria el comportamiento por defecto asociado a esa ruta. Eso ahorra mucho esfuerzo y tiempo y todo es más sencillo. (Kuhn, Zach, 2017)
- Plantillas dinámicas gracias a Handlebars.js. (Kuhn, Zach, 2017)
- Inyección de dependencias entre controladores. (Kuhn, Zach, 2017)

Q. ReactJS

React es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos. (Grados, Julio Giampiere, 2017)

ReactJS está construido en torno a hacer funciones, que toman las actualizaciones de estado de la página y que se traduzcan en una representación virtual de la página resultante. Siempre que React es informado de un cambio de estado, vuelve a ejecutar esas funciones para determinar una nueva representación virtual de la página, a continuación, se traduce automáticamente ese resultado en los cambios del DOM

necesarios para reflejar la nueva presentación de la página. (Grados, Julio Giampiere, 2017)

ReactJS utiliza un concepto llamado el DOM virtual que hace selectivamente sub-árboles de los nodos sobre la base de cambios de estado, desarrollando esto, con la menor cantidad de manipulación DOM posible, con el fin de mantener los componentes actualizados, estructurando sus datos. (Sidorenko, Vladimir, 2017)

React tiene que ver con los renderizados rápidos, pero no se puede crear una aplicación poderosa con el lado del servidor con todas las funciones usando React, es genial solo para la manipulación de vistas. Pasar a otros marcos puede requerir la reescritura de todo el HTML. Al usarlo con Node, necesitará agregar muchos paquetes. Al igual que con Express. (Sidorenko, Vladimir, 2017)

R. Django

Django es un *framework* de código abierto *Full-stack* con tecnología de Python y admite la versión de Python hasta 3.4. *Full-stack* significa que proporciona todo lo que se necesita para construir una aplicación web. Plantillas, del lado del servidor, incluso genera un panel de administración para configurar su sitio web sin editar el código, es como un tipo de CMS (sistema de gestión de contenidos) pero no es un CMS. Por lo tanto, es probable que no necesite ningún otro *framework* de terceros para construir un sitio web potente. (Sidorenko Vladimir, 2017)

Todas las características clave se incluyen inicialmente con el *framework*. Django sigue una arquitectura Modelo-Vista-Controlador (muy similar a MVC) para dividir la lógica y las vistas. Una de sus ventajas es que utiliza un *Object relational mapper* (ORM) para mapear objetos del proyecto con tablas de bases de datos. Dado que Django admite nativamente motores de bases de datos comunes como MySQL, PostgreSQL, SQLite, Oracle, SQL Server, DB2 y otros con controladores adicionales, hace que la migración entre ellos sea muy simple, no será necesario volver a escribir o adaptar el código ya que el *framework* se encarga de mapear todo por sí mismo. (Sidorenko Vladimir, 2017)

Como todos los *frameworks* Django tiene algunas desventajas como, por ejemplo:

- I. Enfocarse demasiado en la compatibilidad reduce de alguna manera la velocidad de desarrollo, a menudo es difícil integrar nuevas palabras claves en el sistema. (Sidorenko Vladimir, 2017)
- II. Django es un marco monolítico y te mantiene dentro de los patrones dados. (Sidorenko Vladimir, 2017)
- III. Siempre se obtiene una estructura relativamente grande, incluso si se realiza una página de aterrizaje simple. (Sidorenko Vladimir, 2017)

- IV. La configuración de procesos asíncronos para sistemas de mensajería y juegos en tiempo real es posible pero bastante compleja. (Sidorenko Vladimir, 2017).

S. Symfony

Symfony es un *framework* para construir aplicaciones web con PHP. En otras palabras, Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web. (Negreira, Javier, 2017)

Para los iniciados en los *frameworks*, Symfony es una de las mejores copias para PHP del famoso framework Ruby on Rails. Symfony ha tomado las mejores ideas de Rails y de muchos otros *frameworks*, ha incorporado ideas propias y el resultado es un framework elegante, estable, productivo y muy bien documentado. (Negreira, Javier, 2017)

Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. (Negreira, Javier, 2017)

La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. (Negreira, Javier, 2017)

Algunas de las ventajas que posee este *framework* son las siguientes:

1. Basado en PHP 5 y, por tanto, lenguaje orientado a objetos. (Negreira, Javier, 2017)
2. Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y unix estándares). (Negreira, Javier, 2017)
3. Independiente del sistema gestor de bases de datos. (Negreira, Javier, 2017)
4. Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. (Negreira, Javier, 2017)
5. Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional. (Negreira, Javier, 2017)
6. Sigue la mayoría de mejores prácticas y patrones de diseño para la web. (Negreira, Javier, 2017)
7. Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. (Negreira, Javier, 2017)

Existen también algunas desventajas que aquejan a este *framework* tales como:

8. Primero que nada, necesitas por lo menos un VPS para poder publicar tus aplicaciones en la web ya que necesitas tener la habilidad de poder descargar e instalar cosas en tu servidor para que Symfony funcione apropiadamente.
9. El otro problema de Symfony es la caché. Gran parte de la velocidad de Symfony se debe a un uso extensivo de la memoria caché por lo que cuando estás desarrollando tiende a ser algo tedioso tener que estar limpiando el caché de vez en cuando.
10. A veces tener tanta flexibilidad son las causas de mal diseño de aplicaciones, pero el que es mal programador va a ser mal programador con cualquier *framework*.
11. Los procesos utilizan demasiada memoria.

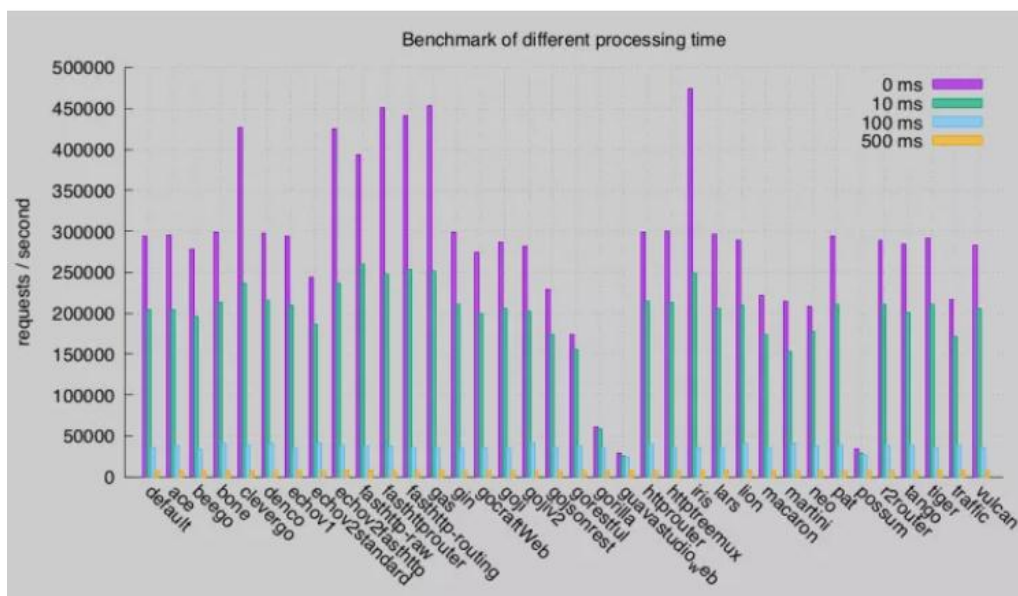
T. Iris

El *framework* Web Iris se considera el Framework Web más rápido para desarrollar en Lenguaje GO, es un IDE muy completo que cuenta con una serie de funcionalidades que nos permiten desarrollar aplicaciones de manera más rápida y organizada utilizando Lenguaje GO. (Toro, Luigys, 2017)

Algunas de las características más importantes de Iris es que está centrado en el alto rendimiento, el IDE en línea, se puede editar y compilar del lado de cliente, en caso de no poder acceder directamente gracias al *pluEdita* y compila el código del lado del cliente cuando no está en casa a través del plugin-editor. Tiene un soporte de enrutamiento robusto y tiene una serie de plantilla que podemos utilizar para nuestros proyectos. Posee una *WebSocket* API altamente escalable con eventos personalizados, también posee sesiones de soporte con GC, *memory*, *Redis providers*, soporta la creación de API REST y la personalizador de errores HTTP. (Toro, Luigys, 2017)

El equipo de desarrollo del *framework* Web Iris a menudo sube pruebas de rendimiento de su herramienta, este conjunto de pruebas tiene como objetivo comparar todo el procesamiento de solicitudes HTTP entre todos los Framework Web de Go. En la gráfica se puede observar una realidad que es la que caracteriza al Framework Web Iris, es sin duda el Framework Web más rápido para desarrollar en Lenguaje GO. (Toro, Luigys, 2017)

Figura 6 Velocidad de desarrollo de cada framework basado en Go



U. Node.js

Node.js es un entorno JavaScript del lado del servidor, basado en eventos. Node ejecuta JavaScript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporcionar un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila JavaScript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo como *bytecode*. Node es de código abierto, y se ejecuta en Mac OS X, Windows y Linux. Al utilizar la sintaxis de JavaScript Node.js puede exprimir todas las ventajas de este lenguaje y una de ellas es poseer un excelente modelo de eventos, ideal para la programación asíncrona. JavaScript también es un lenguaje omnipresente, conocido por millones de desarrolladores. Esto reduce la curva de aprendizaje de Node.js, ya que la mayoría de los desarrolladores no tendrán que aprender un nuevo lenguaje para empezar a construir aplicaciones usando Node.js. (Brad Dayley, 2014)

Además de la alta velocidad de ejecución de JavaScript, la verdadera magia detrás de Node.js es algo que se llama Bucle de Eventos (*Event Loop*). Para escalar grandes volúmenes de clientes, todas las operaciones intensivas I/O en Node.js se llevan a cabo de forma asíncrona. El enfoque tradicional para generar código asíncrono es engorroso y crea un espacio en memoria no trivial para un gran número de clientes (cada cliente genera un hilo, y el uso de memoria de cada uno se suma). (Brad Dayley, 2014)

V. Sqel.js

Squel.js un generador de consultas SQL para JavaScript. Escribir consultas SQL puede ser una tarea sencilla, pero se puede llegar a complicar si se necesitan realizar uniones entre tablas o agregar muchas condiciones incluso aplicar operaciones o agrupar por determinados campos. En ocasiones se pueden crear consultas realizando concatenaciones de texto, pero hay varios problemas con este tipo de implementación entre los cuales están: (Ramesh Nai, 2013)

1. Se necesitan cadenas de varias líneas para la legibilidad.
2. La lógica de consulta puede depender de la lógica del negocio en diferentes partes del código.
3. Concatenaciones de cadenas es desordenado y propenso a errores.

Por lo tanto Sqel.js proporciona una sintaxis sencilla, fácil de usar, donde no importa el orden de las instrucciones y permite cualquier tipo de operación, agrupación, condición, etc. que actualmente soporte SQL. Sqel.js únicamente genera una consulta y la puede convertir a texto, no posee la habilidad de ejecutarla. (Ramesh Nai, 2013).

W.Json

Json (JavaScript *Object Notation*) es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías. El principio básico es con pares atributo-valor, éstos deben estar encerrados entre llaves { , } que es lo que definen el inicio y el fin del objeto. (Esquivá Rodríguez, Alejandro, 2017).

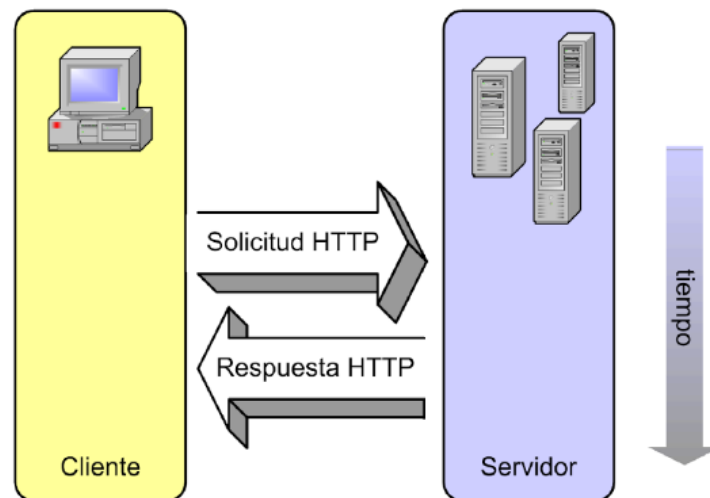
Los tipos de valores que podemos encontrar en Json son los siguientes:(Esquivá Rodríguez, Alejandro, 2017)

- Un número (entero o con punto decimal)
- Un texto (entre comillas simples)
- Un *booleano* (verdadero o falso)
- Una lista (entre corchetes [])
- Un objeto (entre llaves { })
- Valores anulables

X. Protocolo de comunicación

El protocolo HTTP es un protocolo simple de tipo solicitud-respuesta, incluido en la familia de protocolos TCP/IP que se utiliza en internet, lo que quiere decir que cada vez que se accede a una página mediante el uso del HTTP, se establece una conexión diferente a las anteriores. (Fernando Berzal, 2005) A continuación se presenta un diagrama que representa el flujo del protocolo

Figura 7 Flujo del protocolo HTTP



Este protocolo sólo distingue dos tipos de mensajes, solicitudes y respuestas, ambas incluyen cabeceras y cuerpo de mensaje y no necesariamente son iguales, es más, la única diferencia que existe entre ambos es la primera línea que ocupa la cabecera. El cuerpo del mensaje es donde se transmiten los datos en sí, y en las cabeceras es donde se especifica información adicional relevante sobre los datos que se están transmitiendo. (Fernando Berzal,2005)

Existen varios tipos de peticiones dentro del protocolo HTTP algunos de los más reconocidos son:

- GET
- POST
- PUT
- DELETE

Cuando acceden desde el navegador a una página web mediante el protocolo HTTP la solicitud que es enviada al servidor HTTP se ve de la siguiente forma:

Figura 8 Ejemplo de un GET visto desde el navegador

```
GET http://csharp.ikor.org/index.html HTTP/1.1
If-Modified-Since: Fri, 31 Oct 2003 19:41:00 GMT
Referer: http://www.google.com/search?...
```

El final de la solicitud lo marca una línea en blanco, esta línea le indica al servidor HTTP que el cliente ya ha completado la solicitud, por lo tanto, ya le es posible al servidor comenzar a generar una respuesta adecuada a la solicitud que recibe (Fernando Berzal, 2005), dicha respuesta se ve de la siguiente forma:

Figura 9 respuesta de una solicitud enviada mediante el protocolo http

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Sun, 17 Aug 2003 10:35:30 GMT
Content-Type: text/html
Last-Modified: Tue, 27 Mar 2001 10:34:52 GMT
Content-Length: XXX
<html>
  --- Aquí se envía el texto de la página HTML
</html>
```

Y. Gráficos

Un gráfico es la representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que esos datos guardan entre sí y facilitar su interpretación. (Pablo Juan Verdoy 2006)

Un gráfico también puede ser un conjunto de puntos, que se plasman en coordenadas cartesianas, y sirven para analizar el comportamiento de un proceso, o un conjunto de elementos. (Pablo Juan Verdoy 2006)

A continuación, se presenta una tabla que ayuda a comprender de mejor manera el objetivo de cada uno de los gráficos más populares y útiles:

Tabla 2 Objetivo de cada tipo de gráfico

Objetivo del gráfico	Tipo de gráfico que se debe usar
Resaltar la magnitud del cambio a lo largo del tiempo	Gráfico de áreas, gráfico de líneas
Mostrar tendencias a lo largo del tiempo	Gráfico de áreas, gráfico de columnas, gráfico de líneas
Comparar datos	Gráfico de barras, gráfico de columnas, gráfico de indicadores
Mostrar la relación de las partes con todo el conjunto	Gráfico circular, gráfico de anillo, gráficos apilados
Mostrar las partes que contribuyen al total	Gráfico de columnas apiladas
Mostrar grupos de datos Relacionados	Gráfico de barras, gráfico de columnas
Resaltar proporciones	Gráfico circular, gráfico de anillo
Compara diferentes tipos de información cuantitativa.	Gráfico de columnas, gráfico de líneas
Mostrar la distribución de datos	Gráfico de dispersión, gráfico de burbujas, gráfico de puntos

Z. Datos abiertos

Los datos abiertos consisten una política, filosofía y una práctica que requiere que ciertos datos sean de libre acceso para todo el mundo, sin limitaciones técnicas o legales. En el sector público, tener acceso a los datos de la Administración garantiza la transparencia, la eficiencia y la igualdad de oportunidades, a la vez que se crea valor. La transparencia porque se pueden consultar y tratar datos que vienen directamente de las fuentes oficiales, las eficiencias porque ciudadanos y organizaciones pueden crear servicios de forma más ajustada en colaboración con la administración; y la igualdad de oportunidades porque el acceso es el mismo para todo el mundo. Las licencias y los términos de uso de los datos abiertos están sometidos a las leyes de reutilización de la información del sector público, y en algunos casos pueden tener licencias de propiedad intelectual, aunque se tiende a abrirlas sin condiciones siempre y cuando se mantengan sin manipular y con la

obligatoria citación de la fuente y de su última actualización. (Antonio Morales, 2017) (OECD, 2004). (Open Definition, s.f.).

Los datos abiertos permiten la consulta y la construcción de aplicaciones, en especial software y formas de visualización, que utilizan la información liberada como fuente. Las aplicaciones las pueden hacer la misma administración u otros (particulares, organizaciones y empresas) en el momento que se libera su acceso. (Generalitat de Catalunya, 2017)

Los tipos de aplicaciones que se pueden construir con datos liberados son muy diversos y responden a objetivos y finalidades muy diferentes. Van desde los más personales de los ciudadanos que incorporan en su móvil, por ejemplo, los avisos por RSS de las incidencias del transporte, hasta los de asociaciones de determinados enfermos respiratorios que controlan los índices de contaminación o meramente empresariales, como es el caso de actualizaciones automáticas con la información de equipamientos oficiales que pueden alimentar los navegadores GPS para vehículos. (Generalitat de Catalunya, 2017)

La W3C recomienda que se liberen datos útiles y en formatos reutilizables. Cualquier formato es bienvenido, pero cuanto más bien estructurados y enriquecidos estén los datos, más fácil será reutilizarlos y construir aplicaciones que los traten automáticamente. (Generalitat de Catalunya, 2017)

AA. Gobierno abierto

Gobierno abierto es un enfoque que propone una forma particular de entender los procesos de gobierno, a partir de principios como los de la transparencia y la participación ciudadana. (Fundación Telefónica, 2013) como perspectiva surgió del reconocimiento de los problemas de legitimidad y capacidad que enfrentan los gobiernos para responder a las cada vez más numerosas, diversas y complejas de demandas sociales, en un contexto globalizado. En este escenario, las tendencias mundiales de reforma político-administrativa de los últimos años han apuntado hacia la incorporación de los ciudadanos y de organizaciones de la sociedad civil en los procesos de planeación, ejecución y evaluación de las actividades gubernamentales. (Fundación Telefónica, 2013)

El Gobierno Abierto como enfoque y propuesta de cambio del paradigma gubernativo es compatible con marcos teóricos y analíticos –como el de gobernanza- que ponen el acento en la conformación de escenarios de gobierno horizontales y con una presencia plural de actores (privados y sociales) en los procesos decisionales de la política pública, en un plano de coordinación. De igual manera, Gobierno Abierto es una perspectiva compatible con enfoques de corte gerencial que proponen una nueva forma de entender el quehacer público, a partir de la introducción en el sector público de nuevas actitudes y aptitudes que permitan incorporar efectivamente al ciudadano en la gestión pública. (Fundación Telefónica, 2013) (Open

Government Partnership, 2011) (Open Government Partnership, s.f.)

BB. Power BI

Power BI es un conjunto de aplicaciones de análisis de negocios que permite analizar datos y compartir información. Los paneles de Power BI ofrecen a los usuarios una vista de 360 grados con sus métricas más importantes en un mismo lugar. La información se actualiza en tiempo real y está disponible en todos sus dispositivos. Con un solo clic, los usuarios pueden explorar los datos subyacentes del panel mediante herramientas intuitivas que permiten obtener respuestas fácilmente. La creación de un panel es una sencilla operación gracias a los miles de conexiones a conocidas aplicaciones empresariales, que se completan con paneles pre-generados para ayudarlo a ponerse en marcha rápidamente. (Microsoft, 2017)

CC. Tableau

Para un análisis profundo, de utilidad y que le permita realizar descubrimientos, conjugue sus dos activos más importantes: su gente y sus datos (tanto big data como datos de menor volumen). Tableau admite análisis a medida por parte de prácticamente cualquier usuario. De este modo otorga a todo el personal la capacidad para ver y comprender mejor sus datos. Además, permite a sus analistas de negocios publicar KPI de toda la empresa en una plataforma de análisis centralizada y fácil de usar. (Tableau, 2017)

DD. Sisense

Sisense es una solución Inteligencia de Negocios de última generación. Esto significa que Sisense simplifica el análisis de datos complejos de negocios. Además, proporciona una solución completa para preparar, analizar y visualizar grandes conjuntos de datos. Es ágil, flexible y produce resultados viables para las empresas. Como resultado, usted podrá tener su dashboard en cuestión de días, en lugar de meses. Además, los usuarios de negocios y analistas podrán ser capaces de explorar libremente sus datos, sin depender de una amplia asistencia del área de Sistemas. Incluso, y, sobre todo, en escenarios complejos. (Viani, Leonardo, 2017).

EE. *Application programming interface (API)*

Se le conoce así al conjunto de métodos de comunicación entre distintos elementos de software. Tienen como objetivo simplificar el desarrollo de programas que buscan comunicarse con el creador del API.

FF. *Beautiful Soup*

Es una librería de Python que permite extraer información de documentos HTML y XML. Lo que hace es proveer algunos métodos para navegar, buscar, y modificar un parse tree, un kit de herramientas que permite extraer lo que se necesita del documento (Richardson, 2017)

GG. *Go*

Lenguaje de programación hecho para trabajar con máquinas con varios procesadores así como conectadas en la red permite ejecutar de manera más eficiente el trabajo. La modularidad también es parte importante del lenguaje y por lo tanto muchas librerías han sido traducidas desde otros lenguajes.

HH. *Hypertext Markup Language* (HTML)

Se refiere al lenguaje en que están escritas las páginas web y en especial a su estructura. Los navegadores de internet aprovechan las características del *markup language* para desplegar los elementos de manera que sean entendibles para las personas (HTML Introduction, s.f.) (Sweigart, 2015).

II. Lenguaje de marcado (*Markup Language*)

Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación (Lenguaje de marcado, s.f.).

JJ. PHP

Lenguaje de programación creado con el propósito de ser utilizado para el desarrollo web. Actualmente es ampliamente utilizado como lenguaje de propósito general. Tiene como ventaja el hecho de poder ser incluido dentro de documentos HTML (PHP, s.f.).

KK. *Python*

Es un lenguaje de programación orientada a objetos, sencillo, pero de gran potencia, que permite el desarrollo rápido de aplicaciones. Es un lenguaje interpretado y altamente modular, por lo que la etapa de

desarrollo se reduce considerablemente y, además, resulta sencillo añadir nuevo elementos a posteriori al programa (Van Rossum & Drake, 2000).

LL. *Requests*

Librería de *Python* que permite enviar solicitudes de HTTP, a páginas web, de manera sencilla. Es bastante popular debido a la simplicidad de sus métodos y a la flexibilidad que permite al momento de enviar parámetros.

MM. *Web scraping*

Se refiere a la práctica de obtener, recopilar y almacenar información por cualquier medio que no sea un programa interactuando con un API. La manera más común de hacerlo es con un programa que hace solicitudes a un servidor web y este le responde con el HTML de la página solicitada, esta respuesta es analizada y se extrae la información deseada (Mitchell, 2015).

NN. Interacción humano-computador

La interacción Humano-Computador es el área que estudia la forma en la que los seres humanos y los sistemas computacionales intercambian información. En la práctica, la interacción humano-computador se encarga de diseñar, evaluar e implementar la forma en la que el ser humano se comunicara con un sistema computacional específico.

La investigación en HCI lleva a la estandarización de la usabilidad, su mejora y apoyo empírico. El enfoque científico de la HCI incluye una variedad de herramientas y técnicas que ayudan a desarrollar mejores interfaces de usuario.

La interacción humano-computador ha tomado una importancia mayor en los últimos tiempos, mientras que el proceso de desarrollo de sistemas se ha perfeccionado con el propósito de simplificar su ejecución, la forma en la que se presenta la información al usuario y como este interactúa con el sistema se ha vuelto más crítica a la hora de presentar un producto al mercado. Por esta razón, la investigación en el área de HCI ha sido enfocada con el propósito de evaluar el software en una multitud de aspectos que conforman su usabilidad.

Esto ha derivado en que el diseño de los sistemas computacionales se ha centrado en el usuario y procesos de desarrollo de interfaces de interacción humano computador con procesos iterativos ha ganado popularidad.

OO. Estudio de estándar de usabilidad

El estudio de Estándar (Benchmarking) es un estudio de competencia que se lleva a cabo previo al diseño de los sistemas computacionales en los que se busca encontrar similitudes entre el sistema que se quiere diseñar con sistemas ya existentes, de esta manera, este estudio persigue determinar un estándar de diseño el cual en el mínimo de los casos se debe igualar para favorecer el éxito del nuevo sistema a diseñarse. El estudio de estándar de usabilidad consta de tres fases:

1. **Análisis de competencia:** Este análisis tiene como objetivo determinar el estado del arte actual de los sistemas investigados con el propósito de determinar que tanto el nuevo sistema deberá mejorar el estándar. El análisis de competencia se puede realizar de manera empírica o heurística. Además, el análisis de competencia permite determinar cuáles son las características de otros sistemas que conviene recrear en el sistema a diseñar.

2. **Determinar objetivos de usabilidad:** Decide las métricas de usabilidad a utilizar y el nivel deseado del sistema nuevo a desarrollar respecto a los sistemas analizados.

3. **Retorno de inversión:** Es el análisis final que justifica el valor económico del diseño y la usabilidad del sistema, comparando la inversión en el desarrollo de la usabilidad el sistema con el ahorro que representa desarrollar un sistema usable.

PP. Prototipos de diseño

Los prototipos de diseño se realizan luego del análisis de estándar y usabilidad y su objetivo es validar las ideas de diseño que se decidieron implementar luego de obtener los resultados del estudio de estándar y usabilidad. En el ambiente del desarrollo de aplicaciones se manejan por lo general tres fases de prototipos, que se diseñan, construyen y validan en un proceso iterativo. A continuación, se muestran los detalles y metodología de los prototipos antes mencionados.

1. Prototipos dibujados: Los prototipos dibujados son la primera versión de prototipos a diseñarse debido a su simpleza y a su facilidad de cambio, el tipo de retroalimentación que se busca juega un papel importante en la razón de ser de estos prototipos debido a que en las etapas tempranas del diseño de la interfaz de usuario la retroalimentación que se busca es acerca del mapeo de la aplicación, el flujo de aplicación y que es lo que falta o sobra. Debido al hecho de que con esta retroalimentación esperada es posible que se generen grandes cambios en la interfaz que se está diseñando, los prototipos en papel se vuelven muy apropiados debido a la facilidad de cambio que ofrecen, ya que se pueden hacer en vivo durante la etapa de validación de los mismos, o inclusive volver a hacerlos desde el principio sin necesidad de perder mucho tiempo o esfuerzo.

2. Prototipos diseñados: Los prototipos diseñados son la siguiente etapa del proceso de diseño de prototipos de diseño, estos se realizan luego del diseño y validación de los prototipos en papel, de esta manera, los prototipos diseñados se construyen con el objetivo de incorporar la paleta de colores, el tipo de letra y la forma de los botones, se busca retroalimentación respecto del diseño más que respecto del flujo de aplicación, que ya fue previamente validado con la realización de los prototipos en papel.

a. **Teoría del color** La teoría del color es una serie de principios y guías que pueden ser utilizadas para crear combinaciones armoniosas de colores. Se utiliza la teoría del color con el propósito de incorporar estas combinaciones de colores en una paleta de colores y finalmente utilizar esa paleta de colores en cualquier diseño.

En la teoría del color podemos encontrar que además de existir muchas formas de representar el espectro de color, como ruedas, triángulos, entre otros. Los colores tienen sus respectivos análogos, complementarios, complementarios divididos y tríadicos, estos colores muestran que colores podrían ir bien juntos a la hora de diseñar una paleta de colores.

En otro enfoque, uno de los aspectos más conocidos de la teoría del color, es el significado de los colores, que evocan en las personas y como los sentimientos causados por estos colores pueden ayudar a escoger que colores utilizar para un diseño, una combinación de colores por su significado o por las sensaciones que se desea evocar en los usuarios. (Stone, Adams, & Morioka, 2008)

Luego de definir los detalles del diseño como la paleta de colores, la forma de los botones y la fuente o tipo de letra, se procede a la validación de estos prototipos, a diferencia de los prototipos en papel, la retroalimentación que se busca con la validación de los prototipos diseñados es orientada al diseño más que al mapeo o flujo de aplicación, de esta manera se busca validar los detalles del diseño antes de proceder a la construcción de un prototipo final

3. Prototipos interactivos

Los prototipos interactivos son la fase final de la construcción y validación de prototipos, estos incorporan los hallazgos de la validación de los prototipos en papel y los prototipos diseñados con el objetivo de construir un prototipo que sea capaz de validar la experiencia de usuario y la unión del diseño con el flujo de aplicación, de esta manera, el objetivo de los prototipos interactivos es recrear de la mejor manera posible la experiencia del primer producto funcional. Para la construcción de prototipos interactivos existen una variedad de herramientas, muchas de estas herramientas ya tienen como característica la generación de código para la maquetación de los prototipos en una primera versión funcional. Sin embargo, lo más importante es la fidelidad del prototipo respecto de la versión final.

Luego de la posterior validación de los prototipos interactivos, se da paso al desarrollo de la primera versión funcional del producto, que se construye como una versión validada de forma iterativa en la fase de construcción y validación de prototipos, con el objetivo de ser evaluada respecto a su usabilidad mediante las pruebas de usabilidad.

QQ. Pruebas de usabilidad

Las pruebas de usabilidad forman parte importante del proceso de desarrollo de un nuevo sistema, ya que permite analizar los sistemas desde el punto de vista del público objetivo, además de favorecer el proceso iterativo de desarrollo de nuevos sistemas mediante el rápido reconocimiento de errores y mejoras antes de salir al mercado.

Las pruebas de usabilidad se dividen en seis etapas:

1. **Desarrollo de un plan de prueba:** El plan de prueba se realiza con el propósito de identificar las piezas clave que conformaran la prueba, el objetivo de la prueba, el perfil de usuario, el diseño de la prueba entre otros conforma todo el armazón de la prueba que se realizara, desde lo que se espera encontrar hasta el formato de la información a presentar en el informe final.
2. **Selección y adquisición de participantes:** El proceso de selección y adquisición de participantes inicia con la definición de los usuarios finales, estos se dividen entre categorías, se recomienda por lo general contactar 4 personas de cada categoría para realizar la prueba de usabilidad.
3. **Preparación de los materiales de prueba:** Entre los materiales de prueba se encuentra un script de orientación, que permite al sujeto de prueba informarse sobre cómo se realizara la prueba, un cuestionario que permita corroborar que la persona pertenece al público objetivo. Un

consentimiento informado que garantice la confidencialidad de las pruebas, un script de entrenamiento que explique la metodología de la prueba (solamente si es necesario), las tareas a realizar y cualquier cuestionario final que sea necesario.

4. **Realización de una prueba piloto:** La prueba piloto se realiza con un sujeto de prueba solamente, en un ambiente controlado, y con el material final y la versión del prototipo que serán utilizados en la prueba real.
5. **Realización de la prueba real:** La prueba real se realiza a todos los sujetos de prueba adquiridos, con el material final y la versión del prototipo que se utilizó en la prueba piloto, la prueba real se graba para poder sacar conclusiones de lenguaje corporal, expresiones etc...
6. **Análisis y reporte final:** En el análisis final se incluyen todos los descubrimientos de la prueba de usabilidad, la especificación del sistema probado, el lugar en el que se realizó la prueba y los resultados de la prueba. Además se deben tabular los resultados acorde a las métricas definidas, la información de los sujetos de prueba que realizaron la prueba de usabilidad, el análisis cualitativo y cuantitativo de los resultados y una lista de aspectos positivos encontrados durante la prueba de usabilidad.

RR. API

Los servicios del backend se exponen a través de un formato específico para que pueda ser utilizado por la parte visual del sitio web. A este formato se le conoce como API. Application Programming Interface (API) es un contrato que define el protocolo de intercambio de información para el desarrollo de software. (Jacobson, 2012).

El API se considera un API *RESTFUL* porque utiliza JSON como formato de intercambio de información.

El formato *REST* presenta las siguientes restricciones:

1. Interfaz uniforme: Define una interfaz entre el servidor y el cliente. Simplifica y desacopla la arquitectura para que se vuelva independiente. Se basa en recursos y utiliza un protocolo de intercambio de información basado en texto. (Stormpath, 2016)
2. Independiente del estado (*stateless*): cada solicitud de http no depende del resultado de la solicitud anterior o la siguiente. Las solicitudes son procesadas según la información que provee.

No existe sesión entre las solicitudes, cada solicitud debe estar autenticada para manejar la sesión. (Stormpath, 2016)

3. Independencia: Se pueden crear o eliminar ciertas partes del API (en capas) sin afectar a otras ni tampoco teniendo que apagar la aplicación entera, simplemente se agregan o quitan *endpoints* del API. (Stormpath, 2016)
4. Caché: un restful API prioriza y facilita el uso de caché en un cliente, lo cual mejora el rendimiento desde el lado del consumidor del API. . (Stormpath, 2016)
5. Seguridad: Se puede manejar encabezados de http que permiten validar autorización para consumir los recursos del API. (Stormpath, 2016)
6. Encapsulación: Hay ciertas partes del servicio que no necesitan ser expuestas al API, lo cual es posible por la arquitectura que se sigue. (Stormpath, 2016)

SS. Solicitud HTTP

La comunicación en HTTP (*Hyper Text Transfer Protocol*) se centra alrededor de un concepto llamado, *request-response cycle* . Una solicitud http es una petición que se hace al servidor y una respuesta http es la respuesta que se recibe de la petición enviada. Para realizar una petición válida es necesario cuatro elementos:

1. URL (*Uniform Resource Locator*)
2. Método
3. Listado de encabezados
4. Cuerpo

Un URL es la dirección que se usa para decirle para que un cliente se comunique con un servidor y le pida recursos para interactuar. (Petychakis, 2015)

El método utilizado en la petición le dice al servidor que tipo de acción desea el cliente que el servidor realice. Existen principalmente los siguientes métodos:

1. GET: le pide al servidor obtener un recurso.
2. POST: le pide al servidor crear un nuevo recurso.
3. PUT/PATCH: le pide al servidor editar/modificar un recurso existente.
4. DELETE: le pide al servidor eliminar un recurso. (Petychakis, 2015)

Los encabezados de la solicitud http tienen meta data acerca de la solicitud y el cuerpo contiene la información que el cliente desea enviar al servidor. La respuesta del servidor incluye la misma información, con la diferencia que agrega un código de respuesta para indicar el estado de la petición. (Petychakis, 2015)

Actualmente existen dos tipos de formatos para enviar información a través de un API. Existe el JSON, que utiliza un formato de llave – valor y existe el XML que utiliza estructura de árbol donde los nodos tienen atributos que definen los valores. (Petychakis, 2015)

TT. WebSockets

Debido a que el concepto de la plataforma es de una red social es necesario manejar data en tiempo real. Por lo que se necesitará un protocolo que permita una conexión continua basada en TCP para permitir una comunicación activa entre el cliente y el servidor.

WebSocket es un protocolo de red que define como un servidor y un cliente se comunican a través de la web. Los exploradores web implementan el protocolo a través de un WebSocketAPI que permite una conexión de doble vía donde se pueden enviar y recibir mensajes a través de una conexión TCP. Esta comunicación está basada en eventos y cada vez que hay un cambio de estado o evento se envía la información o notificación deseada por el programador. Este protocolo asegura la recepción de la información. (Wang, 2013).

UU. JWT

JWT significa Json Web Tokens y estos son usados como firmas criptográficas para autenticar solicitudes de un servicio. Un JWT contiene ciertas características:

- i. *Issuer*: cliente que creó la JWT.
- ii. *Subject*: que representa la JWT.
- iii. *Audience*: para quien es la JWT.
- iv. *Expiration*: cuando la JWT deja de ser válida.
- v. *Not before*: cuando la JWT empieza a ser válida.
- vi. *Issued at*: fecha de creación.
- vii. *Id*: identificador único de JWT.

Los tokens JWT están firmados criptográficamente pero no están encriptados. Esto quiere decir que cualquiera que tenga acceso a ella podrá ver la información que se está transmitiendo decodificando el *token* en base 64, pero no lo podrá modificar. En caso de transmitir información sensible a través de los

tokens se recomienda utilizar el protocolo HTTPS TLS para encriptar toda la información o también se podría encriptar el token JWT, que se conoce como JWE. Se puede combinar JWT con una encriptación a nivel de https llamada Transport Layer Security (TLS), aunque cabe notar que también es conocido como la nueva versión de SSL (≥ 3.1). (Jones, 2015)

VV. ORM y ODM

Object relational mapping (ORM) es una tecnología de capa abstracción de bases de datos. En otras palabras, es una técnica de programación que soporta la conversión de tipos incompatibles en lenguajes de programación orientados a objetos. Una de las funcionalidades principales de un ORM es el mapeo estructurado que se usa para enlazar los objetos a la base de datos y sus tipos. El mapeo expresa como un objeto y sus atributos se comportan en una o más tablas y sus campos en la base de datos. Usar un ORM implica usar una sintaxis específica dependiendo del proveedor que se esté usando, agregando una capa de complejidad. Un ODM es similar conceptualmente comparado con un ORM. Un ODM está hecho para trabajar con bases de datos basadas en documentos o no estructuradas (NoSQL) en cambio un ORM utiliza bases de datos estructuradas (SQL). (Marty, S, 2012)

WW. Middleware

Middleware es cualquier cantidad de funciones que son invocadas por la capa de enrutamiento de `express.js` antes que llegue a la última función a la que está intencionado la solicitud http. Las funciones de middleware tienen acceso a los objetos de solicitud y respuesta del protocolo de http. Un middleware puede realizar las siguientes tareas: (Shtylman, 2014)

1. Ejecutar cualquier pedazo de código.
2. Realizar cambios sobre los objetos de solicitud y respuesta del protocolo de http.
3. Terminar el ciclo de respuesta de una solicitud.
4. Llamar a la siguiente función guardada en la pila de funciones. (Documentación Express, 2017).

Las funciones de middleware tienen un tercer objeto llamado *next* que sirve como tercer parámetro en las funciones. Para llamar a la siguiente función de la pila (*stack* en inglés) simplemente se realiza una llamada a esta función. Esto se debe porque debido a que Node js se ejecuta de forma asíncrona no sabe cuándo una operación ha sido terminada, entonces hay que decirlo explícitamente. (Shtylman, 2014)

XX. Linter

Un linter es una herramienta que revisa uno o más archivos que han sido escritos en un lenguaje de programación de acuerdo a ciertas reglas definidas. Cada lenguaje de programación tiene sus mejores

prácticas de estilo que son establecidas por el grupo de desarrollo del lenguaje o las que son generalmente aceptadas por la comunidad. Estas reglas se codifican en un linter para convertirlo en un proceso automático. Si una línea del código no cumple con estas reglas, el linter notifica de la regla que no se sigue y en qué línea del código sucede. Existen múltiples herramientas para cada lenguaje de programación y editores de texto especializados para aplicar un linter. (Ellis, 2017).

YY. NPM

Node Package Manager es un manejador de paquetes que tiene un repositorio en línea de librerías de Node JS públicas y privadas y también es una herramienta de comando para interactuar con el registro en línea de librerías utilizando versionamiento para cada una de ellas. (Documentación NPM, 2017)

ZZ. Yarn

Yarn al igual de NPM es un manejador que paquetes que utiliza el repositorio en línea de NPM con la diferencia que tiene ciertas mejoras. Yarn guarda en caché cada paquete que ha descargado para que no tenga que descargar el mismo paquete dos veces. Yarn agrega el uso de un *lockfile* (un archivo especializado) para guardar las versiones exactas a la hora de instalar las librerías para que cuando se instalen en otro sistema se instalen las versiones idénticas, instala las librerías en paralelo y finalmente usa verificaciones de integridad para cada paquete antes de ejecutar el código. (Documentación NPM, 2017)

AAA. SPA

Es una aplicación web que utiliza una sola página con el propósito de dar una experiencia más fluida a los usuarios, reducir los tiempos de carga y reducir los recursos consumidos por petición de archivos JS, HTML, CSS, entre otro tipo de archivos, lo que significa que solo se cargan una vez. (M. Saray, 2015)

BBB. MVC

El *Model-View-Controller* es un patrón de diseño de arquitectura de software que se basa en tres componentes principales (Vistas, modelos y controladores), su objetivo es separar la lógica de negocio de la vista de la aplicación. Los modelos representan un concepto de información lógica de negocio de la vista de estructura de información, básicamente lo que hace es representar y almacenar el estado de la información en algún punto del tiempo en la base de datos. Las vistas son la interfaz de usuario, estas son las encargadas del despliegue de información e interacción con los modelos. Finalmente, los controladores son los encargados

del procesamiento de las peticiones del usuario, lo que hace es tomar la petición de un usuario realizada desde una vista, procesada la petición, realiza toda la lógica de negocio necesaria para cumplir con la petición del usuario y devuelve una respuesta mediante un *response*, que luego se muestra en la vista. (H. Uriel, 2015)

V. DISEÑO E IMPLEMENTACIÓN DE MODELO DE DATOS DE ANÁLISIS DESCRIPTIVO

A. METODOLOGÍA

El trabajo de este módulo del megaproyecto se divide en dos sub módulos principales:

- Diseño de modelo de datos para la plataforma GobHash
- Diseño e implementación de modelo de datos para análisis descriptivo de la información extraída del portal Guatecompras

Para el sub módulo de diseño de modelo de datos para la plataforma GobHash se utilizó la metodología tradicional de desarrollo de bases de datos. Esta metodología se conforma de las siguientes fases:

- Análisis de requerimientos
- Investigación de tecnología
- Modelación de datos

En la fase de análisis de requerimientos se trabajó conjunto a los módulos de diseño y desarrollo de usabilidad para la interacción humano-computador y el módulo de desarrollo e implementación de backend para definir los requerimientos funcionales y no funcionales del sistema. Ya que se contaba con los requerimientos no funcionales, se llevó a cabo una investigación de las tecnologías disponibles en el mercado para encontrar la solución que mejor se adaptara a las necesidades del proyecto. Luego de esto se prosiguió a elaborar un modelo de datos que cumpliera con las necesidades del sistema. Se empezó con un modelo entidad relación, sobre el cual se iteró y adaptó a la tecnología escogida hasta alcanzar el modelo que se puso en producción.

Por otro lado, en el sub módulo de diseño e implementación de modelo de datos para análisis descriptivo de la información extraída del portal Guatecompras se utilizó la metodología CRISP-DM. Se empezó realizando una investigación sobre el proceso de adjudicación de concursos por parte del estado de Guatemala y el papel del portal www.guatecompras.gt en este proceso. Para la fase de comprensión de los datos, se realizó una exploración del portal Guatecompras para familiarizarse con los datos que se despliegan en este y evaluar tanto su variedad como su calidad.

En la fase de preparación de los datos se seleccionaron los campos que se utilizarían en el análisis de las adjudicaciones del estado y se trabajó junto con el módulo de extracción de datos para obtenerlos. Para finalizar esta fase se llevó a cabo un trabajo de limpieza, formateo e integración de los datos. Luego se realizó

la fase de modelación donde se itero sobre un modelo dimensional de la información, hasta alcanzar un punto de representación de la realidad que permitiera llevar a cabo un análisis descriptivo-exitoso.

Para evaluar la exactitud de los datos se generaron reportes basados en la información del modelo para compararlos con reportes que se pueden obtener del portal Guatecompras. Así fue como se obtuvo un nivel de veracidad con relación a la versión oficial. Por último, se realizó la tarea de desarrollo del proyecto, en la cual se puso a disposición del módulo de desarrollo e implementación de widgets el modelo para su utilización en el desarrollo de gráficos y se elaboró un plan de mantenimiento y validación para futuras cargas de datos.

B. DISEÑO DE MODELO DE DATOS PARA PLATAFORMA GOBHASH

1. **Análisis de requerimientos.** Para comenzar el diseño del modelo de datos de la plataforma GobHash se definieron las funcionalidades que esta debía presentar. Para esto se trabajó junto con el módulo de *diseño y desarrollo de usabilidad para la interacción Humano-Computador* para definir qué características del modelo de medio social se implementarían en la plataforma GobHash. Se utilizó como principal caso de estudio la red social Facebook (www.facebook.com).

Requerimientos funcionales del sistema:

- Cada usuario debe contar con las siguientes propiedades:
 - Alias
 - Foto de perfil
 - Biografía
 - Email
 - Contraseña
- Los usuarios establecen Relaciones unilaterales de seguimiento hacia otros usuarios.
- Los usuarios crean contenido por medio de publicaciones.
- Solo los usuarios registrados en la plataforma pueden generar contenido.
- Una publicación se define como un conjunto de widgets, acompañados de un título y una descripción.
- Un widget se define como una gráfica que representa un set de datos extraídos del conjunto de datos para análisis de adquisiciones del estado de Guatemala.
- Un usuario puede comentar o dar Hash a una publicación, sin importar si este usuario sigue al creador de la publicación. (El evento Hash es análogo al evento like en Facebook).
- Los comentarios constan de un usuario emisor y un contenido alfanumérico. No pueden comentarse imágenes. No se puede dar Hash a un comentario.

Por otro lado, también fue necesario identificar requerimientos no funcionales del sistema ya que de estos dependerá la tecnología que se implementa para el manejo de los datos y las métricas de calidad del sistema.

Requerimientos no funcionales:

- Existe un alto riesgo de que el esquema de la base de datos cambie en el tiempo ya que pueden surgir nuevos requerimientos por parte de los diferentes módulos que utilizan o afectan el modelo.
- Se necesita una lectura rápida, ya que debe darse la sensación de ver la creación de contenido en tiempo real. Tiempo máximo: 1 segundo.
- El manejador de bases de datos debe ser compatible con la tecnología utilizada por el módulo de desarrollo de backend.

2. Investigación de tecnología. Al tener identificados los requerimientos del sistema, fue necesario encontrar la solución tecnológica que mejor se adaptara a las necesidades del proyecto. Esta herramienta para el almacenamiento y manejo de datos debía satisfacer tanto los requisitos funcionales como los no funcionales del proyecto. Para esto se identificaron tecnologías que cumplieran con los requisitos del sistema, y se evaluó cuál de ellas era la más adecuada en este caso. Se evaluaron los siguientes manejadores de bases de datos.

- *MySQL*
- *PostgreSql*
- *MongoDB*

Se investigaron las diferentes características de cada manejador y se compararon para encontrar la mejor solución para el proyecto. Las principales características que se evaluaron fueron velocidad de lectura, flexibilidad, escalabilidad y compatibilidad con *Node.js*.

Con respecto a la velocidad de lectura, las tres soluciones cumplirían con el requisito no funcional de tiempo de respuesta ($t < 1s$), pero MongoDB al ser una solución noSql mantendría mejor rendimiento mientras más crece la cantidad de información. También se evaluó la integración que estos manejadores con *node.js*. En este aspecto, los tres manejadores tienen módulos de integración maduros por lo que no resultaría difícil integrar ningún manejador a la tecnología.

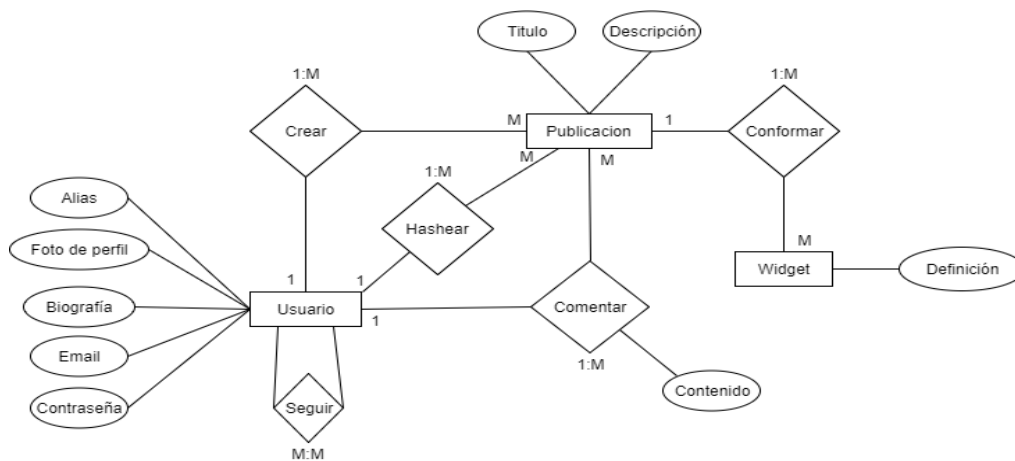
MongoDB al ser un manejador de bases de datos basado en documentos, tiene una flexibilidad de esquema muy superior al de cualquier manejador de bases de datos Relacionales. Por último, al ser una

solución NoSQL también rinde mejor con grandes conjuntos de datos que MySQL y PostgreSQL por lo que tiene un mejor grado de escalabilidad.

Basándose en este análisis se decidió utilizar MongoDB como el manejador de base datos para la plataforma GobHash.

3. Modelación de datos. Para la modelación de datos se utilizó un diagrama de entidad-relación para identificar las entidades que conforman el sistema y las Relaciones entre ellas. Considerando los requerimientos obtenidos en el primer paso de este método se obtuvo el siguiente diagrama.

Figura 10. Diagrama entidad-relación plataforma GobHash

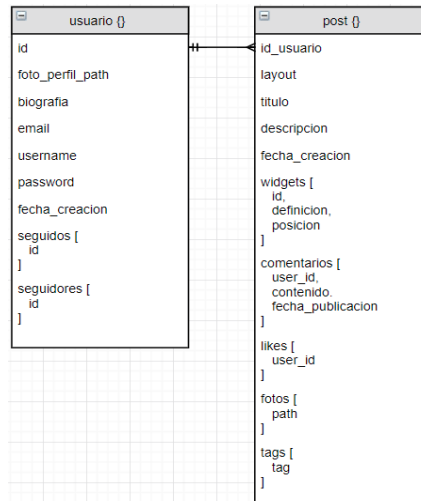


Fuente: Elaboración propia

Basándonos en el diagrama de la Figura 2 se desarrolló el siguiente modelo basado en documentos.

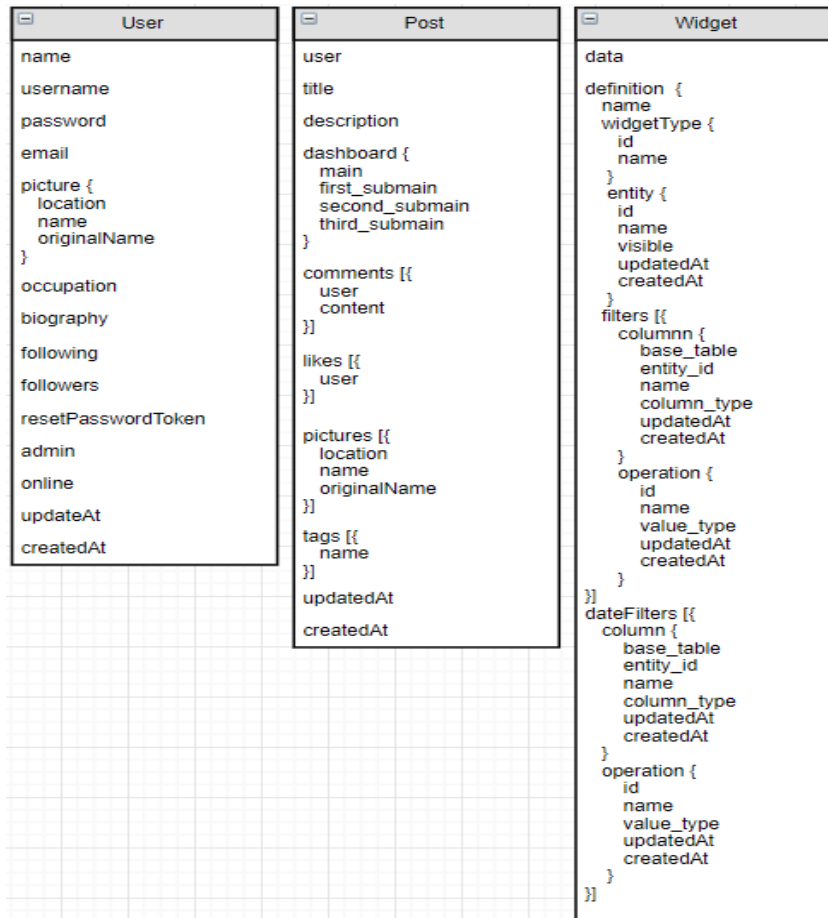
Este esquema fue aceptado por los módulos de desarrollo de backend y de generación de gráficas como guía de como deberían de almacenar los datos de la plataforma GobHash. El modelo en la Figura 3 fue la versión inicial con la que comenzó el proyecto. Debido a diferentes requerimientos de los desarrolladores y las tecnologías, este modelo fue evolucionando. Esta situación fue sencilla de manejar ya que al ser una base de datos NoSQL fue mucho más sencillo alterar el esquema sin repercusiones. La versión final del esquema fue el siguiente.

Figura 11. Primera versión modelo de datos plataforma GobHash



Fuente: Elaboración propia

Figura 12. Versión final modelo de datos plataforma GobHash



Fuente: Elaboración propia

C. DISEÑO DE MODELO DATOS PARA ANÁLISIS

1. **Comprensión del negocio.** El objetivo de esta fase es determinar cuál es el objetivo del análisis de datos que se desea realizar. Para esto fue necesario llevar a cabo una investigación para comprender el papel y los objetivos que el portal Guatecompras tiene en el proceso de adquisiciones del estado de Guatemala.

- Análisis del negocio

Guatecompras.gt es el nombre asignado al Sistema de Información de Contrataciones y Adquisiciones del Estado. Es administrado por el Ministerio de Finanzas Públicas, específicamente por la dirección normativa de contrataciones y adquisiciones del estado o DNCAE. El Estado de Guatemala lo utiliza para comprar y contratar bienes y servicios, dando así cumplimiento a las disposiciones del Decreto 57-92 del Congreso de la República de Guatemala, Ley de Contrataciones del Estado y su reglamento.

El proceso para que un concurso sea adjudicado empieza cuando las instituciones estatales - municipalidades, ministerios, federaciones, etcétera- publican en el portal los anuncios o convocatorias con toda la información Relacionada con la compra, venta o contrataciones de bienes, suministros, obras y servicios que requieran. Dicha información es de carácter público y su puede acceder a ella por medio de la dirección en internet www.guatecompras.gt. Todo esto con base en el artículo 9 del acuerdo gubernativo No. 1056-92 del reglamento de la ley de contrataciones del estado.

En el momento que se publica la información, se abre una ventana de tiempo en la cual los proveedores interesados pueden enviar los documentos pertinentes a la unidad compradora encargada del concurso. Luego que la ventana de presentación de ofertas termina, la unidad compradora lleva a cabo un proceso de selección en el cual se decide quién de los proveedores será a quien se le adjudique el concurso. Por último, se publica en guatecompras.gt los detalles del concurso adjudicado.

a. Objetivos de Guatecompras

- Publicidad
- Evidencia
- Igualdad de competencia
- Eficiencia
- Elegibilidad
- Transparencia

b. Objetivo del proyecto

- Extraer información de los datos de adjudicaciones del estado de Guatemala.

2. **Comprensión de los datos.** En esta fase la metodología CRISP-DM establece que es necesario una primera inmersión en la carga inicial de datos para que el analista se familiarice con estos y pueda determinar la calidad inicial de la información. Dado que el conjunto de datos para este proyecto no fue provisto por la fuente original, se utilizó esta fase para determinar junto con el módulo de extracción de datos qué información sería necesaria extraer del portal para cumplir con los objetivos del proyecto.

El proceso de comprensión de los datos se realizó mediante un análisis de la información que se presenta en el www.guatecompras.com. Mediante este análisis se identificaron 5 entidades principales dentro del proceso de adjudicación de concursos:

- Concurso
- Unidad compradora
- Entidad compradora
- Proveedor
- Representante legal

Cada una de estas entidades es representada por un conjunto de datos que se despliegan en Guatecompras. Estos mismos datos describen la relación entre cada una de las entidades. Los datos identificados por entidad fueron los siguientes.

a. Concurso

- El NOG del concurso (Numero de operación de Guatecompras)
- Una descripción del servicio o bien que se desea adquirir
- Categoría a la que pertenece el concurso
- Modalidad de compra del concurso. Un concurso puede pertenecer a una o más categorías. Existen 14 diferentes categorías
- Tipo del concurso
- El nombre de la entidad compradora, que también funciona como link hacia el perfil de la entidad como usuario tipo comprador dentro de Guatecompras
- La clasificación de tipo a la que pertenece la entidad compradora
- La unidad compradora. Se refiere al departamento dentro de la entidad compradora que se hizo cargo del concurso
- La fecha de publicación del concurso en Guatecompras

- La fecha presentación de ofertas. Es la fecha desde cuando el comprador empieza a recibir ofertas de proveedores
- La fecha de cierre de recepción de ofertas. Es la fecha cuando el comprador deja de recibir ofertas de proveedores
- El tipo de recepción de ofertas. Es el medio que el comprador establece para la recepción de ofertas de proveedores
- La fecha de finalización. Se refiere a la fecha en la que el comprador adjudicó el concurso a alguno de los compradores que ofertaron
- Estatus del concurso. En las adjudicaciones este siempre será *Terminado Adjudicado* ya que es el estado al que un concurso entra al ser adjudicado

b. Unidad compradora

- Nombre de la entidad de la que forma parte esta unidad compradora
- NIT de la entidad superior
- Nombre de la unidad compradora
- Departamento geográfico en el que se encuentra
- Municipio en el que se encuentra
- Dirección
- Teléfonos
- Números de fax
- Apartado Postal
- Páginas web
- Direcciones de correo electrónico
- Link hacia la información que la SAT tiene registrada sobre esta unidad compradora
- Nombre de los representantes legales actuales de la unidad compradora

c. Entidad compradora

- Nombre
- NIT
- Tipo
- Entidad origen de los fondos
- Departamento geográfico donde se encuentra la oficina central de la entidad
- Municipio donde se encuentra la oficina central de la entidad
- Dirección de la oficina central

- Teléfonos de la oficina central
- Números de fax
- Apartado Postal
- Páginas web
- Direcciones de correo electrónico
- Link hacia información que la SAT tiene registrada sobre la entidad
- Nombre de los representantes legales de la entidad

d. Proveedor

- Nombre o razón social
- NIT
- Nombre comercial. Puede tener más de uno.
- Estatus actual del proveedor
 - Habilitado o Inhabilitado
 - Adjudicado o No adjudicado
 - Participa o no en Contrato Abierto
 - Con o Sin Contraseña
 - Estatus en la SAT
 - Estatus del RTU
 - Solvencia del NIT en RTU
- Domicilio Fiscal
 - Departamento geográfico donde está registrado el proveedor.
 - Municipio donde está registrado el proveedor.
 - Dirección.
 - Teléfonos.
 - Números de fax.
- Datos de inscripción en la SAT
 - Tipo de organización.
 - Número de escritura de constitución.
 - Fecha de constitución.
 - Actividad económica.
 - NIT Notario

e. Representante legal

- CUI. Código Único de Identificación.
- Nombre o razón social.
- NIT.
- Estatus en Guatecompras
 - Habilitado o Inhabilitado
 - Adjudicado o No adjudicado
 - Participa o no en Contrato Abierto
 - Con o Sin contraseña
 - Estatus en la SAT
 - Estatus del RTU
 - Solvencia del NIT en RTU
- Datos de inscripción
 - Tipo de organización
 - Actividad económica

La relación entre las entidades se puede describe en el siguiente enunciado:

“Una entidad compradora está conformada por una o más unidades compradoras. Estas unidades compradoras son responsables de la adjudicación de concursos a proveedores que son representados legalmente por personas jurídicas o individuales”.

El detalle del proceso de comprensión de los datos y las páginas que se visitaron se encuentran en la sección de anexos bajo el título Proceso de comprensión de datos.

3. Preparación de los datos Basado en esta exploración de Guatecompras, se realizó una primera propuesta de campos a descargar al módulo de extracción de datos. Los datos que se propusieron fueron los siguientes.

Tabla 3. Primera propuesta de campos a extraer

Concurso Adjudicado	Unidad Compradora	Entidad Compradora	Proveedor	Representante Legal
NOG	NIT	NIT	NIT	CUI
Descripcion	Nombre	Nombre	Nombre	NIT
Categoria	Nombre entidad superior	Departamento	Departamento	Nombre
Modalidad de compra	Departamento	Municipio	Municipio	
Tipo concurso	Municipio	Representantes Legales	Actividad Economica	
Nombre entidad compradora	Representates Legales		Fecha de constitucion	
Tipo entidad compradora			Representante Legal	
Nombre unidad compradora				
Nit unidad compradora				
Fecha publicacion				
Fecha presentacion ofertas				
Fecha cierre recepcion ofertas				
Monto				
Fecha finalizacion				
Estatus del concurso				

Debido a los alcances del módulo y la dificultad que implicaba descargar estos datos para el módulo de extracción de datos, se consensó que los datos a extraerse serían los siguientes.

Tabla 4. Campos extraídos por el módulo de extracción de datos

Concurso Adjudicado	Unidad Compradora	Proveedor
NOG	NIT	NIT
Descripcion	Nombre	Nombre
Categoria	Nombre entidad superior	Departamento
Modalidad de compra	Departamento	Municipio
Tipo concurso	Municipio	Actividad Economica
Nombre entidad compradora		Fecha de constitucion
Tipo entidad compradora		Nit/Nombre Representante legal
Nombre unidad compradora		
Nit unidad compradora		
Fecha publicacion		
Fecha presentacion ofertas		
Fecha cierre recepcion ofertas		
Monto		
Fecha finalizacion		
Estatus del concurso		

El resultado del módulo de extracción de datos fueron una serie de archivos CSV que se dividían en tres grupos:

- adjudicaciones.csv

- proveedores.csv
- compradores.csv

Cada uno de estos contiene información relativa a lo que su título indica en el año 2016. Debido a la cantidad de información que se maneja en guatecompras.gt, cada uno de estos grupos se dividió mensualmente, excepto por los datos de los proveedores. En total se obtuvieron 12 archivos CSV con los datos de concursos adjudicados, 12 archivos CSV con la información de los compradores y 3 archivos CSV con la información de los proveedores.

Esta fase de la metodología se divide en tres tareas:

1. Limpieza de datos
2. Formateo de datos
3. Integración de datos

1. Limpieza de datos. En esta tarea se pretende elevar la calidad de los datos al nivel mínimo requerido por las técnicas de análisis seleccionadas. En el caso de este proyecto implicó la inserción manual de valores en campos o registros que se descargaron con defectos.

Los registros que fueron insertados en el CSV de [adjudicaciones.csv](#) fueron los siguientes:

- Los datos de la adjudicación con NOG 53385151 debido a que la descripción del conjunto contiene el carácter delimitador del CSV, lo que ocasiona problemas de sintaxis en el archivo.

Los registros que fueron insertados en el CSV [proveedores.csv](#) fueron los siguientes:

- Los datos del proveedor con NIT 84666114 debido a que el nombre del proveedor contiene el carácter delimitador del CSV, lo que ocasiona problemas de sintaxis en el archivo.

2. Formateo de datos. El objetivo de esta tarea es unificar los formatos de los valores de los diferentes registros. De manera que sea posible operar estos valores entre sí, y estas operaciones tengan sentido. Por lo que se definieron cinco tipos diferentes de datos:

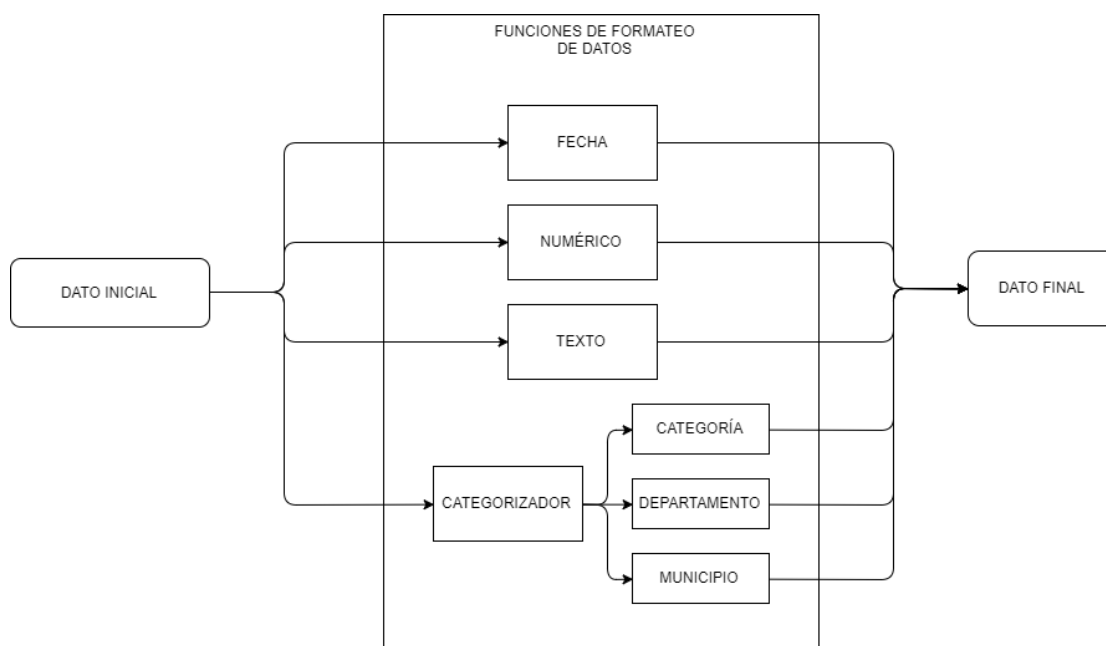
- a. Fecha, con formato dd/mm/aaaa
- b. Numérico, con valores en dominio de los reales
- c. Texto
- d. Geográfico
- e. Categoría

Los valores de tipo fecha, numérico y texto expresan explícitamente este tipo de valores y no es necesario hacer ninguna interpretación de estos. Por otro lado, los valores de tipo geográfico y de categoría

son textos, pero en este contexto deben ser interpretados correctamente. Los valores de tipo geográfico se refieren a nombres de departamentos y municipios de Guatemala mientras los valores de tipo categoría se refieren a las categorías de concursos que existen en Guatecompras.

Todos estos procesos se automatizaron mediante el programa de Python *sanitizer.py*, el cual se encuentra en el capítulo de anexos bajo el título de Código de automatización para la preparación de datos. Cada uno de los diferentes tipos de valores es procesado por una función definida en el módulo *sanitizer.py*. En el caso de los valores de tipo geográfico y categoría se implementó un categorizador que utiliza la distancia de Levenshtein para encontrar el valor al que se hace referencia dentro de un conjunto de catálogos. Este categorizador se encuentra en la sección de anexos bajo el título Código de automatización para la preparación de datos con el nombre *classifier.py*.

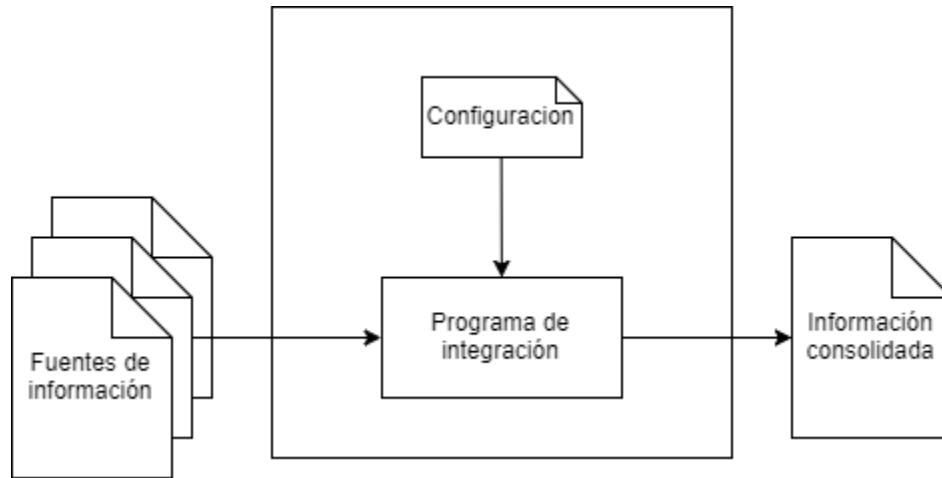
Figura 13. Flujo de formateo de datos



Fuente: Elaboración propia

3. Integración de datos El objetivo de esta tarea es centralizar la información de las diferentes fuentes existentes, en este caso, de los diferentes archivos que genera el módulo de extracción de datos. Para esto se utilizó un programa escrito en Python. Este programa se configura para que sepa que archivos debe integrar bajo que esquemas y creo un nuevo archivo CSV donde consolida toda la información semejante.

Figura 14. Flujo de integración de datos



Fuente: Elaboración propia

La intención de automatizar los procesos de integración y formateo de datos es que, para futuras cargas de información, estas tareas puedan ejecutarse con las mismas transformaciones que se efectuaron en este proyecto. Todo el proceso de automatización se encuentra en la sección de anexos bajo el título de Código de automatización para la preparación de datos.

1. **Modelación.** El objetivo de esta fase es obtener un modelo lógico de datos que cumpla con los objetivos del proyecto.

Esta fase se divide en dos tareas:

a. Selección de técnica de modelado. Para este proyecto se seleccionó la técnica de modelación dimensional. Se escogió esta técnica ya que se orienta a los procesos de negocio de tipo transaccional. En el caso de este proyecto podemos modelar las adquisiciones como compras por parte del estado y a los compradores y proveedores como dimensiones de la adquisición.

b. Construcción lógica del modelo

La metodología para diseñar un modelo dimensional tiene 4 pasos:

- i. Elegir el proceso de negocio
- ii. Decidir la granularidad de la información
- iii. Identificar las dimensiones
- iv. Identificar los hechos

a) Elegir el proceso de negocio. Para este proyecto el proceso de negocio que se desea modelar es la adjudicación de concursos por parte de unidades compradoras a proveedores del estado.

b) Declarar la granularidad. Granularidad se refiere la descripción exacta en qué se debería enfocar el modelo dimensional. Esta se define en una sola oración. Para este proyecto sería:

- “El monto pagado a cada proveedor por adjudicación”.

c) Identificar dimensiones

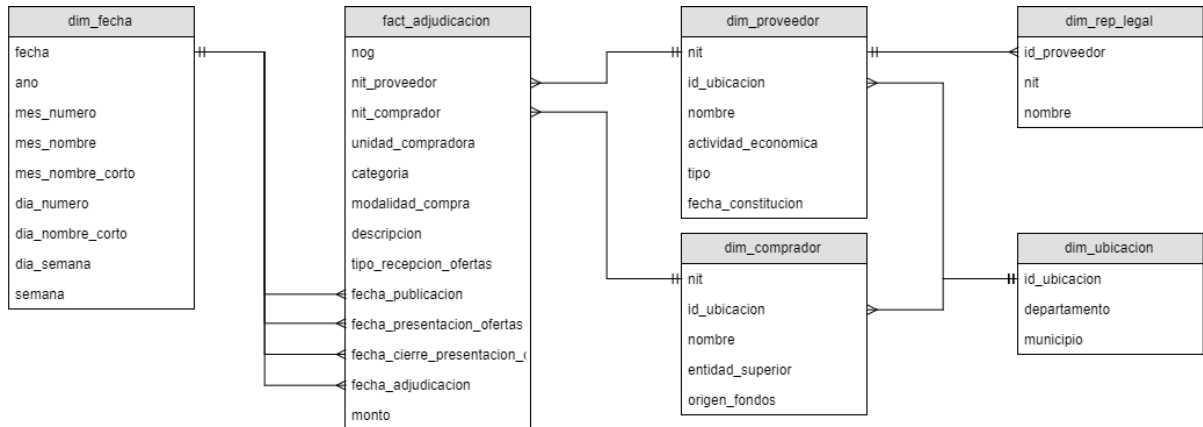
- Comprador
- Proveedor
- Fecha
- Municipio, Departamento
- Representante legal

d) Identificar los hechos. Se identifica los valores que representan una transacción en la tabla de hechos. Para representar adjudicaciones se definieron los siguientes valores:

- NOG del concurso
- NIT proveedor
- NIT comprador
- unidad compradora
- categoría
- modalidad de compra
- descripción
- tipo de recepción de ofertas
- fecha de presentación del concurso
- fecha de presentación de ofertas
- fecha de cierre de recepción de ofertas
- fecha de adjudicación
- monto

Utilizando los conceptos desarrollados por esta técnica se consigue el desarrollo del siguiente modelo.

Figura 15. Modelo dimensional de adjudicaciones del estado de Guatemala



Ya que se obtuvo el modelo que representa los datos del proyecto es necesario implementar este modelo en una herramienta de manejo de datos. La herramienta seleccionada para este proyecto fue el manejador de bases de datos PostgreSQL. El módulo de desarrollo de backend puso a disposición de este módulo una instancia t2.micro de Amazon EC2, la cual cuenta con 1 vCPU y 1 GiB de memoria RAM. Esta instancia se configuró con el sistema operativo Ubuntu 16.04 y la versión de PostgreSQL 9.5.

En PostgreSQL se creó la base de datos ghanalysis. En esta se crearon tres tablas principales, las cuales almacenan la información de los documentos consolidados que se obtuvieron la fase de preparación de datos. Los nombres las tablas son:

- raw_adjudicaciones
- raw_compradores
- raw_proveedores

Luego se utilizó la función de materialized views de PostgreSQL para crear las tablas del modelo dimensional. Las definiciones de estas se encuentran en la sección de anexos bajo el título de Creación de modelo de datos.

2. Evaluación. En esta fase se evalúa el modelo de datos obtenido utilizando métricas del negocio. Para este proyecto se evaluaron la precisión y la completión de los datos. Para esto se utilizaron reportes obtenidos del portal Guatecompras. Se descargaron los reportes generados en la sección *Estadística > Concursos Adjudicados por Modalidad y Año*. La intención fue generar los mismos reportes utilizando nuestro modelo de datos y utilizar como métrica el porcentaje de error entre el dato de Guatecompras y el dato obtenido por el modelo.

La única consideración que tuvo que tomarse para comparar los datos del modelo contra los datos del reporte público fue que la información en el modelo no contiene adjudicaciones hechas a proveedores

extranjeros. Las evaluaciones se hicieron contra reportes mensuales de las adjudicaciones realizadas en el transcurso del año 2016. De la evaluación inicial se obtuvieron los siguientes resultados.

Tabla 5. Resumen reporte de validación inicial

	Cantidad de Adjudicaciones				Monto total			
	Dato Guatecompras	Dato GobHash	Diferencia	%Error	Dato Guatecompras	Dato GobHash	Diferencia	%Error
Enero	3145	3145	0	0.00%	Q 1,354,223,236.56	Q 1,354,223,236.56	Q -	0.00%
Febrero	6484	6484	0	0.00%	Q 286,437,746.86	Q 286,437,746.86	Q -	0.00%
Marzo	6393	6393	0	0.00%	Q 339,338,500.17	Q 339,338,500.17	Q 0.00	0.00%
Abril	5893	5892	1	0.02%	Q 429,897,095.85	Q 414,684,069.97	Q 15,213,025.88	3.54%
Mayo	7701	7701	0	0.00%	Q 787,342,878.47	Q 787,342,878.47	Q 0.00	0.00%
Junio	8429	8427	2	0.02%	Q 709,557,436.96	Q 708,763,016.96	Q 794,420.00	0.11%
Julio	6493	6493	0	0.00%	Q 807,042,060.14	Q 807,060,218.14	Q (18,158.00)	0.00%
Agosto	7121	7122	-1	-0.01%	Q 818,024,094.92	Q 818,056,659.74	Q (32,564.82)	0.00%
Septiembre	10532	10533	-1	-0.01%	Q 1,402,573,291.09	Q 1,398,537,511.98	Q 4,035,779.11	0.29%
Octubre	8788	8778	10	0.11%	Q 1,226,864,252.69	Q 1,226,145,861.34	Q 718,391.35	0.06%
Noviembre	7806	7805	1	0.01%	Q 844,620,356.95	Q 833,014,211.12	Q 11,606,145.83	1.37%
Diciembre	4405	4406	-1	-0.02%	Q 692,844,217.22	Q 691,962,194.67	Q 882,022.55	0.13%

De este reporte podemos observar que únicamente los meses enero, febrero, marzo, mayo y julio obtuvieron un cero por ciento de porcentaje de error en la comparación de cantidad de adjudicaciones y monto total entre los reportes de Guatecompras y los datos obtenidos en el modelo. De esta evaluación se obtuvo el dato de que no todas las adjudicaciones en el modelo de datos contaban con fecha de adjudicación asignada, problema que se heredó desde el módulo de extracción de datos. Por lo que fue necesario buscar y asignar manualmente estas fechas. Estos ajustes se pueden ver en la definición de la materialized view fact_adjudicacion. Por otro lado, también se descubrió que en los reportes de Guatecompras no figuran adjudicaciones que fueron adjudicadas nuevamente después de una adjudicación inicial.

Luego de hacer los ajustes necesarios en el modelo, y encontrar los datos respectivos a los proveedores internacionales, la evaluación obtuvo los siguientes resultados.

Tabla 6. Validación final modelo de datos

	Cantidad de Adjudicaciones				Monto total			
	Dato Guatecompras	Dato GobHash	Diferencia	%Error	Dato Guatecompras	Dato GobHash	Diferencia	%Error
Enero	3145	3145	0	0.00%	Q 1,354,223,236.56	Q 1,354,223,236.56	Q -	0.0000%
Febrero	6484	6484	0	0.00%	Q 286,437,746.86	Q 286,437,746.86	Q -	0.0000%
Marzo	6393	6393	0	0.00%	Q 339,338,500.17	Q 339,338,500.17	Q -	0.0000%
Abril	5893	5893	0	0.00%	Q 429,897,095.85	Q 429,897,095.85	Q -	0.0000%
Mayo	7701	7701	0	0.00%	Q 787,342,878.47	Q 787,342,878.47	Q -	0.0000%
Junio	8429	8429	0	0.00%	Q 709,557,436.96	Q 709,557,436.96	Q -	0.0000%
Julio	6494	6494	0	0.00%	Q 807,066,003.14	Q 807,066,003.14	Q -	0.0000%
Agosto	7122	7122	0	0.00%	Q 818,056,659.74	Q 818,056,659.74	Q -	0.0000%
Septiembre	10534	10534	0	0.00%	Q 1,403,532,678.09	Q 1,403,532,678.09	Q -	0.0000%
Octubre	8789	8789	0	0.00%	Q 1,226,868,752.69	Q 1,226,868,752.69	Q -	0.0000%
Noviembre	7808	7808	0	0.00%	Q 844,624,865.95	Q 844,624,865.95	Q -	0.0000%
Diciembre	4406	4408	-2	-0.05%	Q 692,863,281.02	Q 692,872,074.68	Q (8,793.66)	-0.0013%

Con esta evaluación se pudo determinar que el modelo tiene una exactitud del 99.99991% en relación con los datos que muestra el portal de Guatecompras. Un desglose más detallado de esta evaluación se encuentra en la sección de anexos bajo el nombre Reporte de Validación de datos.

3. **Desarrollo.** En este último paso se puso la base de datos ghanalysis a disposición del módulo de generación de gráficas. Para esto fue necesario crear un usuario con nombre **widgets** en PostgreSQL, el cual solamente tiene permisos de lectura.

También se desarrolló un plan de mantenimiento de los datos que puede utilizarse para agregar nueva información o para revalidar la información que se encuentra actualmente en el sistema. Este plan detalla todo el proceso que se siguió de la metodología CRISP-DM y observaciones con respecto a problemas que se encontraron en esta iteración de la metodología.

Los capítulos de este plan son:

1. Proceso de selección de los datos
2. Proceso de integración de los datos
3. Proceso de validación de los datos

El manual se encuentra en el capítulo de anexos bajo el título Protocolo de mantenimiento de los datos.

B. DISCUSIÓN

Aunque ambos submódulos tienen como objetivo principal el diseño de modelos de datos, se utilizaron dos metodologías diferentes para el desarrollo de cada uno debido a que el objetivo de cada modelo era diferente. En el caso del modelo de datos de la plataforma GobHash su objetivo era definir la manera en la que la información de los usuarios iba a ser almacenada. Esto implicaba que este modelo iba a estar fuertemente Relacionado con el desarrollo de la plataforma y por lo tanto existía una alta probabilidad que la definición de la información que se almacenaría en este modelo cambiara con el tiempo, ya fuera por requerimientos del equipo de desarrollo, de la tecnología o del estudio de HCI.

Por otro lado, tenemos el modelo de análisis de información de adquisiciones del estado de Guatemala, información que se obtuvo antes del desarrollo del modelo y que era necesario estructurar de manera que fuera posible sacarles la mayor información posible a los datos sin deformar el proceso de negocio. Este sería más estable en su estructura y el modelo tendría que buscar simplificar la comprensión de la información.

Debido a los requerimientos de cada modelo, las metodologías que se utilizaron se adaptaron muy bien. Por el lado modelo de datos de la plataforma Gobhash la metodología tradicional de diseño de bases de

datos consiguió completar el objetivo del módulo con un modelo y una tecnología que fueron exitosos al momento de su implementación. El mayor aporte de este módulo a la plataforma fue la correcta selección de mongoDB como manejador de base de datos.

MongoDB cumple con los requerimientos del proyecto tanto funcionales como no funcionales, de manera que, aunque los requerimientos funcionales del modelo fueron cambiando a lo largo del desarrollo del proyecto, esta tecnología permitió que no fuera costoso no conocer desde el principio el modelo de datos final

El uso de la metodología CRISP-DM para el diseño del modelo para análisis de datos se pudo implementar muy bien. El mayor aporte de esta metodología fue la manera en la que hizo necesario comprender el negocio que se quería modelar, en este caso las adjudicaciones del estado de Guatemala, antes de definir una técnica de modelado y tecnología para el desarrollo del proyecto. Debido a que desde el principio conocíamos tanto el objetivo del proceso del negocio como el del proyecto de desarrollo, todas las decisiones que se tomaron fueron con la intención de cumplir los objetivos definidos en el primer paso de la metodología.

Dado que el objetivo del proyecto extraer información de un proceso de negocio, se seleccionó la técnica de modelación dimensional. Esta técnica es la que mejor conserva la esencia de las reglas del negocio y hace más sencillo el entendimiento del usuario que utiliza el modelo, en este caso el módulo de diseño de gráficos.

Para la implementación de este modelo se utilizó el manejador PostgreSQL debido a la cantidad de información que se maneja y los métodos que se necesitan para crear el modelo. Se evaluaron diferentes tecnologías como MySQL, PostgreSQL, MongoDB, Hadoop. Al momento de evaluar las tecnologías ya se sabía que los datos que se almacenarían en ella tendrían un tamaño alrededor de 150MB.

Los requisitos no funcionales para este módulo eran alta velocidad de lectura, baja curva de aprendizaje e integración con Node.js. Aunque las soluciones NoSQL (MongoDB, Hadoop) son más veloces que los manejadores tradicionales, el tamaño pequeño y variabilidad nula de la información hace que en este caso de uso los 4 presenten una velocidad que cumple con el requisito de tardar menos de un segundo en recuperar la información.

Por otro lado, MySQL, PostgreSQL y MongoDB cuentan con módulos de integración maduros en Node.js. Esto es algo que Hadoop aún no tiene, ya que el módulo de integración, según la comunidad, aun no es estable. Por último, se evaluó la curva de aprendizaje. Debido a su popularidad y experiencia previa, la curva de aprendizaje es menor en los manejadores de bases de datos Relacionales que en las tecnologías NoSQL. Debido a este análisis, se determinó que la mejor solución a este problema sería un manejador

Relaciónal entre MySQL y PostgreSQL. Se termino utilizando PostgreSQL por la facilidad de creación de materialized views.

El set de datos y su estructura que se obtuvieron como resultado de este desarrollo es en sí un producto que podría servir como herramienta a la sociedad civil. Aunque toda la plataforma GobHash se basa en las ideas de gobierno abierto y datos abiertos, el producto de este módulo es similar a los sets de datos que en otros países se liberan al público, y que instituciones y personas utilizan para auditar a estas instituciones.

VI. DISEÑO E IMPLEMENTACIÓN DE INDICADORES ESTADÍSTICOS REPRESENTADOS A TRAVÉS DE GRÁFICOS CON BASE EN LA INFORMACIÓN EXTRAÍDA DEL PORTAL DE GUATECOMPRAS

A. METODOLOGÍA

La metodología de desarrollo utilizada en este proyecto fue una implementación de SCRUM, ya que se realizarán reuniones semanales con todo el equipo del megaproyecto donde cada uno de los integrantes dará sus actualizaciones y se realiza una minuta como constancia de la reunión, estas reuniones para el SCRUM deben ser rápidas porque la actualización de cada miembro debe ser concisa y general, estas reuniones también se utilizan para consultar cualquier duda con respecto a módulos externos e integración. Además, se estará llevando el control de todas las tareas pendientes y en progreso y completada en una herramienta llamada “*Meistertask*” la cual es una herramienta que ayuda con la planificación de las tareas y sprint para apoyar la implementación de la metodología SCRUM en el proyecto.

Para el controlador de versiones durante el desarrollo de la plataforma GobHash se utilizó un repositorio privado en GitHub, la cual es una herramienta muy útil para llevar un seguimiento de las versiones del desarrollo de cada uno de los módulos y que la integración sea mucho más sencilla ya que todos los módulos se están manejando en el mismo repositorio. Las tecnologías que se utilizaron para el desarrollo de este módulo fueron elegidas luego de hacer comparaciones entre diversos *frameworks* que persiguieran ciertos objetivos en común y basados en ciertos indicadores que tienen como características cada una de las tecnologías se llegó a la conclusión de que para el manejo del *backend* en este módulo se iba a utilizar Node.JS y para el *frontend* se utilizó AngularJS, para la visualización de los gráficos se utilizó una librería llamada Angular-Chart.js.

Síntesis y conceptualización. En base a las reuniones presenciales que se realizaron en donde participaron los estudiantes, el asesor general de megaproyecto, y el stakeholder del mismo, se delimitaron los requisitos funcionales, el alcance y las áreas de tecnología y administración de proyectos que están involucradas.

Elaboración de módulos. Para poder cubrir las distintas áreas de tecnología y administración de proyectos se propusieron siete módulos, los cuales son:

- “Diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico”.
- “Desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de Guatecompras”
- “Diseño y desarrollo de estudio de usabilidad para la interacción Humano-Computador.”
- “Desarrollo e Implementación del *Backend*”
- “Diseño, Análisis y Desarrollo de *Frontend*”
- “Extracción de datos de adjudicaciones, proveedores y compradores del sistema Guatecompras, para su utilización en la plataforma GobHash”
- “Gestión del desarrollo de la Plataforma, diseño de sus estrategias de comunicación y análisis financiero de la implementación de un modelo de negocio *Freemium*.”

Cada uno de los módulos cumple con el objetivo de satisfacer una necesidad que lleva a cumplir con todos los requisitos funcionales.

Específicamente el trabajo de este módulo se dividió en 4 fases:

- **Análisis**
 - Análisis del modelo y estructura de la base de datos donde están almacenados los datos que se extrajeron del portal de Guatecompras.
 - Análisis de las estructuras para los metadatos que se necesitan.
 - Análisis de los pasos para crear una gráfica.
 - Análisis de los requerimientos funcionales que fueron establecidos.
 - Análisis de la planeación.
 - Análisis de las tecnologías a utilizar.
- **Investigación y diseño**
 - Estudio de plataforma que brinda el servicio para crear gráficos como Tableau, Power Bi y Sisense.
 - Estudio y revisión de plataformas que existen y brindan un servicio igual o parecido al que ofrece este módulo.
 - Diseño de la estructura de la plataforma

- Estados, servicios y controladores
 - Definición de los metadatos.
 - Diseño de las tareas para cumplir con los requisitos funcionales.
 - Definición de la estructura de los pasos para crear una gráfica
 - Diseño final de los metadatos.
 - Diseño de la estructura para la “definición” de una gráfica
 - Diseño y definición de los *endpoint* que se deben crear para consultar los metadatos y crear la previsualización.
 - Definición de las tecnologías que se utilizaran.
 - Definición de la librería para manejo de Gráficas en AngularJS
 - Diseño y definición de los pasos y requerimientos para la integración con los módulos de “Diseño Análisis y Desarrollo de *frontend*” y “Desarrollo e Implementación de *Backend*”
- Validación del diseño
 - Validar los pasos para crear los gráficos con base en la revisión de plataformas como Tableau, Power Bi y Sisense.
- Desarrollo
 - Desarrollo de la aplicación para crear gráficas
 - Crear el repositorio del proyecto
 - Iniciar el proyecto
 - Crear estructura base de una aplicación de AngularJS
 - Crear una función para el manejo de modales.
 - Crear la base de datos para almacenar los metadatos.
 - Crear los metadatos basado en el esquema, “materialized views” y columnas de la base de datos donde están almacenados los datos extraídos de la plataforma de Guatecompras.
 - Crear las vistas para cada paso que se necesita para crear una gráfica.
 - Crear los controladores, servicios y repositorios.
 - Los Requisitos funcionales del módulo eran los siguientes:
 - Permitir al usuario seleccionar los campos que estén disponibles en la base de datos donde se almacenan todos los datos extraídos de Guatecompras.
 - Permitir al usuario crear por lo menos tres tipos diferentes de gráficos.
 - Permitir al usuario crear gráficos que impliquen Relacionar varias tablas entre sí.

- Permitir al usuario crear gráficos agrupando datos por cualquier columna registrada en los metadatos.
- Permitir al usuario aplicar operaciones sobre una determinada columna.
- Permitir al usuario agregar filtros por cualquier columna registrada en los metadatos.
- Permitir al usuario ver una previsualización de la gráfica que represente los datos que resultaron de los campos que seleccionó.

B. DELIMITACIÓN

El presente módulo contempla la entrega de un editor que permite a un usuario crear un gráfico a la vez y una ventana para la previsualización del mismo.

Cuando se hace referencia a que la plataforma GobHash permitirá al usuario generar indicadores estadísticos, realmente se habla de gráficos, los cuales por definición son una clase de indicadores estadísticos. En la plataforma el usuario podrá crear por lo menos tres tipos de gráficos. Asimismo, no existe un límite para el número de gráficos que el usuario pueda crear, los mismos serán creados con los datos que se logren extraer de la plataforma Guatecompras. Los gráficos pueden ser filtrados, ordenados, agrupados, etc. por lo menos por dos entidades esenciales que se extraigan la información relevante con respecto a las adquisiciones del estado.

No estará disponible la opción de reeditar un gráfico y no se podrá hacer ningún tipo de copia automática de algún otro gráfico que esté contenido en alguna otra publicación, lo que quiere decir que si desea replicar algún gráfico que haya encontrado en la plataforma deberá hacerlo desde cero.

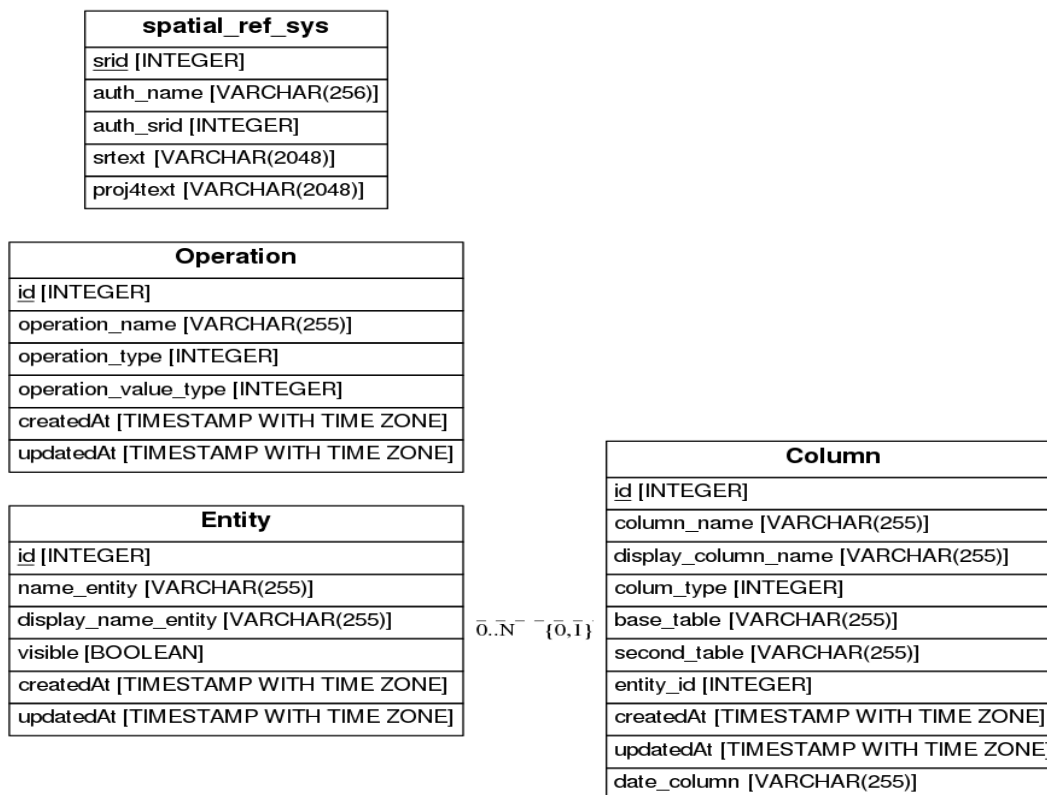
Los datos que conformen los gráficos pueden variar dependiendo de las actualizaciones por parte del módulo encargado de extraer datos del portal de Guatecompras, la variación de estos datos no afecta de forma directa a los gráficos, los cuales se actualizarán únicamente cuando se creen nuevos y no en tiempo real mediante vayan variando los datos.

C. RESULTADOS

Al analizar de la base de datos donde se encuentran almacenados los datos que fueron extraídos del portal de Guatecompras que lleva por nombre “ghanalysis”, fue necesario crear una base de datos Relacional, llamada “udvcomgt_gobhash”, donde fueran almacenados metadatos cuyos valores están basados en los *materialized views* y las columnas de la base de datos descrita anteriormente. Las tres tablas que se pueden apreciar en la “Figura 9” almacenan todos los metadatos que son necesarios para apoyar la implementación del *backend* y crear una consulta que represente todo lo que el usuario configuró en la interfaz de usuario. En la tabla

“Column” se almacén los nombres, tipos y a los *materialized view* que pertenecen las columnas de la base de datos “ghanalysis”. En la tabla “Entity” se almacenan los nombres de los *materialized view*, y por último en la tabla “Operation” se almacenan los nombres y tipos de las operaciones que se pueden aplicar a las diferentes columnas que están almacenadas en la tabla “Column”.

Figura 16 Modelo entidad relación de la tabla “udvcomgt_gobhash”



Como parte de los resultados se obtuvo una interfaz con tres pasos que permite al usuario crear de forma sencilla un gráfico, los tipos de gráficos que el usuario puede crear son los siguientes:

- Gráfico de Pastel
- Gráfico de Barras
- Gráfico de Líneas.

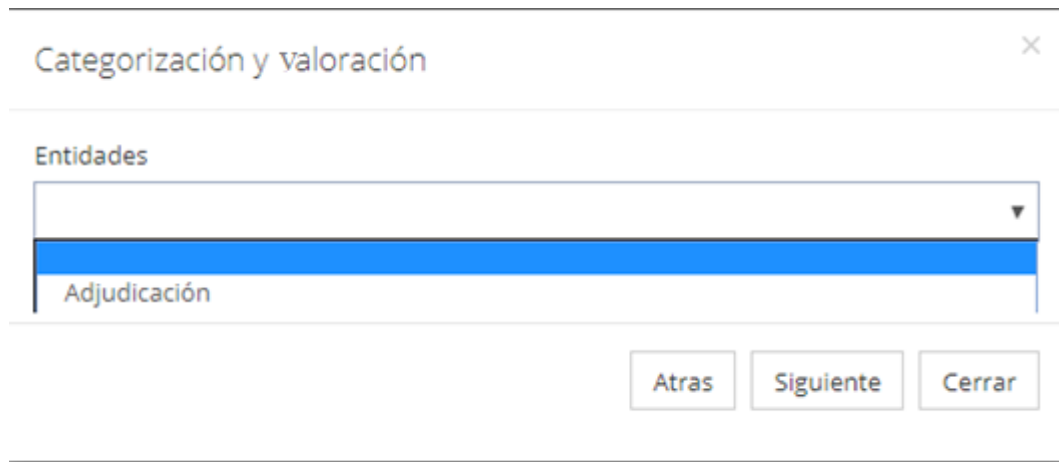
En el primer paso para crear una gráfica el usuario le debe poner un título y seleccionar que tipo de gráfica, de las anteriormente mencionadas, desea crear como se puede observar en el en la “Figura 10”. Solo hasta que el usuario completa los campos que se le solicitan en esta vista puede pasar a la siguiente

Figura 17 Primera vista del editor para crear una gráfica.

La imagen muestra una ventana de diálogo con el título "Creando gráfico" y un botón de cierre "X" en la esquina superior derecha. Dentro de la ventana, hay un campo de texto etiquetado "Titulo" que contiene el texto "Monto total de adjudicaciones por departamento del comprador". Debajo de esto, hay un menú desplegable etiquetado "Tipo de Gráfico" que muestra tres opciones: "Gráfico de Pie", "Gráfico de Barras" y "Gráfico de Líneas". La opción "Gráfico de Pie" está seleccionada y resaltada en azul. En la parte inferior derecha de la ventana, hay dos botones: "Siguiete" (con un error ortográfico) y "Cerrar".

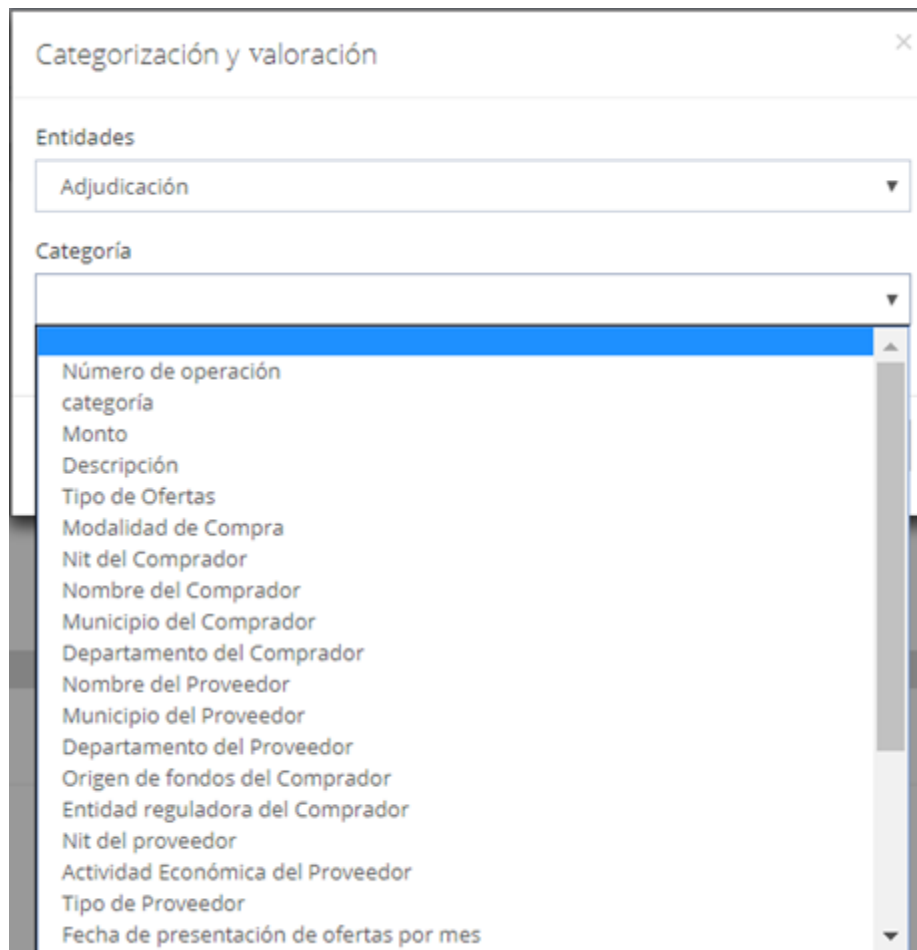
En el segundo paso para crear una gráfica el usuario debe seleccionar la entidad sobre la cual desea ver los campos disponibles. Todos los metadatos que el usuario puede seleccionar son obtenidos de la base de datos “udvcomgt_gobhash”. Como se puede apreciar en la “Figura 11”, los campos que el usuario puede seleccionar para categorizar dependen de la entidad (*materialized view*) que seleccione. El módulo de “Diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico”, fue el encargado de definir las *materialized views* que existen y las columnas que pertenecen a cada uno de los *materialized views*. La columna que el usuario elija para categorizar implica un agrupamiento en la consulta que se forma para desplegar datos. En la “Figura 12” se ve que las columnas aparecen dependiendo de la identidad, lo que quiere decir que por cada valor distinto que exista en la base de datos en el campo que el usuario seleccionó para categorizar representará un posible valor para graficar. Luego el usuario debe elegir la columna para calcular, como se ve en “Figura 13”, entonces se procede a elegir la operación que se aplicará a la columna. Están disponibles para las columnas de tipo fecha y texto la operación “Contar”, y para las columnas numéricas, ya sean enteros o con punto decimal las operaciones “Suma” y “Promedio” en la “Figura 14” se logra apreciar el tipo de operaciones para una columna, en la consulta, esto determina el campo por el cual se aplica la operación en el “Select” por ejemplo “SUM(monto)” o “Count(categoría)”. Cuando el usuario haya completado todos los campos requeridos en este paso puede pasar al siguiente.

Figura 18 Segunda vista del editor para crear una gráfica seleccionado entidades.



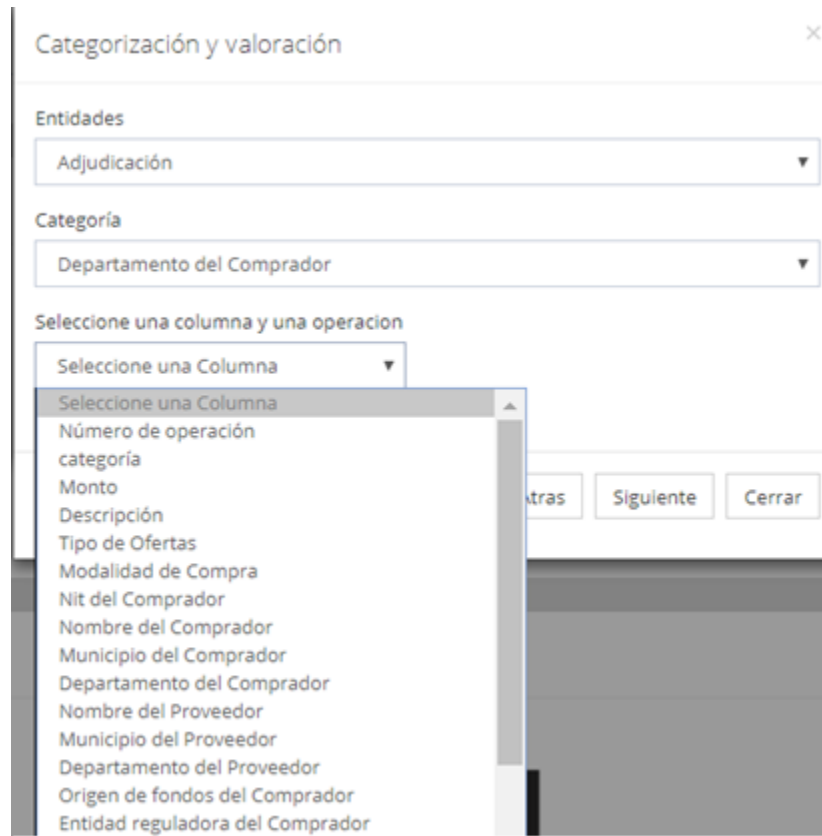
The image shows a software window titled "Categorización y valoración" with a close button (X) in the top right corner. Below the title bar, there is a section labeled "Entidades" containing a dropdown menu. The menu is open, and the option "Adjudicación" is highlighted in blue. At the bottom of the window, there are three buttons: "Atras", "Siguiete", and "Cerrar".

Figura 19 Segunda vista del editor para crear una gráfica, campos disponibles por entidad.



The image shows the same software window "Categorización y valoración". In this view, the "Entidades" dropdown menu is set to "Adjudicación". Below it, the "Categoría" dropdown menu is open, displaying a list of available fields. The first item in the list, "Número de operación", is highlighted in blue. The list includes the following items: "Número de operación", "categoría", "Monto", "Descripción", "Tipo de Ofertas", "Modalidad de Compra", "Nit del Comprador", "Nombre del Comprador", "Municipio del Comprador", "Departamento del Comprador", "Nombre del Proveedor", "Municipio del Proveedor", "Departamento del Proveedor", "Origen de fondos del Comprador", "Entidad reguladora del Comprador", "Nit del proveedor", "Actividad Económica del Proveedor", "Tipo de Proveedor", and "Fecha de presentación de ofertas por mes".

Figura 20 Segunda vista del editor para crear una gráfica, seleccionar una columna por entidad

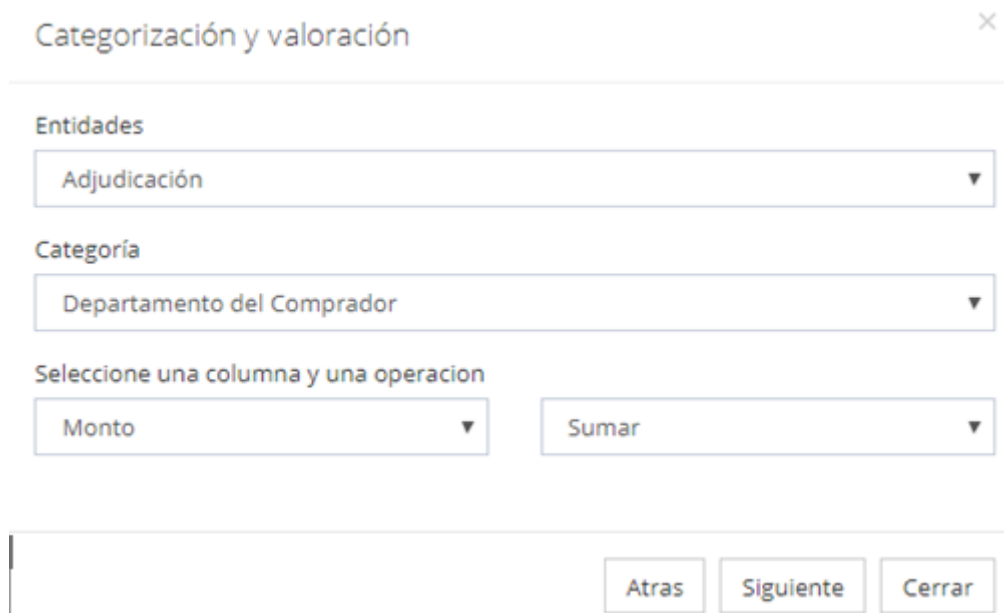


The screenshot shows a web interface titled "Categorización y valoración" with a close button (X) in the top right corner. It contains three main sections:

- Entidades:** A dropdown menu with "Adjudicación" selected.
- Categoría:** A dropdown menu with "Departamento del Comprador" selected.
- Seleccione una columna y una operación:** This section has two dropdown menus. The first, "Seleccione una Columna", is open, displaying a list of options: "Seleccione una Columna", "Número de operación", "categoría", "Monto", "Descripción", "Tipo de Ofertas", "Modalidad de Compra", "Nit del Comprador", "Nombre del Comprador", "Municipio del Comprador", "Departamento del Comprador", "Nombre del Proveedor", "Municipio del Proveedor", "Departamento del Proveedor", "Origen de fondos del Comprador", and "Entidad reguladora del Comprador". The second dropdown menu, "Seleccione una operación", is currently empty.

At the bottom right of the form, there are three buttons: "Atras", "Siguiete", and "Cerrar".

Figura 21 Segunda vista del editor para crear una gráfica, todos los campos completados



The screenshot shows the same "Categorización y valoración" interface, but now all fields are filled out:

- Entidades:** "Adjudicación" (dropdown)
- Categoría:** "Departamento del Comprador" (dropdown)
- Seleccione una columna y una operación:** "Monto" (dropdown) and "Sumar" (dropdown)

At the bottom right, the buttons "Atras", "Siguiete", and "Cerrar" are visible.

En el tercer paso para crear una gráfica el usuario puede aplicar filtros. Están disponibles dos clases de filtros, los filtros por cualquier campo de tipo numérico, con o sin punto decimal, o texto, y los filtros por campos de tipo fecha los cuales se aplican en un apartado diferente como se aprecia en la “Figura 15” y en la “Figura 16”.

Figura 22 Vista para agregar filtros normales y filtros por fecha

The screenshot shows a dialog box titled "Seleccionar filtros" with a close button (X) in the top right corner. Below the title bar, there is a section labeled "Filtros" with a blue plus icon. Underneath, there are two rows of filter controls. The first row has a dropdown menu set to "Monto", a dropdown menu set to "M€", a text input field containing "1200.01", and a blue minus icon. The second row has a dropdown menu set to "Departamento del", a dropdown menu set to "Co", a yellow text input field containing "s", and a blue minus icon. Below these rows is a section labeled "Filtro por Fecha" with a blue plus icon. At the bottom of the dialog, there are four buttons: "Atrás", "Cerrar", "Previsualización", and "Guardar".

Figura 23 Vista con algunos filtros por fecha agregados

This screenshot is similar to the previous one but includes an additional filter. Under the "Filtro por Fecha" section, there is a dropdown menu set to "Fecha de", a dropdown menu set to "Últ", a calendar icon, a text input field containing "03/10/", and a blue minus icon. The rest of the interface, including the "Filtros" section and the bottom buttons, remains the same as in Figure 22.

Todos los metadatos que el usuario va seleccionando durante el proceso de crear un gráfico se van almacenando en memoria formando algo que se conoce como “definición” de un gráfico, esta definición representa cada uno de los valores de los metadatos que el usuario seleccionó en la interfaz de usuario. Esta “definición” es un objeto json, toda la información estructurada que forma parte del json se puede ver en la “Figura 17”, toda esta información dentro de la “definición” es vital para la implementación, ya que permite construir una consulta en SQL en base a lo que el usuario seleccionó en la interfaz de usuario, una gran parte del proceso de implementación se basa en traducir la “definición” a una consulta que se puede ejecutar a

través del ORM en PostgreSQL.

Figura 24 “definición” de un gráfico en formato Json

```

{
  "baseColumn": {
    "base_table": "fact_adjudicacion",
    "createdAt": "2017-08-08T06:00:00.000Z",
    "date_column": "ano",
    "display_name": "Adjudicaciones por año",
    "entity_id": 2,
    "id": 13,
    "name": "fecha_adjudicada",
    "second_table": "dim_fecha",
    "type": 2,
    "updatedAt": "2017-08-08T06:00:00.000Z"
  },
  "category": {
    "column": {
      "base_table": "fact_adjudicacion",
      "createdAt": "2017-08-08T06:0000.000Z",
      "date_column": null,
      "display_name": "Monto",
      "entity_id": 2,
      "id": 7,
      "name": "monto",
      "second_table": "",
      "type": 4,
      "updatedAt": "2017-08-08T06:00:00.000Z",
    },
    "operation": {
      "createdAt": "2017-08-08T06:00:00.000Z",
      "id": 1,
      "name": "Sumar",
      "type": 2,
      "updatedAt": "2017-08-08T06:00:00.000Z",
      "value_type": 2,
    }
  },
}

```

```

"dateFilters": [
  {
    "column": {
      "base_table": "fact_adjudicacion",
      "createdAt": "2017-08-08T06:00:00.000Z",
      "date_column": "mes_nombre",
      "display_name": "Adjudicaciones por mes",
      "entity_id": 2,
      "id": 12,
      "name": "fecha_adjudicada",
      "second_table": "dim_fecha",
      "type": 2,
      "updatedAt": "2017-08-08T06:0000.000Z",
    },
    "date1": "1/1/2017",
    "date2": null,
    "operation": {
      "createdAt": "2017-08-08T06:0000.000Z",
      "id": 18,
      "name": "Último mes",
      "type": 1,
      "updatedAt": "2017-08-08T06:0000.000Z",
      "value_type": 5,
    }
  }
],
"entity": {
  "createdAt": "2017-08-08T06:0000.000Z",
  "display_name": "Adjudicación",
  "id": 2,
  "name": "fact_adjudicacion",
  "updatedAt": "2017-08-08T06:0000000Z",
  "visible": "true"
},

```

```

"filters": [
  {
    "column": {
      "base_table": "fact_adjudicacion",
      "createdAt": "2017-08-08T06:00:00.000Z",
      "date_column": null,
      "display_name": "Monto",
      "entity_id": 2,
      "id": 7,
      "name": "monto",
      "second_table": "",
      "type": 4,
      "updatedAt": "2017-08-08T06:00:00.000Z",
    },
    "operation": {
      "createdAt": "2017-08-08T06:00:00.000Z",
      "id": 15,
      "name": "Mayor que",
      "type": 1,
      "updatedAt": "2017-08-08T06:00:00.000Z",
      "value_type": 4
    },
    "value": "1200.01"
  },
  {
    "column": { ... },
    "operation": { ... },
    "value": "S"
  }
],
"name": "Test",
"widgetType": {
  "id": 1,
  "name": "Gráfico de Pie"
}
}

```

Para transformar la “definición” del gráfico a una consulta que se puede ejecutar mediante el ORM en PostgreSQL se utilizó una librería llamada “Squel.js” la cual permite transformar texto simple a una consulta sin que se ejecute, únicamente genera el texto pero en la sintaxis de SQL, esto permite poder tomar todos los metadatos y mediante un algoritmo pasar de metadatos que están en json a una consulta en texto con la sintaxis de SQL, “Squel.js” se utilizó en el *backend*, ya que por medio de una petición desde Angular en el

frontend es enviada la “definición” del gráfico a un API “endpoint”, el cual procesa la llamada y mediante un algoritmo que fue implementado, verifica que los metadatos que recibe está en el formato que se requiere y construye la consulta por partes.

Luego de que la “definición” ha pasado por el proceso de validación, es extraída la información que el usuario provee, por ejemplo si el usuario no puso ninguna clase de filtros, la propiedad “filters” dentro de json de la “definición” tiene un valor “null” lo mismo que para el campo “dateFilters”, también se extrae la información con la que la consulta se irá formando, por ejemplo la columna sobre la cual se aplicará la agrupación en la consulta, o si son necesarios “joins” dentro de la consulta para obtener datos de los campos que el usuario seleccionó en la interfaz de usuario, etc.

Cuando toda la información que está en la “definición” ha sido extraída se da inicio a la construcción de la consulta, que es manejada por un algoritmo que se divide en tres pasos, el primero de ellos es la construcción de la base de la consulta. Esto quiere decir que se realizan los “joins” necesarios para obtener los datos que el usuario solicita, el segundo es la aplicación de los filtros a la consulta, primero se aplican todos los filtros que no tengan nada que ver con fechas, y por último se aplican los filtros que son en base a fechas. Por último y como tercer paso se une la consulta que se generó en los pasos anteriores a una consulta general donde se aplica la operación sobre la columna que el usuario seleccionó para categorizar y se aplica la agrupación sobre toda la consulta según el campo que el usuario haya seleccionado en la interfaz de usuario. Sqel.JS permite obtener el texto que representa esa consulta, como se puede observar en la “Figura 18”, y por medio de una “row query” ejecutarlo mediante Sequelize.JS que es el ORM que se utilizó. El “endpoint” encargado de procesar, crear y correr la consulta, retorna una lista con dos objetos, el primero es un conjunto de datos de la forma “clave”:”valor”, y el segundo objeto contiene información sobre la consulta final y sobre los campos que están en el primer objeto, además incluye también un conjunto de datos idéntico al del primer objeto en la lista, en la “Figura 19” se puede apreciar toda la información descrita con anterioridad.

Figura 25 Ejemplo del texto generado por Sqel.JS

```
SELECT SUM(q.monto) as value, q.departamento_comprador FROM (SELECT DISTINCT a.nit_proveedor,
a.nog, a.monto, a.departamento_comprador FROM consolidated_table `a` INNER JOIN dim_fecha `b`
ON (b.id_fecha = a.fecha_adjudicada) WHERE (a.departamento_comprador like '%S%') AND (a.monto <
1200.01)) `q` GROUP BY q.departamento_comprador
```

Figura 26 Lista de datos que retorna el “endpoint” luego construir y ejecutar la consulta

```

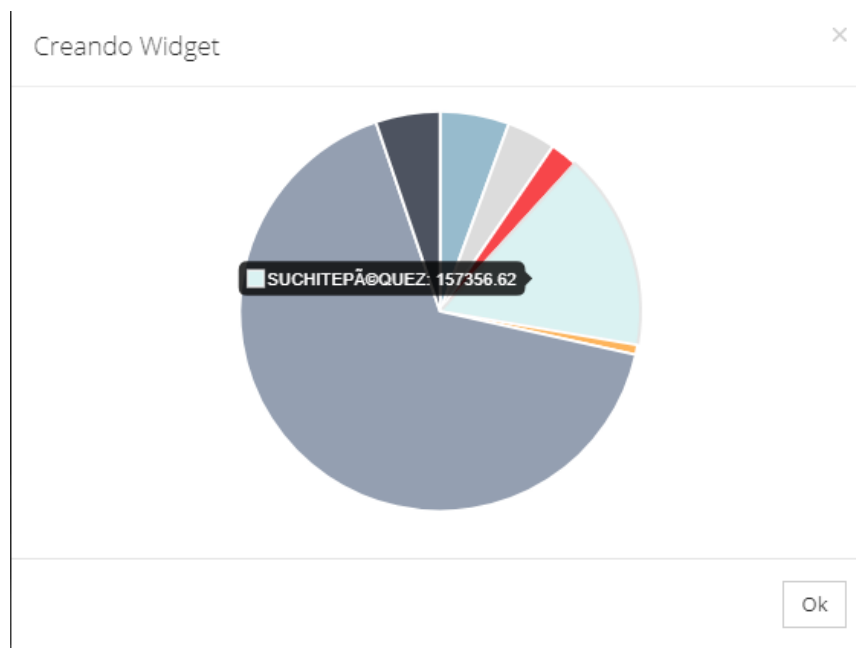
▼ (2) [Array(7), {...}] ⓘ
  ▼ 0: Array(7)
    ▶ 0: {value: 55121.07, departamento_comprador: "SANTA ROSA"}
    ▶ 1: {value: 39000.62, departamento_comprador: "SAN MARCOS"}
    ▶ 2: {value: 21920.38, departamento_comprador: "EL PROGRESO"}
    ▶ 3: {value: 157356.62, departamento_comprador: "SUCHITEPÃ@QUEZ"}
    ▶ 4: {value: 7825.64, departamento_comprador: "SOLOLÃ;"}
    ▶ 5: {value: 654891.01, departamento_comprador: "ESCUINTLA"}
    ▶ 6: {value: 51305.39, departamento_comprador: "SACATEPÃ@QUEZ"}
    length: 7
    ▶ __proto__: Array(0)
  ▼ 1:
    command: "SELECT"
    ▼ fields: Array(2)
      ▼ 0:
        columnID: 0
        dataTypeID: 701
        dataTypeModifier: -1
        dataTypeSize: 8
        format: "text"
        name: "value"
        tableID: 0
        ▶ __proto__: Object
      ▼ 1:
        columnID: 21
        dataTypeID: 25
        dataTypeModifier: -1
        dataTypeSize: -1
        format: "text"
        name: "departamento_comprador"
        tableID: 16541
        ▶ __proto__: Object
    length: 2
    ▶ __proto__: Array(0)
    oid: null
    rowAsArray: false
    rowCount: 7
    ▼ rows: Array(7)
      ▶ 0: {value: 55121.07, departamento_comprador: "SANTA ROSA"}
      ▶ 1: {value: 39000.62, departamento_comprador: "SAN MARCOS"}
      ▶ 2: {value: 21920.38, departamento_comprador: "EL PROGRESO"}
      ▶ 3: {value: 157356.62, departamento_comprador: "SUCHITEPÃ@QUEZ"}
      ▶ 4: {value: 7825.64, departamento_comprador: "SOLOLÃ;"}
      ▶ 5: {value: 654891.01, departamento_comprador: "ESCUINTLA"}
      ▶ 6: {value: 51305.39, departamento_comprador: "SACATEPÃ@QUEZ"}
      length: 7
      ▶ __proto__: Array(0)
    ▶ _parsers: (2) [null, null]
    ▶ __proto__: Object
    length: 2

```

Los botones que accionan la llamada al “endpoint” que procesa y ejecuta la consulta están en la tercera vista de la interfaz de usuario, como se aprecia en la “Figura No.14” o en la “Figura 15”. Si es presionado el botón para generar una previsualización, entonces, es ejecutado un método para dar el formato

necesario a los datos que se van a desplegar en el gráfico, el formato requerido es una lista con todos los valores que podrán ser vistos, el gráfico y una lista con la etiqueta que corresponde a cada valor, como resultado se puede apreciar una previsualización con datos que reflejan la consulta que se forma en base a lo que el usuario seleccionó en los pasos previos para crear un gráfico que se puede apreciar en la “Figura 20”.

Figura 27 Ejemplo de previsualización de un gráfico



Por último, si el usuario está satisfecho con la gráfica que generó, la puede adjuntar a su publicación pulsando el botón “Guardar” que se encuentra en la vista para aplicar filtros como se observa en la “Figura 15”, luego de adjuntarlo a la publicación el gráfico se puede observar en la vista principal donde se construye una publicación. Como se puede apreciar en la “Figura 21”, el proceso de desplegar el gráfico dentro de la vista para construir una publicación es parte de la integración con el módulo de “Diseño, Análisis y Desarrollo de *Frontend*”.

Figura 28 Vista para crear una publicación

The image shows a web interface for creating a publication. It consists of three main sections:

- Titulo y Descripción:** This section has a title bar and an orange 'Crear' button. It contains two text input fields. The first is labeled 'Titulo' and contains the text 'Prueba'. The second is labeled 'Descripción' and contains the text 'Esta es una prueba para crear una publicación'.
- Gráfica Principal:** This section displays a single, solid blue pie chart.
- Gráfica 1:** This section displays a multi-colored pie chart with several segments in shades of blue, red, green, and grey.

D. DISCUSIÓN

1. **Tecnología para *backend*.** Para el *backend* se utilizó la tecnología de Node JS (versión 8) sobre el *framework* de Express (versión 5). El ORM a fue Sequelize que soporta PostgreSQL, MySQL, SQLite y MSSQL. La base de datos utilizada de SQL fue PostgreSQL y la base de datos NoSQL fue MongoDB. Se escogió la tecnología de Node JS por su compatibilidad con bases de datos SQL y NoSQL, ya que mantiene una sintaxis muy similar para el ODM y el ORM. Node JS utiliza JavaScript como lenguaje de programación lo que hace más fácil su entendimiento con los módulos que se dedican al *frontend* ya que también están basados en JavaScript. Se utilizó la nueva especificación de JavaScript ES6 para tener una sintaxis más actualizada, funcional y estandarizada junto con la especificación ES7 que utilizó *async/await* para eliminar el fenómeno llamado *callback hell*, el cual consiste en que una función tiene funciones anidadas como parámetros. NodeJS tiene un ambiente asíncrono lo que permite ejecutar tareas en forma no secuencial sino simultánea y también tiene una operación *non-blocking* que permite ejecutar tareas sin que se hayan terminado otras que puedan bloquear ciertas operaciones. Por eso se dice que NodeJS con express a nivel de rendimiento es mayor que las otras tecnologías que existen hoy en mercado como se puede apreciar en la comparación entre tecnologías en la “Tabla 3”

En esta tabla se consideraron métricas relevantes al proyecto. Otra de las razones por la cual se escogió NodeJS con express es por su flexibilidad ya que no tiene dependencias que restrinjan la arquitectura con la que se debe trabajar. Esto permite tener un nivel de personalización profunda porque se pueden hacer cambios sobre aspectos específicos. Esto no es posible en otros *frameworks* como Django que ya traen incluido el ORM y modificar ciertas partes del *framework* no es posible o es muy complejo. Otras de las consideraciones que se tomaron fue la facilidad para integración y comunicación con el módulo de “Desarrollo e Implementación de *Backend*”.

Tabla 7 Comparación entre frameworks para desarrollo de backend

Aspecto	NodeJS (Express)	Python (Django)	PHP (Symfony)	Go (Iris)
Documentación	Minimalista, conceptos de JavaScript la completan.	Completa	Completa	Solo hay ejemplos de código.
ORM	Bookshelf (PostgreSQL, MySQL, SQLite). [6] Node-ORM(MySQL, MariaDB, PostgreSQL, redshift, SQLite, mongodb) [7] Cammit (SQL and NoSQL) [8] Sequelize (MySQL, PostgreSQL, SQLite, mssql) [10]	Django's ORM	Doctrine (mssql, MySQL, pgsql, SQLite, mongo, Oracle) [9]	GORM [5] (MySQL, SQLite, PostgreSQL, mssql) XORM (MySQL, mssql, SQLite, Oracle, pgsql) [11]
Soporta Socket.IO	Sí	Sí	Hay versiones portadas o sobre NodeJS [1], [2], [3]	Sí [4]
ODM	Mongoose	No existe una versión estable.	Doctrine soporta ODM pero no al 100%.	GoBongo

cont. Tabla 7

Aspecto	NodeJS (Express)	Python (Django)	PHP (Symfony)	Go (Iris)
Clasificación según performance.	2	3	4	1
Sintaxis	Similar a otros módulos. Fácil de leer y compatible a través del equipo.	Identación obligatoria. No compatible con el módulo de <i>frontend</i> .	Acceso de métodos de forma diferente. No compatible con el módulo de <i>frontend</i> .	Básica a nivel de alto nivel. Es necesario especificar las estructuras a usar. Parecida a C.

2. Tecnología para *Frontend*. Para la implementación del *frontend* fue escogida la tecnología AngularJS, la cual es un *framework* basado en el lenguaje JavaScript, AngularJS se basa en el patrón MVC (Modelo, Vista, Controlador), con el objetivo de separar las capas de presentación, lógica y componentes de una aplicación. AngularJS no es el único *framework* especialmente diseñado para trabajar *frontend*, pero cada uno tiene sus ventajas y desventajas y está enfocado en satisfacer un objetivo específico, por ejemplo, algunos se enfocan en la seguridad informática, algunos en el rendimiento, la escalabilidad, etc. así mismo varían en complejidad, usabilidad, documentación disponible, curva de aprendizaje, etc. Los *frameworks* comparados para ser usados en este módulo, persiguen los siguientes criterios:

- Seguridad informática
- Escalabilidad
- Compatibilidad
- Mantenimiento

Se llegó a la conclusión basado en los criterios especificados con anterioridad que los mejores tres *frameworks* que cumplen con las características ya explicada, era AngularJS, ReactJS y EmberJS luego de comparar estos *frameworks* bajo los siguientes indicadores.

Comunidad

- Soporte
- Curva de aprendizaje

Se determinó que el uso de AngularJS era el más indicado para el manejo del *frontend* en el presente módulo. Ya que al ser un *framework* diseñado, por ingenieros de Google la comunidad que lo ha adoptado es muy grande incluso es uno de los más usados en el presente año. Lo mismo para con el soporte, al Google utilizarlo para brindar soporte al *frontend* de algunas de sus plataformas AngularJS cuenta con un soporte excelente. Por último, la curva de aprendizaje es uno de los factores que más peso tiene en la elección, ya que en comparación con los otros dos *frameworks* la curva de aprendizaje de AngularJS es baja. Por último, uno de los factores que también pesó a favor de la selección de AngularJS es la compatibilidad y la extremada

facilidad con la que se pueden integrar dos aplicaciones que se trabajan por separado, esto es vital ya este módulo y el módulo de “Diseño, Análisis y Desarrollo de *Frontend*” se desarrollaron por separado.

Tabla 8 Evaluación de frameworks según indicadores

Indicadores	AngularJS	ReactJS	EmberJS
Comunidad	*****	***	***
Soporte	*****	*****	*****
Curva de Aprendizaje	**	*****	**

3. **Librería para creación de gráficos.** Para visualizar los gráficos se utilizó una librería diseñada específicamente para la visualización de gráficos en AngularJS la cual se llama “Angular-Chart.js”, ahora bien, existen una gran cantidad de librerías para el manejo de gráficos con AngularJS, pero cada una varía en complejidad, usabilidad, documentación disponible, etc. se llegó a la conclusión que el uso de “Angular-Chart.js” era el más adecuado para poder implementarlo en el proyecto luego de que se realizara una comparación entre varias librerías que brinda soporte para el manejo de gráficas en AngularJS. Los indicadores que se utilizaron para llegar a tomar una decisión fueron los siguientes:

- Usabilidad
- Manejo e integración con AngularJS
- Documentación
- Rango funcional
- Uso técnico.

En cuanto a la usabilidad la librería mostró ser realmente muy usable, ya que el formato de los datos que serán graficados, la forma de renderizar el gráfico en cualquier página HTML y la simplicidad con la que se puede ser usada y comprendida realmente facilita su implementación. Por otro lado, la librería fue especialmente diseñada para ser usada dentro de un proyecto de AngularJS por lo tanto la implementación no requiere más que instalar las dependencias, y la forma en la que los datos son insertados en la gráfica que se renderiza en la página HTML; es siguiendo los patrones de AngularJS para el manejo de datos que se quieren compartir entre una página HTML y un controlador.

La documentación y los ejemplos de implementaciones fueron otros de los factores que pesaron mucho en la decisión, ya que Angular-Char.js cuenta con amplia documentación y ejemplos de su correcta implementación.

El rango funcional fue incluso mejor que el esperado, ya que para el desarrollo de este módulo solo se esperaba poder crear por lo menos tres tipos diferentes de gráficos, y Angular-Chart.js permite crear más de

ocho tipos diferentes de gráficos, los cuales son interactivos, ya que cuando el “mouse” para por encima de alguna parte del gráfico este tiene alguna animación cómo dar detalles sobre el valor que representa lo que se está apuntando, y esta capacidad era una de las cosas que se esperaba, la librería que fuera a ser utilizada, permitiera.

Realmente es muy sencillo entender cómo funciona Angular-Chart.js incluso para crear gráficas personalizadas, por ejemplo, especificar ciertos colores o con alguna escala en especial, etc. El uso técnico si bien no es la mejor característica de esta librería tampoco es muy malo. Al comprar todas estas las características con otras librerías como se puede apreciar en la “Tabla 5” es claro que la librería Angular.Chart.js es la mejor opción para el manejo de gráficos en AngularJS.

Tabla 9 Comparación entre librerías para el manejo de gráficas en AngularJS

Librería	Rango funcional	Uso técnico	Usabilidad	Manejo e integración con AngularJS	Documentación
d3.js	*****	*****	*	*	*****
NVD3	****	****	*	*	****
Angular-ndv3-directives	***	**	****	****	***
Angular-nvd3	****	**	*****	*****	***
Angular- Chart	*****	****	*****	*****	****

4. **Implementación.** El proceso para crear un gráfico se espera que sea simple y hasta cierto punto intuitivo, esto por el hecho de que la plataforma GobHash será utilizada por cualquier persona y para crear una gráfica o una publicación no se espera que la persona tenga un amplio manejo sobre estadística o economía, sin embargo sí se esperaría que las personas que vayan a crear un gráfico, no tanto las personas que se registren en GobHash, sepan o tengan una noción de términos básicos que se manejan en cuanto a adquisiciones del estado, ya que los campos que se presentan en los metadatos para armar gráficos están Relacionados con conceptos que van de la mano con adjudicaciones y los datos que se manejan en el portal de Guatecompras.

El hecho de que solo sean tres pasos para crear un gráfico fue establecido para que no sea un proceso engorroso y que demore mucho tiempo, los campos que se presentan al usuario para que sean completados por el fueron seleccionados estratégicamente en cada una de las vistas, por eso es que en el primer paso debe elegir un título para su gráfico y luego seleccionar que tipo de gráfico desea crear, esto con el fin de que tenga ya en mente lo que espera ver en su gráfico o por lo menos una vaga noción de lo que espera, para que de esta forma se haga una idea que campos seleccionar en los próximos pasos para poder llegar a lo que desea.

En el segundo paso el usuario debe seleccionar la información que le interesa ver, por eso el primer paso es elegir una entidad, para que se establezca el hecho sobre el que le gustaría saber. Luego debe elegir una columna la cual es de vital importancia porque será sobre la que se agruparán los datos sobre la que se categorizan y aquellos valores diferentes entre sí que pertenezcan a esa categoría de datos serán los que tendrán una representación visual dentro del gráfico. Por último, el usuario selecciona sobre qué columna quiere operar y que operación quiere aplicar, estos pasos se encuentran sobre una misma vista ya que van de la mano y prácticamente es lo que define lo que al final el usuario verá en la gráfica y al seleccionar tantos campos al usuario se le da una noción de la importancia que tienen los mismos.

Por último, el usuario puede o no agregar filtros, y esta parte de los filtros se dividió en dos para que el usuario tuviera más claridad sobre lo que podría hacer con los filtros y lograba diferenciar de manera sencilla como aplicar filtros por fecha. en la misma vista se encuentran los botones para realizar una previsualización, y el botón para guardar. El botón para la previsualización aparece primero y es un poco más grande para dar al usuario la idea de que primero debería ver una previsualización y así asegurarse de estar creando lo que necesita y en un principio quería ver.

Es importante mencionar que específicamente sobre este módulo no fueron realizados diseños, prototipos ni test de usabilidad, ya que las opiniones sobre diversas plataformas que se utilizan en la actualidad para realizar esta clase de actividades, tales como Tableau, Sisense, Power Bi, etc. son muy variadas y muy susceptibles a la capacidad de los usuarios de crear gráficas y de su manejo de términos de estadística, esto sumado a que la plataforma está pensada para que cualquiera la pueda utilizar, los resultados de los diseños, y los test de usabilidad podrían llevar a no tener ninguna conclusión en particular, porque algunos usuarios podrían estar satisfechos con la interfaz de usuario para crear un gráfico y otros que quizá no están familiarizados con la estadística o con los datos que se manejan en Guatecompras podrían estar en desacuerdo con la interfaz de usuario.

Por otro lado en las pruebas de usabilidad aquellos usuarios que tengan una noción del significado de los datos que se tratan en Guatecompras y con conocimientos básicos de estadística podrían lograr crear gráficos con algún valor en términos de que reflejaran lo que ellos quisieran ver, pero los usuarios que no sepan nada quizá al momento de aplicar filtros lleguen a un punto donde no existen datos para lo que pretendían ver, o quizá si logran formar una gráfica no tengan ni idea de lo que están viendo. No obstante, se revisaron a detalle plataformas como Tableau y Sisense las cuales, como ya se mencionó, son plataformas que permiten crear gráficos en base a cualquier tipo de datos, para tener una idea y aprovechar la interfaz que estas plataformas luego de tantos años en el mercado han ido perfeccionando.

La “definición” que se construye en base a los metadatos que el usuario va seleccionando en la interfaz de usuario, tiene un formato Json, el cual se ha convertido en uno de los más usados hoy en día, se

trabaja con este formato para aprovechar la facilidad con la que se puede manipular esta clase de objetos con JavaScript, así mismo para facilitar la integración con los módulos de “Desarrollo e Implementación de *Backend*” y “Diseño, Análisis e Implementación de *Frontend*” ya que en el caso de módulo de “Desarrollo e Implementación de *Backend*” la “definición” se guarda en una base de datos MongoDB, y que la “definición” tenga un formato Json facilita este proceso. Por otro lado, todos los metadatos provienen de consultas que se realizan desde Angular hacia los *endpoints* en el *backend* y los metadatos que provienen de los endpoints tiene un formato json, todos esos metadatos como ya se especificó son guardados en memoria dentro de una “definición” la cual tiene un formato json, lo que facilita agregar estos metadatos a la “definición”. Al construir la consulta que se corre en PostgreSQL que la “definición” que es enviada al *backend* tenga un formato json es de gran ayuda para poder acceder a todos los metadatos que formar parte de la “definición”.

5. ¿Por qué la elección de los tipos de gráficos?

Los tres tipos de gráficas que se permiten crear son los siguientes:

- Gráfico de pastel
- Gráfico de barras
- Gráfico de líneas.

Se puede crear únicamente tres tipos de gráficas por limitaciones de tiempo. Estos tres tipos de gráficos que se pueden crear no fueron seleccionados al azar, uno de los factores que fue primordial para considerar el uso de estos tres tipos de gráficas es el hecho de que la plataforma será utilizada por cualquier tipo de usuario sin ningún requerimiento como conocimientos sobre estadística. Esto es algo clave porque no sería aconsejable usar gráficas sobre las cuales es difícil llegar a conclusiones, y por esta razón se optó por una gama de gráficos de los más prácticos y conocidos. De estos tres tipos de gráficos se puede extraer mucha información y son hasta cierto punto intuitivos.

Así mismo luego de hacer el análisis de datos con los que se contaba, se llegó a la conclusión que una de las mejores formas de representar los datos es con Gráficos de tipo pastel y con Gráficos de tipo Barras. Dentro de los datos con los que se cuentan provenientes del portal de Guatecompras existen varios campos que tratan con fechas, y las gráficas de líneas son adecuadas para representar este tipo de datos

VII. Extracción de datos de adjudicaciones, proveedores y compradores del sistema Guatecompras

A. METODOLOGÍA

1. Comprender el funcionamiento de Guatecompras. El primer paso para extraer la información de cualquier sitio de internet, es tener claro el flujo que se necesita para acceder a esta. Es por eso que en esta fase el objetivo era establecer la serie de pasos que permiten acceder a la información. Inicialmente se trabajó con la herramienta de búsqueda de adjudicaciones de Guatecompras (en adelante ‘el sitio’) con el objetivo de determinar las limitaciones que esta presenta al momento de utilizarla.

La opción 1 presenta una lista de todos los concursos con estatus de adjudicado terminado hasta el día en que se hace la consulta, a pesar de brindar la información solicitada sin mayores complicaciones, presenta información que no siempre es útil, i.e. al momento de querer buscar las adjudicaciones hechas en determinado periodo de tiempo hay que pasar primero por las más recientes y eso implica una carga innecesaria tanto para el servidor web del sitio como para el programa que se quiere diseñar. Es por esta razón que este componente de la herramienta de búsqueda fue descartado.

Teniendo en cuenta el requerimiento de obtener únicamente la información para el periodo de tiempo relevante para la búsqueda se encontró que las opciones del 2 al 4 no permiten este tipo de filtrado y fueron descartadas porque simplemente ofrecen un subconjunto de la información de la opción 1.

Al llegar a la opción 5 se pudo notar que ofrece la posibilidad efectuar búsquedas con los parámetros de las opciones 1 a la 3 y además en un período de tiempo específico lo cual es valioso para minimizar la carga de trabajo. En la se aprecian los distintos tipos de filtro que se permiten aplicar. El hecho de poder acceder un subconjunto específico de adjudicaciones hechas en un periodo de tiempo determinado reduce la carga de manera considerable.

La opción 6 fue descartada debido a que se necesita conocer el NIT de cada proveedor adjudicado durante el periodo de tiempo necesario y aun así esta opción muestra todas las adjudicaciones hechas al proveedor durante toda la historia del sitio, con lo que se cae en una situación similar a la de la opción 1.

Figura 29. Detalle de la opción 5 para búsquedas

Buscar Adjudicaciones

GUATECOMPRAS contiene todas las adjudicaciones por concursos efectuadas, tanto a proveedores con NIT como a proveedores extranjeros sin NIT, desde el 19.Jul.2004, fecha en que comenzó a funcionar el registro de adjudicaciones.

Por favor elija una opción:

- Opción 1: Buscar TODAS las adjudicaciones
- Opción 2: Buscar adjudicaciones a PROVEEDORES con NIT
- Opción 3: Buscar adjudicaciones a EXTRANJEROS sin NIT
- Opción 4: Buscar adjudicaciones a proveedores INHABILITADOS
- Opción 5: Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor
- Opción 6: Buscar por NIT del proveedor
- Opción 7: Buscar por NOG del concurso
- Opción 8: Buscar adjudicaciones con contratos de infraestructura con recursos públicos
- Opción 9: Buscar adjudicaciones por Entidad compradora, segmentadas por modalidad **inactivo**

Opción 5: Fecha de adjudicación, monto adjudicado o tipo de proveedor (seleccione una o más opciones)

(Fecha) Del Al

(Monto) De A

Escrito sin comas, puede usar símbolo decimal.(Ejemplo: 12500.50)

Tipo de proveedor

Con NIT Sin NIT

Para la opción 7 se vuelve a caer en lo mismo que la anterior solo que este caso en vez de NIT, hay que conocer el Número de operación de Guatecompras (NOG) que identifica a cada transacción de manera individual dentro del sitio. Fue descartada por presentar las mismas limitaciones que la opción 1. La opción 8 cuenta con un filtro para fechas al igual que la opción 5, pero presenta solamente los contratos de infraestructura. Es por esto que no se utilizó. Hay que resaltar que la opción 9 no estaba disponible al inicio del proyecto, razón por la cual no forma parte de este análisis.

Finalmente teniendo en consideración las ventajas y desventajas que presenta cada una se optó por utilizar la opción 5 para iniciar el flujo de búsqueda dentro del sitio. El siguiente paso en la es la información Relacionada con las adjudicaciones, obtenidas con la opción seleccionada, que se muestra en la . Cabe mencionar que para todas las búsquedas futuras se limitan a proveedores con NIT, esto se debe a que este se considera el identificador único de estos y en el caso de los extranjeros no se puede determinar una manera de hacerlo.

De la Figura 30 es claro que los resultados tienen dos links, el primero (de izq. a der.) sirve para acceder a la información del proveedor ganador de la adjudicación, mientras que el segundo es utilizado para consultar el proceso así como la información de dicha adjudicación. Otra consideración que se tuvo fue con respecto a la paginación, las solicitudes de las páginas con un número mayor al 10 no se hacen con su número, sino que se utiliza un algoritmo que le asigna un número relativo entre 1 y 11 para hacer la petición al sitio de la página correspondiente.

Figura 30. Resultados de la búsqueda con opción 5

Fecha de adjudicación	Nombre o razón social	NIT o país	Monto (Q.)	NOG
14.jul.2016	QUEVARA,GONZALEZ,ELMER,AUGUSTO	12494798	230,386.80	4742680
14.jul.2016	AGUILAR,RUIZ,HELENE,RAQUEL	53013077	334,123.20	47418526
14.jul.2016	CONSTRUCTORA COLUMBIA SOCIEDAD ANONIMA	69290830	304,871.10	4741852
14.jul.2016	CONSULTORES CIVILES SOCIEDAD ANONIMA	23233225	1,128,800.00	4687353
14.jul.2016	GODINEZ MANSILLA, WILLIAM RAMON	8428689	1,280,269.30	4688413
14.jul.2016	HERNANDEZ DE LA CRUZ, MORELIA, ELIZABETH	38492288	1,160,270.00	4633091
14.jul.2016	GOMEZ LOPEZ, EDGAR, JEREMIAS	42126983	89,390.00	5101506
14.jul.2016	LUX, HERNANDEZ, JOSE	2837765	19,969.00	5062454
14.jul.2016	INDUSTRIA DE PAPEL, SOCIEDAD ANONIMA	25212087	8,973.56	5063264
14.jul.2016	GRUPO CENTROAMERICANO GER, SOCIEDAD ANONIMA	52143759	21,600.00	5032172
14.jul.2016	LIBRERIAS Y PAPELERIAS SCRIBE, SOCIEDAD ANONIMA	66658675	10,112.13	5064783
14.jul.2016	AMIPRO, SOCIEDAD ANONIMA	25903640	3,500.00	5077826
14.jul.2016	LIBRERIA E IMPRENTA VIVIAN SOCIEDAD ANONIMA	4851498	5,444.40	5116643
14.jul.2016	JOSO, SOCIEDAD ANONIMA	25294016	8,365.53	5113296
14.jul.2016	DE LEON MARTINEZ, EVER, ALEXANDER	53597478	50,356.00	5106893
14.jul.2016	AVENTURAS TURISTICAS SOCIEDAD ANONIMA	7550022	1,090.00	5111617
14.jul.2016	SYSTEMS ENTERPRISE, SOCIEDAD ANONIMA	48327581	13,870.25	5119316
14.jul.2016	INDUSTRIA DE PAPEL, SOCIEDAD ANONIMA	25212087	2,617.50	5116333
14.jul.2016	JOSO, SOCIEDAD ANONIMA	25294016	2,778.04	5112105
14.jul.2016	LOPEZ, TZOY, JESUS	41698401	24,035.00	5184238
14.jul.2016	DISTRIBUIDORA GIRON SOCIEDAD ANONIMA	88729508	3,400.00	5077532
14.jul.2016	CORPORACION PETENERA DE TURISMO SOCIEDAD ANONIMA	16896663	34,582.00	5123253
14.jul.2016	NANDI MERCANTIL, SOCIEDAD ANONIMA	37853295	5,472.32	5112443
14.jul.2016	ELECTRONICA PAN AMERICANA SOCIEDAD ANONIMA	5230136	61,084.00	5107830
14.jul.2016	CHAVEZ ALVAREZ, LUIS OTTONIEL	42657415	10,800.00	5115906
14.jul.2016	MUNDITROFEDS, SOCIEDAD ANONIMA	4605586	12,000.00	5081475
14.jul.2016	MUNDITROFEDS, SOCIEDAD ANONIMA	4605586	19,050.00	5081610
14.jul.2016	SUMINISTRO INTERNACIONAL DE MERCADERIAS SOCIEDAD ANONIMA	25397400	32,625.00	5112435
14.jul.2016	MAQUILLACCESORIOS, SOCIEDAD ANONIMA	38626926	4,821.43	5086751
14.jul.2016	VALDEZ GUZMAN, VICTOR GERARDO	4050177	20,982.14	5003310
14.jul.2016	CHAI, CHAI, EDVIN OSVALDO	73117366	106,000.00	5019516
14.jul.2016	ROLDAN YATE, CARLOS GIOVANNI	7165110	105,000.00	5019516
14.jul.2016	IMPRENTA LITOGRAFIA Y EDITORIAL OVER PRINT, SOCIEDAD ANONIMA	46297715	801.00	5097452
14.jul.2016	RICOH DE GUATEMALA, SOCIEDAD ANONIMA	4925343	181,090.83	5023718
14.jul.2016	MAQUILLACCESORIOS, SOCIEDAD ANONIMA	38626926	8,437.50	5066794
14.jul.2016	INDUSTRIAS DE LA RIVA SOCIEDAD ANONIMA	3736598	6,450.00	5091322
14.jul.2016	DISTRIBUIDORA MEDICA, SOCIEDAD ANONIMA	35460180	1,835.00	4998286
14.jul.2016	DE LEON, CIJUENTES, OMAR MOISES	27172961	198,700.00	5052250
14.jul.2016	ISETEC, SOCIEDAD ANONIMA	7110561	26,407.14	5089131
14.jul.2016	PETROLERA GUATEMALTECA SOCIEDAD ANONIMA	1455311	60,087.00	5039738
14.jul.2016	GRUPO TULAN, SOCIEDAD ANONIMA	81251645	4,945.33	5118891
14.jul.2016	DISTRIBUIDORA MEDICA, SOCIEDAD ANONIMA	35460180	2,508.75	4998235
14.jul.2016	CLINICA LONDRES, SOCIEDAD ANONIMA	22930418	32,000.00	5091519
14.jul.2016	PESQUEROS DE GUATEMALA, SOCIEDAD ANONIMA	35733926	37,800.00	5111013
14.jul.2016	DISTRIBUIDORA MEDICA, SOCIEDAD ANONIMA	35460180	1,554.14	4985966
14.jul.2016	DISTRIBUIDORA MEDICA, SOCIEDAD ANONIMA	35460180	1,345.89	4981803
14.jul.2016	CORPORACION DE SERVICIOS OFTALMOS, SOCIEDAD ANONIMA	26250764	74,330.36	5092000
14.jul.2016	AMECO, SOCIEDAD ANONIMA	57911681	11,625.00	5124093
14.jul.2016	SANATORIO CUILAPA, SOCIEDAD ANONIMA	86786326	90,000.00	5105234
14.jul.2016	GRUPO RADIAL VERTICAL, SOCIEDAD ANONIMA	89765354	22,500.00	5111331

2. Investigar las posibles herramientas. La parte correspondiente a esta fase se enfocó en buscar las herramientas más comunes para hacer *web scraping*, una de las primeras delimitaciones que se puso en este enfoque es que fueran ampliamente utilizadas y no estuvieran en fase experimental. Esto se debe que a la naturaleza del proyecto no se presta para utilizar herramientas que puedan ser descontinuadas y repercutan de manera negativa. De esta cuenta se pudo encontrar que la herramienta de *BeautifulSoup* es ampliamente usada por varias empresas alrededor del mundo para trabajar con páginas web. El siguiente componente que fue encontrado es *Requests*, que es una herramienta utilizada para comunicarse con los servidores web y obtener la página web para ser analizada (Sweigart, 2015) (Mitchell, 2015).

Por otra parte se encontró que ya hay *frameworks*, que le permiten a uno generar su *scraper* en base a modelos ya establecidos, por ejemplo *Scrapy* (Scrapy, s.f.). El problema con este tipo de acercamientos es que se encuentra limitado a las funcionalidades ofrecidas por el *framework*. En conjunto con el módulo de diseño e implementación de modelo de datos se llegó al consenso de utilizar archivos CSV para el almacenamiento de la información obtenida del sitio y de esta manera simplificar su posterior carga a la base de datos.

3. Investigar los posibles lenguajes de programación. Una vez encontradas las herramientas más comunes se comenzó a revisar para que lenguajes de programación estaban disponibles. Los lenguajes que tomaron como candidatos para ser revisados fueron PHP, *Python* y *Go*. Para comenzar la consideración de PHP se hizo con base en un proyecto similar existente llamado Transparente GT¹, al discutir los aprendizajes obtenidos por el equipo de Transparente se llegó a la conclusión de, que al ser ejecutado en un solo proceso, la velocidad de procesamiento era lenta.

A su favor juega que al ser un lenguaje orientado a la programación web, no tiene la necesidad de usar librerías externas y ha sido ampliamente utilizado para hacer *web scraping*. Su tiempo de ejecución prolongado, sumado a la necesidad de aprender a programar en este lenguaje hicieron que fuera descartado para la ejecución del proyecto.

El siguiente candidato en la lista era *Python*. La simplicidad de su sintaxis, que contribuye a disminuir la complejidad para futuros colaboradores del proyecto; el conocimiento previo que se tenía, permitía invertir más tiempo en desarrollo del programa y menos en aprendizaje del lenguaje. Una consideración adicional con este lenguaje es que se encuentra en un punto donde la versión 2.7 tiene previsto ser reemplazada por la versión 3.x en el mediano plazo (Wichmann, 2017), a pesar de esto se escogió trabajar con la versión 2.7 debido a la cantidad de documentación es mayor con la versión 2.7. Las herramientas investigadas en la fase anterior se encuentran disponibles para este lenguaje por lo cual no es necesario usar alternativas. Otro punto que se consideró para usar *Python 2.7* es que la sintaxis cambia con respecto al 3.x y por lo tanto si se diera el caso de tener que cambiar funcionalidades de las herramientas utilizadas es más sencillo hacerlas si no se tiene que considerar el tiempo de aprendizaje a fondo de la nueva sintaxis.

El caso de *Go* fue especial, fue sugerido por el catedrático de tecnologías web. En el caso de las librerías para hacer *web scraping*, se encontró que ya había varios proyectos²³ que había generado herramientas similares a las investigadas, el problema radicaba en que parecía estar sin soporte debido a que sus actualizaciones más recientes eran de 2014-2015. Sumado a la necesidad de aprender a programar en este lenguaje sin dejar que lado que de surgir la necesidad de ampliar las capacidades de las herramientas se debería tener un conocimiento profundo del lenguaje para poder lograrlo.

4. Selección del lenguaje de programación y las herramientas. Con las características de cada lenguaje de programación analizadas se tomó la decisión de utilizar *Python* en conjunto con *BeautifulSoup* y *Requests*. Esto con el objetivo de tener el control total sobre las funcionalidades ofrecidas.

¹ <https://transparente.gt/about/>. El proyecto de Transparente GT obtuvo la información de Guatecompras correspondiente al año 2015.

² <https://github.com/PuerkitoBio/goquery> . *GoQuery* es una librería para recorrer documentos HTML

³ <https://github.com/yhat/scrape>. *Scrape* es un programa que permite recolectar información de páginas web.

5. Diseñar el esquema de obtención de datos. Llegado este punto se conocía el flujo a seguir para obtener información de manera manual se procedió a escribir el programa que se interactuara con el sitio y obtuviera la información para su almacenamiento. El primer paso fue considerar cuales eran los periodos de tiempo más adecuados y manejables para obtener la información usando la búsqueda de la opción 5 (Figura 29). Debido a que en el sitio se agrega información de manera diaria, no es aconsejable dejar que se acumule una gran cantidad de datos porque el tiempo de procesamiento se vuelve largo. Debido a lo anterior se diseñó un método que es capaz de obtener las adjudicaciones, sus compradores y proveedores asociados, de un día determinado. Con este enfoque se hizo posible obtener información de manera diaria. Al ser establecido que la información que se deseaba obtener era la correspondiente al año 2016 la limitación más grande se encontró en el tiempo, la persona que ejecuta el programa tiene que estar atenta para ejecutar nuevamente el programa. Para solucionar esta desventaja se procedió a generar otro método que se encargaba de ejecutar al de obtención de información diaria, pero de manera continua hasta completar un mes calendario, con este nuevo enfoque se resuelve el problema anterior.

El problema que se presenta con el nuevo método es que al ser ejecutado durante un período de tiempo extendido, por lo general alrededor de 12 horas para obtener la información de un mes completo, es mucho más propenso a fallos, tales como pérdida de conexión a internet; fallas en la energía eléctrica. Para lidiar con estos inconvenientes se dotó al programa de un método que se encarga de repetir la solicitud de información al servidor hasta que este conteste de manera afirmativa o se supere un número de intentos determinado y se cancele la ejecución. La parte que se encarga de lidiar con los errores de conexión está basada en el concepto de *congestion avoidance*⁴⁵ que utiliza el protocolo TCP para manejo red. Básicamente lo que se hace es aumentar el tiempo de espera conforme se van detectando errores en la conexión y de esta manera se evita hacer múltiples solicitudes de contenido innecesarias lo que permite aprovechar la mayor parte de estas.

Figura 31. Método de procesamiento de solicitudes HTML

```
def obtain_html_content(request_type, url, data=None):
    """
    metodo encargado de realizar el request hacia la pag solicitada
    es capaz de volver a pedir la pag si la conexion falla o esperar
    si es que se sobrecarga al servidor destino
    :param request_type: si es un POST, GET, etc.
    :param url: link que vamos a consultar
    :param data: payload que se envia con el request
    :rtype string: que tiene el resultado
    """
    continuar = True
    resp = ''
    espera = FACTOR_ESPERA #segundos de espera inicial
    threshold = FACTOR_ESPERA*20
    max_ciclos = 1000
    ciclo_actual = 1
    while continuar and ciclo_actual <= max_ciclos:
        try:
            HEADERS['User-Agent'] = USER_AGENTS[randint(0, len(USER_AGENTS)-1)]
            req = requests.Request(request_type, url, headers=HEADERS, data=data)
            prepped = SESSION.prepare_request(req)
            response = SESSION.send(prepped, timeout=TIMEOUT)
            if response.status_code == OK_CODE:
                resp = response
                continuar = False
                # activarlo si comienzan a bloquear muy frecuentemente
                time.sleep(1) # espera 2 segundos para evitar cargar el server con el siguiente
            else: # hay que volver a pedir la info al server
                ciclo_actual += 1

        except (ex.ConnectTimeout, ex.ChunkedEncodingError, ex.ConnectionError) as la_exception:
            logging.exception(la_exception.message)
            logging.exception('voy a esperar %s seg', espera)
            time.sleep(espera)
            espera = ceil(espera*1.5)
            if espera > threshold:
                espera = ceil(espera/randint(1, 6))
                threshold = randint(floor(threshold*3/4), floor(threshold*7/5))
            ciclo_actual += 1
    #se quita el caracter de la flacha que causa error al parsear
    return resp.content.replace('&#9660;', '&#33;')
```

⁴ <http://www.mathcs.emory.edu/~cheung/Courses/558a/Syllabus/6-transport/TCP.html>

⁵ <https://www.cs.umd.edu/~shankar/417-F01/Slides/chapter3b/sld027.htm>

Por último se agregó un método de obtención de datos aún más general que permite obtener la información de un año calendario completo y resuelve el problema, mismo que se da entre días y meses, encontrado al querer obtener un año entero y utilizar solamente la obtención por mes.

El siguiente paso se enfocó en diseñar los métodos que se encargan de extraer la información obtenida por definidos anteriormente, como se ve en la hay dos links hacia páginas con información adicional. El primero es el proveedor y es aquí donde se encontró la primera diferencia notable, hay dos tipos de proveedores del Estado, las personas individuales y las personas jurídicas, Figura 34. Junto con el módulo de diseño e implementación de modelo de datos se acordó que los siguientes campos serían los obtenidos: NIT, nombre, tipo, representantes legales, municipio y departamento. De estos hay las personas individuales no presentan representante legal ni municipio y departamento, Figura 32, por lo cual se colocaron como 'null' en este caso, es importante resaltar que durante la obtención de la información de estos se tuvo que considerar que la información presentada para las empresas no estaba estandarizada esto es que en algunas se presentaba la fecha de inscripción provisional al Registro Mercantil y en otras la fecha de inscripción definitiva, razón por la cual se decidió obtener la fecha que estuviera disponible en lugar de buscar solamente la definitiva y en caso esta no estuviera presente dejarla en blanco. Otro de los campos que tuvo una consideración especial fue la dirección, en algunos casos se presentaba domicilio fiscal y comercial y en otros solamente el domicilio fiscal, en este caso para mantener la uniformidad se optó por extraer el municipio y departamento del domicilio fiscal. Este manejo de casos especiales hace que se la ejecución del programa haga cálculos que sería innecesarios de tener datos estandarizados por parte del sitio.

Figura 32. Detalle de proveedor, persona individual

Registro de Proveedores - GALVEZ,MUÑOZ,,MARIO,ERNESTO	
Identificación (Datos recibidos de la SAT el: 08.feb..2017 18:34:36)	
CUI:	1722356341201
Nombre o razón social:	GALVEZ,MUÑOZ,,MARIO,ERNESTO
Número de Identificación Tributaria (NIT):	1027040K
Nombre comercial 1:	SERVICIOS INTEGRALES DE CONSTRUCCION "SERINCO"
Nombre comercial 2:	SERVICIOS TECNICOS INTEGRALES "SERINT"
Estatus Actual del Proveedor (lea antes ¿Qué significa estar inscrito, habilitado, adjudicado y con contraseña?)	
Estatus en Guatecompras	
Habilitado o Inhabilitado:	HABILITADO
Adjudicado o No adjudicado:	ADJUDICADO
Participa o no en Contrato Abierto:	NO PARTICIPA (no tiene productos en el catálogo)
Con o Sin contraseña:	CON CONTRASEÑA
Estatus en la SAT(08.feb..2017 18:34:36)	
Estatus del RTU	ACTIVO
Solvencia del NIT en RTU	SOLVENTE
Datos de inscripción (Datos recibidos de la SAT el: 08.feb..2017 18:34:36)	
Tipo de organización:	INDIVIDUAL
Actividad Económica:	CONSTRUCCIÓN DE OBRAS PARA INGENIERIA CIVIL
Inconformidades e Inhabilitaciones	
Cantidad de INCONFORMIDADES presentadas:	[--NINGUNA--]
Cantidad de INHABILITACIONES recibidas:	[--NINGUNA--]

El siguiente link que se exploró fue el que detalla el proceso de la adjudicación y la información presentada en esta, una vez más en conjunto con el módulo de diseño e implementación de modelo de datos se acordó obtener la siguiente información: NOG, descripción, estatus, cantidad de productos adquiridos, fecha de publicación, fecha de presentación de ofertas, fecha de cierre de ofertas, fecha de finalización, tipo de ofertas, modalidad de compra y categorías. De estos campos hay ciertas adjudicaciones que no tienen fecha de finalización por lo cual se procedió a reemplazarlas por el valor 'null'. Durante la obtención de la información de las adjudicaciones se notó que en algunos casos no se encontraba presente la cantidad de productos o esta no estaba presente en el mismo elemento del documento HTML del sitio, por lo cual se implementó una búsqueda que considerara ambos casos y encontrara el dato esperado.

Finalmente queda obtener la información del comprador, para esto se decidió obtener la siguiente información: NIT, nombre, departamento, municipio, entidad general, origen de los fondos y tipo de entidad. En este caso en la información general muestra dos campos: entidad y unidad compradora, la primera se refiere a la entidad compradora general, *e.g.* Ministerio de Salud. Mientras que la unidad compradora se refiere a la dependencia específica dentro de la entidad que hace la compra, *e.g.* oficina de acceso a la información. Ambas entidades comparten datos tales como NIT y origen de los fondos.

Figura 33. Detalle de adjudicación, información general

Detalle de concurso	
Datos del Concurso	
NOG:	4380401
Categoría:	Construcción y materiales afines
Descripción:	CONSTRUCCIÓN SISTEMA DE AGUA POTABLE CASERIO GRANADILLAS, ALDEA CHANJULE DEL MUNICIPIO DE TACANÁ DEPARTAMENTO DE SAN MARCOS.-
Modalidad:	Cotización (Art. 38 LCE)
Tipo de concurso:	Público
Entidad:	MUNICIPALIDAD DE TACANÁ, SAN MARCOS
Tipo de entidad:	Gobiernos Locales (Municipalidades, Mancomunidades, etc.)
Unidad compradora:	TESORERIA MUNICIPAL DE TACANÁ, SAN MARCOS
Fecha de publicación:	08.enero.2016 Hora:9:42:18 a. m.
Fecha de presentación de ofertas:	21.enero.2016 Hora:9:00:00 a. m.
Fecha de cierre de recepción de ofertas:	21.enero.2016 Hora:9:30:00 a. m.
Recepción de ofertas:	Sólo en papel. Todas las ofertas deben recibirse en papel y no se permite recibirlas en forma electrónica.
Fecha de finalización:	27.enero.2016 Hora:12:34:02 p. m.
Estatus:	Terminado adjudicado

Para el nombre se decidió guardar el de la unidad compradora. Finalmente el tipo de entidad es un caso especial, las unidades compradoras no tienen este campo, pero aporta variedad a los posibles análisis que se pueden llevar a cabo. Es por esto que se obtiene de la información de la cantidad general.

Figura 34. Detalle de proveedor, persona jurídica

Registro de Proveedores - BAXTER DE GUATEMALA SOCIEDAD ANONIMA

Identificación
(Datos recibidos de la SAT el: 25.sep..2017 20:31:01)

CUI:
Nombre o razón social: BAXTER DE GUATEMALA SOCIEDAD ANONIMA
Número de Identificación Tributaria (NIT): 7505043
Nombre comercial 1: BAXTER DE GUATEMALA, S.A.

+ Estatus Actual del Proveedor
(lea antes ¿Qué significa estar inscrito, habilitado, adjudicado y con contraseña?)

- Domicilio Fiscal
(Datos recibidos de la SAT el: 25.sep..2017 20:31:01)

Departamento: GUATEMALA
Municipio: GUATEMALA
Dirección: 16 CALLE 0-55 9NIVEL Z.10 EDIF.TORRE INTERNACIONAL
Teléfonos: 2424-1111
Números de fax: 2424-1130

- Representantes legales
(Datos recibidos de la SAT el: 25.sep..2017 20:31:01)

	Proveedor	Plazo de Nombramiento
Representante Legal 1	(6693999) ESCALONA,CACERES,,ROBERTO,JOSE	05/04/2019
Representante Legal 2	(7680872) LOPEZ,LETRAN,,BRENDA,MARIENA	05/04/2019
Representante Legal 3	(76828182) RIVAS,CERNA,,BETZY,MARICEL	30/04/2025
Representante Legal 4	(87929112) TOLEDO,BARCENAS,,KARINE,ISABEL	30/04/2025

- Datos de inscripción
(Datos recibidos de la SAT el: 25.sep..2017 20:31:01)

Tipo de organización: SOCIEDAD ANÓNIMA
Número de escritura de constitución: 14
Fecha de constitución: 08.feb..0095
Inscripción PROVISIONAL en el Registro Mercantil: 07.mar..1995
Inscripción DEFINITIVA en el Registro Mercantil: 24.mar..1995
Inscripción en la SAT: 07.mar..1995
Actividad Económica: FABRICACIÓN DE PRODUCTOS FARMACÉUTICOS, SUSTANCIAS QUÍMICAS MEDICINALES Y PRODUCTOS BOTÁNICOS
NIT Notario: (3821773) PANIAGUA,CORZANTES,ROSALES,CAROLINA,

6. Probar el esquema definido. Al momento de diseñar los métodos que forman el programa se consideró que iba a ser necesario hacer una gran cantidad de solicitudes de información al sitio, razón por la cual se hizo un método específico para esta funcionalidad (Figura 31) y de esta manera si fuera necesario implementar un cambio en la forma de comunicarse con el sitio, esta se puede hacer en un solo punto sin afectar al resto del programa. Adicionalmente se decidió que la extracción de información de los 3 grandes entes (adjudicaciones, compradores, proveedores) tuviera su propio método (Figura) esto primero aporta claridad para la comprensión del código, pero aún más importante es que debido a que la forma de extracción de los datos está amarrada a la estructura de la página web y que esta se mantenga inalterada. Razón por la cual al separar en métodos distintos las entidades es posible adaptarlos a futuros cambio sin tener afectar al resto del programa.

Para esta fase se puso en funcionamiento el conjunto de métodos definidos en la fase anterior, para la ejecución del programa se hizo por medio de la terminal de comandos de *Windows*. Al obtener la información del sitio se procedió a almacenarla en archivos *CSV*. Adicionalmente se tomó en cuenta el tiempo de ejecución del programa para adaptarlo y obtener un menor tiempo.

Cabe resaltar que las pruebas iniciales se llevaron a cabo en una computadora con sistema operativo *Windows*, la razón por la cual se hizo así es que el servidor donde se alojó el programa en su versión final fue puesto a disposición durante la etapa final del proyecto.

Para evitar el consumo excesivo de RAM se decidió que al terminar de obtener la información para un día el programa elimine de caché la información de las adjudicaciones mas no de los proveedores y compradores. Esto se debe a que las referencias a compradores y proveedores no se limitan al mismo día donde fue obtenida su información por primera vez, mientras que con las adjudicaciones solamente se referencian dentro del mismo día.

7. **Validar los datos obtenidos.** Esta fase consistió en cotejar la información obtenida contra los reportes que ofrece el sitio, con el objetivo de validar si se obtuvo la información de manera de manera correcta. Esta fase se llevó a cabo en colaboración con el módulo de diseño e implementación de modelo de datos debido a que la validación era más sencilla de llevarse a cabo desde una base de datos estructurada a diferencia de los archivos CSV, los cuales no tienen una estructura apta para esta tarea. Como primer paso se validó que el número de adjudicaciones coincidiera con las mostradas en el sitio. El siguiente dato que se validó fue el monto total de las adjudicaciones. Una vez superadas ambas validaciones el siguiente paso fue asegurar que la no existieran adjudicaciones sin comprador ni proveedor. Este proceso se hizo para cada mes y para cada una de las distintas modalidades, esto con el objetivo de focalizar la información que debía ser obtenido nuevamente, en caso fuera necesario.

B. RESULTADOS

La extracción de la información se hizo en distintos ambientes, el primero fue en laptop con las siguientes especificaciones: procesador Core i7 con 2 núcleos, 8 GB de RAM, 256 GB SSD y sistema operativo *Windows* el cual fue el ambiente de desarrollo (en adelante 'PC'); el segundo fue un servidor de *Amazon Web Services Lightsail* con las siguientes especificaciones: procesador de 1 núcleo, 1 GB de RAM, 30 GB SSD y sistema operativo Linux que representa el ambiente donde se alojó el programa para su ejecución en producción(en adelante 'Server').

Para la obtención de las cifras presentadas en esta sección se utilizaron variantes del algoritmo de extracción de datos, una con tiempo de espera y otra sin este. En la PC únicamente se utilizó la primera, mientras que en el Server se utilizaron ambas (Server wait y Server no wait). Para la interpretación cuando se haga referencia a Server, se asumirá la variante con tiempos de espera a menos que sea haga referencia explícita a la otra.

Tabla 10. Cantidad de adjudicaciones obtenidas por mes en 2016

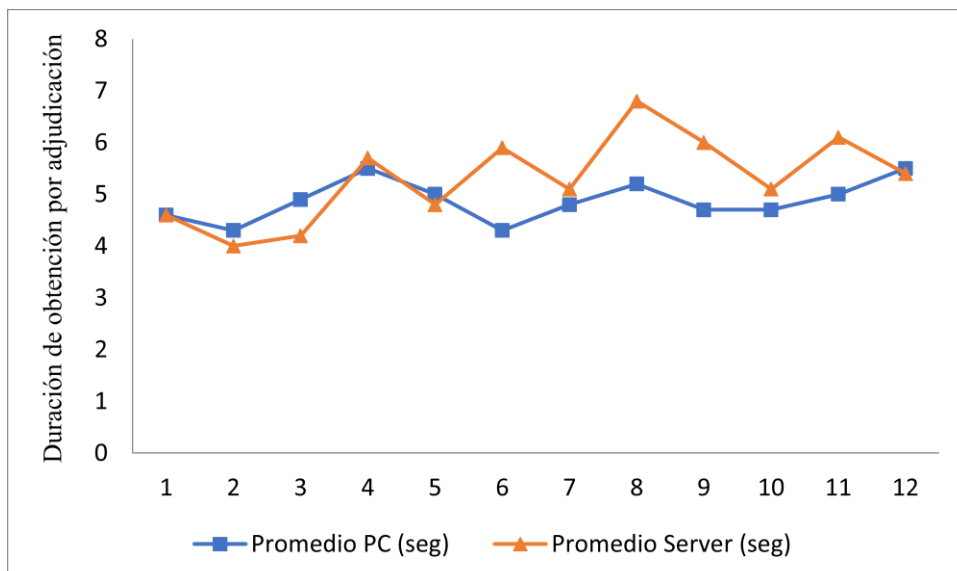
Mes	Adjudicaciones
1	3,323
2	7,172
3	7,438
4	6,626
5	8,591
6	9,283
7	7,287
8	7,853
9	11,370
10	9,960
11	8,877
12	5,039
Total	92,819

En la Tabla 10 se tienen la cantidad de adjudicaciones que se obtuvieron por cada mes de 2016. Estas cantidades se obtuvieron como resultado de las búsquedas hechas con la opción 5 de búsqueda en el sitio.

Tabla 11. Duración promedio de obtención por adjudicación

Mes	Promedio PC (seg)	Promedio Server (seg)
1	4.6	4.6
2	4.3	4.0
3	4.9	4.2
4	5.5	5.7
5	5.0	4.8
6	4.3	5.9
7	4.8	5.1
8	5.2	6.8
9	4.7	6.0
10	4.7	5.1
11	5.0	6.1
12	5.5	5.4

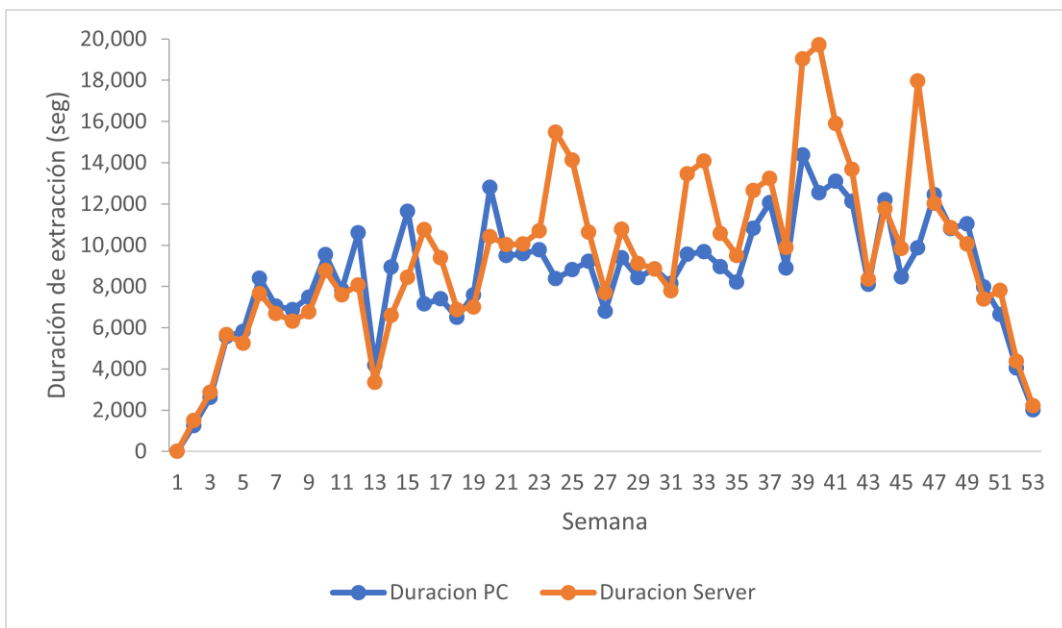
Figura 35. Duración de extracción datos de 2016 por mes, PC vs Server



En la Tabla 11 se encuentra la duración promedio para obtener las adjudicaciones de cada mes de 2016 obtenidas en PC como en el Server, para calcularlo se utilizó la información de la Tabla 10 en conjunto con la presentada en la Figura 35.

La Figura 35 se encuentra la comparación de la duración de la obtención de datos por mes de 2016, tanto en PC como el Server.

Figura 37. Duración de extracción datos de 2016 por semana, PC vs Server



La Figura 37 muestra la variación de la duración de la obtención de los datos conforme a la semana del año que se obtuvo.. La Figura 36 muestra la cantidad de adjudicaciones obtenidas por cada semana de 2016.

Figura 36. Adjudicaciones obtenidas por semana

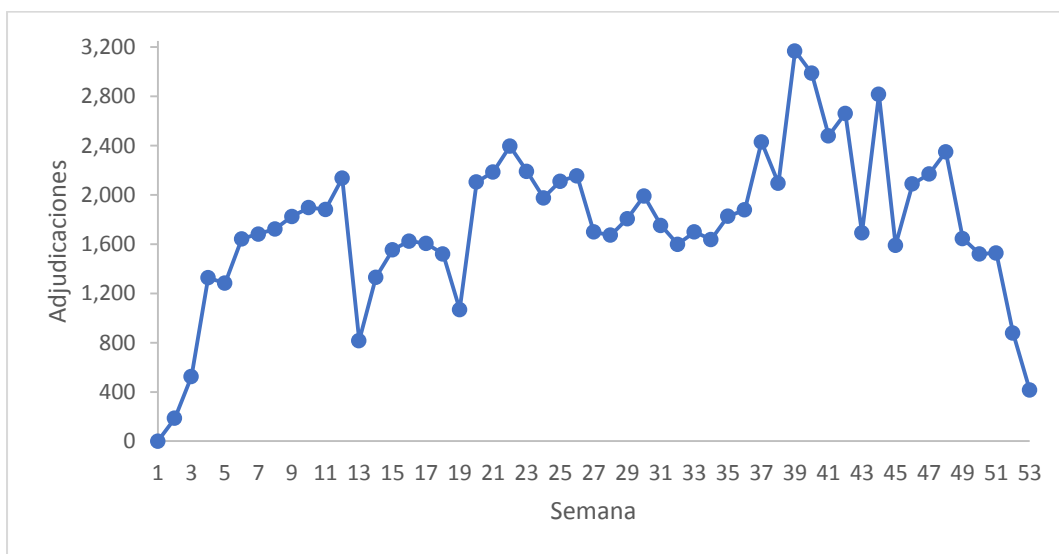
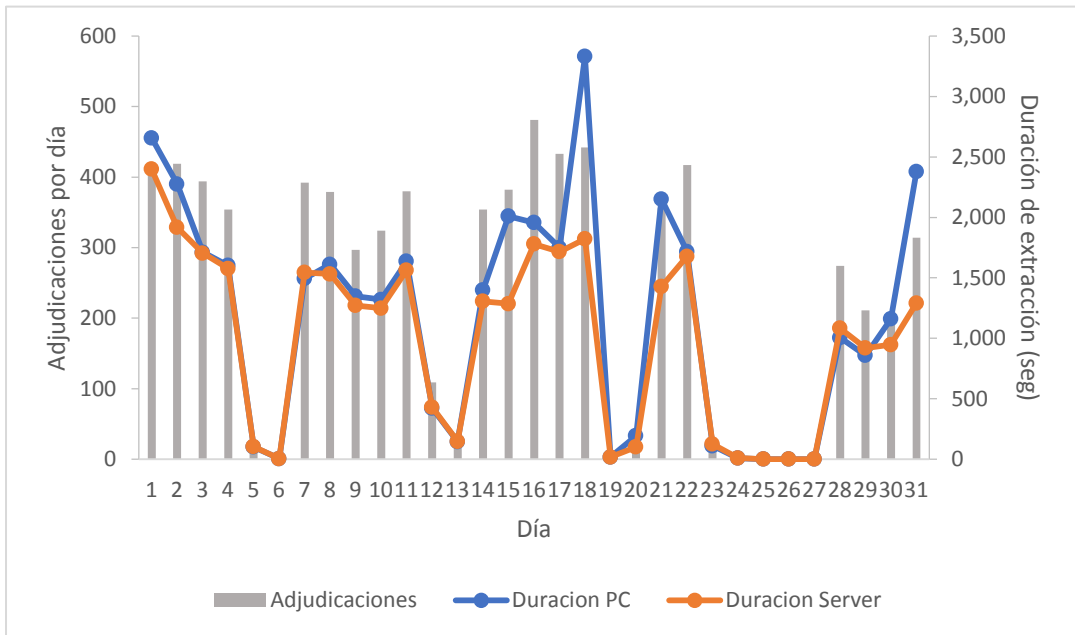
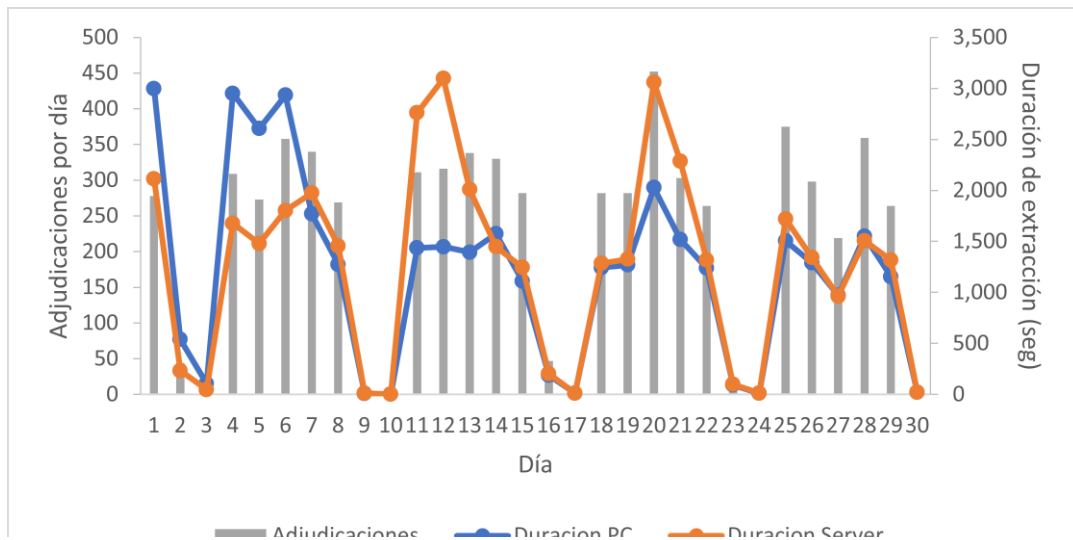


Figura 38. Duración de extracción de datos marzo 2016, PC vs Server



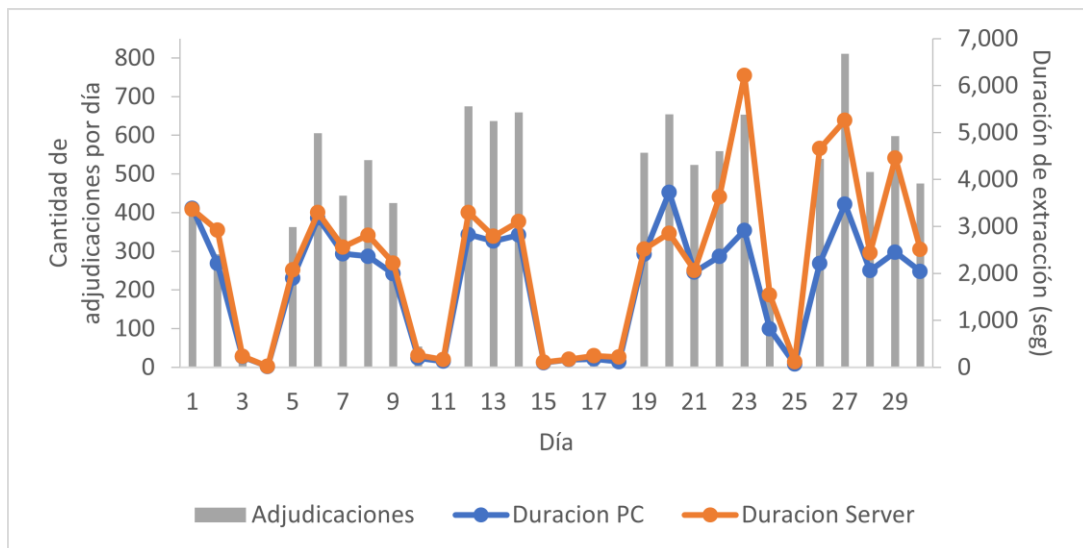
La Figura 38 muestra la relación de la cantidad de adjudicaciones por día y la duración de la extracción de datos de marzo de 2016, tanto para PC como para el Server.

Figura 39. Duración de extracción de datos abril 2016, PC vs Server



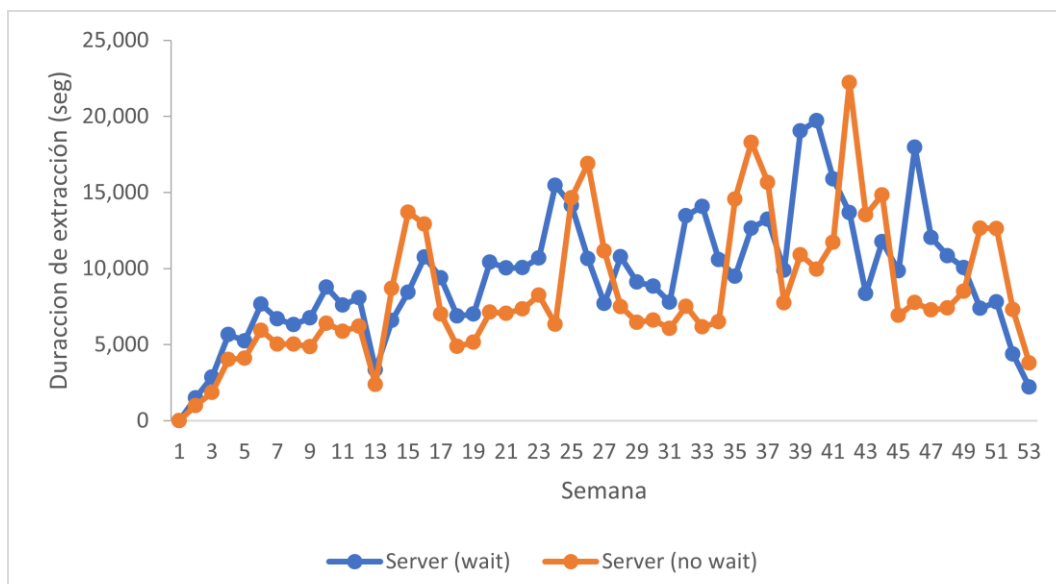
En la Figura 39 se aprecia la relación entre la cantidad de adjudicaciones y la duración en su obtención para abril de 2016, para PC y el Server.

Figura 40. Duración de extracción de datos septiembre 2016, PC vs Server



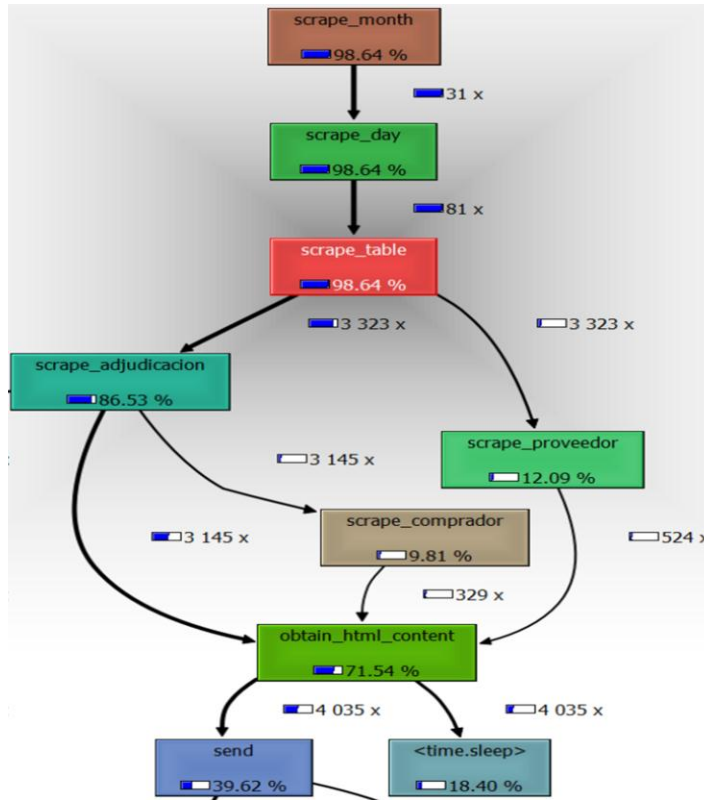
La Figura 40 muestra la cantidad de adjudicaciones para los días de septiembre de 2016 y la duración de su obtención en PC y Server.

Figura 41. Comparación de algoritmos para extracción de datos, ejecutados en Server



En la Figura 42 se encuentra la comparación de la duración de la extracción de información por semana, en el Server, entre las variantes del algoritmo con tiempo de espera entre solicitudes de información al sitio y sin este.

Figura 42. Diagrama de ejecución del programa

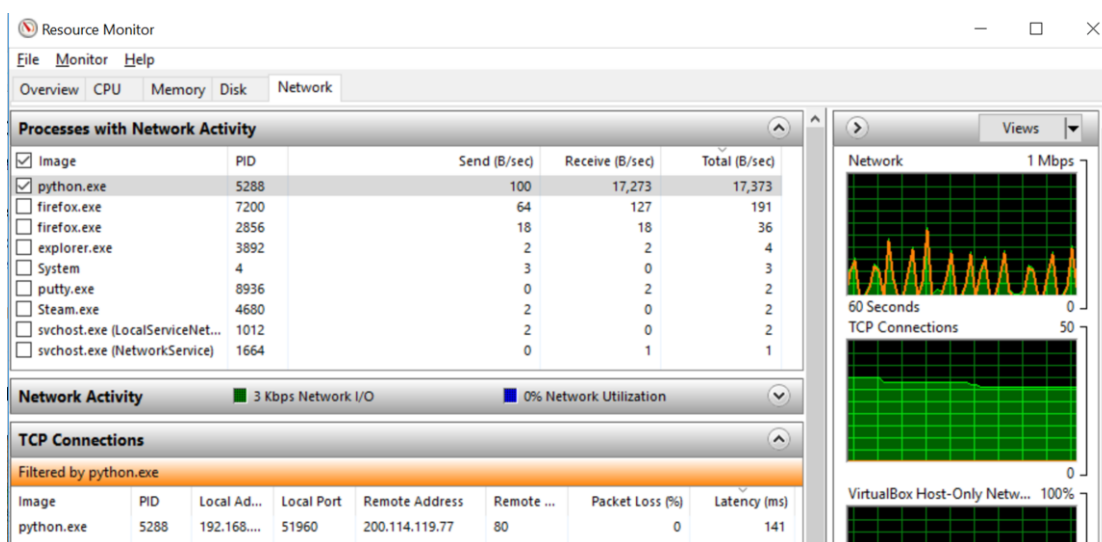


L

a

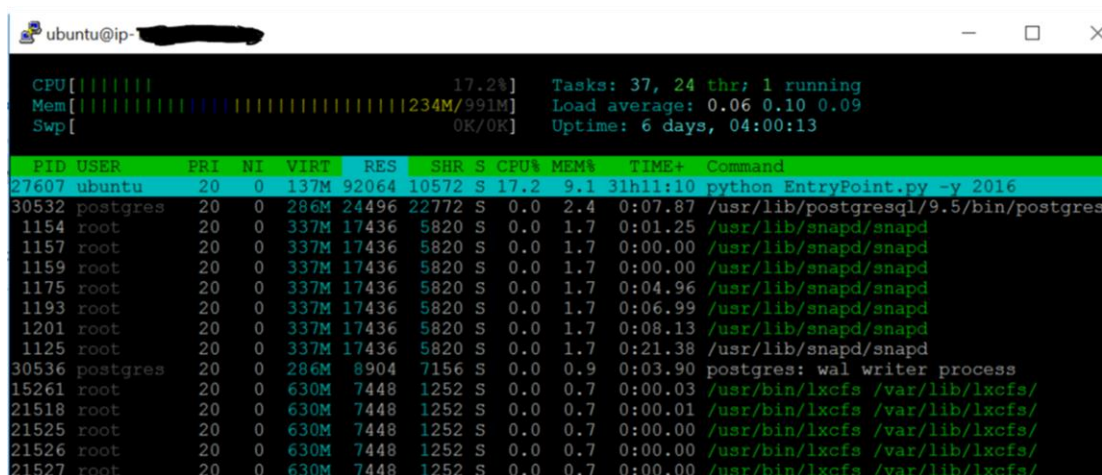
Figura muestra el flujo de ejecución, así como el número de llamadas entre los métodos del programa durante la obtención de enero de 2016.

Figura 43. Uso de red en PC



En la 43 se muestran estadísticas referentes al consumo de red por parte del programa mientras se ejecutó en la PC.

Figura 43. Consumo de recursos en Server



En la Figura 43 se muestran las estadísticas de consumo de recurso del programa durante su ejecución en el Server. La columna RES se refiere a la cantidad de RAM (en kB) que utilizó el proceso, la columna MEM% se refiere al porcentaje de RAM que utilizó el proceso, en la columna TIME+ se muestra el tiempo que el procesador ejecutó tareas del programa

C. DISCUSIÓN

Asegurar que los datos obtenidos coincidieran con los presentados en el sitio es de vital importancia para el proyecto, porque las personas que utilicen estos datos deben tener la seguridad de que cualquier análisis que se genere a partir de estos sea válido. La obtención íntegra de los datos correspondientes a los proveedores, compradores y adjudicaciones del año 2016 fue la tarea que debía cumplir el programa durante su ejecución y la manera de asegurar el cumplimiento de esta fue comparar las estadísticas de los datos obtenidos contra los reportes que se ofrecen en el sitio. Los datos de mayor relevancia son los de adjudicaciones y por lo tanto fueron los primeros que se validaron.

El total de adjudicaciones obtenidas para el año 2016 no coincidió, ni en cantidad de adjudicaciones ni en el monto total de las mismas con el reporte obtenido del sitio (ver anexos), esto se debe a que el reporte generado por el sitio incluye las adjudicaciones a proveedores sin NIT, las cuales no fueron obtenidas para este proyecto. Una vez descartadas estas adjudicaciones el reporte siguió sin coincidir en cuanto a cantidad, esto se debe a que cuando se da el caso de que un concurso es adjudicado a distintos proveedores el NOG es el mismo para todos pero al ser desplegados en los resultados de la búsqueda por día (Figura 30) se muestra la información de cada proveedor con el monto correspondiente a su adjudicación así como el link hacia la información del concurso, Figura 33. Es decir, en los resultados de la búsqueda se muestran como entradas diferentes pero todos tienen el link hacia la misma página de información del concurso. Una vez tomadas en cuenta estas particularidades, la cantidad de adjudicaciones obtenidas así como el monto de las mismas coincidió con el reporte ofrecido por el sitio.

Una vez asegurada la veracidad de las adjudicaciones se debía asegurar la confiabilidad de los datos compradores y los proveedores. Al no contar el sitio con reportes específicos para compradores, la validación consistió en buscar compradores que no tuvieran ninguna referencia desde las adjudicaciones. Para esta validación no se encontraron adjudicaciones sin comprador ni compradores sin adjudicaciones lo cual indica que la extracción de datos se llevó a cabo de manera correcta. En el caso de los proveedores se encontró una circunstancia similar a la anterior, razón por la cual se procedió a validar con la misma metodología que los compradores. Nuevamente se encontró que no había proveedores sin adjudicaciones ni adjudicaciones sin proveedor. Sumando esto al conocimiento de que las adjudicaciones de 2016 fueron obtenidas de manera íntegra, se puede asegurar que la información obtenida por el programa es fidedigna. La razón por la cual la información de los proveedores y compradores fue obtenida de manera correcta es que los links hacia estos se obtenían de la página de información de la adjudicación, Figura 33. Esto provocó que al solicitar una adjudicación y su información se asegurara la obtención del comprador y proveedor asociados a la misma de una vez. La importancia de haber utilizado este enfoque radica en el aseguramiento de la obtención de los datos de manera correcta.

La obtención de los datos solamente es una parte de los requisitos del programa, la obtención de manera eficiente y efectiva fue clave para el proyecto. De nada sirve que el programa obtenga los datos para un día si esta obtención tarda 24 horas, es por esta que razón que durante el diseño del programa se contemplaron distintas formas de disminuir el tiempo de ejecución. Finalmente se llegó al diseño que se observa en la Figura . De las 3,323 veces que se ejecutó el método para obtener los datos de una adjudicación, únicamente se hicieron 3,145 consultas al sitio y esto se debe a la existencia de adjudicaciones hechas a múltiples

proveedores. Así mismo se hicieron 3,145 ejecuciones del método que obtiene los datos de los proveedores pero solamente 329 consultas al sitio. Esto se debe a que se decidió almacenar en una caché local la información de los proveedores y antes de hacer una solicitud al sitio revisar si ya se contaba con la información necesaria, con esto se logró hacer una disminución drástica del tiempo de ejecución del programa. Ya que cada vez que se hace una solicitud al sitio el programa está diseñado para esperar 1 segundo antes de continuar con su ejecución para evitar sobrecargar al sitio de solicitudes, de igual manera esta espera no garantiza evitar el bloqueo por parte del sitio pero contribuye a minimizar la probabilidad de que ocurra. Lo mismo ocurre con el método de obtención de datos de proveedores, de las 3,323 que este fue ejecutado, únicamente en 524 fue necesario hacer consultas al sitio para obtener la información. Estas cifras son relevantes en el hecho de que por cada consulta al sitio se deben esperar la respuesta del mismo y no se tiene la certeza de que esta espera sea corta, mientras que hacer una consulta a la información guardada en caché más eficiente. Por lo cual al recortar las consultas hechas, también se disminuyó la posibilidad de que ocurra un bloque por parte del sitio y además el tiempo de ejecución del programa se reduce como consecuencia directa del recorte de consultas.

Al hablar de bloqueo por parte del sitio lo que sucede es lo siguiente, al realizar muchas solicitudes en un periodo de tiempo corto, el servidor del sitio deniega nuevas conexiones de la IP de origen (nuestro equipo) y es necesario esperar antes de enviar una nueva solicitud, porque al seguir enviando solicitudes el servidor, este simplemente las ignora. Si después de ese compás de espera el servidor vuelve a denegar la solicitud, se debe aumentar la espera y volver a intentar. Este compás de espera es de 9 segundos inicialmente pero aumenta rápidamente de acuerdo a un algoritmo (Figura 31), en contraste esperar 1 segundo después de cada solicitud no parece ser significativo.

La extracción de todos los datos de 2016 se llevó cabo inicialmente en la PC, lo que obligó a esto fue que el Server fue adquirido hasta la fase final del proyecto y era de vital importancia contar con los datos para la plataforma. Durante esta extracción inicial solamente se contaba con la versión del algoritmo que espera 1 segundo después de cada solicitud y al terminar con la ejecución de este se consideró la posibilidad de alterarlo, lo que dio paso a la versión alternativa del mismo. Durante la ejecución en la PC también se monitoreó el consumo de recursos, (Figura). Una vez se tuvo el Server disponible se procedió a ejecutar la extracción de datos para todo 2016 y de esa manera poder comparar el rendimiento en distintos ambientes de ejecución. Esto fue importante para asegurar que la diferencia de componentes y recursos no afectara de manera exagerada el rendimiento del programa.

La obtención de datos de marzo 2016 que se encuentra en la Figura 38, muestra detalles importantes. Para comenzar la extracción de datos en los primeros días del mes es más lenta que en el resto, esto queda claro al ver que a pesar de tener casi la misma cantidad de adjudicaciones los días 1 y 7 tienen una diferencia de casi 1,000 segundos, tanto para PC como Server. Esta situación se puede explicar debido a que al tener almacenada en caché la información de los proveedores como de los compradores, se reducen la cantidad de solicitudes de información al sitio, lo que se traduce en un ahorro significativo de tiempo. Otro detalle importante que se debe resaltar se encuentra en los días 9, 10, 15 y 16. A pesar de que el segundo y el cuarto tienen una mayor cantidad de adjudicaciones que los otros, su tiempo de extracción fue menor, para los primeros días dicha situación se dio en ambos ambientes y en los últimos se da solamente en PC, mientras que para los días 14 y 15 se manifiesta únicamente en el Server. Este comportamiento de ahorro de tiempo solamente en alguno de los ambientes es más claro en la Figura 39, los días 4,5 y 6 necesitaron de más tiempo para su obtención en PC,

situación que se revirtió en los días 11, 12, 13, 20 y 21. Este comportamiento deja claro no todo el tiempo alguno de los ambientes es totalmente más rápido que el otro para obtener los datos a pesar de la diferencia en los recursos disponibles. Esto es muestra que a pesar de haber sido desarrollado en un ambiente distinto al cual sería puesto en producción no hubo una diferencia excesiva en cuanto al rendimiento observado.

En la Figura 35, se muestran que la diferencia en el tiempo que ejecución fue mayor en la PC durante 4 meses, mientras que en el Server fue mayor durante 7 meses, el mes de enero fue exactamente el mismo tiempo de ejecución para ambos ambientes. De los 7 meses en los cuales el Server tardó más extraer los datos, son realmente 4 meses los que acentúan esta diferencia. La Figura 37 muestra que durante la ejecución el server fue más lento que el PC durante bastante tiempo pero resaltaron 4 picos en distintos meses. Siendo uno de estos meses septiembre (Figura 40), se puede apreciar que el pico que genera el mayor diferencial se encuentra el día 23, el cual toma aproximadamente el doble que la ejecución en PC. Aún más sorprendente es que superó en duración la extracción del día 27, el cual tiene el mayor número de adjudicaciones para cualquier día de 2016.

Al haber finalizado esta extracción inicial se procedió a buscar puntos de mejora en el rendimiento del programa. El punto obvio de mejora se encontró en el algoritmo que se encarga de hacer las solicitudes y los tiempos de espera asociados a este. Dicha variante se probó únicamente en el Server, esto se debe al hecho de que será desde este que tendrá lugar la extracción de datos y por lo tanto no es importante el cambio en el rendimiento durante la ejecución en la PC.

Es por esto que se decidió comparar la ejecución de ambas versiones del algoritmo en el Server (Figura 42) y se encontró que durante la mayor parte del tiempo la versión sin el segundo de espera tras cada solicitud al sitio fue más rápida en la obtención de los datos. Sin embargo se observa que alrededor de la semana 13 comienza el aumento en la duración de la extracción de datos y en su pico de mayor diferencia cercano a la semana 15 es aproximadamente 1.5x más lento que el algoritmo que con tiempo de espera después de cada solicitud, es importante mencionar que ese 1.5 es también el factor de aumento del tiempo de espera cuando el algoritmo detecta un bloqueo por parte del servidor, con lo cual podemos asegurar que en este punto ocurrieron bloqueos a las solicitudes hechas por el programa y por lo tanto se activó el modo de espera antes de seguir generando solicitudes. Otro factor que se consideró fue que la el aumento de adjudicaciones de la semana 15 con respecto a la 14 es de menos de 400 la (Figura 36) asumiendo que la mayoría de las adjudicaciones obtenidas fueron obtenidas durante el mes de abril, se hizo un estimado simple en el cual se asumió un incremento de 400 adjudicaciones y el tiempo promedio de 5.5 segundos se obtuvo como máximo un aumento de 2,200 segundos que representan menos de la mitad de la diferencia entre las semanas en cuestión.

Un argumento similar se puede utilizar para analizar el fenómeno que se aprecia en las semanas 23 y 24 (Figura 36), solamente que en este caso el afectado es el algoritmo con tiempo de espera incluido, lo que demuestra que al igual que el otro fue bloqueado, pero se retrasó la aparición de esta situación en este caso.

Por otra parte durante la ejecución del algoritmo con tiempo de espera se midió el consumo de los recursos del Server usando una herramienta externa para no alterar el desempeño del programa. La medición hecha durante esta ejecución no se ve afectada por la diferencia de tiempos de ejecución entre algoritmos debido a que el número de solicitudes al sitio y extracción de dichos datos que se ve reflejado en la utilización de memoria RAM donde se almacena la información obtenida antes de ser escrita en un CSV y almacenado en

el disco duro, es exactamente el mismo. En la Figura 43 se ve en la columna del uso de RAM que se está utilizando el 10% aproximadamente del recurso disponible, considerando que la ejecución lleva al menos 31 horas, la columna de uso del procesador no mide el tiempo total de la ejecución del proceso, solamente el tiempo acumulado de sus operaciones en el procesador, se considera que se es eficiente con el manejo de este recurso. Al disminuir el consumo de RAM se abre la puerta para ejecutar el programa en sistemas con menos recursos y por lo tanto se amplía la cantidad posible de ambientes donde el programa puede funcionar.

VIII. DISEÑO Y DESARROLLO DE ESTUDIO DE USABILIDAD PARA LA INTERACCIÓN HUMANO-COMPUTADOR

A. METODOLOGÍA

El módulo de Interacción Humano Computador para la plataforma GobHash fue llevado a cabo durante todo el desarrollo de la plataforma de manera que cada una de las fases que conforman el proceso de diseño y pruebas de la plataforma se llevaran a cabo en el momento más oportuno para de esta manera aportar la retroalimentación más útil y ayudar lo más posible al desarrollo del proyecto, el desarrollo del módulo fue llevado a cabo en tres fases generales, cada una con un propósito específico dentro del proceso de desarrollo y pruebas de la plataforma GobHash, estas fases del proceso de diseño y pruebas de la plataforma GobHash se muestran a continuación.

1. Fase I: Desarrollo de estudio de estándar y usabilidad. En la primera fase del módulo de Interacción Humano Computador para la plataforma GobHash se llevó a cabo un estudio de estándar y usabilidad con el objetivo de definir aspectos primordiales de la plataforma y de esta manera iniciar el desarrollo del proyecto de la forma correcta, en el estudio de estándar y usabilidad se definieron temas relativos a funcionalidad, idea de proyecto y referencia respecto de plataformas actuales en el ámbito internacional. Los aspectos definidos durante el desarrollo del estudio de estándar y usabilidad son los siguientes:

a. Proposición de valor. La proposición de valor es muchas veces la mejor forma de iniciar un nuevo proyecto, ya que esta encierra muchos aspectos básicos del mismo, tales como definición, objetivos y funcionalidades principales. En el desarrollo de la plataforma GobHash, la proposición de valor fue de las primeras cosas que se realizaron, ya que fue de mucha ayuda para aterrizar la idea y abrió paso para la definición de los requisitos funcionales de la plataforma, ya con un propósito de ser del proyecto definido por la proposición de valor.

b. Análisis de competencia. El análisis de competencia se realizó con el objetivo de definir el estándar de lo que se quería lograr con la plataforma GobHash, al plantear un punto de referencia con plataformas existentes actualmente, se buscaron plataformas que cumplieran con la filosofía de datos abiertos, de la misma manera se buscó que estas plataformas ofrecieran la funcionalidad de mostrar los datos de una manera amigable al usuario, y no que solamente ofrecieran los datos en crudo al público, de esta manera se encontraron 3 plataformas que cumplían con los requisitos para formar parte del análisis de competencia,

se muestra una breve descripción de estas plataformas a continuación y en la sección de resultados los hallazgos del análisis de competencia.

1. Data.gov. Data.gov es el portal de datos abiertos del gobierno de los Estados Unidos de América, ya que aquí se centralizan los datos públicos la plataforma cumple con los requisitos para ser tomada en cuenta en el análisis de competencia.

2. Data.gov.uk. Data.gov.uk es el portal de datos abiertos del Reino Unido, aunque a simple vista pueda parecer una plataforma análoga a data.gov, data.gov.uk no es solamente el portal de centralización de datos abiertos del gobierno, sino también un portal abierto en donde los ciudadanos británicos pueden aportar sus propios datos además de publicar aplicaciones creadas a partir de los datos disponibles en la página.

3. Data.imf.org. Data.imf.org es el portal de datos abiertos del Fondo Monetario Internacional (IMF por sus siglas en inglés), en este portal se guarda registro de todos los datos referentes a la inversión de los fondos que se proveen desde el Fondo Monetario Internacional, esta plataforma además de cumplir con el requisito de datos abiertos, cuenta con potentes herramientas para la creación de gráficos interactivos e inclusive modificables por los usuarios del portal.

a. Diagrama de flujo de aplicación. Luego de finalizar el análisis de competencia, además de tener la definición y propósito de la plataforma, se contaba con un panorama amplio respecto de plataformas que implementaban las funcionalidades que se querían dentro de la plataforma final, de esta manera se creó un diagrama de flujo de aplicación que mostraba las distintas pantallas de la interfaz de usuario y como estas estaban interconectadas logrando así un flujo de aplicación lo más natural y sencillo posible.

b. Definir el perfil de los usuarios finales: En la plataforma GobHash se definió el perfil de los usuarios finales para las siguientes etapas de diseño y pruebas de la plataforma, más adelante en la sección de resultados se muestra como estos grupos objetivo se dividieron entre las etapas de validación de prototipos y pruebas de usabilidad respecto del tipo de retroalimentación y resultados que se necesitan en cada etapa del diseño y pruebas de la plataforma

c. Definición de curva de aprendizaje. Finalmente antes de terminar todo lo relacionado con la definición de un estándar respecto del cual se va a diseñar y desarrollar la plataforma, es necesario definir una curva de aprendizaje que ayude a identificar el grado de ayuda que el usuario necesitara a la hora de una prueba de usabilidad con una primera versión funcional.

d. Definición de métricas a cumplir con las pruebas de usabilidad de la nueva plataforma. El último paso del estudio de estándar y usabilidad fue la definición de métricas a utilizar en la medición de resultados de las pruebas de usabilidad a realizarse al momento en el que exista una primera versión funcional del proyecto. La definición de estas métricas es el resultado más importante del análisis de

estándar y usabilidad ya que plantea las metas del proyecto respecto de la usabilidad de la plataforma y provee de un método de medición de resultados del estudio de usabilidad. Sin las métricas no habría forma de justificar la razón por la cual se lleva a cabo el estudio de usabilidad.

En la definición del estudio de estándar y usabilidad por lo general también se define el retorno de inversión (ROI por sus siglas en inglés) del estudio de usabilidad, por medio de las métricas definidas se define como el estudio de usabilidad provee una mejora lo suficientemente significativa como para cubrir los costos de su realización, sin embargo, por el ámbito académico del proyecto, no se definió retorno de inversión para el estudio de usabilidad de la plataforma GobHash.

2. Fase II: Desarrollo de prototipos de diseño. En la segunda fase del módulo se llevaron a cabo los prototipos de diseño, mejor conocidos como *mockups*, que luego fueron validados por miembros del grupo objetivo con el propósito de validar el diseño, colores, y flujo de la plataforma GobHash. Durante el proceso de desarrollo y validación de prototipos de diseño se llevaron a cabo 3 tipos diferentes de prototipos, cada uno con un nivel de complejidad y detalle mayor, de manera que sea más fácil realizar cambios al principio con los primeros prototipos y de esta manera ir elevando la complejidad de los mismos conforme los cambios a realizar se reducen. El principal objetivo del desarrollo de prototipos de diseño fue proveer al módulo de diseño, análisis y desarrollo de Frontend de una idea validada respecto del diseño, colores y flujo de la plataforma. A continuación se muestra la descripción de los tipos de prototipos desarrollados y la validación de los mismos.

a. Prototipos en papel. Estos fueron dibujados en hojas de papel, favoreciendo la simpleza y la ausencia de colores, para obtener retroalimentación sobre aspectos generales del diseño más que en pequeños detalles, los prototipos en papel son fáciles de cambiar por la simpleza de su diseño.

b. Prototipos diseñados. Estos prototipos fueron diseñados utilizando Gimp, una herramienta de manejo de imágenes, estos ya contaron con la presencia de un logo, colores y tipo de letra específicos, el objetivo de estos prototipos fue obtener retroalimentación respecto de los detalles del diseño, los colores utilizados, y contemplo agregar y quitar elementos del diseño. Luego de la validación de estos prototipos diseñados, se dio inicio la fase de desarrollo de Frontend por parte del módulo de Diseño, Análisis y Desarrollo de Frontend.

c. Prototipos interactivos. Este prototipo fue desarrollado en conjunto con el módulo de Diseño, Análisis y Desarrollo de Frontend utilizando Silex, una herramienta de desarrollo de páginas web simples, con el propósito de crear un prototipo interactivo y al mismo tiempo avanzar en la maquetación HTML de la plataforma final, de manera que el prototipo fuera lo más parecido posible a lo que sería la primera versión funcional, este prototipo contenía el flujo de aplicación deseado para un prototipo funcional, además de desplegar *dummy data*, con este prototipo se buscó obtener retroalimentación respecto del flujo y

funcionamiento del sistema, la validación de este prototipo dio paso al desarrollo completo de la primera versión funcional del sistema a utilizarse en las pruebas de usabilidad.

Luego del desarrollo de estos prototipos se dio paso a la validación de los mismos, para la validación de los prototipos se diseñó una ficha de validación de prototipos, en donde se recolectaron datos de las personas entrevistadas para la validación así como su retroalimentación cualitativa respecto al diseño, flujo de aplicación y paleta de colores. (Ver anexo)

Esta validación se realizó con un total de seis personas pertenecientes a dos de los tres grupos objetivo definidos en el estudio de estándar y usabilidad, por la naturaleza cualitativa de la retroalimentación que se buscaba en el proceso de validación de prototipos de diseño, no fue necesario validar los prototipos con una gran cantidad de personas, sin embargo, cuando la retroalimentación requerida sea cuantitativa, es necesario elevar la cantidad de usuarios a no menos de 15 personas durante el test de usabilidad.(Nielsen, 2000)

3. **Fase III: Pruebas de usabilidad.** Las pruebas de usabilidad se realizaron cuando se dispuso de la primera versión funcional de la plataforma GobHash, se realizó la prueba de usabilidad con un total de 15 personas pertenecientes al grupo objetivo restante de la validación de prototipos de diseño, se modeló la prueba de usabilidad por medio de un script con tareas específicas a realizarse en GobHash.

Además de las plataformas evaluadas en el análisis de competencia en el estudio de estándar y usabilidad, previo al diseño del script y demás documentación de las pruebas, se desarrolló un plan de pruebas de usabilidad que define los detalles de la prueba de usabilidad realizada a los usuario expuestos con anterioridad y definió el lugar, la cantidad de personas necesarias y la organización de la prueba.

B. RESULTADOS

1. Fase I: Estudio de estándar y usabilidad

a. **Proposición de valor.** GobHash es una plataforma de análisis descriptivo sobre las adquisiciones del estado de Guatemala, construida como una red social, que permite a los usuarios, sin necesidad de conocimientos o habilidades específicas crear, compartir, modificar y discutir indicadores descriptivos creados a partir de conjuntos de datos extraídos del portal de Guatecompras.

b. **Análisis de competencia.** Como se describió en la metodología, el análisis de competencia se realizó en tres plataformas que cumplen los requisitos de datos abiertos, gobierno abierto, y funcionalidad en la muestra de los datos al usuario. A continuación se detallan los hallazgos del análisis de competencia para cada una de las plataformas evaluadas.

1) Data.imf.org. El Fondo Monetario Internacional tiene una plataforma de acceso a información macroeconómica y financiera. Ver Figura 46.

Figura 44 Encabezado de la plataforma del FMI.



Esta plataforma ofrece conjuntos de datos predeterminados en categorías generales, que luego se pueden analizar en categorías más pequeñas. Ver Figura 47

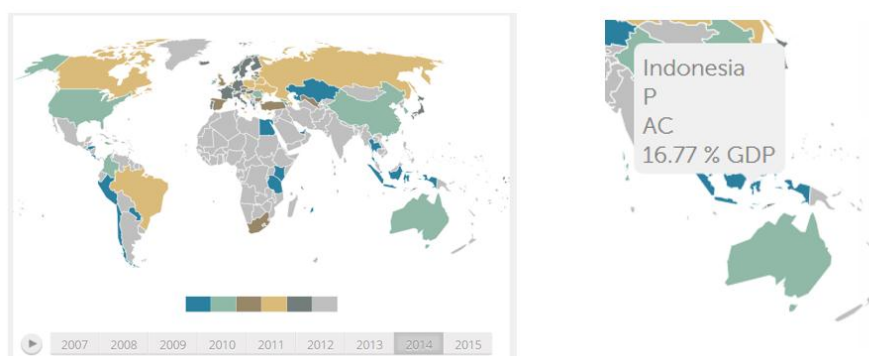
Figura 45 Categorías de datos.

GFS is divided into seven databases, as indicated below. To query and download data from any of the databases, simply click on the links below:

1. Government Finance Statistics (GFS), Main Aggregates and Balances
2. Government Finance Statistics (GFS), Revenue
3. Government Finance Statistics (GFS), Expense
4. Government Finance Statistics (GFS), Expenditure by Function of Government (COFOG)
5. Government Finance Statistics (GFS), Integrated Balance Sheet (Stock Positions and Flows in Assets and Liabilities)
6. Government Finance Statistics (GFS), Financial Assets and Liabilities by Counterpart Sector
7. Government Finance Statistics (GFS), Statement of Sources and Uses of Cash

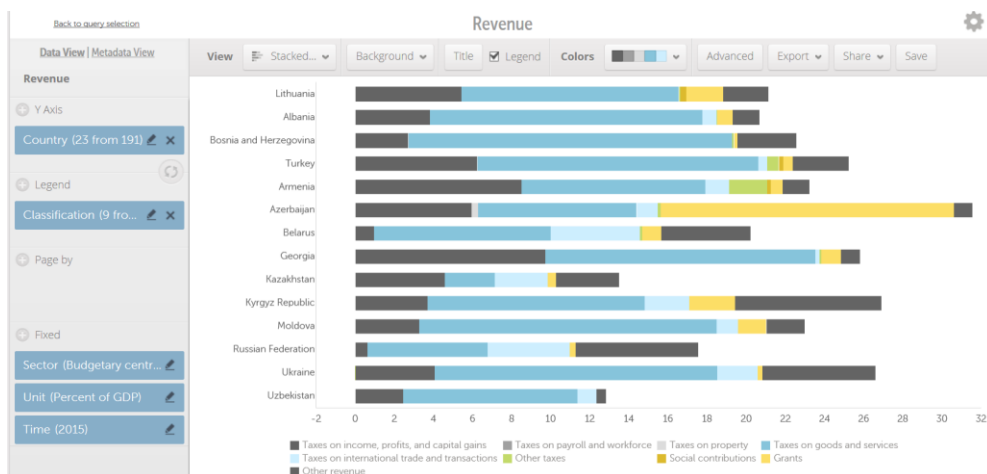
Estas categorías finalmente pueden analizarse a nivel mundial o por países específicos. Ver Figura 48.

Figura 46 Vista de datos por país.



La plataforma provee un editor de graficas en donde los indicadores se asocian con países específicos para cada una de las categorías o subcategorías para cualquier conjunto de datos. Ver Figura 49.

Figura 47 Editor de gráficas



a. Usabilidad de data.imf.org. A pesar de ser una plataforma muy avanzada y que maneja conjuntos muy grandes de datos, la plataforma del FMI es lenta, y la ayuda disponible para el usuario consiste en enviar un mensaje para luego esperar una respuesta, no existe un manual o glosario que explique el significado de los indicadores que se pueden seleccionar.

La organización de los datos y como se muestran al usuario es de primer nivel, se podría tomar de la plataforma del IMF la manera en la que dividen la data en categorías y subcategorías, para ayudar al usuario a encontrar de una manera más sencilla lo que están buscando.

Finalmente, respecto de la plataforma de análisis financiero del IMF, GobHash podría lograr un funcionamiento más rápido respecto de esta plataforma, también la capacidad de generar graficas diferentes a los gráficos de barras, y así proveer más puntos de interpretación.

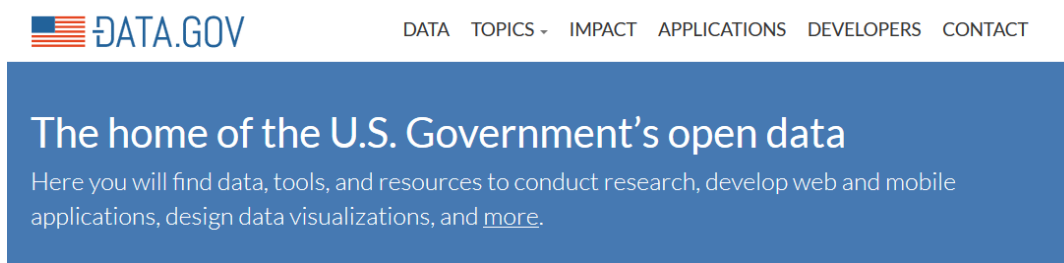
b. Funcionalidades de data.imf.org.

- Registro para creación de nuevo usuario
- Inicio de Sesión para autenticarse en IMF
- Búsqueda de sets de datos pre categorizados
- Creación de gráficos a partir de sets de datos seleccionados
- Vistas geográficas para sets de datos regionales

2) Data.gov

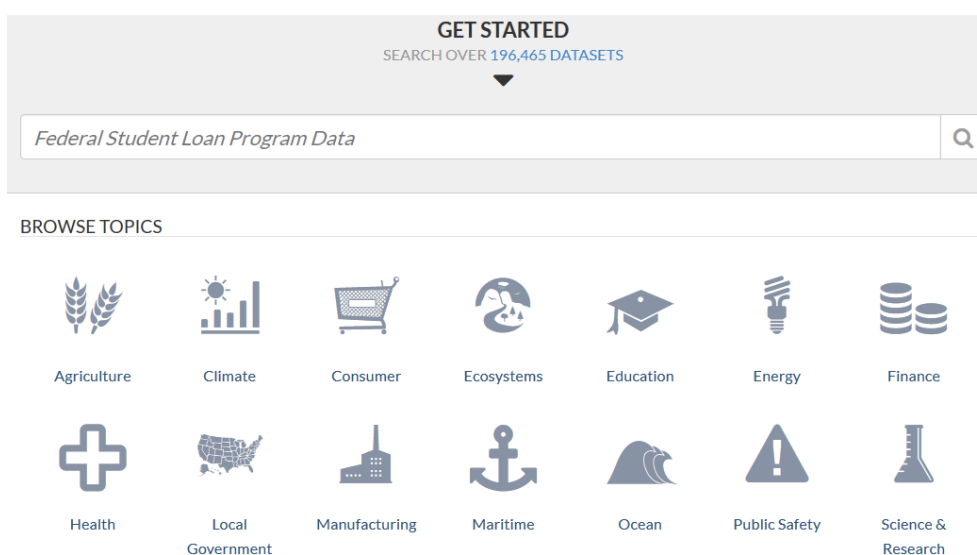
El Gobierno de los Estados Unidos de América tiene un portal donde provee toda la data pública a cualquier persona interesada en hacer uso de ella llamado data.gov. Ver Figura 50.

Figura 48 Encabezado de plataforma de datos abiertos de EEUU



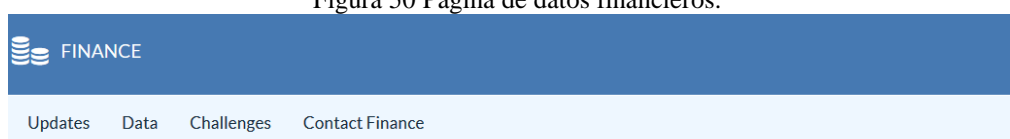
En este portal se encuentra que los sets de datos están divididos en diferentes categorías, facilitando la manera en la que los usuarios acceden a los datos. Ver Figura 51.

Figura 49 Categorías de sets de datos.



A su vez dentro de estas categorías se encuentran noticias y actualizaciones respecto a la data nueva y existente.

Figura 50 Página de datos financieros.



Explore hundreds of free data sets on financial services, including banking, lending, retirement, investments, and insurance. Start using these data sets to build new financial products and services, such as apps that help financial consumers and new models to help make loans to small businesses.

A la hora de buscar la data que se necesite hay varios filtros que se pueden aplicar además de una barra de búsqueda, incluyendo una búsqueda por mapa. Ver Figura 53.

Figura 53 Búsqueda de datos financieros.

The screenshot shows a search interface with a sidebar on the left and a main content area on the right. The sidebar includes a 'Filter by location' section with a search box and a map of the United States. Below the map is a 'Topics' section with a 'Clear All' button and a list of topics: Finance (129), Consumer (64), BusinessUSA (39), Energy (39), and AAPI (26). The main content area displays '129 datasets found' and lists several datasets with their respective descriptions and view counts. Each dataset entry includes a 'Federal' label in a blue ribbon on the right side.

- Housing Affordability Data System (HADS)** (456 recent views): Department of Housing and Urban Development – The Housing Affordability Data System (HADS) is a set of files derived from the 1985 and later national American Housing Survey (AHS) and the 2002 and later Metro...
- American FactFinder II** (317 recent views): Department of Commerce – American FactFinder is the Census Bureau's online, self-service tool designed to search a variety of population, economic, geographic and housing information.
- National Flood Hazard Layer (NFHL)** (230 recent views): Department of Homeland Security – The National Flood Hazard Layer (NFHL) is a compilation of GIS data that comprises a nationwide digital Flood Insurance Rate Map. The GIS data and services are...
- Economic Census** (217 recent views): Department of Commerce – Provides a detailed portrait of business activities in industries and communities once every five years, from the national to the local level.

Dentro del detalle de cada set de datos se encuentra fecha de creación, referencias, etiquetas, y reseñas de usuarios sobre estos datos, todo mostrado de una manera amigable como un perfil.

Figura 51 Detalle de set de datos.

The screenshot shows the detail page for the 'American Community Survey' dataset. The page is organized into several sections:

- Bureau of the Census**: Includes the logo and the text 'Bureau of the Census'.
- Topics**: A list of topics including AAPI, BusinessUSA, Consumer, Education, Energy, Finance, and Manufacturing.
- Publisher**: US Census Bureau, Department of Commerce.
- Contact**: Sharon Stern.
- Share on Social Sites**: Links for Google+, Twitter, and Facebook.
- American Community Survey**: Title of the dataset, with 'Metadata Updated: September 17, 2015'.
- Description**: An annual nationwide survey that collects information such as age, race, income, commute time to work, home value, veteran status, and other data.
- Access & Use Information**: Includes 'Public: This dataset is intended for public access and use.' and 'License: U.S. Government Work'.
- Downloads & Resources**: Includes a 'Web Page' link with '8363 views' and a 'Visit page' button.
- References**: A text box containing the URL 'http://www.census.gov/acs/www/methodology/methodology_main'.
- Dates**: A table with the following data:

Metadata Created Date	May 20, 2015
Metadata Updated Date	September 17, 2015
Data Update Frequency	R/P1Y
- Metadata Source**: Includes a 'Data.json Metadata' link and a 'Download Metadata' button.

Como un agregado, el portal de data.gov publica competencias para todos los usuarios, a manera de competencia publica para el desarrollo de soluciones tecnológicas a partir de la data pública.

Figura 52 Competencia de área financiera.

Finance Challenges

Completed Challenges

MyMoneyAppUp Challenge

The Department of Treasury wants your ideas and designs for next generation mobile apps to help Americans shape their financial futures.



Winner Announced!



En el caso de data.gov, muchas veces no es necesario guardar la información en los servidores del portal, ya que todas las entidades de gobierno ofrecen su data publica en sus propios sitios, sin embargo, data.gov sirve a manera de motor de búsqueda para estos datos y muchas veces solamente ofrece la redirección a los sitios de las entidades de gobierno que tiene la información que se necesita.

a. Usabilidad de data.gov. El sitio data.gov es simple, intuitivo, y bastante rápido, es importante mencionar que al funcionar solamente como un motor de búsqueda de datos en muchos casos, la carga que el sitio recibe no es mucha, sin embargo, realiza muy bien esta tarea.

Entre sus funcionalidades no contiene la autenticación, ya que solamente los funcionarios y trabajadores de gobierno tienen acceso mediante su identificación del gobierno de USA. Lo que nos hace pensar que a pesar de ser un portal que ofrece data publica, su uso no está enfocado en la sociedad en general sino en el uso de los empleados de gobierno.

Es muy interesante también las competencias que liberan para realizar un *crowdsourcing* para sus ideas a la hora de realizar aplicaciones que utilicen estos datos para facilitar el uso de su población, esta es una prueba clara de gobierno abierto, y de cómo la población se incluye en los procesos de gobierno mediante esta clase de sitios.

b. Funcionalidades de data.gov

- Búsqueda de sets de datos categorizados.
- Vistas geográficas para sets de datos regionales.
- Redirección a sitios de gobierno para la extracción de datos.
- Uso de etiquetas a la hora de crear nuevos sets de datos

- Capacidad de reseña por parte de los usuarios

3) Data.gov.uk. El Reino Unido tiene un portal de datos y gobierno abierto llamado data.gov.uk. Ver Figura 56.

Figura 53 Encabezado de la plataforma de datos abiertos del Reino Unido.



Search for data

Find data published by government departments and agencies, public bodies and local authorities. You can use this data to learn more about how government works, carry out research or build applications and services.

En este portal se encuentra la data segmentada en distintos ámbitos. Entre los cuales podemos encontrar las adquisiciones del Gobierno.

Figura 54 Categorización de datos.

Browse data by theme

Business and economy

Small businesses, industry, imports, exports and trade

Crime and justice

Courts, police, prisons, offenders, borders and immigration

Defence

Armed forces, health and safety, search and rescue

Education

Students, training, qualifications and the National Curriculum

Environment

Weather, flooding, rivers, air quality, geology and agriculture

Government

Staff numbers and pay, local councillors and department business plans

Government spending

Includes all payments by government departments over £25,000

Health

Includes smoking, drugs, alcohol, medicine performance and hospitals

Mapping

Addresses, boundaries, land ownership, aerial photographs, seabed and land terrain

Society

Employment, benefits, household finances, poverty and population

Towns and cities

Includes housing, urban planning, leisure, waste and energy consumption

Transport

Airports, roads, freight, electric vehicles, parking, buses and footpaths

Además de esta segmentación, existe una barra de búsqueda en donde se encuentra por medio de palabras clave lo que se necesite.

Figura 55 Búsqueda de datos.

Como una característica interesante, el usuario puede realizar una búsqueda geográfica detallada por medio de un mapa con el propósito de encontrar toda la información de un área específica deseada.

Figura 59 Búsqueda en mapa.

De esta manera, la búsqueda no se hace por un ámbito o serie de palabras clave en específico, sino por una región especificada.

Figura 56 Búsqueda de zona específica.

5,183 results

Sort by: Location

Air Quality Monitoring Area

South Cambridgeshire District Council

Displays the area covered by the South Cambridgeshire District Council 'Air Quality Monitoring Area' (AQMA).

Environment

HTML WMS

[Preview on Map](#)

[Add to Preview List](#)

Article 4 Directions

South Cambridgeshire District Council

This dataset shows the currently active Article 4 Directions issued by South Cambridgeshire District Council.

Towns & Cities

HTML WMS

[Preview on Map](#)

[Add to Preview List](#)

Finalmente, dentro de cada publicación se puede encontrar toda la información respecto de su origen y uso, además de un ranking de apertura para cada uno. Ver Figura 62.

Figura 57 Detalle de set de datos.

Los datos están disponibles en una variedad de formatos, dependiendo de la publicación.

Figura 58 Formatos de set de datos.

The screenshot displays two sections of data resources. The first section, 'DATA RESOURCES (2)', contains two items: 'INSPIRE View Service' with a 'WMS' icon and links for 'Details' and 'Link to WMS'; and 'INSPIRE Download Service' with an 'HTML' icon and links for 'Details' and 'Link'. The second section, 'ADDITIONAL LINKS (2)', contains two items: 'Source GEMINI2 record' with an 'XML' icon and a 'Link to XML' link; and 'Source GEMINI2 record (formatted)' with an 'HTML' icon and a 'Link'.

También se encuentra información adicional detallada.

Figura 59 Información adicional de set de datos.

ADDITIONAL INFORMATION	
Added to data.gov.uk	19/03/2015
Theme	Environment
Themes (secondary)	Government
Temporal coverage	17/3/2015
Harvest URL	http://inspire.misoportal.com/metadata/files/South_Cambridgshire
Harvest date	19/03/2015 09:50
Metadata date	17/3/2015
Harvest GUID	083433ba-0135-4fd4-94b2-5264fbc11794
Harvest source reference	http://inspire.misoportal.com/metadata/files/South_Cambridgshire/Air_Quality_Monitoring_Area.xml
Extent	Latitude: 52.3963391168° to 51.9584371983° Longitude: -0.2395648488° to 0.4333477488°
Spatial reference system	http://www.opengis.net/def/crs/EPSG/0/27700
Dataset reference date	17/3/2015 (publication)
Frequency of update	asNeeded
Responsible party	South Cambridgeshire District Council (owner)
Access constraints	None
ISO19139 resource type	dataset
Metadata language	English
Schema/Vocabulary	No value
Code list	No value
Service Level	No value

Como una característica nueva, data.gov.uk, presenta una sección donde todas las aplicaciones desarrolladas por usuarios haciendo uso de los datos disponibles en el portal, se presentan.

Figura 60 Vista de aplicaciones.

The screenshot shows a web interface for viewing applications. On the left is a sidebar with a search bar and filters for 'CONTENT TYPE' (App) and 'CATEGORY' (Transportation, Location, Society, Health, Finance, Environment, Education, Government, Linked data, Administration, Spending data, Policy). The main area displays '421 Apps' sorted by 'Last updated'. Two app cards are visible: 'Travel Tools Dashboard' and 'Is the Grass Greener?'. Each card includes submission details, a description, and a set of three small screenshots.

Dentro de cada una de estas aplicaciones se puede encontrar su uso, que información o sets de datos específicos del portal utiliza, su precio y reseñas de usuarios. Ver Figura 66.

Figura 61 Vista detallada de App.

This is a detailed view of the 'Is the Grass Greener?' app. It features a title, submission information (submitted by OnlineMortgageA... on Wed, 26/07/2017 - 11:45 | Updated on Wed, 26/07/2017 - 12:45), a star rating (Average: 5 (3 votes)), and a category tag 'Location'. The developer is 'Impression' and the app is free. A row of tags includes 'Cost of Living', 'Quality of Life', 'compare UK cities', 'compare UK house prices', 'compare GDHI', 'compare gross disposable household income', 'compare crime rates', and 'london boroughs'. The description explains that the app compares property listing against government data and national statistics to find the cost and quality of living in any two UK cities or London boroughs. It concludes that users can find out if 'the grass really is greener on the other side' with a few clicks. The 'Uses dataset' section lists 'House Price Index', 'Gross Disposable Household Income (GDHI)', and 'CCTV'.

a. Usabilidad de data.gov.uk. El portal de data.gov.uk es una iniciativa completamente enfocada en la población en general, tiene varias características que la vuelven incluyente, además de sencilla de usar. En data.gov.uk, es posible tener un usuario para autenticarse, esta cuenta puede ser utilizada para registrar la búsqueda de datos, publicar aplicaciones como desarrollador, entre otros.

Entre las características más destacables de data.gov.uk, resalta el uso de un mapa para realizar una búsqueda regional, en el caso de estudios o análisis respecto de solamente una región específica, es interesante lo sencillo que es encontrar cualquier tipo de datos que pertenezcan a esta región en específico por medio de esta búsqueda. Vale la pena mencionar que en países descentralizados y grandes como este es una muestra clara de descentralización de la información.

Finalmente, la naturaleza de los datos y los usos que se les da, además de la manera en la que los ofrecen, es muy cotidiana y favorece su uso para la población en general al ser fácil de utilizar y al contener datos de interés para la población en general.

b. Funcionalidades de data.gov.uk

- Autenticación
- Búsqueda por medio de palabras clave
- Capacidad de publicar aplicaciones desarrolladas que usen estos datos
- Información segmentada para su fácil navegación
 - Ranking de apertura para las publicaciones

Las plataformas evaluadas son grandes exponentes de los conceptos de datos abiertos y gobiernos abierto, sin embargo, tienen algunas falencias además de muchas ventajas, antes de definir que funcionalidades de las vistas con anterioridad se implementaran en GobHash es importante hacer un análisis comparativo de las plataformas evaluadas con el propósito de replicar las mejores características y evitar los mismos errores. A continuación, se muestra el análisis de las plataformas evaluadas en el análisis de competencia.

Con base en lo expuesto en el cuadro anterior se propone hacer de GobHash una plataforma con autenticación y creación de gráficos y además agregar interacción entre los usuarios y métodos de búsqueda de información simplificados, se propone también desde el punto de vista técnico hacer que la plataforma sea lo más rápida y simple posible para no causarle problemas al usuario.

Tabla 12 Comparación de plataformas evaluadas en análisis de competencia.

Plataforma	Descripción	Funcionalidades	Ventajas	Desventajas
data.imf.org	data.imf.org es el portal de datos abiertos del fondo monetario internacional (IMF)	a) Registro para creación de nuevo usuario b) Inicio de Sesión para autenticarse en IMF c) Búsqueda de sets de datos pre categorizados d) Creación de gráficos a partir de sets de datos seleccionados e) Vistas geográficas para sets de datos regionales	data.imf.org provee una potente herramienta de visualización y manipulación de datos en donde se pueden generar los indicadores deseados a partir de un set de datos que luego pueden ser exportados en formatos apropiados para su uso y manipulación	La plataforma es demasiado lenta y muy cargada de información por lo que le dificulta mucho al usuario nuevo poder entender o operar la plataforma.
data.gov.uk	data.gov.uk es la plataforma de datos y gobierno abiertos del gobierno del Reino Unido	a) Autenticación b) Búsqueda por medio de palabras clave c) Capacidad de publicar aplicaciones desarrolladas que usen estos datos d) Información segmentada para su fácil navegación e) Ranking de apertura para las publicaciones	data.gov.uk provee muchos datos al público. Además de varias formas de búsqueda amigable como la búsqueda en el mapa para encontrar los datos disponibles de una región, además provee un espacio donde los ciudadanos pueden mostrar los frutos de su trabajo con la data en forma de sistemas o aplicaciones.	El riesgo que tiene data.gov.uk al permitir que cualquier persona suba datos a la plataforma es que muchas veces dentro de las publicaciones de sets de datos en los comentarios se puede observar que la gente denuncia que los datos están mal por lo que la plataforma pierde credibilidad
data.gov	data.gov es la plataforma de datos y gobierno abiertos del gobierno de los Estados Unidos de América	a) Búsqueda de sets de datos categorizados. b) Vistas geográficas para sets de datos regionales. c) Redirección a sitios de gobierno para la extracción de datos. d) Uso de etiquetas a la hora de crear nuevos sets de datos e) capacidad de reseña por parte de los usuarios	Data.gov ofrece una categorización de datos más amigable para el público que no sabe que busca, además el gobierno de los Estados Unidos de América utiliza esta plataforma como un <i>crowdfunding</i> de ideas para como solucionar problemas con sus datos por medio de competencias organizadas por el gobierno.	la gran desventaja de data.gov es que el ciudadano común no puede tener una cuenta, aunque cualquier persona puede acceder a los datos, la operación y mantenimiento del contenido de la plataforma es exclusiva de empleados del Gobierno de los Estados Unidos de América

d. Usuarios finales. Entre los usuarios finales de GobHash, se definieron tres perfiles de usuarios que son de importancia durante el proceso de desarrollo y prueba de la plataforma.

- Medios de comunicación, periodistas, reporteros o estudiantes. Forman parte del perfil de usuario ya que pueden darle un uso práctico a la plataforma para generar indicadores que puedan usar en noticias o investigación periodística, además pueden usar la plataforma como un vehículo de exploración en búsqueda de nuevas notas.

- Académicos en el área de Economía, analistas, profesionales o estudiantes. Los Economistas son un perfil de usuario importante ya que pueden utilizar la plataforma con un propósito académico, generando indicadores que los pueden ayudar a hacer análisis económico y econométrico. Ayudándolos en su vida académica y profesional.

- Sociedad civil en general. El enfoque de red social de GobHash convierte a la sociedad civil en General en el perfil de usuario final, ya que son los usuarios que crearan sus cuentas, generaran contenido y discusión, dotando de verdadera riqueza a la plataforma y logrando los objetivos planteados.

e. Definición de métricas

- Relación de éxito. La relación de éxito refiere al número de éxitos contra el número de intentos para una tarea específica dentro del estudio de usabilidad, esta es una de las métricas más importantes para medir funcionamiento y el planeamiento del flujo de aplicación. Para la plataforma Gobhash, se apunta a tener un 90% de relación de éxito para todas las tareas, más adelante se definirá el valor exacto de esta métrica para cada tarea específica.

- Relación de error. Al igual que la relación de éxito, la relación de error refiere a la cantidad de errores por cantidad de intentos para cualquier tarea, en el estudio de usabilidad, al utilizar una primera versión funcional del producto final, se busca una relación de error crítico menor al 10%, donde error crítico

refiere a un error que no permite al usuario completar la tarea que se le presenta, siendo todos los demás errores no críticos.

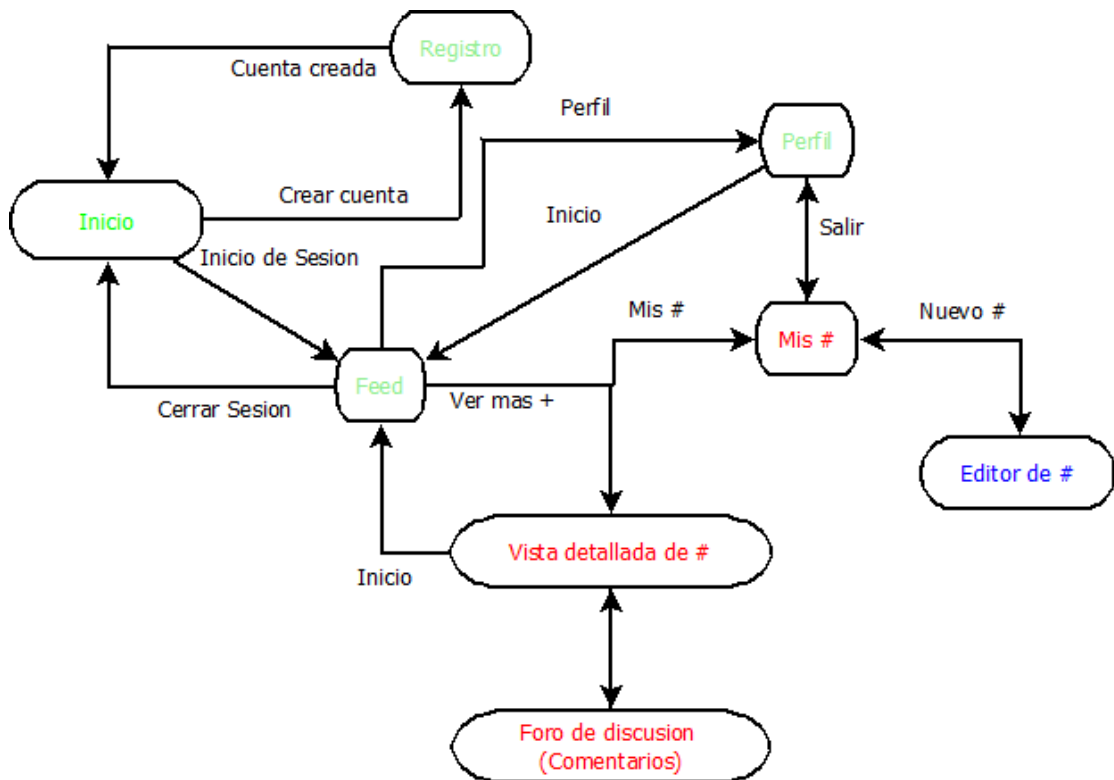
- Tiempo de finalización. Para cada tarea también se medirá el tiempo en el que las mismas se completan, fijando un tiempo límite de 5 minutos, los tiempos esperados para cada tarea se definirán luego de la prueba piloto, el tiempo de finalización es importante ya que es una métrica muy útil para medir la usabilidad más que la funcionalidad.

- Medidas subjetivas. Las medidas subjetivas se refieren a una escala en la que el usuario mide la facilidad para completar cada tarea, en una escala de 1-10, se espera tener una media mayor a 6 para todas las tareas luego de las pruebas de usabilidad.

f. Curva de aprendizaje. La característica abierta y social de GobHash nos da cierta responsabilidad respecto de la usabilidad y facilidad de operación de la plataforma, la curva de aprendizaje debe ser lo más baja posible para los usuarios sin experiencia o conocimientos profundos en el área.

g. Diagrama de flujo de aplicación. Como parte del estudio de estándar y usabilidad se construyó un diagrama de flujo de aplicación que muestra la conexión entre partes de la plataforma, este diagrama definió el flujo de aplicación a diseñar en los prototipos de diseño. Ver Figura 67.

Figura 62 Diagrama de flujo de aplicación.



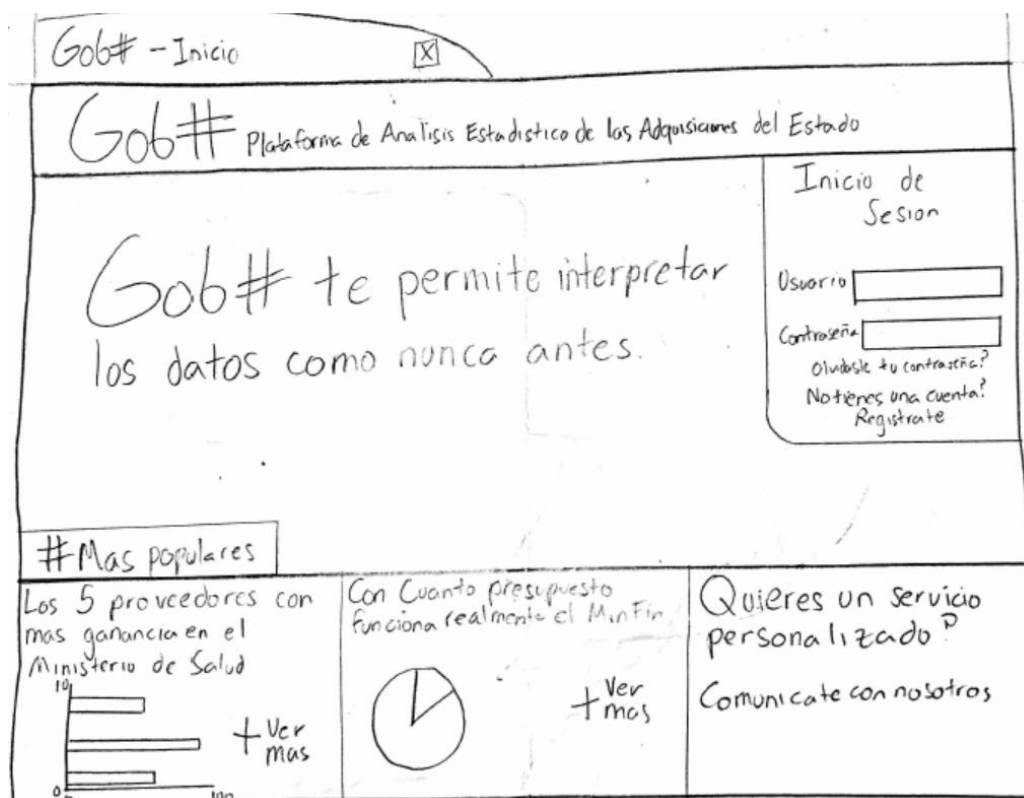
2. Fase II: Desarrollo de prototipos de diseño

a. Prototipos en papel. El desarrollo de prototipos en papel se llevó a cabo inmediatamente después del estudio de estándar y usabilidad con el objetivo de desarrollar la primera versión de los prototipos y enfocándose principalmente en el mapeo de botones, diseño general de la plataforma y el flujo de aplicación. A continuación se muestra el detalle de pantallas diseñadas en esta primera versión del prototipo.

- Inicio. La pantalla de inicio es la pantalla que el usuario ve primero cuando ingresa a la plataforma y contiene características clave en el correcto funcionamiento de la plataforma, además de esto tiene que tener contenido que interese al usuario y que lo haga quedarse en la plataforma, tiene que ser simple pero informativa y tiene que ser capaz de transmitir el mensaje del funcionamiento de la plataforma de una forma sencilla. A continuación se presenta un listado de características de la página de inicio. Ver Figura 22.

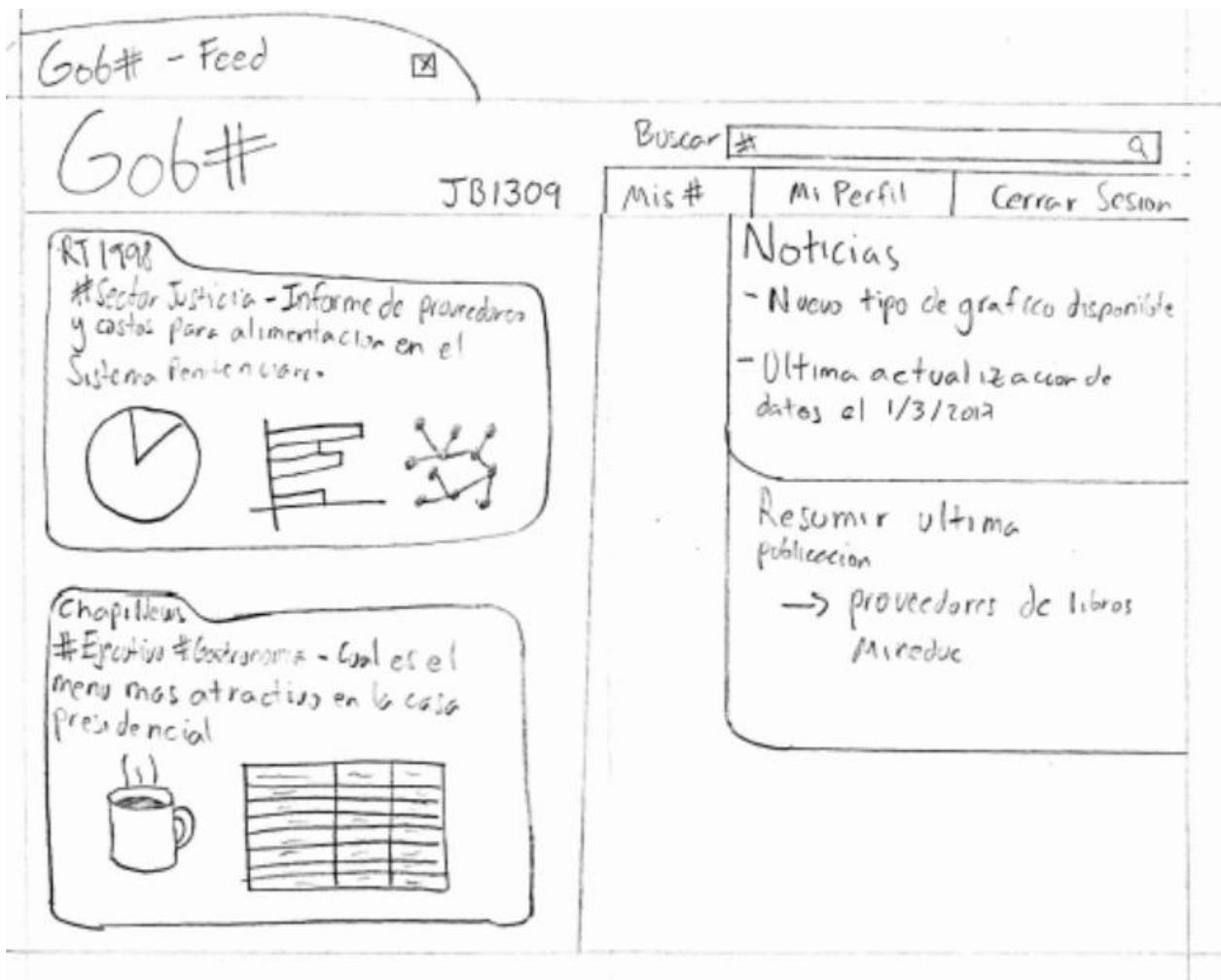
- Inicio de sesión
- Registro de nuevos usuarios
- Muestra de publicaciones más populares

Figura 63 Pantalla de inicio – Prototipo dibujado.



- Pantalla principal. La pantalla principal es la pantalla a la que llega el usuario inmediatamente después de iniciar sesión, en el diseño de esta pantalla se apegó a la memoria física de los usuarios respecto de redes sociales existentes, en búsqueda de que no se confundan al ingresar a la plataforma. La pantalla principal albergara todas las publicaciones que el usuario debería ver, además de permitir el acceso al resto de la plataforma. A continuación se muestra el detalle de las características de la pantalla principal. Ver Figura 69.

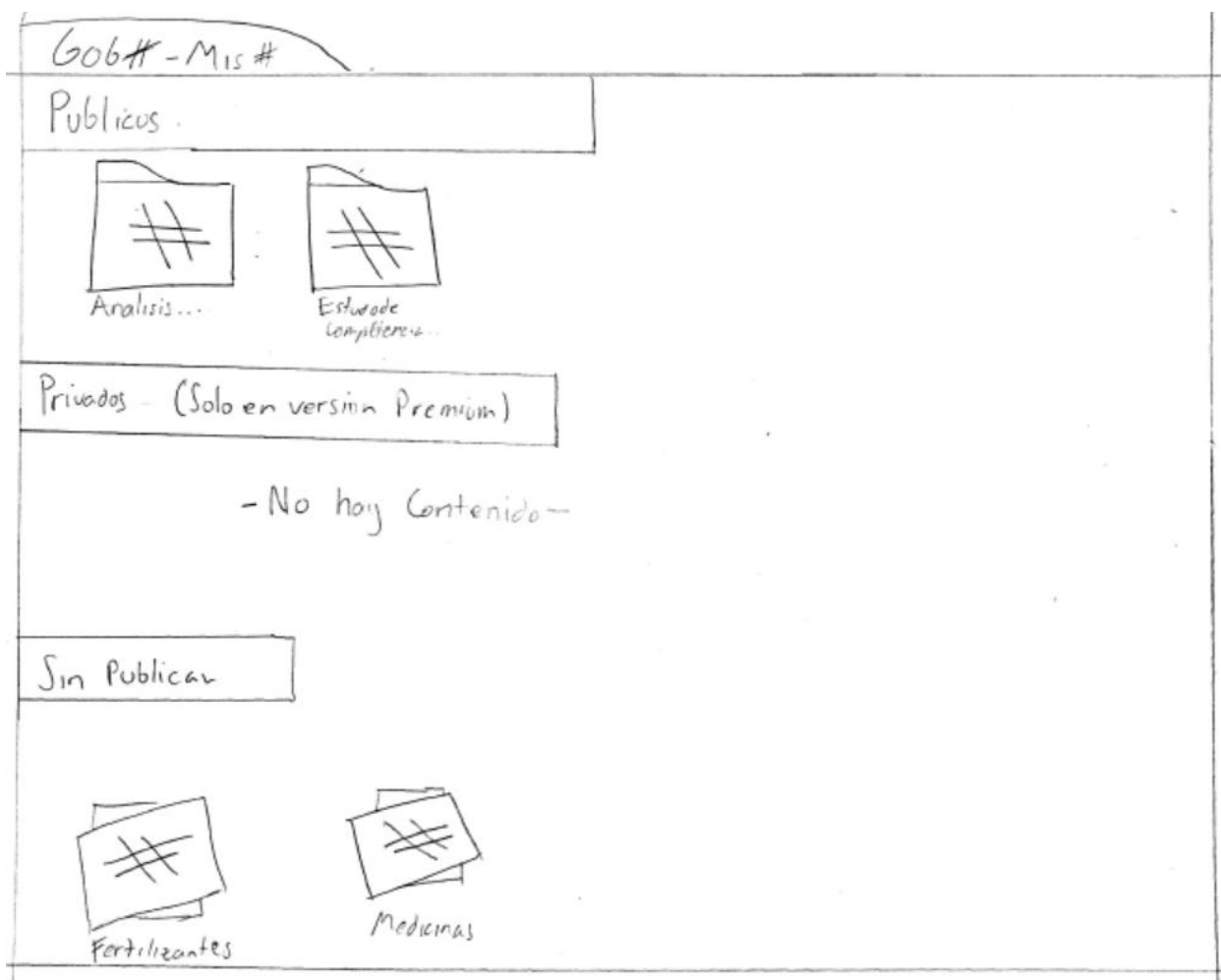
Figura 64 Pantalla principal – Prototipo dibujado.



- Mis Hash. En la pantalla de mis hash se tienen todos los hash de un usuario, un hash es el análogo de una publicación en cualquier otra red social. En la pantalla de mis hash podemos ubicar todas nuestras publicaciones. En el prototipo en papel, aun se manejaba el concepto de un esquema de monetización *Freemium* por medio del cobro por la capacidad de que algunos de los hash sean privados, sin embargo, esta característica ya no se manejó en ninguno de los siguientes prototipos, a continuación se muestra el detalle de las características de la pantalla de mis hash.

- Sección de hash públicos
- Sección de hash privados
- Sección de hash sin publicar

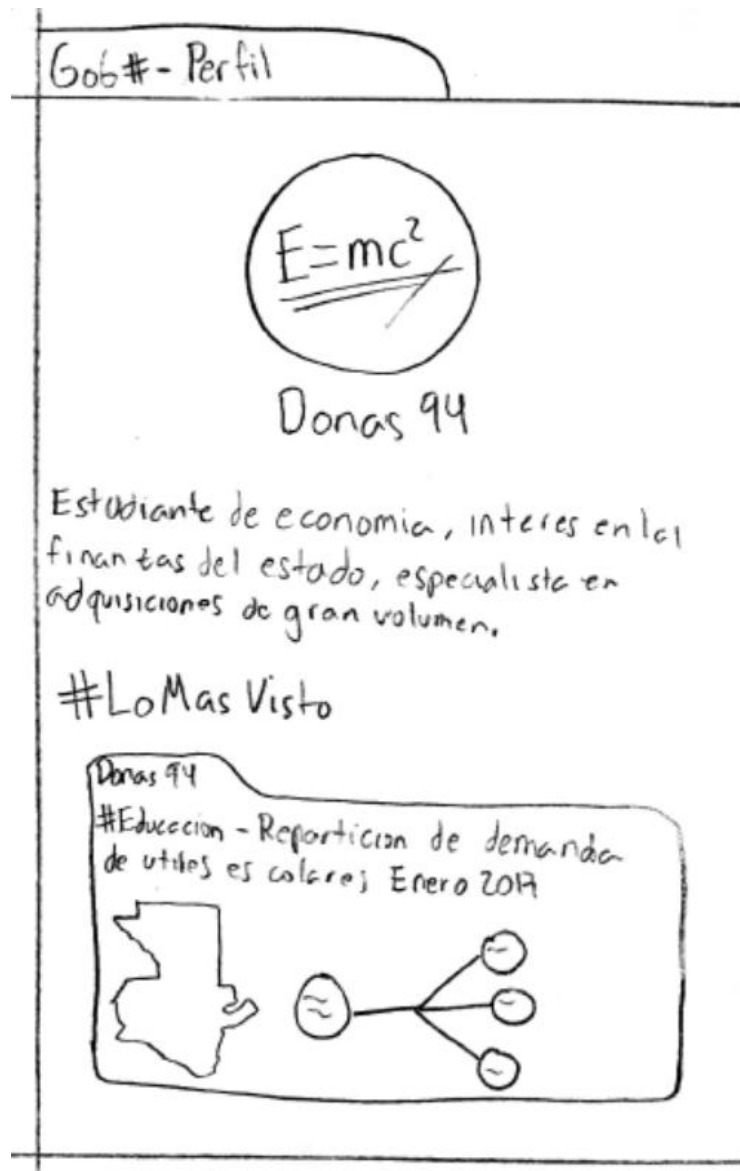
Figura 65 Pantalla mis hash – Prototipo dibujado.



• Pantalla de perfil de usuario. La pantalla de perfil de usuario contiene toda la información necesaria del usuario en la plataforma, en GobHash, el usuario puede tener un alias, una descripción y una imagen en su perfil. El objetivo es que el usuario pueda expresar sus intereses sin necesidad de mostrar mucha de su información personal, además, el perfil contiene el hash más visto del usuario. En los prototipos en papel aún no se manejaba el concepto de seguidor/seguido, que se implementó en los siguientes prototipos. A continuación, se muestra el detalle de las características de la pantalla del perfil de usuario.

- Imagen o avatar del usuario
- Alias o nombre de usuario
- Descripción de perfil
- Lo más visto del usuario.

Figura 66 Pantalla de perfil de usuario – Prototipo dibujado.



- Pantalla de registro de usuario

En la pantalla de registro de usuario es donde se solicita la información necesaria a cualquier usuario que quiera tener una cuenta en la plataforma, por la naturaleza de GovHash, no se solicita mucha información al usuario por lo que es una pantalla relativamente simple. Sin embargo, el registro es una pieza clave de la plataforma, por lo que su diseño era importante sin importar la simpleza de su función. A continuación se muestran las características de la pantalla de registro de usuario.

- 1) Registro del usuario
- 2) Clausula de términos y condiciones

Figura 67 Pantalla de registro de usuario – Prototipo dibujado.

Gob# - Registro

Crea una nueva
Cuenta de Gob#

- Nombre de usuario
—

- Contraseña
—
Confirmar Contraseña
—

- Correo Electronico
—

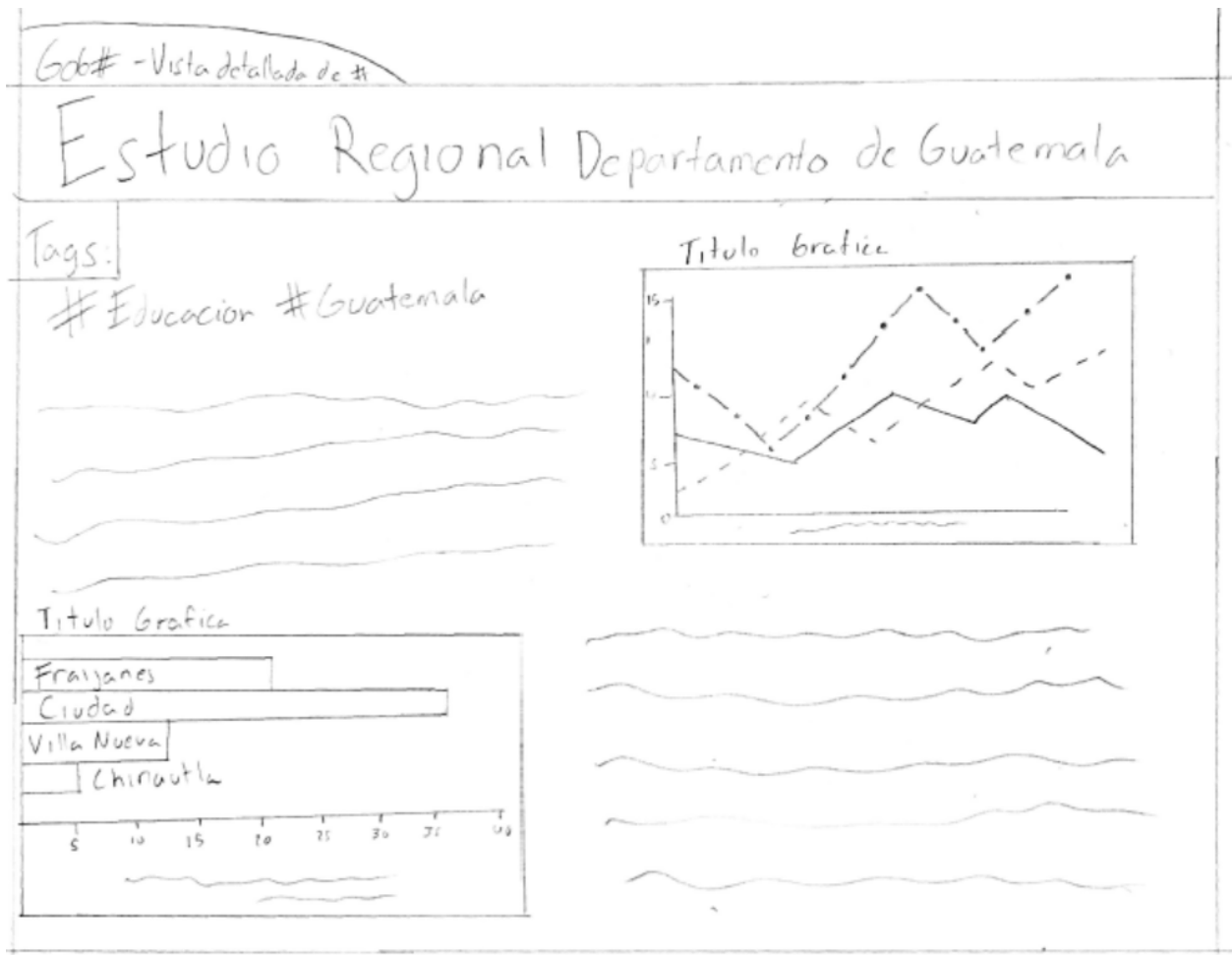
Acepto los terminos y condiciones de Gob#

Crear Cuenta

- Pantalla de vista detallada de Hash. La pantalla de vista detallada de hash es la pantalla que ve el usuario cuando ingresa a un hash desde la pantalla principal o desde el perfil de un usuario. En el prototipo en papel aún no se tenía la certeza de cómo iba a ser la plantilla de visualización detallada de hash, sin embargo, se tenían las etiquetas para hacer que el hash fuera encontrado fácilmente por su contenido. A continuación, se detallan las características de la pantalla de vista detallada de hash. Ver Figura 27.

- Título de Hash
- Etiquetas de Hash
- Espacio para distintas gráficas y descripciones.

Figura 68 Vista de detalle de hash – Prototipo dibujado.



b. Prototipos diseñados. En el desarrollo de los prototipos diseñados se incorporó la fuente y paleta de colores al diseño de la plataforma, La paleta de colores seleccionada para la plataforma fue una mezcla de naranjas, grises y blancos. Las razones principales de esto son dos, la primera refiere al hecho de que GobHash no es una página estatal ni creada por el gobierno, razón por la cual evitamos usar colores de la bandera nacional o similar, la segunda razón fue que queríamos dar una sensación de modernismo y creatividad, de esta forma, al ser el gris claro un color que evoca una sensación de tranquilidad y da la sensación de un ambiente sofisticado al usuario, todo esto mezclado con el naranja conocido por ser un color alegre y un disparador natural de la creatividad, transmitimos la sensación de querer generar contenido al usuario, logrando el cometido de que utilice la plataforma. Ver Figura 74. (Bourn, 2011)

Figura 69 Logo de GobHash con la paleta de colores.

The logo consists of the text "GOBHASH#" in a bold, sans-serif font. The letters "GOB" are white, and "HASH#" are orange. The text is set against a dark grey background.

A continuación, se muestra el detalle de las pantallas del prototipo diseñado, solamente se resaltarán los cambios respecto al prototipo en papel en cada una de las pantallas.

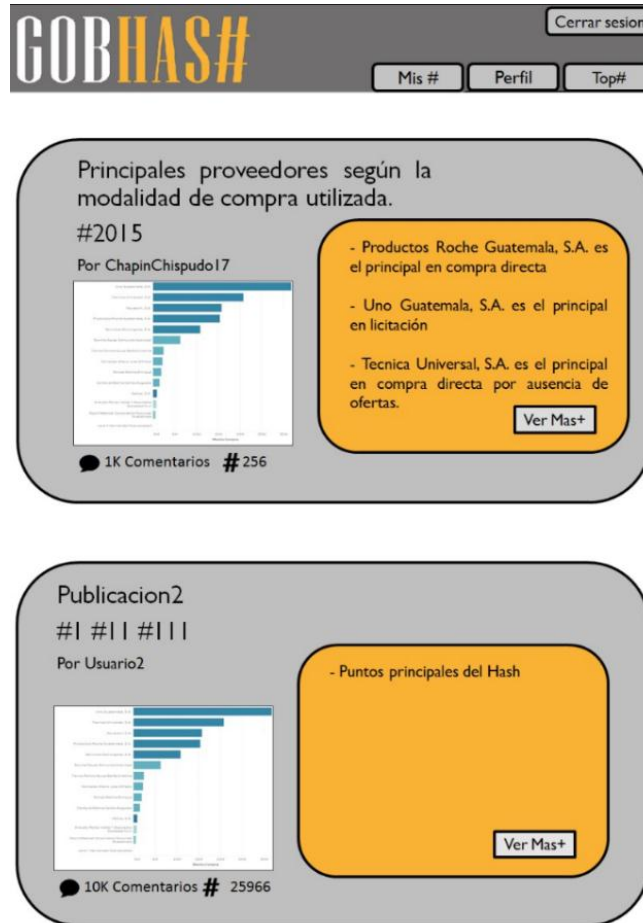
1. Pantalla de inicio. La pantalla de inicio prácticamente se mantiene igual respecto a los prototipos en papel, sin embargo, se agregan botones para ingresar a la vista detallada de los hash que salen en la pantalla de inicio. Ver Figura 75.

2. Pantalla principal La pantalla principal si sufre varios cambios respecto del prototipo dibujado, la ubicación de las publicaciones cambia y ahora ocupa toda la pantalla la barra de búsqueda se cambia por un botón de *top#*, a manera de exploración de nuevas publicaciones. En la pantalla de inicio se agregó un fondo blanco para arreglarle claridad a la vista de la pantalla principal. Ver Figura 75.

Figura 70 Pantalla de inicio – Prototipo diseñado.



Figura 71 Pantalla principal – Prototipo diseñado.



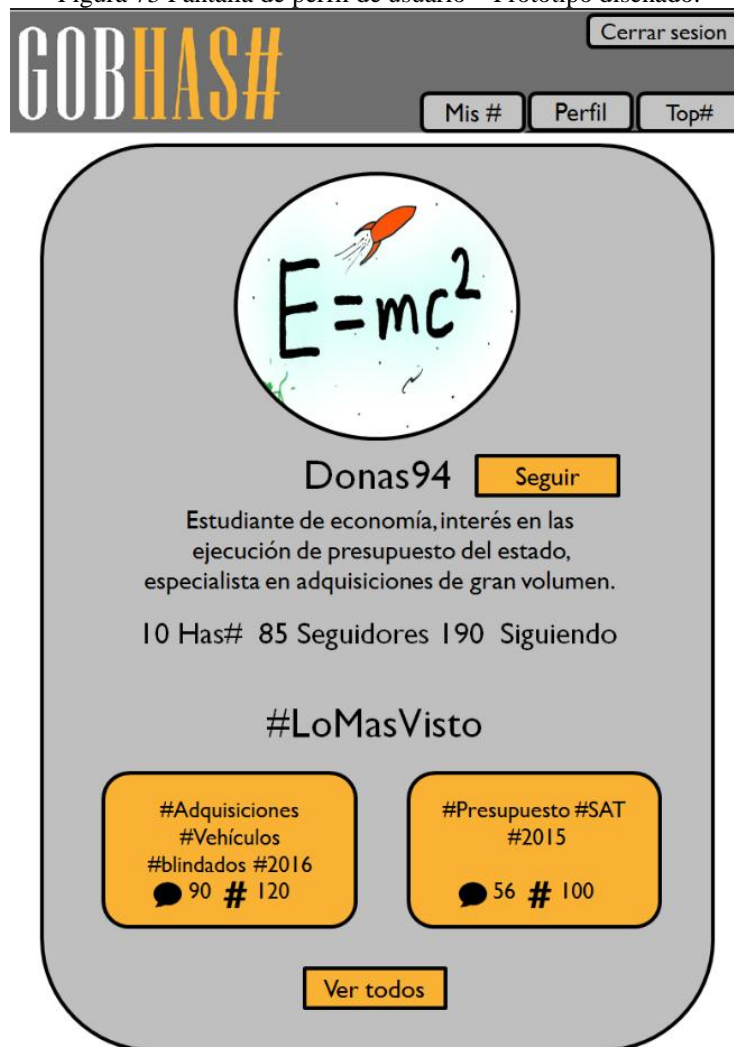
3. Mis hash. La pantalla de mis hash cambia respecto del prototipo dibujado eliminando las secciones de hash privados y no completados. La pantalla de mis hash se simplifica dejando solamente los hash publicados y a diferencia del prototipo dibujado ya tiene el botón para crear un nuevo hash, que fue agregado luego de las validaciones. Ver Figura 77.

Figura 72 Pantalla de mis Hash – Prototipo diseñado.



4. Pantalla de perfil de usuario. En la pantalla de perfil de usuario no se ven muchos cambios respecto del prototipo dibujado, sin embargo el único gran cambio fue la introducción de un modelo de seguidores a la plataforma, de forma que en el perfil del usuario se tenga un contador de seguidores, seguidos y de publicaciones creadas. Ver Figura 78.

Figura 73 Pantalla de perfil de usuario – Prototipo diseñado.



5. Pantalla de registro de usuario. En la pantalla de registro de usuario, se mantienen todas las características del prototipo diseñado, se eliminaron los términos y condiciones ya que por la naturaleza pública de los datos no es necesario que el usuario este de acuerdo con ningún termino o condición. Ver Figura 33.

Figura 74 Pantalla de registro de usuario – Prototipo diseñado.



Registro

Nombre de usuario

Correo Electrónico

Contraseña

Confirmar contraseña

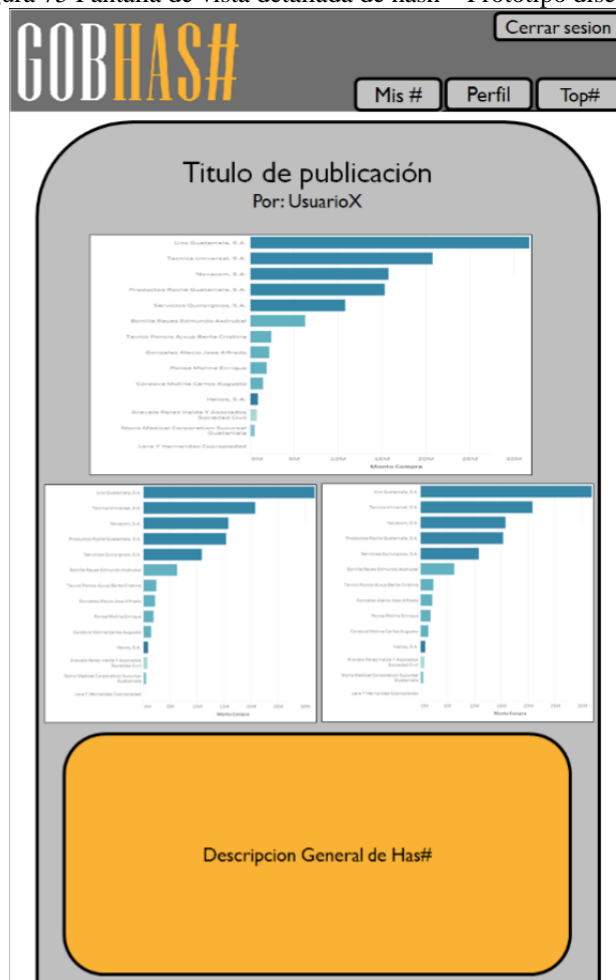
Crear Cuenta

GOB #

The image shows a user registration form with four input fields and a 'Crear Cuenta' button. To the right of the form is a logo featuring the letters 'GOB' in white on a dark grey background, with a white line graph showing an upward trend and a yellow hashtag symbol below it.

6. Pantalla de vista detallada de hash Durante el proceso de diseño de los prototipos diseñados se manejaron varias opciones para el diseño del detalle de hash, sin embargo se optó por un diseño que orientara los gráficos arriba y la descripción abajo. Ver Figura 34. En el anexo se pueden observar las opciones que se tenían para la vista detallada de hash.

Figura 75 Pantalla de vista detallada de hash – Prototipo diseñado.



Finalmente, en la sección de anexos se pueden apreciar el resto de diseños del prototipo diseñado, tales como la sección de comentarios, la confirmación de cierre de sesión y la recuperación de la contraseña.

c. **Prototipos interactivos.** Los prototipos interactivos fueron la última iteración del desarrollo y validación de prototipos, estos fueron realizados con la misma maquetación de la primera versión funcional de la plataforma y el funcionamiento es prácticamente el mismo excepto por el hecho de que por ser un prototipo interactivo no utiliza datos reales para su funcionamiento. A continuación, se detallan los aspectos de este prototipo final.

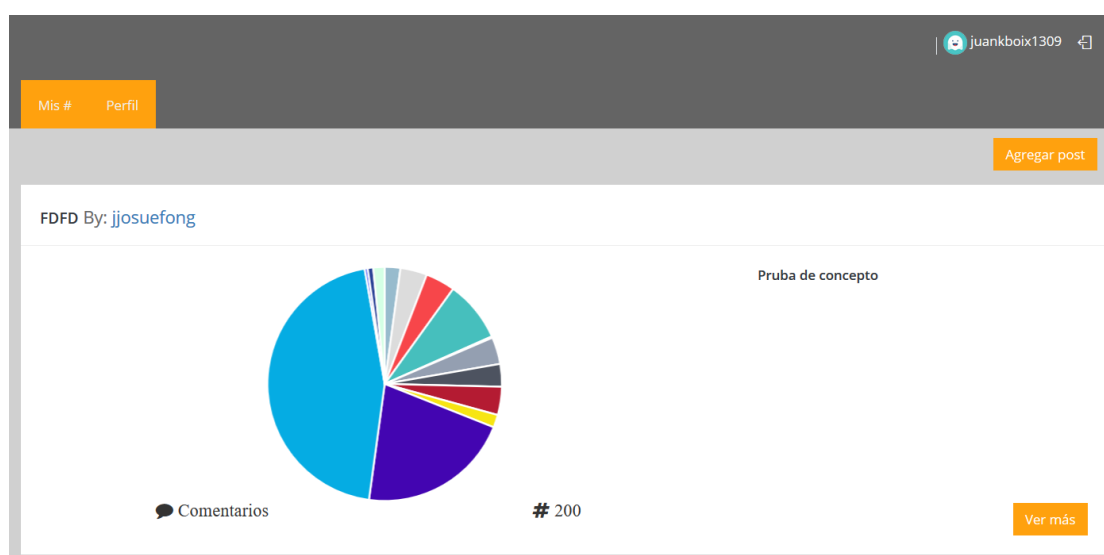
1. **Pantalla de inicio.** La pantalla de inicio no sufrió prácticamente ningún cambio respecto del prototipo diseñado, el tipo de letra y el logo sufrieron cambios entre un prototipo y otro, sin embargo, la esencia del sitio sigue siendo la misma. Ver figura 81.

Figura 76 Página de inicio – Prototipo interactivo



2. Pantalla principal. La pantalla principal mejora mucho en términos de simpleza y del color de las publicaciones, con la adición del blanco para agregar claridad a la plataforma final, se tiene un botón para agregar nuevos hash a la plataforma, además de información del usuario y el cierre de sesión. Ver Figura 82.

Figura 77 Página principal – Prototipo interactivo.



3. Pantalla de registro de usuario. El registro definitivo de usuario requirió solamente del correo electrónico, usuario y contraseña para la creación de la cuenta. Ver Figura 83.

Figura 78 Pantalla de registro de usuario – Prototipo interactivo.

Registro

Email

Usuario (Alias)

Contraseña

Registrar Cancelar

4. Pantalla de perfil de usuario. La pantalla de perfil de usuario integro todo lo que se realizó en el prototipo dibujado con el prototipo diseñado para tener al final una pantalla donde se encuentra toda la información del usuario así como la opción de editar esta información si el perfil es el propio. Ver Figura 84.

Figura 79 Pantalla de perfil de usuario – Prototipo interactivo.

PERFIL

Información personal Change Avatar Contraseña

Nombre de usuario

juankboix1309

Ocupación

Estudiante

Acerca de ti

Guardar cambios Cancelar

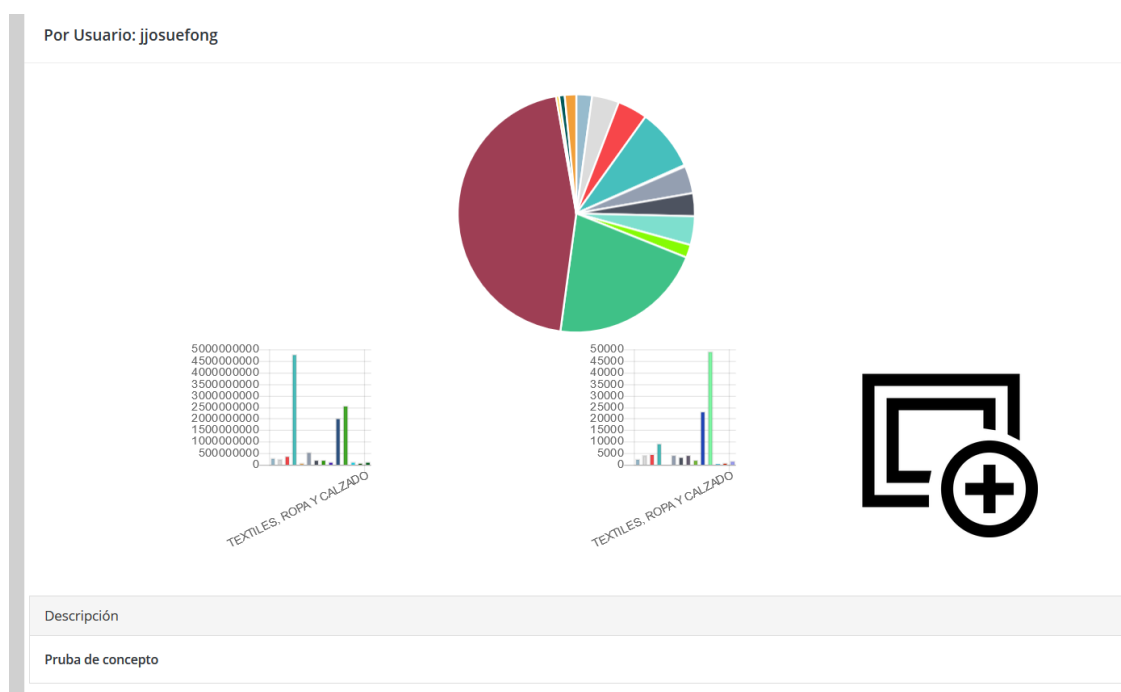
1 POSTS 1 FOLLOWERS 1 FOLLOWING

5. Pantalla de creación de hash. La pantalla de creación de hash tiene el espacio para el ingreso de toda la información necesaria para la creación de un nuevo hash, en ninguno de los prototipos se había contemplado la construcción de las publicaciones, sin embargo, partiendo de la información necesaria para su creación y de la forma como se vería el detalle de las publicaciones la pantalla de creación de hash fue sencilla de diseñar. Ver Figura 39.

Figura 80 Pantalla de creación de hash – Prototipo interactivo.

6. Pantalla de detalle de hash La pantalla de detalle de hash se diseñó en el prototipo interactivo con la plantilla y los colores elegidos en el prototipo diseñado, de forma que los resultados finales se ven muy similares a los del prototipo previo. Ver Figura 86.

Figura 81 Pantalla de detalle de hash – Prototipo interactivo.



d. Validación de prototipos de diseño. La validación de los prototipos de diseño se realizó con un total de seis personas pertenecientes a dos de los tres grupos objetivo, definidos en el estudio de estándar y usabilidad, la cantidad de personas se debe a la clase de retroalimentación cualitativa que se necesitaba de esta validación de prototipos, en la metodología se menciona que si se busca una retroalimentación cualitativa no es necesario un gran número de personas entrevistadas. Se buscó más que una gran cantidad de sujetos de prueba, personas que trabajen o estudien en áreas Relacionadas con el propósito

de la plataforma, tales como periodistas y/o economistas. A continuación se muestran los resultados de la validación de los prototipos de diseño. Ver Tabla 13.

Tabla 13 Validación de prototipos de diseño.

Validación de prototipos							
Nombre	Institucion	Puesto	Telefono	Correo	Fecha	Evaluacion	Comentarios
Ivania Castro	Deloitte Guatemala	Asistente precios de Transferencia	58789410	ivania.castro.rosell@gmail.com	26-Sep	10	Es muy intuitiva, grafica y sería una opción viable para buscar información relevante para hacer análisis económico y econometrico
Stefani Villeda	Universidad Rafael Landivar	Estudiante	41288696	stefanivilleda@gmail.com	26-Sep	7	Los colores son un poco aburridos
Jorge Sapon	Banco G&T Continental	Gestor Administrativo	50104545	jsay011@gmail.com	26-Sep	9	N/A
Regina Vazquez	UNIS	Estudiante	24794708	reginagalich@gmail.com	26-Sep	8	Es una plataforma muy sencilla y facil de entender. Le falta el boton para crear. Sería conveniente un boton de ayuda para los que no conocen muy bien el tema.
Maria Olga Men	UNIS	Estudiante	50162712	vega151104@unis.edu.gt	26-Sep	8	Recuerden crear el boton para crear. Boton de ayuda en las graficas
Krista Ramirez	UNIS	Estudiante	50173820	ramirez151286@unis.edu.gt	26-Sep	8	Faltan algunos elementos como el boton de inicio y el buscador. Botones de ayuda para entender algunos elementos de la pagina

La forma en la que se le solicito la información a los sujetos de prueba a la hora de la validación fue por medio de una ficha de validación de prototipos que se diseñó con el propósito de recolectar la información específica requerida en esta fase del estudio. La evaluación de la plataforma en un rango de 1 a 10 y los comentarios sobre la plataforma fue la retroalimentación cualitativa que se buscó con esta validación de prototipos y que dio paso a la solicitud de cambios al módulo de diseño análisis y desarrollo de frontend. Ver anexos.

3. Fase III: Pruebas de usabilidad

a. **Objetivos de las pruebas de usabilidad.** El inicio en la planeación de cualquier prueba de usabilidad debe ser la definición de los objetivos de estas pruebas, ya que un objetivo bien definido ayuda a definir la retroalimentación correcta a buscar a la hora de la realización de la prueba además del ambiente, cantidad de sujetos de prueba, entre otros.

Los objetivos de las pruebas de usabilidad para la plataforma GobHash fueron los siguientes.

- Cumplir con las métricas definidas en el estudio de estándar y usabilidad.
- Encontrar posibles errores de funcionamiento, desempeño o flujo de información en la primera versión funcional de la plataforma.
- Retroalimentación sobre la aceptación de la plataforma respecto de la usabilidad.

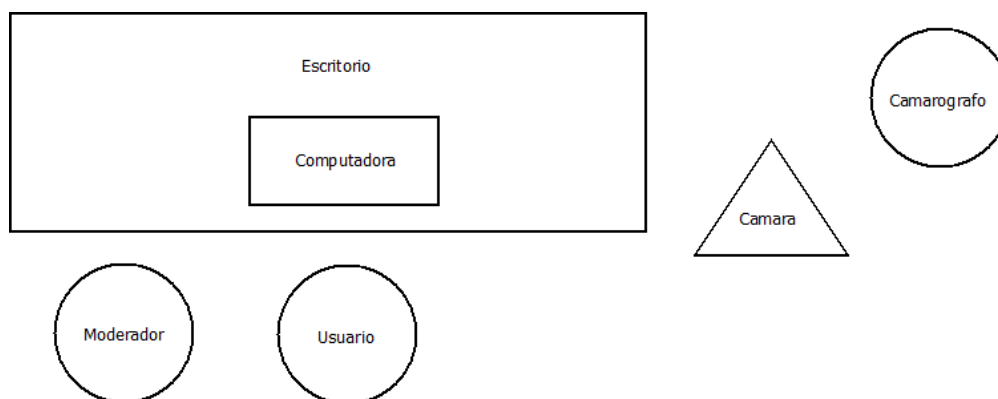
Se busca cumplir estos objetivos con la realización de las pruebas de usabilidad, de manera que se justifique la ejecución de las mismas como una parte importante del proceso de desarrollo de la plataforma.

b. Perfil del usuario. A diferencia de la validación de los prototipos de diseño, que se realizó con un grupo específico de usuarios objetivo definido en el estudio de estándar, las pruebas de usabilidad se realizaron con el público general, la razón de esto es porque la validación de prototipos tiene un acercamiento cualitativo y un enfoque correctivo, mientras que las pruebas de usabilidad buscan ser lo más fieles posibles al escenario final, que es el de una plataforma abierta al público, además buscan retroalimentación cuantitativa que ayude a medir el cumplimiento correcto de las métricas definidas en el estudio de estándar y usabilidad. Razón por la que se realizó la prueba a un total de 15 personas con el objetivo de lograr una muestra suficiente para cubrir el análisis de las métricas.

c. Ambiente de prueba.. Las pruebas de usabilidad se llevaron a cabo en la Universidad del Valle en el Salón J205, previamente reservado con las autoridades del departamento de ciencias de la computación. El salón cumplía con los requerimientos necesarios para llevar a cabo una prueba de usabilidad, ya que cuenta con conexión a internet, sistema de control de temperatura, sillas y mesas cómodas para las pruebas y la privacidad para evitar que las pruebas sean realizadas en un ambiente con exceso de gente o ruido.

Las pruebas se llevaron a cabo en una laptop, localizada en un escritorio en donde la persona se sentaba y estaba expuesta a 2 tomas de cámara, una lateral registrando sus movimientos en búsqueda de lenguaje corporal de algún tipo y la webcam de la computadora registrando sus expresiones faciales, finalmente, una tercera grabación de la pantalla de la computadora nos permitió monitorear sus movimientos en todo momento. Ver Figura 87.

Figura 82 Esquema de pruebas de usabilidad.



d. Diseño de prueba. Las pruebas de usabilidad fueron llevadas a cabo con un script, que contiene 5 tareas que se realizaron en la plataforma, además de las plataformas evaluadas en el estudio de competencia en el estudio de estándar y usabilidad, excluyendo data.gov ya que debido a la incapacidad de crear un usuario si la persona no labora en el gobierno de EEUU, la interacción posible con la plataforma es muy pequeña.

La prueba fue moderada, es decir, hubo una persona disponible para brindar ayuda o solucionar dudas en caso existieran, para cada tarea se midieron las métricas definidas en el estudio de estándar, de forma que

los resultados provean de una visión global al equipo de desarrollo sobre el desempeño y usabilidad de la plataforma respecto de la competencia. A continuación, se detallan las tareas que forman parte del script y las plataformas en la que fueron realizadas, para ver el script de pruebas completo favor revisar anexos.

- Tarea 1: Creación de nuevo usuario

Esta tarea consiste en el ejercicio de crear un usuario nuevo e iniciar sesión. Se le solicitara al usuario que la realice en las siguientes plataformas:

- 1) GobHash
- 2) Data.gov.uk
- 3) Data.imf.org

- Tarea 2: Crear una nueva publicación

Esta tarea consiste en la creación y publicación de un nuevo hash. Se le solicitara al usuario que la realice en las siguientes plataformas:

- 1) GobHash
- 2) Data.imf.org (En este caso con ubicar un set de datos y crear/manipular una gráfica es suficiente)

- Tarea 3: dejar un comentario o reseña en una publicación.

Esta tarea consiste en encontrar una publicación ajena, ingresar al detalle y dejar un comentario o reseña. Se le solicitara al usuario que la realice en las siguientes plataformas:

- 1) Gobhash
- 2) Data.gov.uk

- Tarea 4: seguir a otro usuario en la plataforma.

Esta tarea consiste en encontrar un usuario ajeno, ingresar en su perfil y seguirlo. Esta tarea será exclusiva de la plataforma Gobhash.

- Tarea 5: Cerrar sesión

Esta tarea consiste en cerrar la sesión actual en todas las plataformas en las que se inició.

- 1) Gobhash

2) Data.gov.uk

3) Data.imf.org

e. Resultados de la prueba de usabilidad. Los resultados de la prueba de usabilidad se tabularon y cuantificaron respecto de cada tarea en específico y de todas las tareas en un todo, con el objetivo de mostrar la mayor cantidad de información útil posible, además, los resultados se cuantificaron respecto de las métricas definidas, el resumen completo de la tabulación de las pruebas puede encontrarse en los anexos.

1. Respecto de cada tarea y métrica en específico. Ver Tabla 14-19. Se tabulo primero para cada tarea y cada métrica en específico, aunque los verdaderos resultados que estamos buscando para un correcto veredicto para la usabilidad de la plataforma serán de todas las tareas en un condensado, por lo menos a lo que el tiempo respecta, es necesario revisarlo en cada tarea.

Tabla 14 Métricas en tarea 1.

Métricas	Tarea 1: Creación de un nuevo usuario		
	GobHash	Data.gov.uk	Data.imf.org
Relación de éxito en tareas (%)	100%	100%	92%
Tiempo Promedio de Finalizacion (s)	60.27	99.67	177.92
Medidas Subjetivas (1-10)	9.20	8.33	6.42

Figura 83 Relación de éxito tarea 1.

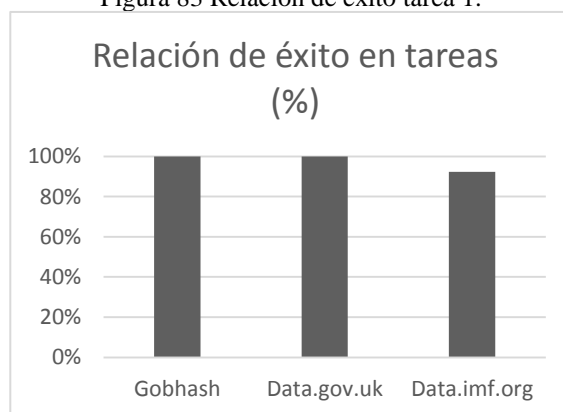


Figura 84 Tiempo promedio de finalización tarea 1.

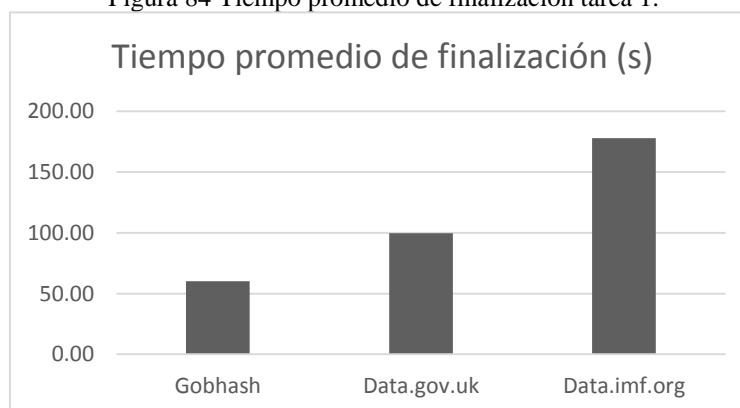


Figura 85 Promedio medidas subjetivas tarea 1.

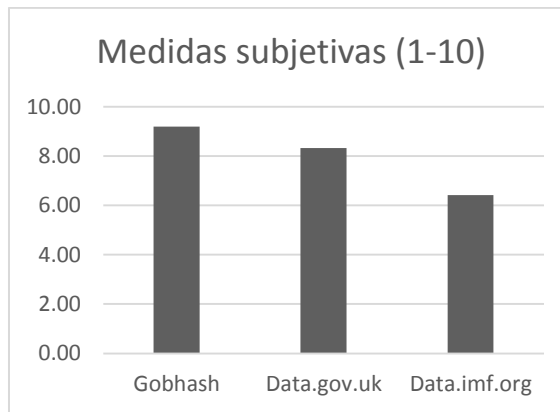


Tabla 15 Métricas en tarea 2.

Métricas	Tarea 2: Crear una nueva publicación	
	GobHash	Data.imf.org
Relación de éxito en tareas (%)	73%	77%
Tiempo Promedio de Finalizacion (s)	110.27	114.30
Medidas Subjetivas (1-10)	9.27	7.70

Figura 86 Relación de éxito tarea 2.

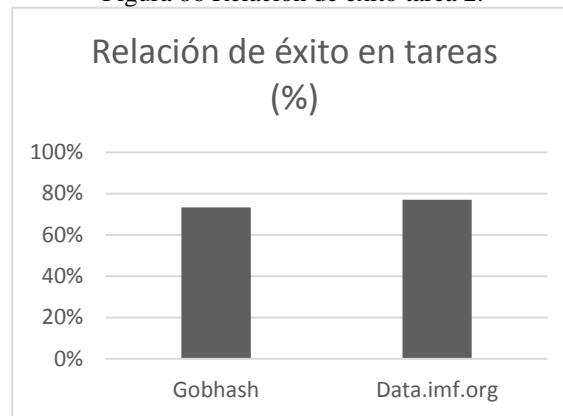


Figura 87 Tiempo promedio de finalización tarea 2.

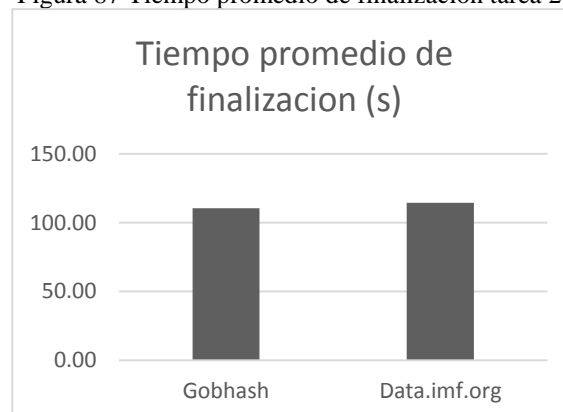


Figura 88 Promedio medidas Subjetivas tarea 2.

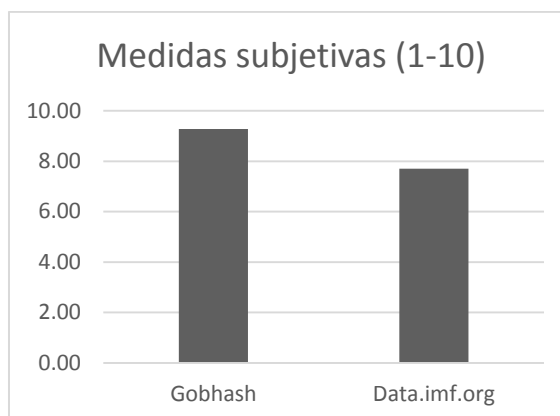


Tabla 16 Métricas en tarea 3.

Métricas	Tarea 3: dejar un comentario o reseña en una publicación	
	GobHash	Data.gov.uk
Relación de éxito en tareas (%)	100%	87%
Tiempo Promedio de Finalizacion (s)	65.27	177.31
Medidas Subjetivas (1-10)	9.47	6.54

Figura 89 Relación de éxito en tarea 3.

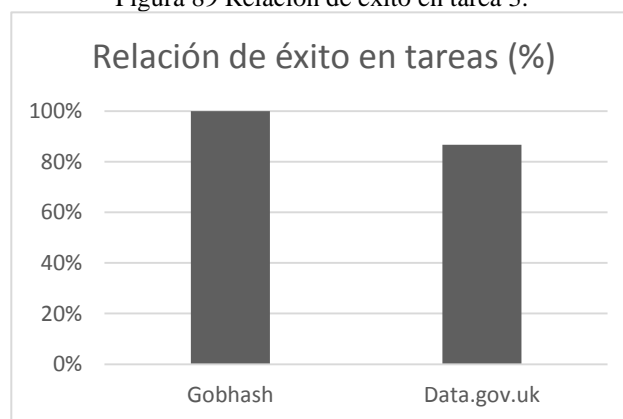


Figura 90 Tiempo promedio de finalización tarea 3.

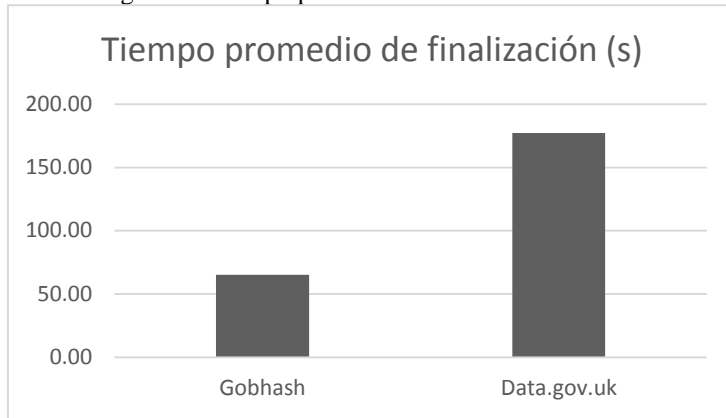


Figura 91 Promedio medidas subjetivas tarea 3

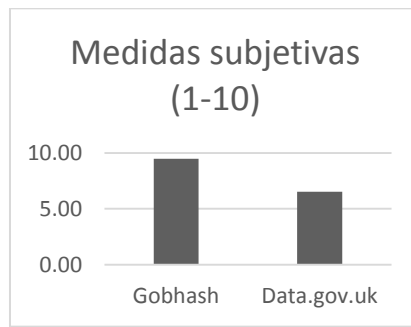


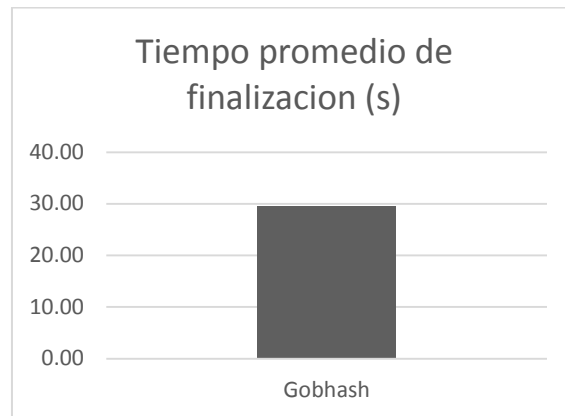
Tabla 17 Métricas en tarea 4.

	Tarea 4: seguir a otro usuario
Métricas	GobHash
Relación de éxito en tareas (%)	100%
Tiempo Promedio de Finalizacion (s)	29.47
Medidas Subjetivas (1-10)	9.87

Figura 92 Relación de éxito en tarea 4



Figura 93 Tiempo promedio de finalización tarea 4



. Figura 94 Promedio de medidas subjetivas tarea 4.

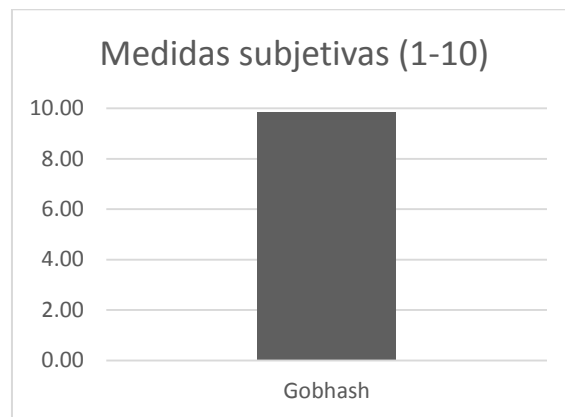


Tabla 18 Métricas en tarea 5

Métricas	Tarea 5: Cerrar sesión		
	Gobhash	Data.gov.uk	Data.imf.org
Relación de éxito en tareas (%)	100%	100%	92%
Tiempo Promedio de Finalizacion (s)	5.47	10.20	17.17
Medidas Subjetivas (1-10)	9.93	9.80	9.75

Figura 95 Relación de éxito en tarea 5.

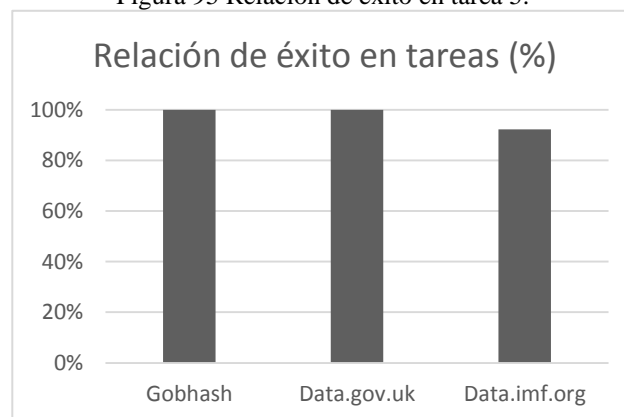


Figura 96 Tiempo promedio de finalización tarea 5.

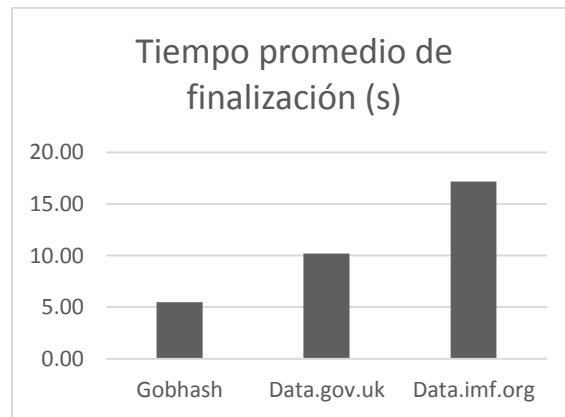
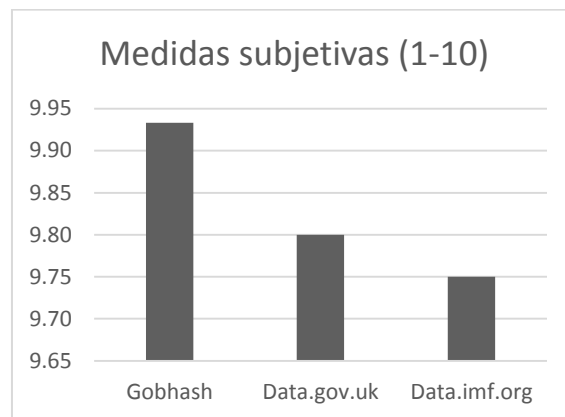


Figura 97 Promedio medidas subjetivas tarea 5



2. Respecto de las métricas en todas las tareas juntas. En este resumen de los datos se dejó fuera el tiempo ya que un dato de tiempo promedio por usuario por tarea por todas las tareas no dice nada respecto del uso de la plataforma ya que las tareas son distintas. Ver Tabla 19

Tabla 19 Métricas en todas las tareas.

Métricas	Todas las tareas		
	GobHash	Data.gov.uk	Data.imf.org
Relación de éxito en tareas (%)	95%	96%	87%
Tiempo promedio de finalizacion (s)	N/A	N/A	N/A
Medidas subjetivas (1-10)	9.55	8.22	7.96

Figura 98 Relación de éxito todas las tareas.

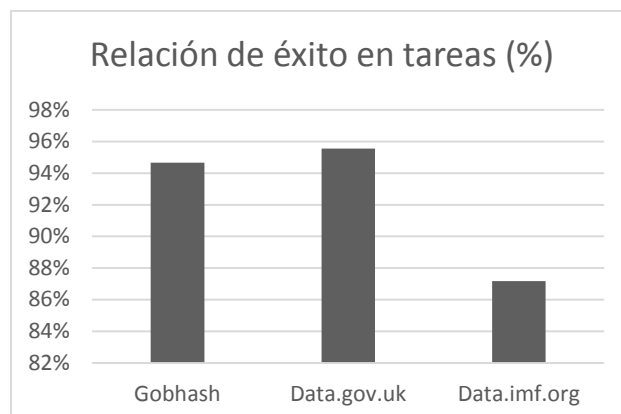
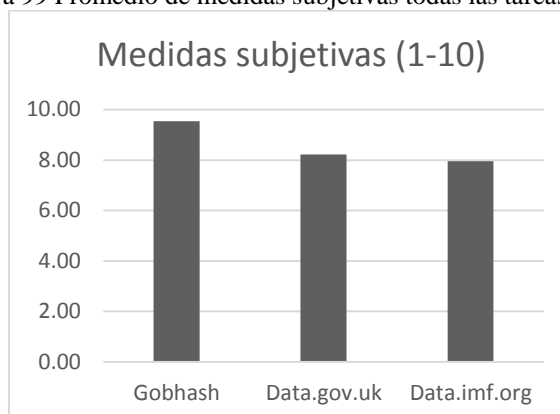


Figura 99 Promedio de medidas subjetivas todas las tareas.



3. Respecto de los errores críticos y no críticos. Los resultados respecto de los errores críticos y no críticos se tabularon por aparte dado que no son datos complementarios a la relación de éxito en tareas, es decir, no tener éxito en una tarea no implica un error, puede que el fracaso en una tarea se deba a que alcanzo el límite de tiempo más que por algún error en la plataforma. Ver cuadros 20-22. Y Figuras 104-106

Tabla 20 Relación de error total en pruebas.

Relación de error total (%)		
GobHash	data.gov.uk	data.imf.org
23%	2%	4%

Figura 100 Relación de error total.

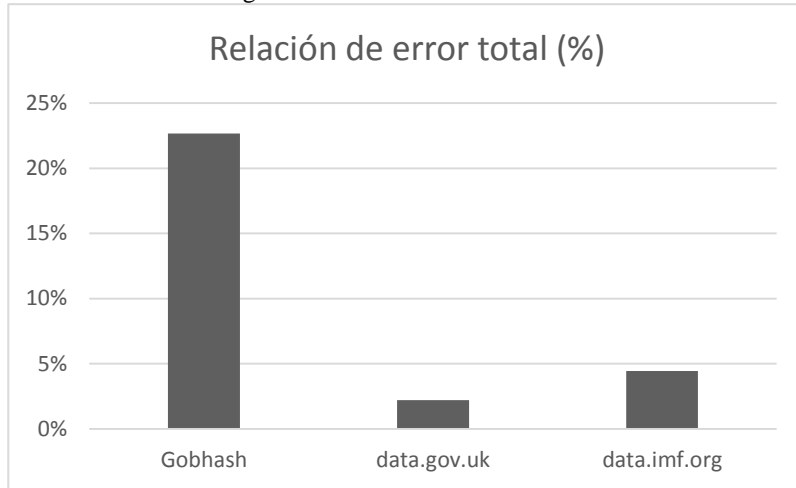


Tabla 21 Relación de error crítico en pruebas.

Relación de error crítico (%)		
GobHash	data.gov.uk	data.imf.org
5%	0%	4%

Figura 101 Relación de error crítico.

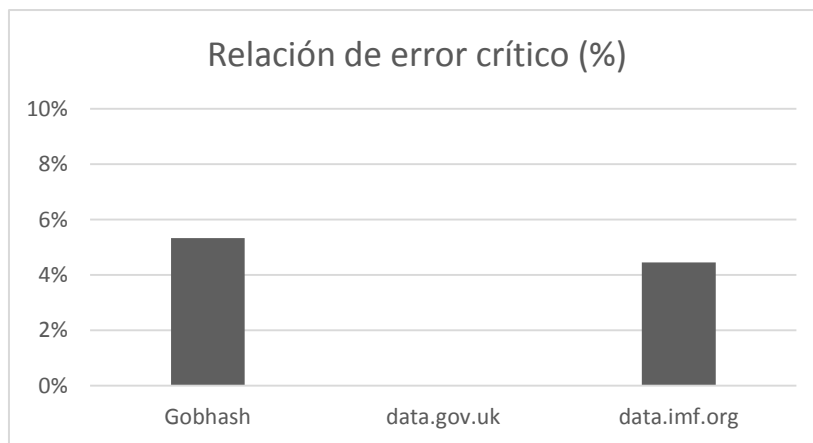


Tabla 22 Relación de error no crítico en pruebas

Relación de error no crítico (%)		
GobHash	data.gov.uk	data.imf.org
17%	2%	0%

Figura 102 Relación de error no crítico.



C. ANÁLISIS DE RESULTADOS

1. Fase I: Estudio de estándar y usabilidad. El estudio de estándar y usabilidad fue clave para el desarrollo del módulo, debido a que aterrizo el proyecto y despejo muchas dudas respecto de cómo debería de funcionar y cuáles deberían ser sus requisitos funcionales. La proposición de valor cumple con las expectativas del equipo de desarrollo respecto del proyecto, además de ser la mejor definición para la plataforma. Respecto de las otras plataformas evaluadas, es interesante lo adelantados que están los países más desarrollados en temas de gobierno abierto y datos abiertos, plataformas como data.gov y data.gov.uk son muestras de como gobiernos centralizan sus datos públicos y los ponen al servicio de la gente, no solamente para consulta sino para el desarrollo de herramientas tecnológicas y no tecnologías que provengan de la sociedad civil, esta clase de iniciativas ayuda a que el gobierno sea inclusivo con el pueblo y este se sienta parte de algo mayor en su facultad de ciudadano.

A pesar de que GobHash es una iniciativa que no viene del Estado, es importante poner a mano de la ciudadanía esta clase de herramientas y plataformas que ayudan a que el mecanismo de consulta y visualización de información pública llegue al siguiente nivel, y esa es exactamente la razón por la que se definió a la sociedad civil como el gran grupo objetivo y objetivo de usabilidad del proyecto, al igual que las plataformas analizadas en el estudio de estándar y usabilidad, en donde a pesar de tener información compleja o en formatos que no todos comprenden, las plataformas en si están diseñadas para el uso de la población en general. Razón por la que nosotros quisimos con GobHash, aunque la plataforma estuviera orientada solamente a los datos de las adquisiciones del estado, tener una plataforma a la que todos pudieran entrar, tener un usuario y crear comunidad con otras personas que tengan el mismo interés de saber más sobre como el Estado de Guatemala invierte el dinero de los contribuyentes.

Finalmente, los resultados del estudio de estándar y usabilidad que ayudaron más al resto del módulo fueron el diagrama de flujo de aplicación y la definición de métricas. La definición de las métricas ayudo a orientar la creación de los prototipos, la validación de los mismos y el diseño y desarrollo de las pruebas de usabilidad al cumplimiento de las métricas, se utilizaron estrictamente las métricas necesarias para las circunstancias de desarrollo de la aplicación. La relación de éxito se utilizó para medir el correcto funcionamiento de la plataforma, ya que además de ser la prueba de usabilidad, fue la primera vez que los usuarios finales se encontraron con la primera versión funcional de la plataforma. La relación de error tenía el mismo propósito de medir el desempeño, pero ahora desde el punto de vista de encontrar errores que pudieron haber pasado por alto en la etapa de desarrollo, ya que los errores no críticos, a pesar de no interrumpir el proceso de cumplimiento de funcionalidades, le resta calidad a la plataforma.

Para las siguientes métricas, se enfocó el proceso de definición y cumplimiento de las mismas desde el punto de vista de la usabilidad de la plataforma, razón por la que se tomaron en cuenta los factores de completitud y corrección de la plataforma, los tiempos de finalización y las medidas subjetivas son métricas específicamente definidas para medir la usabilidad. Ya que los tiempos de finalización nos permiten comparar la usabilidad de la plataforma en el primer acercamiento en contra de las demás plataformas analizadas en el estudio de estándar y usabilidad. Por otro lado, las medidas subjetivas, factor que media que tan fáciles de realizar son las tareas en las diferentes plataformas, más que medir funcionalidad, valida el diseño de la plataforma, el mapeo y el flujo de aplicación de la misma, desde el punto de vista de si está hecha para ser fácil de utilizar, más que rápida de utilizar.

Finalmente, la curva de aprendizaje viene más o menos implícita con la definición de los usuarios objetivo ya que si a pesar de que tenemos usuarios en grupos específicos que podrían utilizar bastante la plataforma el objetivo es que la plataforma sea apta para la población en general, la curva de aprendizaje tiene que ser lo más pequeña posible para que cualquier persona que la utilice se familiarice rápidamente con ella y pueda volverse un usuario asiduo más fácilmente. Además, los usuarios objetivos también influyeron en el diagrama de flujo de aplicación ya que se optó por apegar mucho a la memoria de los usuarios para simplificarles el uso de la plataforma. Se tomaron ideas de flujo de aplicación de las redes sociales más grandes en el mundo tales como Facebook, Instagram y Twitter, de forma que los usuarios pudieran familiarizarse rápidamente con la plataforma para su buen uso.

2. Fase II: Desarrollo de prototipos de diseño El proceso de desarrollo de los prototipos de diseño fue clave para el producto final en materia de diseño y flujo de aplicación ya que tomo todos los resultados del estudio de estándar y usabilidad trato de plasmarlos de la mejor manera en un prototipo que luego sería validado por los usuarios finales para sus cambios en cada una de sus fases.

El prototipo dibujado marco la referencia para toda la plataforma y más que cualquier retroalimentación respecto a diseño nos ayudó a encontrar que faltaba, que estaba mal y como esto se vería reflejado en cambios para el prototipo diseñado. Específicamente, el prototipo dibujado valido el flujo de aplicación que fue sencillo de comprender por parte de las personas entrevistadas, y se encontraron ausencias

como el botón de crear publicación. Luego, el prototipo dibujado fue bastante útil a la hora de la validación ya que era simple de modificar y se podía modificar en el momento.

Luego se pasó al desarrollo del prototipo diseñado, con el prototipo diseñado se tomó una decisión muy importante respecto de la paleta de colores ya que al escogerla por el efecto que la mezcla de los colores tendría en el usuario y luego de haberla plasmado en el logotipo de la plataforma. Había que combinar los colores de la forma y proporción correcta para que la plataforma se viera bien además de tener los colores correctos, razón por la que el diseño de los prototipos diseñados tal vez haya sido el más largo de los tres, sin embargo, durante todo el proceso se tuvo presente el hecho de que realizar de forma correcta los prototipos diseñados ayudaban mucho a la consecución de un prototipo interactivo y una maquetación final de la plataforma de muy alta calidad. Luego de terminar el diseño se validó el prototipo diseñado y esto confirmó que a los usuarios el color no les causaba ningún problema, contrario a lo que se había pensado en un principio.

Finalmente, el desarrollo de los prototipos interactivos se llevó a cabo en conjunto con el módulo de Diseño, Análisis y Desarrollo de Frontend, ya que en mutuo acuerdo decidimos que era mucho más ágil diseñar un prototipo interactivo con una herramienta de fabricación de prototipos en HTML como lo es Silex. Esto comprobó ser una buena decisión ya que a la hora del diseño y validación de este prototipo no había ninguna preocupación respecto de cómo el módulo de Diseño, Análisis y Desarrollo de Frontend iba a poder replicar lo creado en los prototipos interactivos, razón por la cual se logró un muy alto nivel de consistencia entre la última iteración de prototipos y la primera versión funcional de la plataforma, tanto en temas de diseño, mapeo y flujo de aplicación, logrando que el paso de usar dummy data en un prototipo a usar datos reales en una primera versión funcional del producto fuera prácticamente transparente, ayudando así a tener un alto nivel de consistencia y calidad. La validación de los prototipos interactivos necesito de una posterior solicitud de cambios al módulo de Diseño, Análisis y Desarrollo de Frontend debido a que el prototipo era trabajado en conjunto. Ver anexos.

3. **Fase III: Pruebas de usabilidad.** Las pruebas de usabilidad fueron la última fase del módulo, pero a pesar de ser la última fase, es la más importante ya que es la que representa verdadero valor al proyecto por parte del módulo. Las pruebas de usabilidad amarran todo el proceso llevado a cabo en el estudio de estándar y usabilidad con el diseño refinado en el desarrollo y validación de prototipos, de esta manera, al haber llegado a una primera versión funcional de la plataforma. Las pruebas de usabilidad nos dieron la retroalimentación necesaria para validar que el proceso de Interacción Humano Computador cumpliera con las métricas definidas y que las pruebas de usabilidad en si cumplieran con sus propios objetivos.

Respecto del cumplimiento de las métricas definidas en el estudio de estándar y usabilidad, si analizamos la primera métrica definida, la relación de éxito, podemos observar que la meta definida para esta métrica en el estudio de estándar y usabilidad era tener un 90% de relación de éxito como mínimo. Luego de las pruebas de usabilidad, la relación de éxito final fue de un 95% para todas las tareas, teniendo como máxima relación de éxito las tareas de inicio de sesión, cierre de sesión, seguir a un usuario y comentar una publicación, con un 100% de relación de éxito, sin embargo, como puntaje menor tenemos a la tarea de creación de hash,

con un 73% de relación de éxito, en la parte de la plataforma dedicada a esta tarea se encontraron muchos errores críticos y no críticos. El total de errores críticos encontrados en GobHash durante las pruebas de usabilidad (4), se encontraron a la hora de hacer esta tarea. Probablemente debido a que era la parte de la plataforma en la que todos los módulos del proyecto debían integrar.

De esta forma cumplimos también uno de los objetivos de las pruebas de usabilidad, el encontrar errores, ya que ahora por medio del informe de errores y la solicitud de cambios la plataforma se puede mejorar y corregir antes de salir al mercado. Sin embargo, durante el proceso de análisis de los resultados de las pruebas de usabilidad se encontró una relación de error de 23% para toda la plataforma GobHash, este dato puede contrastar a simple vista con el dato de la relación de éxito total de la plataforma (95%) sin embargo, la mayoría de los errores encontrados fueron no críticos, razón por la que pudimos cumplir con el objetivo de la métrica de relación de éxito. A la vez, pudimos cumplir con la métrica de relación de error ya que solamente se tuvo un 4% de relación de error crítico en Gobhash.

Finalmente, para las métricas de usabilidad, Tiempo de finalización y medidas subjetivas el único objetivo era ser dignos competidores de las plataformas evaluadas en el estudio de estándar y usabilidad, de manera que la meta era estar más o menos mejor en términos de usabilidad por lo menos. Comparando en las métricas evaluadas con anterioridad, en términos de relación de éxito data.gov.uk mejora el desempeño de GobHash por una pequeña diferencia de 1%, mientras que data.imf.org está mucho más abajo, sin embargo, donde se nota la diferencia es en la relación de error en donde GobHash tiene la relación de error más alta con un 23%, sin embargo, esto es comprensible ya que GobHash es una plataforma nueva.

En lo que a los tiempos promedio de finalización respecta, GobHash maneja tiempos de finalización menores que el resto de plataformas en todas las tareas, sin embargo, a pesar de que GobHash es mucho más pequeña y simple que estas otras plataformas, la métrica se complementa con las medidas subjetivas ya que con un promedio de medición subjetiva mayor al 9.5 somos la plataforma mejor calificada de las pruebas, lo que comprueba que a pesar de ser una plataforma más pequeña y simple que el resto, el funcionamiento cumple con las expectativas de usabilidad de plataformas mucho más grandes, serias y con más tiempo de desarrollo.

De esta manera podemos concluir que todas las métricas planteadas fueron satisfechas con éxito por parte de la plataforma GobHash en las pruebas de usabilidad, que los errores críticos y no críticos fueron encontrados y luego documentados mediante el informe de errores y la solicitud de cambios a Frontend. Ver Anexo. Finalmente, la retroalimentación sobre la aceptación del producto final fue positiva y respaldada por la métrica de medidas subjetivas, al ser GobHash la plataforma mejor medida en términos de facilidad de uso alrededor de todas las pruebas de usabilidad

IX. DISEÑO E IMPLEMENTACIÓN DEL BACKEND

A. METODOLOGÍA

Requisitos funcionales

1. **Usuario:**
 - a) Registro
 - b) Inicio de sesión
 - c) Solicitar cambio de contraseña
 - d) Recuperar contraseña
 - e) Perfil de usuario
 - f) Fotografía del usuario
 - g) Seguir a otro usuario
2. **Feed:**
 - a) Feed en vivo: Mostrar en tiempo real las publicaciones de las personas a las que sigue el usuario.
 - b) Feed: Mostrar las publicaciones de los usuarios que sigue y los del mismo usuario.
3. **Post:**
 - a) Crear publicación:
 - (1) Agregar título
 - (2) Agregar descripción
 - (3) Agregar gráfico (3 máximo)
 - (4) Estado de “me gusta” a la publicación
 - (5) Comentar publicación

- b) Eliminar publicación
- c) Requisitos no funcionales
 - i. Seguridad: El API debe ser lo suficientemente seguro para que una persona no autorizada no tenga acceso.
 - ii. Rendimiento: El API responderá en máximo 1 segundo.
 - iii. Disponibilidad: El API estará disponible los 7 días de la semana y 24 horas al día.
 - iv. Interfaz: La documentación debe ser intuitiva para al menos el 80% de desarrolladores que la consulten.

Diseño

Para diseñar el API se siguieron estándares de la industria que se basan en el principio de una interfaz estándar para que pueda ser entendido por cualquier desarrollador.

Un endpoint sirve para realizar ciertas acciones sobre un recurso a través de un URL. Las URL deben seguir cierto estándar para describir las acciones sobre los recursos. Nótese que las acciones están definidas por verbos.

Las acciones están definidas como las siguientes posibles:

1. GET: para obtener recursos.
 - a. Solicita transferencia de recursos al destino
2. POST: para crear nuevos recursos
 - a. Solicita que el destino procese la solicitud y su cuerpo de acuerdo a la semántica definida por el *endpoint*.
3. PUT: para actualizar un recurso
 - a. Solicita que el estado del recurso de destino sea creado o reemplazado por el estado definido en el cuerpo definido en la semántica del *endpoint*.

4. PATCH: para actualizar un recurso

- a. La diferencia entre patch y put es que el payload del put es considerado una versión modificada del recurso guardado en el servidor y el cliente está pidiendo que esa versión sea reemplazada. A cambio, el patch el cuerpo se considera un conjunto de instrucciones describiendo como un recurso actual del servidor debe ser modificado para crear una nueva versión. Patch puede ser utilizado para cambiar solo una parte del recurso a modificar.

5. DELETE: para eliminar un recurso.

Una vez definido la semántica de las posibles acciones es necesario definir la estructura de las URL. Las URL deben ser basadas en sustantivos que describan los recursos.

Por ejemplo:

El método GET en la url /users debería obtener un listado de los usuarios.

El método GET en la url /users/12 debería obtener el usuario con el id 12.

El método DELETE en la url users/12 debería de eliminar el usuario con el id 12.

Los códigos de respuesta son también parte del diseño de API pues indican el resultado de una llamada a un endpoint.

- Categoría 20x. Estos códigos representan que la acción fue procesada correctamente. El código 200 (ok) es la respuesta estándar para los métodos de http. El código 201 (Created) es el código de respuesta que indica que se ha creado un nuevo recurso, comúnmente utilizado en un método POST. El código 204 (No content) representa que la solicitud http ha sido procesada correctamente pero no ha regresado nada de contenido. Esto se puede dar en los métodos de tipo DELETE.

- Categoría 30x. Esta categoría representa el código de respuesta cuando el cliente ya tiene un recurso en la caché, por lo que no es necesario volver a mandarlo.

- Categoría 40x. Esta categoría representa los códigos de respuesta cuando una solicitud http ha fallado. El código 400 (*Bad request*) indicar que la solicitud no ha sido procesado correctamente porque el servidor no puede interpretar la información enviada. El código 401 (*Unathorized*) indicar que el cliente no tiene permitido acceder a un recurso. El código 403 (*Forbidden*) indica que la solicitud está autenticada pero no tiene permisos para acceder a un recurso. El código 404 (*Not Found*) indica que no se ha podido encontrar el recurso solicitado.
- Categoría 50x. Esta categoría representa los códigos de respuesta cuando un solicitud es válido pero al procesarlo a ocasionado un error fatal.

A los endpoints del API también se le agregan ciertos parámetros estándares para consultar información. Estos parámetros son el ordenamiento, cantidad máxima de registros y a partir de que registro buscar. El parámetro de ordenamiento no fue implementado sino que se dejó por defecto regresar los registros más recientes y se parametrizó la cantidad máxima de registros a retornar y a partir de que registro empezar a filtrar.

A nivel de programación se utiliza una linter, para verificar el estilo de la sintaxis de cada línea de código. Los archivos se nombraron de forma *underscore* y se incluyó el módulo al que pertenece el archivo. Por ejemplo, nombre_archivo.controller.js para poder ordenar los archivos de forma estándar y eficiente a la hora de buscar un archivo. Como manejador de paquetes o dependencias se utiliza Yarn.

Parte del diseño del API también incluye la división de las funcionalidades que se desarrollaron y se manejó el estándar de programación en inglés . Se dividió el desarrollo los siguientes módulos.

1. Auth: Módulo donde se desarrollaron las funcionalidades de autenticación y autorización de sesiones.
2. User: Módulo donde se desarrollaron las funcionalidades del usuario.
3. Post: Módulo donde se desarrollaron las funcionalidades de las publicaciones y el protocolo de tiempo real.
4. Stats: Módulo donde se desarrollaron las funcionalidades de las estadísticas, por ejemplo, las publicaciones con más preferencia de los usuarios.

5. Search: Módulo de búsqueda de publicaciones y usuarios.

- Versionamiento. El API usará *semantic versioning*, un sistema de versionamiento creado por el fundador de Github, que es se está volviendo un estándar a la hora de desarrollar un API.

Sea dado un número de versión: *MAJOR.MINOR.PATCH*

1. *MAYOR*: Cuando se realiza un cambio que hace incompatible el API
2. *MINOR*: Cuando se agrega una nueva funcionalidad que es compatible con el resto del API.
3. *PATCH*: Cuando se arregla un error, o mejor conocido como un *bug*.

Por esto se incluye la versión en el URL del API.

- Pruebas unitarias. Se realizaron pruebas unitarias a cada endpoint para asegurar el correcto funcionamiento a nivel individual para cada función creada. Esto mantendrá confianza a nivel del API para los clientes que lo consuman.
- Pruebas de estrés. Se realizaron pruebas de estrés para conocer si la infraestructura podía mantener cierta cantidad de solicitudes/usuarios definidos y para identificar los cuellos de botella de una aplicación. En general sirven para tener manejar el rendimiento de una aplicación.

En el API hay dos endpoints que son críticos dentro del API. El *feed* que es el más utilizando dentro de la red social y el endpoint de los gráficos que obtiene la información de GuateCompras. Según el módulo de diseño y desarrollo de estudio de usabilidad para la interacción humano-computador la cantidad de usuarios en a soportar deberían de ser al menos mil y la cantidad de usuarios concurrentes a soportar es de cien. Se define que una prueba de estrés exitosa no puede usar más del 80% de los recursos de la infraestructura.

- Servicio de terceros. El API dependerá de otros servicios que en conjunto realizarán tareas importantes que garantizarán la funcionalidad. A continuación se describen los servicios externos que se usarán.

- i. Servicio de correos. El servicio de correos será utilizado para enviar notificaciones, recuperaciones de contraseña, confirmación de registro, etc. Sendgrid tiene un plan de 40 mil correos por un período de 30 días gratuito y después de ese período tiene un costo \$9.95 al mes. Mailgun tiene un plan de 10 mil correos al mes por un período indefinido y después de los 10 mil tiene un costo de \$0.00050 por correo. Por fines prácticos para este proyecto se utilizará Mailgun. Se evaluó también MailChimp pero este tiene un enfoque más de correos de marketing (a pesar que ya tiene un servicio transaccional MailDrill) y Sendgrid y Mailgun tiene un enfoque más transaccional que será lo servirá para este proyecto.

- ii. Backups. Los backups se habilitaron solamente para la base de datos de la red social, que se aloja en la plataforma de mLab. La base de datos de GuateCompras se dejó a cargo del módulo de modelo de datos.

- iii. Archivos estáticos. Se utiliza Amazon S3 para subir los archivos estáticos que son generados por el lado del cliente de GobHash. El cual tiene toda un cuadro de costos por almacenamiento y por cantidad de solicitudes dependiendo del método de http que se realice.

- iv. DNS y seguridad. Para manejar los DNS se usó CloudFlare como servicio para manejar las direcciones del dominio y poder poner los nombres de los dominios a cada aplicación bajo el dominio de gobhash.com. El dominio fue comprado en GoDaddy debido a la funcionalidad permitida por la consola de administración de dominios.

- v. Infraestructura. Se utiliza una plataforma como servicio para proveer la infraestructura GobHash. La bases de datos no Relacional MongoDB se encuentra alojada en mLab y la base de datos Relacional se encuentra en servidor EC2 de Amazon Web Services. Adicionalmente, se utilizan las aplicaciones de Librato, LogEntries, New Relic, Redis

Monitor, Blaze Meter, Travis, BitHound, CodeCov y Artillery como herramientas de monitoreo y pruebas.

B. RESULTADOS

1. Endpoints. Debido a que el backend es puramente código, el mayor resultado visible es la exposición de los servicios a través del API y su documentación. A continuación se listan los endpoints disponibles.

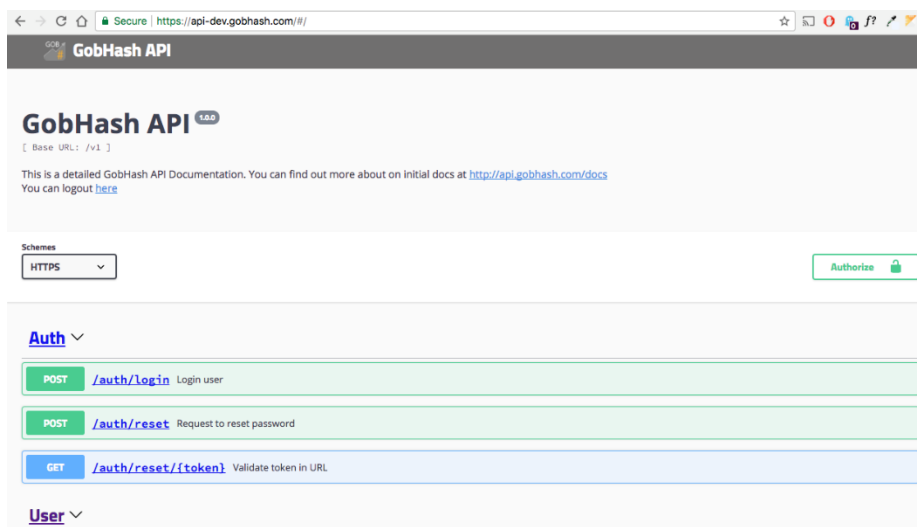
Tabla 23. Listado de endpoints disponibles en el API.

No.	Módulo	Método	Endpoint /v1/	Descripción
1	Auth	POST	/auth/login	Iniciar sesión
2	Auth	POST	/auth/reset	Pedir recuperar la contraseña
3	Auth	GET	/auth/reset/{token}	Verificar un token de recuperación de contraseña
4	User	GET	/users	Listar todos los usuarios
5	User	POST	/users	Crear un nuevo usuario
6	User	GET	/users/{id}	Obtener un usuario específico
7	User	PUT	/users{id}	Modificar un usuario
8	User	DELETE	/users/{id}	Eliminar un usuario
9	User	PATCH	/users/password/change	Cambiar la contraseña al recuperarla.
10	User	PATCH	/users/password/update	Cambiar la contraseña validando la contraseña actual.
11	User	PATCH	/users/biography	Modificar la biografía
12	User	POST	/users/pciture	Subir foto de perfil
13	User	POST	/users/follow	Seguir a otro usuario
14	User	POST	/users/unfollow	Dejar de seguir a un usuario
15	User	PATCH	/users/profile	Actualizar el perfil de un usuario
16	User	GET	/users/profile	Obtener el perfil de un usuario
17	User	GET	/users/follow/{userId}/check	Revisar si un usuario puede seguir a otro usuario.
18	Post	GET	/posts/	Obtener listado de posts
19	Post	POST	/posts/	Crear un nuevo post

cont. Tabla 23				
No.	Módulo	Método	Endpoint /v1/	Descripción
20	Post	DELETE	/posts/	Eliminar un post
21	Post	GET	/posts/{postId}	Obtener un post específico
22	Post	GET	/posts/feed/{userId}	Obtener el feed de un usuario
23	Post	POST	/posts/comment	Crear un comentario a un post
24	Post	POST	/posts/tag	Agregar un tag a un post
25	Post	DELETE	/posts/tag	Eliminar un tag a un post
26	Post	POST	/posts/like	Agregar un like a un post
27	Post	DELETE	/posts/like	Eliminar un like a un post
28	Post	GET	/posts/like/validation/{postId} /{userId}	Verificar si un usuario puede dar like a un post
29	Search	GET	/search/users	Buscar a un usuario por nombre o usuario
30	Search	GET	/search/posts	Buscar a un post por título o por autor
31	Stats	GET	/stats/user/{userId}	Obtener la cantidad de posts, followers y following de un usuario
32	Stats	GET	/stats/posts/top	Obtener los posts con mayor cantidad de likes.

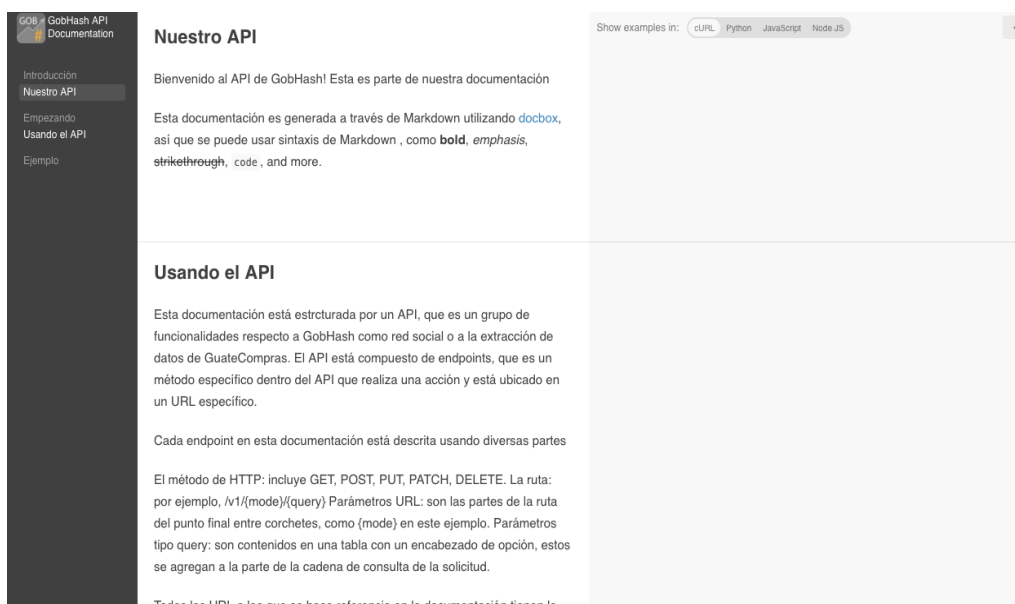
2. Documentación. El API se encuentra documentado utilizando una versión modificada de la librería de Swagger UI. Se publicó en NPM la librería de Swagger UI para que sea descargada con el repositorio principal.

Figura 103. Documentación principal del API.



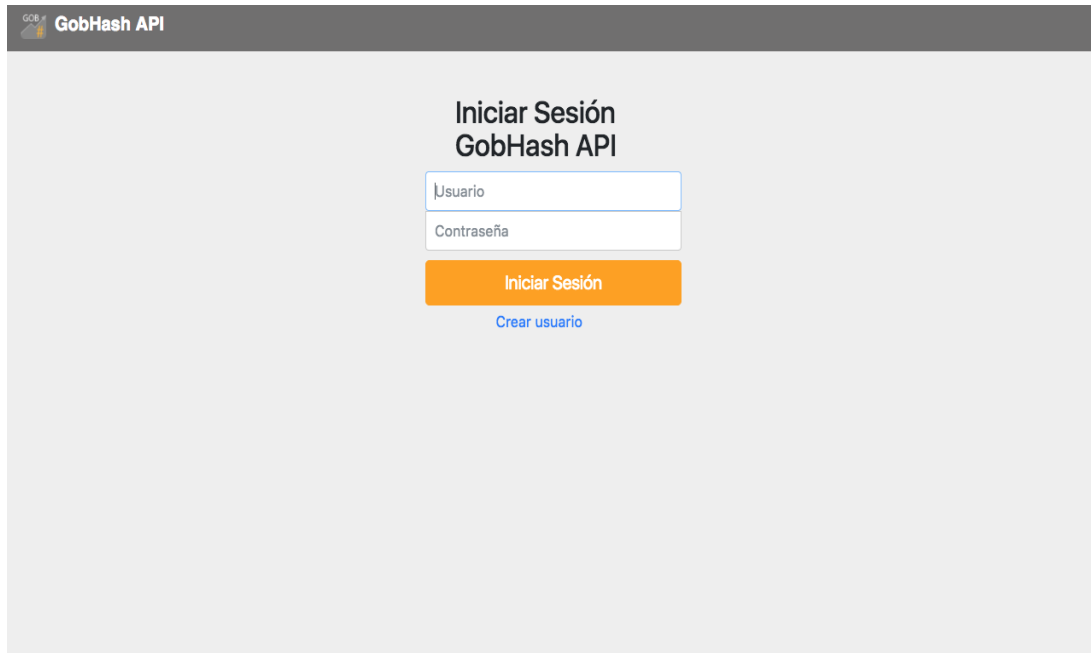
El API cuenta con una segunda documentación para explicar cómo funciona, como empezar a usarlo y explica el protocolo en tiempo real.

Figura 104. Documentación secundaria del API.



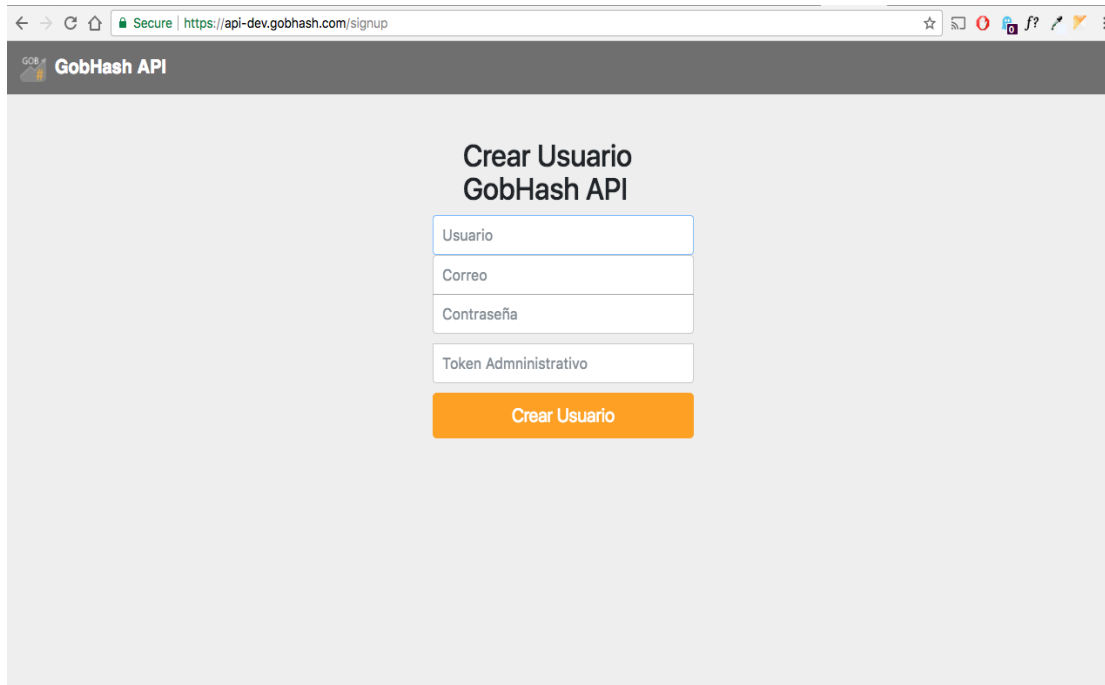
El API cuenta con autenticación para poder acceder a la información ya que no está destinado a ser accedido por el público.

Figura 105. Inicio de sesión a la documentación del API.



The screenshot shows the login page for GobHash API. At the top left, there is a logo with 'GobHash API' text. The main heading is 'Iniciar Sesión GobHash API'. Below the heading are two input fields: 'Usuario' and 'Contraseña'. Underneath these fields is an orange button labeled 'Iniciar Sesión' and a blue link labeled 'Crear usuario'.

Figura 106. Creación de usuario para acceder a la documentación del API.

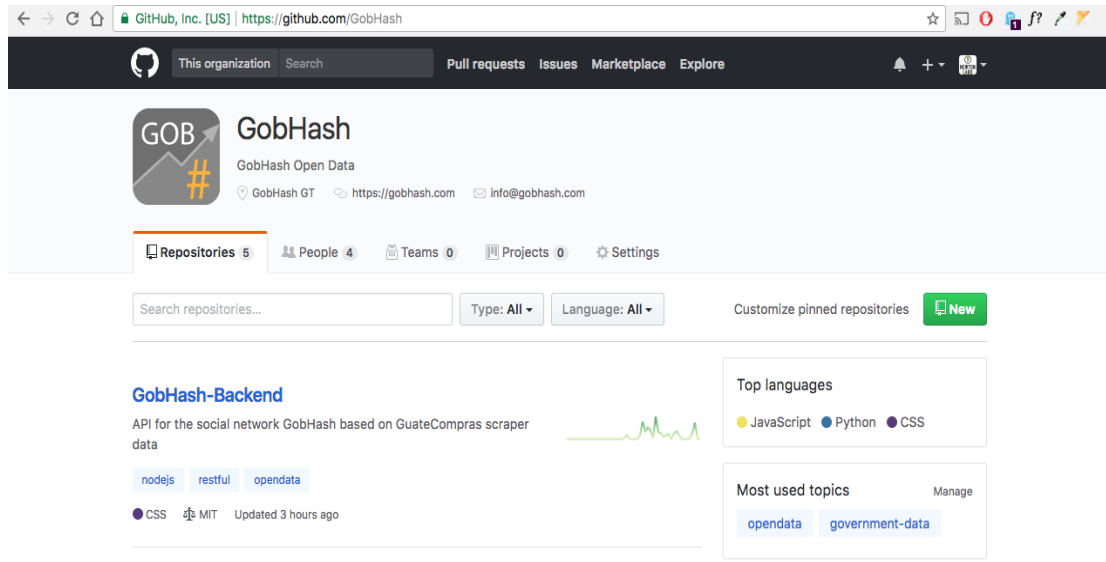


The screenshot shows the user creation page for GobHash API. At the top left, there is a logo with 'GobHash API' text. The main heading is 'Crear Usuario GobHash API'. Below the heading are four input fields: 'Usuario', 'Correo', 'Contraseña', and 'Token Administrativo'. Underneath these fields is an orange button labeled 'Crear Usuario'. The browser's address bar shows the URL 'https://api-dev.gobhash.com/signup'.

Para crear un usuario con acceso a la documentación es necesario un token que autoriza la creación del usuario.

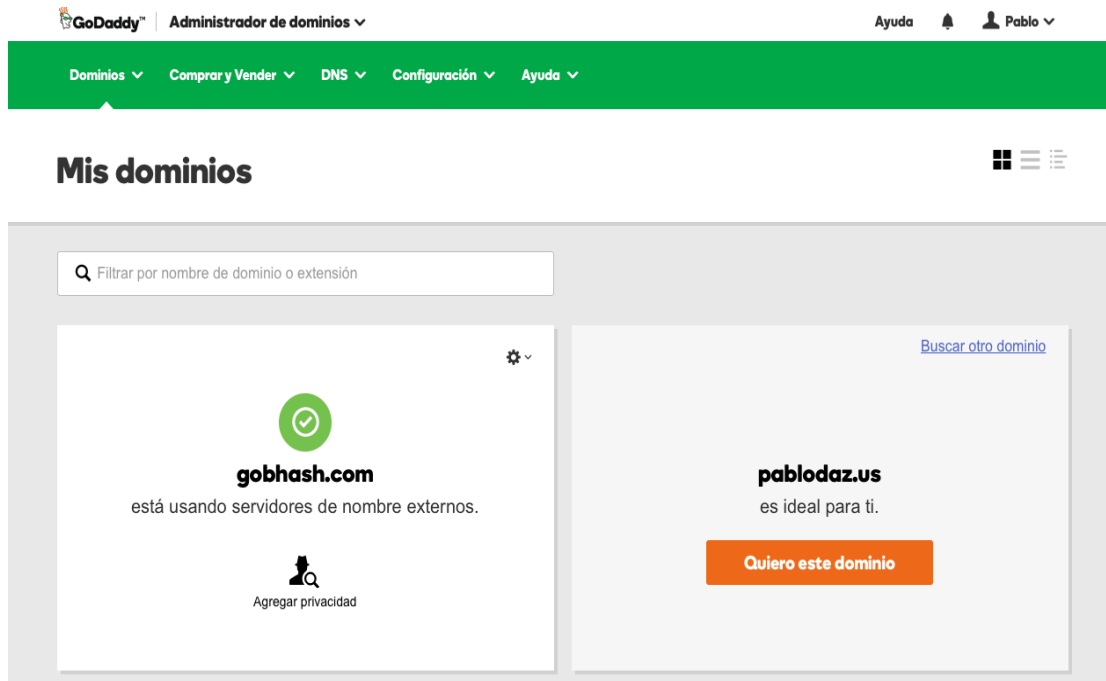
3. Repositorio

Figura 107. Organización GobHash con los repositorios del proyecto.



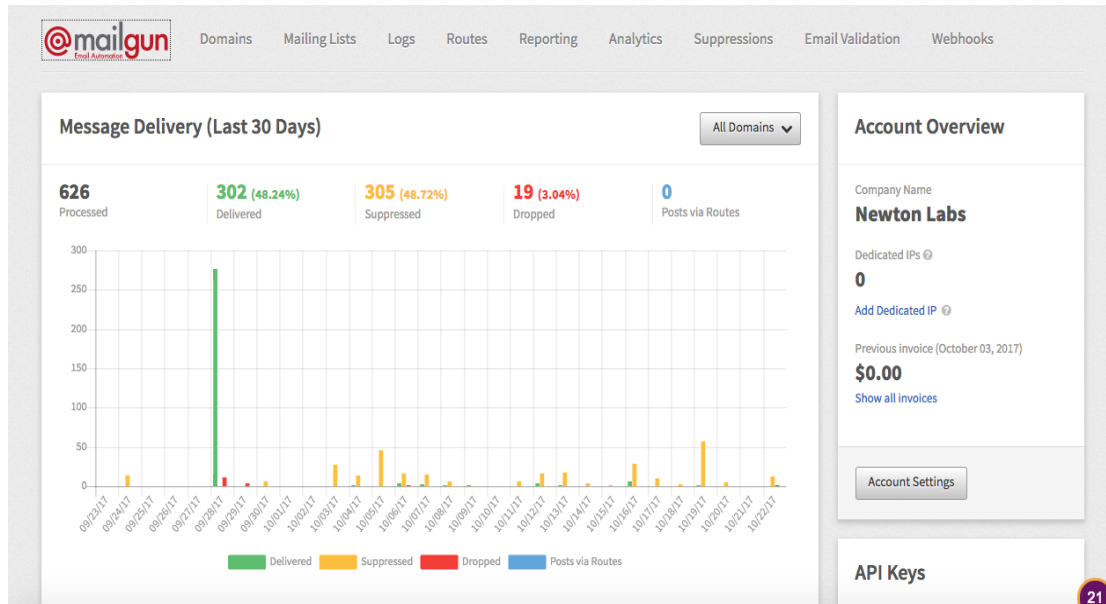
4. Dominio

Figura 108. Compra del dominio en godaddy.com



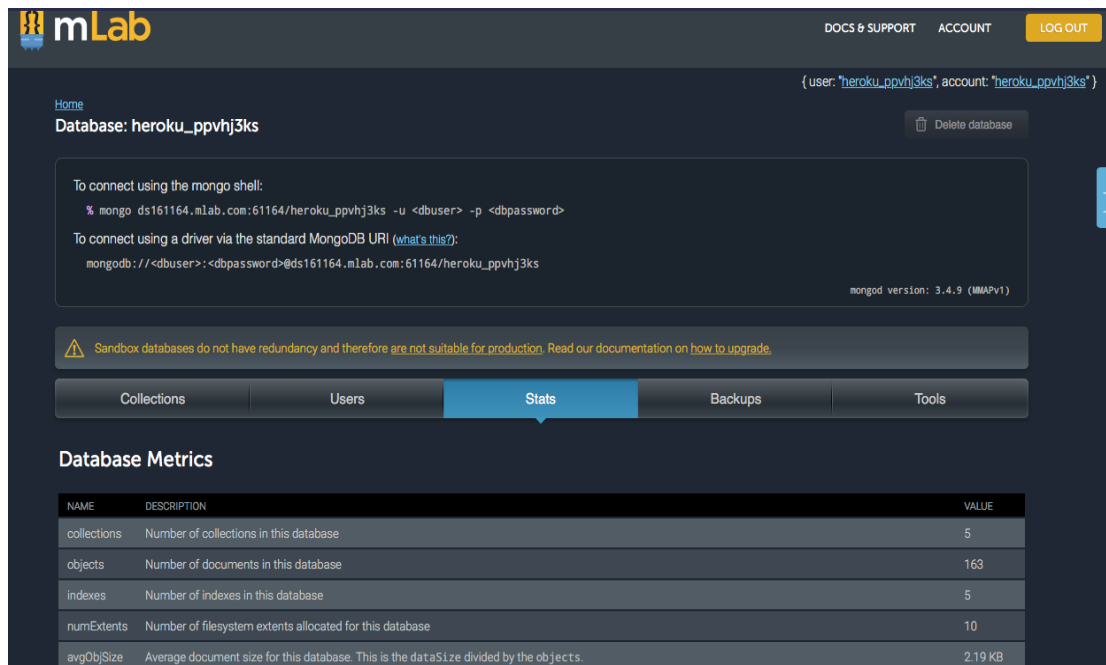
5. Servicio de correo

Figura 109. Servicio de envío de correos Mailgun.



6. Base de datos

Figura 110. Base de datos MongoDB almacenada en mLab.



7. Plataforma como servicio.

Figura 111. Panel de ambientes del API en Heroku.

The screenshot shows the Heroku dashboard for the application 'api-dev-gobhash'. The top navigation bar includes 'Personal apps', 'api-gobhash', and 'api-dev-gobhash'. Below the navigation, there are tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The main content area is divided into several sections:

- Installed add-ons:** Shows a list of add-ons with their status and pricing. Add-ons include Librato (Development), Logentries (TryIt), mLab MongoDB Sandbox, New Relic APM (Wayne), Opbeat APM (Free), and RedisMonitor (Free).
- Dyno formation:** Shows the current formation configuration, indicating that the app is using free dynos. The configuration is 'web' with 'npm start' command, and the status is 'ON'.
- Collaborator activity:** Shows a list of collaborators, including 'pablodiaz@gobhash.com' with 97 deploys.
- Latest activity:** A list of recent deployment and build activities, including 'Deployed' and 'Build succeeded' events with timestamps and version numbers.

8. Integración continua

Figura 112. Servicio de pruebas automatizadas Travis CI.

The screenshot shows the Travis CI dashboard for the repository 'GobHash / GobHash-Backend'. The dashboard includes a search bar, navigation links, and a list of repositories. The main section displays the build status for the 'integration-widget' branch, which is currently failed.

My Repositories:

- GobHash/GobHash-Backend # 86:** Duration: 6 min 46 sec, Finished: about an hour ago.
- fcpauldiaz/ChapiVentas # 20:** Duration: 5 min 22 sec, Finished: 8 days ago.
- diegojacobs/ChapiVentas-Angu # 53:** Duration: 1 min 34 sec, Finished: 10 days ago.
- Newton-Labs/Learn-in-UDV # 118:** Duration: 3 min 49 sec, Finished: 8 months ago.

Default Branch:

✓ master	# 2 passed	f9be048	✓	✗			
2 builds	17 days ago	GitHub					

Active Branches:

✗ integration-widget	# 86 failed	c0d3451	✗	✗	✗	✗	✗
83 builds	about an hour ago	Pablo Díaz					
✗ widgets	# 3 failed	c2fdcd0	✗				
1 build	17 days ago	willys7					

Figura 113. Resumen de pruebas unitarias ejecutadas.

```

## User APIs
# POST /v1/users (387ms)
✓
✓ (281ms)
# POST /v1/users/password/update (401ms)
✓
✓ (152ms)
# GET /v1/users/:userId (150ms)
✓
# POST /v1/users/biography (133ms)
✓
# GET /v1/users/follow/:userId/check (118ms)
✓
# POST /v1/users/follow (121ms)
✓
# POST /v1/users/unfollow (119ms)
✓
# PUT /v1/users/:userId (282ms)
✓
# GET /v1/users/ (127ms)
✓
# PATCH /v1/users/profile (110ms)
✓
# GET /v1/users/profile (130ms)
✓
# DELETE /v1/users/ (126ms)
✓

47 passing
Done in 16.67s.

```

9. Pruebas de rendimiento feed

Figura 114. Estado base de la aplicación en reposo en Keymetrics

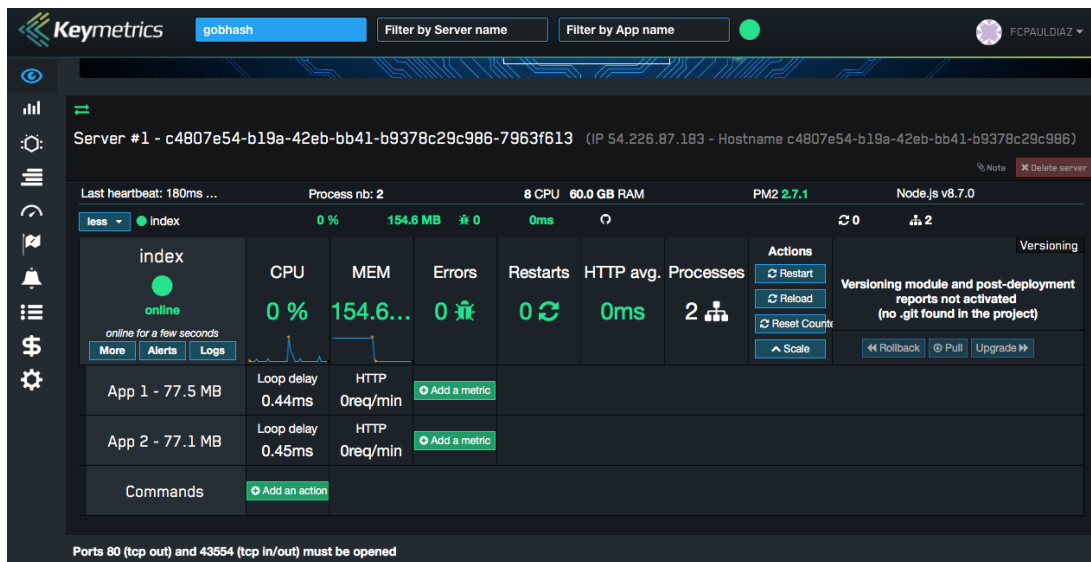


Figura 115. Resultado prueba de estrés feed.

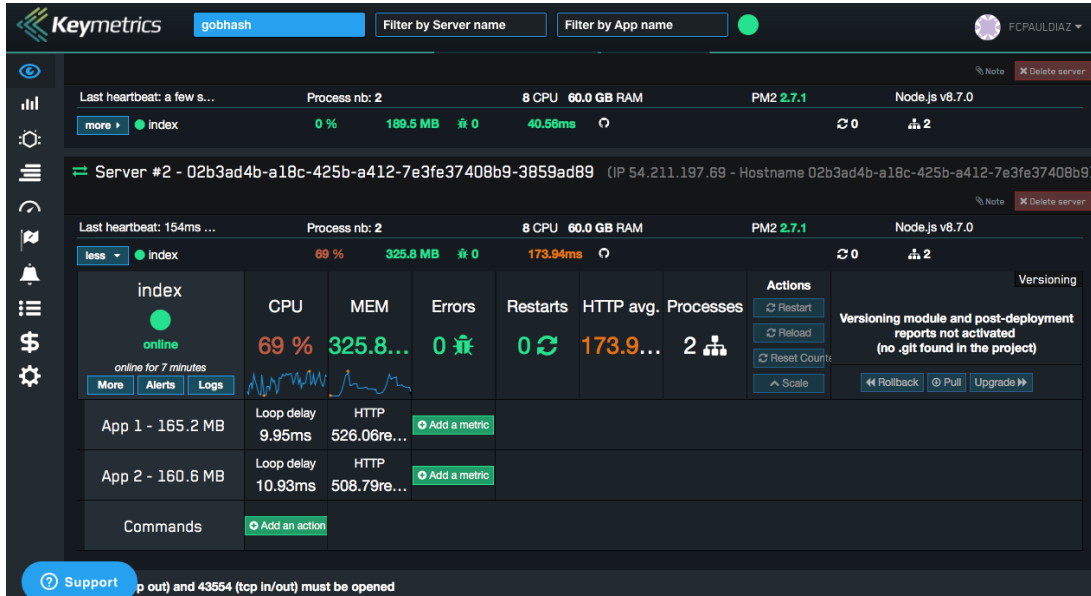
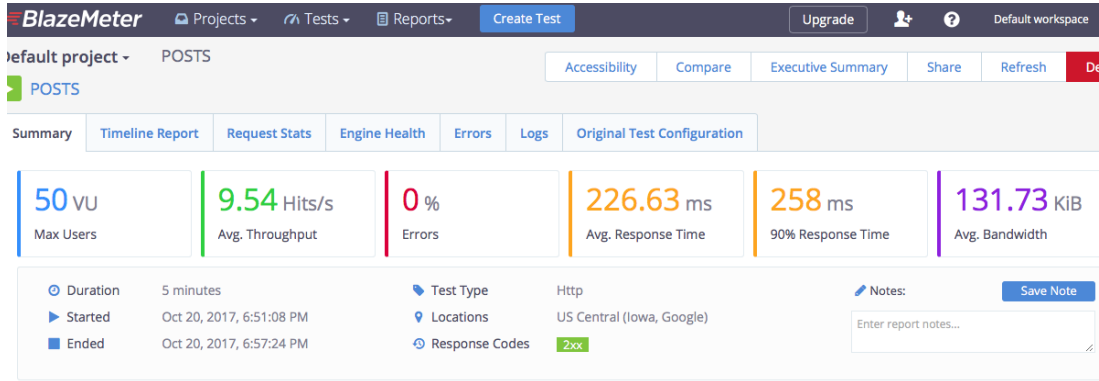


Figura 116. Resultado prueba de estrés feed .



Figura 117. Resumen resultado de prueba de estrés feed.



10. Pruebas de rendimiento del endpoint de gráficos

Figura 118. Resultado prueba de estrés endpoint gráficos

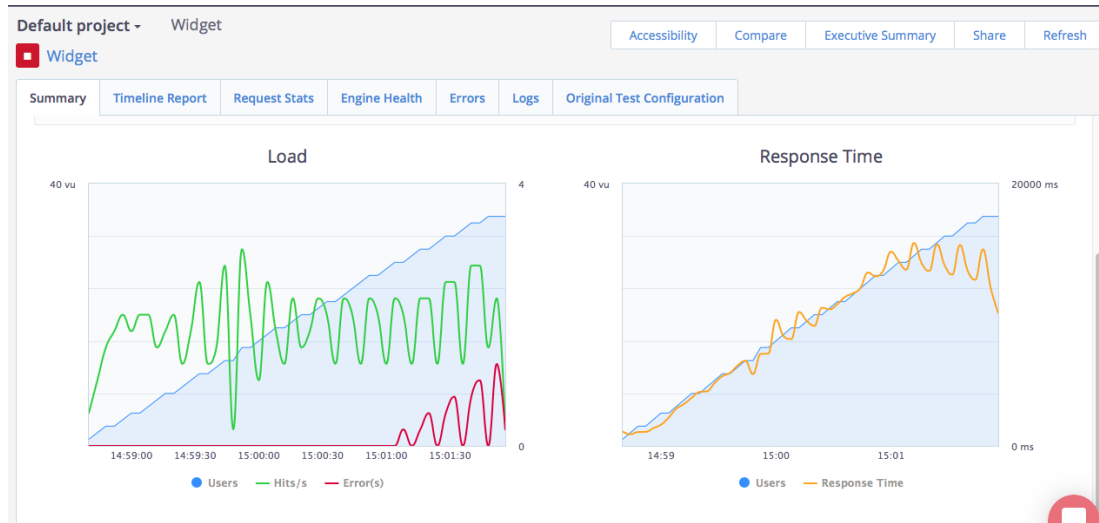
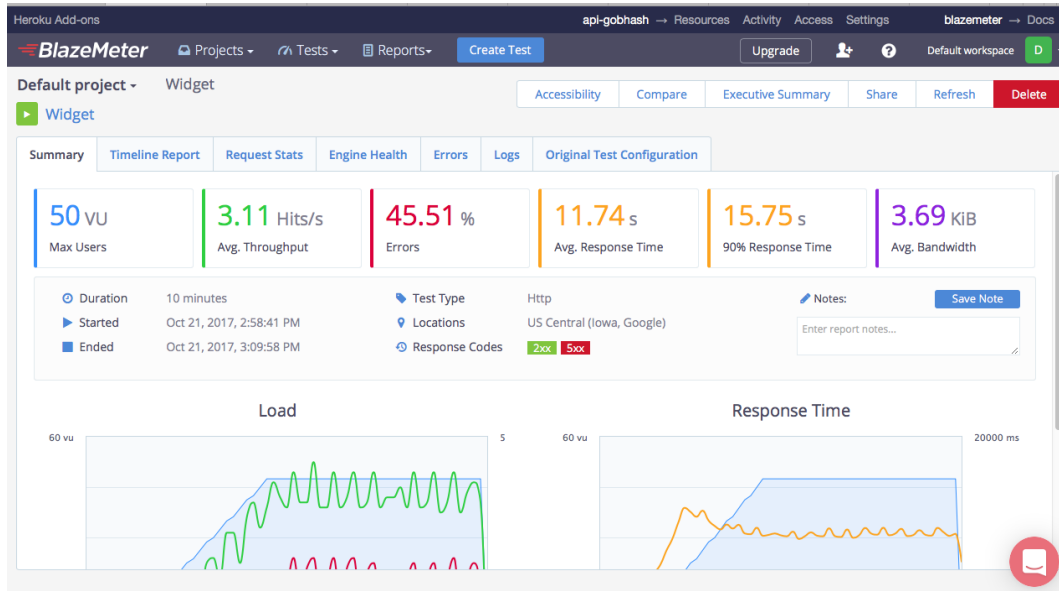


Figura 119. Resumen prueba de estrés endpoint gráficos.



11. Pruebas de rendimiento del protocolo del tiempo real.

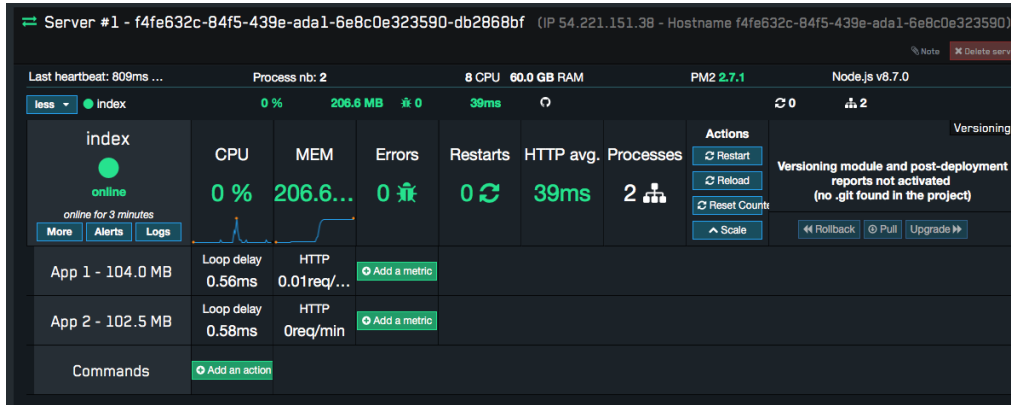
Figura 120. Prueba de estrés del protocolo de tiempo real con Artillery

```

All virtual users finished
Summary report @ 21:40:07(-0600) 2017-10-22
Scenarios launched: 100
Scenarios completed: 100
Requests completed: 100
Concurrent users: 41
RPS sent: 10.24
Request latency:
  min: 0.2
  max: 5
  median: 0.3
  p95: 0.9
  p99: 3.2
Scenario duration:
  min: 210.6
  max: 1740.4
  median: 319.5
  p95: 1101.4
  p99: 1722.3
Scenario counts:
  0: 100 (100%)
Codes:
  0: 100

```

Figura 121. Resultado de prueba de estrés del protocolo de tiempo real con Artillery.



C. ANÁLISIS DE RESULTADOS

1. Tecnologías. Se escogió la tecnología de Node JS por su compatibilidad con bases de datos SQL y NoSQL, ya que mantiene una sintaxis muy similar para el ODM y el ORM. Node JS utiliza JavaScript como lenguaje de programación lo que hace más fácil su entendimiento con los módulos que se dedican al frontend ya que también están basados en JavaScript. JavaScript al ser un lenguaje muy utilizado en el desarrollo web, disminuye la curva de aprendizaje y el costo de aprender un nuevo lenguaje para un nuevo desarrollador, lo que disminuye el tiempo de desarrollo. Se utilizó la nueva especificación de JavaScript ES6 para tener una sintaxis más actualizada, funcional y estandarizada junto con la especificación ES7 que se utilizó *async/await* para eliminar el fenómeno llamado *callback hell*, el cual consiste en que una función tiene funciones anidadas como parámetros. Node Js tiene un ambiente asíncrono lo que permite ejecutar tareas en forma no secuencial sino simultánea y tiene una operación *non-blocking* que permite ejecutar tareas sin que se hayan terminado otras que puedan bloquear ciertas otras operaciones. Por eso se dice que Node JS con Express JS a nivel de rendimiento es una de las mejores según la investigación del marco teórico. Otro aspecto relevante evaluado fue el soporte para realizar transacciones en línea en tiempo real. No todas las tecnologías evaluadas tenían un soporte nativo del protocolo para la tecnología de tiempo real.

Se hizo una comparación con otros frameworks donde se consideraron métricas relevantes al proyecto. Otra de las razones por la cual se escogió Node JS con Express JS es por su flexibilidad ya que no tiene dependencias que restrinjan las arquitectura con la que se debe trabajar. Esto permite tener un nivel de personalización profunda porque se pueden hacer cambios sobre aspectos específicos. Esto no es posible en otros frameworks como

Django que ya traen incluido el ORM y modificar ciertas partes del framework no es posible o es muy complejo. Una de las desventajas del ORM seleccionado es que no tiene migraciones automáticas como otros ORMs en otras tecnologías que hace muy sencillo las alteraciones en las bases de datos Relacionales.

2. Bases de datos. La base de datos que almacena la información de GuateCompras es una base de datos PostgreSQL, escogida por el módulo de diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico. La base de datos para la red social GobHash es una base de datos de MongoDB debido a su flexibilidad ya que no utiliza tablas para almacenar la información sino que utiliza el concepto de documentos para guardar la información. Esto permitió agregar funcionalidades no planificadas y según era necesario escalar la aplicación con una flexibilidad total que permitió un desarrollo más ágil y con menos costos. Al principio de este proyecto la base de datos de la red social se comenzó con una base de datos Relacional por la necesidad de empezar a desarrollar pero el módulo de diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico no había terminado su análisis y cuando lo terminó se decidió que la mejor opción era usar un modelo no Relacional, esto conllevó a realizar un cambio en todo el API para hacerlo compatible. Debido a la tecnología usada y la sintaxis actualizada de JavaScript escogida el cambio solo duro 4 horas porque la sintaxis era muy similar. Esto se considera una ventaja ya que el tiempo fue relativamente corto para hacer un cambio drástico en el backend.

El backend utiliza una base de datos llamada Redis, que se basa en campos de llave – valor para guardar información temporal.

3. Diseño e implementación del API. El API se diseñó para utilizar prácticas comúnmente usadas para describir lo mejor posible un endpoint utilizando la URL como sustantivo y los métodos de http como verbos para indicar las acciones que se podían realizar en un endpoint. Un API es tan bueno como lo es su documentación por lo que se utilizaron dos librerías para documentar el API, Swagger UI y Docbox, respectivamente. Debido a que el API era privado para los desarrolladores se implementó una versión modificada de estas librerías para implementar autenticación por lo que el código del backend se dividió en dos repositorios, uno para el código y otro para la documentación. Se utilizó Docbox para explicar cómo empezar a usar el API, como autenticar las solicitudes

http y cómo implementar el protocolo de tiempo real. En la interfaz de Swagger se listaron los endpoints disponibles para consultar junto con la información necesaria para consultarlos.

Se realizaron 47 pruebas unitarias para probar la funcionalidad individual de cada función del API, lo cual otorga cierto nivel de confianza para los valores esperados al consultar los endpoints para poder mantener funcional la aplicación con cada commit realizado. Hay que notar que las pruebas unitarias son tan buenas como hayan sido programadas y las pruebas unitarias no son suficientes para garantizar un funcionamiento correcto de toda la plataforma pero no fueron implementadas por la delimitación y alcance de este módulo. Antes de realizar un commit se ejecutan las pruebas unitarias y se ejecuta el linter que verifica la sintaxis de código. Cuando esto se cumple se procede a ejecutar las pruebas en un ambiente de pruebas usando los servidores del servicio de Travis CI. Si estas pruebas son ejecutadas correctamente y se mantiene un mínimo de 80% del código sujeto a pruebas se procede a actualizar el código en el ambiente de Heroku.

Respecto al dominio se procedió a comprarlo en GoDaddy porque incluye un consola de administración sencilla para administrar el dominio. El dominio fue re dirigido a CloudFlare, que es una plataforma enfocada en seguridad, que sirvió para administrar los DNS a diferentes plataformas. Por ejemplo, se configuraron los DNS para para el servicio de correo de Mailgun y los dominios de gobhash.com, api.gobhash.com y api-dev.gobhash.com. Aparte de eso CloudFlare agrega un certificado SSL gratuito para cifrar la información transmitida y ofrece disminución en el tamaño de los archivos para incrementar el rendimiento.

En todos los endpoints del API se incluye la versión del API, en este momento debido a que no ha salido a producción solo existe la versión 1.

Respecto a la implementación del protocolo del transporte de información en tiempo real se tuvo dos acercamientos. El primero consistía en que un usuario esperaba a través de un sistema de colas esperando a recibir las publicaciones de los usuarios que estén en su listado de *following*, esto agregaba complejidad a nivel de lógica e infraestructura por lo que se descartó este acercamiento. El acercamiento más sencillo era verlo desde el lado de los *followers*. Lo que se implementó fue que cuanto un usuario creaba un post se revisa que

usuarios están su lista de followers y se revisa cuales están online (disponibles para mandarles la información). Si estaban online se enviaba la información a un canal exclusivo donde solo se encuentra el usuario. Fue importante crear un canal exclusivo para cada usuario online porque no se le tenía que notificar a todos los usuarios online, sino solo a uno en específico. También se implementó autenticación en el protocolo porque este no usa la sesión guardada en el explorador web. Entonces cada vez que un usuario ingresa al *feed* de publicaciones por primera vez, se autentica con el mismo token de JWT para autenticar al usuario en el protocolo de *websockets* y crear un canal con un id único que le pertenece al usuario.

Con respecto a los requisitos funcionales, se lograron implementar todos los definidos. Cabe mencionar que el módulo de análisis, diseño e implementación del frontend no incluyó todos los requisitos funcionales desarrollados en su delimitación. El requisito funcional de búsqueda de usuarios y publicaciones fue implementado en el backend pero no fue integrado en el frontend. Los servicios de terceros incorporados funcionaron para completar los requisitos funcionales, mantener el correcto funcionamiento de las dependencias, mantener la calidad de código y verificar posibles vulnerabilidades de seguridad.

4. Rendimiento. Las pruebas de estrés se realizaron acorde a los endpoints con más carga que pueden ser un cuello de botella, es decir los endpoints que son los más críticos para el backend. Como referencia se utilizó el API en estado de reposo donde consume 0% de CPU y 154 mb de RAM según la figura 120. También se toma la cantidad de solicitudes realizadas antes de las pruebas de estrés en CloudFlare, donde se indica que son 2, 913 según la figura 121.

Las tres primeras pruebas de estrés (figura 22 hasta la figura 27) fueron pruebas que fueron escalando la cantidad de solicitudes http al endpoint del feed hasta las 1, 200 solicitudes. Los recursos llegaron hasta 69% de uso del CPU en un momento y 325 mb de RAM. Se considera que las pruebas de estrés pasan exitosamente porque no consume el 80% de los recursos de la infraestructura actual. La prueba de estrés con Blaze Meter en la figura 30 ilustra gráficamente el uso concurrente de 50 usuarios. Se aprecia que el tiempo de respuesta se mantiene constante mientras aumenta la cantidad de solicitudes al servidor linealmente. En esta gráfica se puede apreciar el performance que provee Node JS a la hora

de recibir múltiples solicitudes. El tiempo de respuesta promedio registrado fue de 226 ms. Se cumple el requisito no funcional de un tiempo de respuesta máximo de 1 segundo. Para comparar que tan bueno es este tiempo de respuesta, se agregó en los anexos el tiempo de respuesta promedio de un ping a google y el tiempo de respuesta del API de *amazon products*. Los tiempos fueron de 69 ms para el ping y 644 ms para el API de Amazon. El tiempo de respuesta se encuentra menor que el de Amazon por lo que se considera suficientemente bueno.

La siguiente prueba de estrés realizada fue al endpoint de los gráficos. Este podía ser un endpoint que requiere recursos porque es el que consulta la base de datos de GuateCompras. Este fue desarrollado por el módulo de gráficos pero se procedió a probar. Según la gráfica de la figura 104 a partir de los 22 usuarios concurrentes empieza a generar errores el endpoint. En la misma figura se puede apreciar que con una mayor cantidad de solicitudes empieza a incrementar el tiempo de respuesta del servicio, en promedio fue de 11.54 s. Estos parámetros no cumplen con los requisitos no funcionales de la metodología, debido a que no es parte de este módulo solo se procedió a investigar la causa y documentarlo. Al revisar la figura 106 se pudo ver que la RAM tenía un consumo de 245 mb y CPU de 0% por lo que no era problema de recursos de infraestructura de la aplicación. Al revisar la gráfica de la figura 37 se encontró que el proceso de Node era el que más tomaba tiempo. Se aumentó la cantidad de conexiones concurrentes para la base de datos de GuateCompras para probar si era eso pero el resultado fue el mismo. Al investigar más a profundidad se encontró que el problema era la configuración de la conexiones con el ORM utilizado por el módulo de gráficos. La configuración actual solo permitía 5 conexiones concurrentes por lo que falló en la prueba de estrés, se comunicó los resultados al responsable para arreglar la configuración.

Finalmente, la última prueba de estrés se realizó al protocolo de comunicación de tiempo real. El módulo de interacción humano – computador definió una cantidad de 100 usuarios usando la plataforma al mismo tiempo por lo que se procedió a utilizar este parámetro como prueba. La figura 118 muestra 100 usuarios concurrentes utilizando el protocolo y la figura 119 muestra los recursos utilizados, 0% de CPU y 206 de RAM. Por lo tanto, se considera que las pruebas fueron completadas exitosamente ya que no utilizó más del 80% de los recursos de la infraestructura.

5. Infraestructura. Se utilizó la plataforma de Heroku para tener una infraestructura fácil de usar, de esta forma se minimizó el tiempo invertido en subir las aplicaciones a los servidores ya que fue un proceso automatizado por la plataforma. Se utilizaron un ambiente de desarrollo y un ambiente de producción para mantener el API. Esto permite que solo la versión ya probada, es decir, la estable sea pasada a producción para que sea usada por el frontend. Al principio la aplicación estaba configurada para utilizar 4 nodos pero al realizar las pruebas de estrés cada nodo consumía la misma cantidad de RAM por lo que al estar bajo carga la RAM de la infraestructura empezó a llegar a su límite por lo que se decidió que era suficiente tener 2 nodos de la aplicación para soportar a la cantidad de usuarios definida por el módulo de diseño y desarrollo de estudio de usabilidad para la interacción humano-computador. Para manejar las solicitudes de http se utiliza un balanceador de carga *round robin* que distribuye la carga uniformemente entre los nodos disponibles. Con la infraestructura se tuvo dificultades ya la configuración de tipo clúster no permitió el manejo de sesiones, ya que la sesión se creaba en un nodo pero no se compartía en los demás. Entonces al realizar la siguiente solicitud no se completaba porque el siguiente nodo no tenía la sesión guardada. Este fenómeno se dio solamente en el protocolo de tiempo real. Para solucionar este problema se instanció una base de datos Redis, que guardó las sesiones de los usuarios para que pudiesen ser compartidas a través de los nodos. Resolver este problema tuvo una duración de 12 horas, lo cual se considera bastante tiempo para resolverlo.

X. ANÁLISIS, DISEÑO Y DESARROLLO DE FRONTEND

A. METODOLOGÍA

Inicialmente se definieron los requisitos funcionales y no funcionales, esto se hizo junto con los módulos de *Diseño y desarrollo de usabilidad para la interacción Humano-Computador* y *Desarrollo e implementación de backend*. El primer módulo definió los requisitos funcionales ya que en dicho módulo se listan todas las funcionalidades de la plataforma GobHash y con el segundo módulo se definió todo lo que se iba a necesitar para poder cumplir con las funcionalidades listadas.

Requisitos funcionales definidos:

- Usuario:
 - Registro
 - Inicio de sesión
 - Solicitar cambio de contraseña
 - Recuperar contraseña
 - Perfil del usuario
 - Seguir a otro usuario
- Feed:
 - Feed en vivo: Mostrar en tiempo real los posts de las personas a las que sigue el usuario
 - Feed inicial: Mostrar los posts de los usuarios que sigue y los del mismo usuario
- Post
 - Crear post:
 - Agregar título
 - Agregar descripción
 - Agregar *dashboard*:
 - Gráfico principal
 - Gráfico 1
 - Gráfico 2
 - Gráfico 3
 - Hash post: Esto es el equivalente a “Me gusta el post”

▪

Una vez definidos los requisitos funcionales fue necesario definir los requisitos no funcionales, requisitos a cargo del buen funcionamiento de la plataforma

Requisitos no funcionales definidos:

- Seguridad informática
- Escalabilidad
- Compatibilidad
- Mantenimiento
- Comunidad
- Soporte
- Curva de aprendizaje
- Control de versiones

1. ANÁLISIS

a. Análisis de requerimientos funcionales. Tomando como referencia los requisitos definidos se realizaron los diagramas de flujo para cada una de las funcionalidades con el fin de establecer lo que se harían en cada una ellas. Se analizó desde dos perspectivas, desde la perspectiva del usuario e *interacción frontend-backend*, durante el análisis de la interacción *frontend-backend* se definieron los *endpoints* que se utilizarían y la estructura de ellos (parámetros e información a recibir si aplica).

Una vez definida la interacción *frontend-backend* y flujo de funcionalidad desde la perspectiva del usuario, se llevó a cabo una investigación para saber cómo funciona una red social y cómo desarrollar una. Para esto se realizó una ingeniería inversa a la red social *Facebook*, en donde se encontraron resultados interesantes sobre cómo desarrollar la red social (ver Apéndice), por lo cual se decidió utilizar como referencia a *Facebook*.

Luego se definieron los conceptos o terminología a utilizar:

- Usuario: Esto representa a cada una de las personas que se registra en la plataforma.
- *Layout*: Esto representa el conjunto de gráficos generados por el módulo de *Desarrollo e implementación de indicadores estadísticos (Gráficos/Widgets) con base en la información extraída del portal de GuateCompras*.
 - Post: Esto representa el estudio descriptivo que realiza un usuario, por lo cual incluye un *Layout*

- Comentario: Esto representa la discusión de un usuario hacia el post de otro usuario o de sí mismo.

b. Diagramas de flujo por funcionalidad

Figura 122. Diagrama de flujo: Inicio de sesión

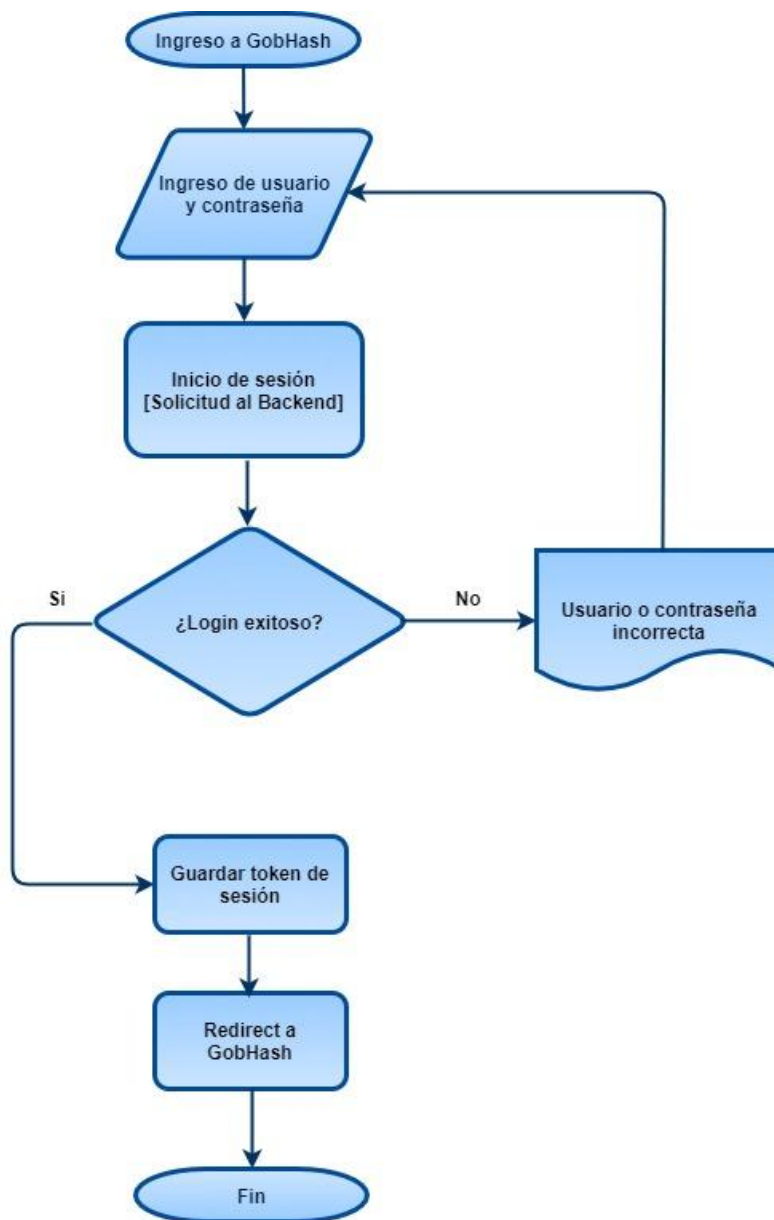


Figura 123. Diagrama de flujo: Registro de usuario

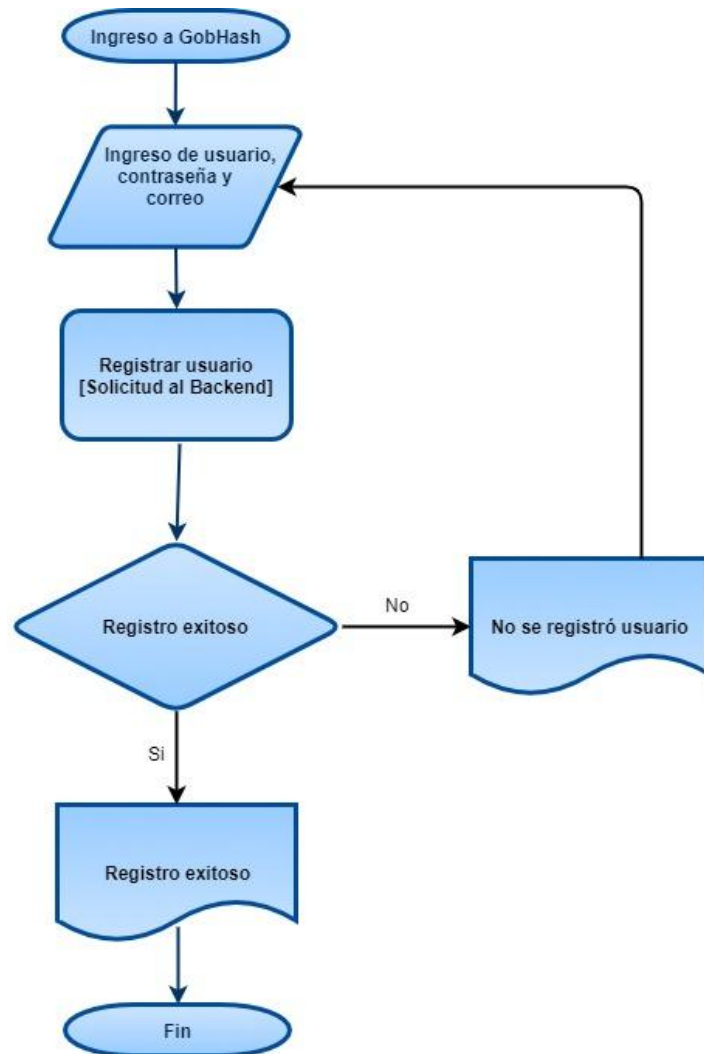


Figura 124. Diagrama de flujo: Perfil del usuario

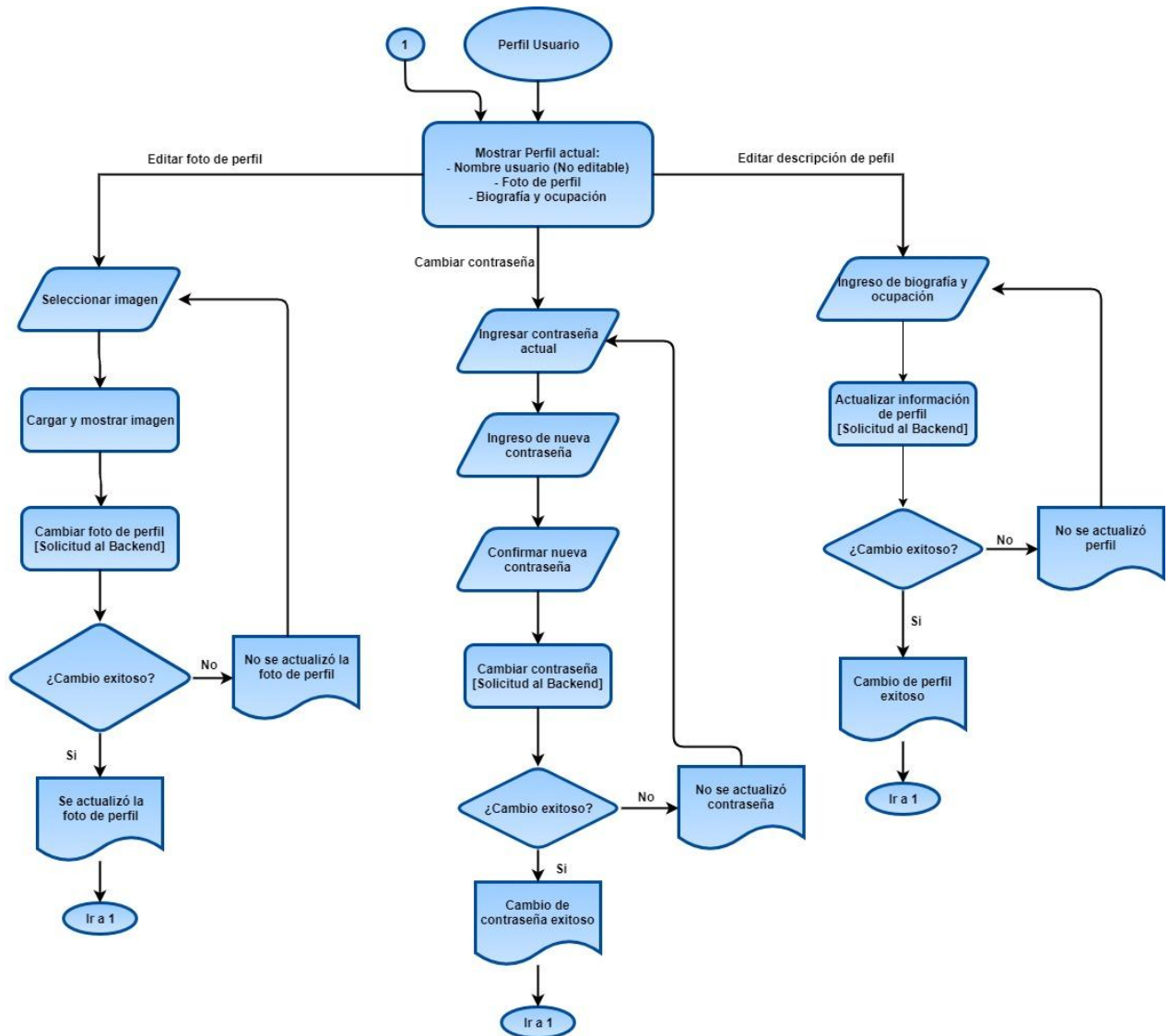


Figura 125. Diagrama de flujo: Solicitar cambio de contraseña

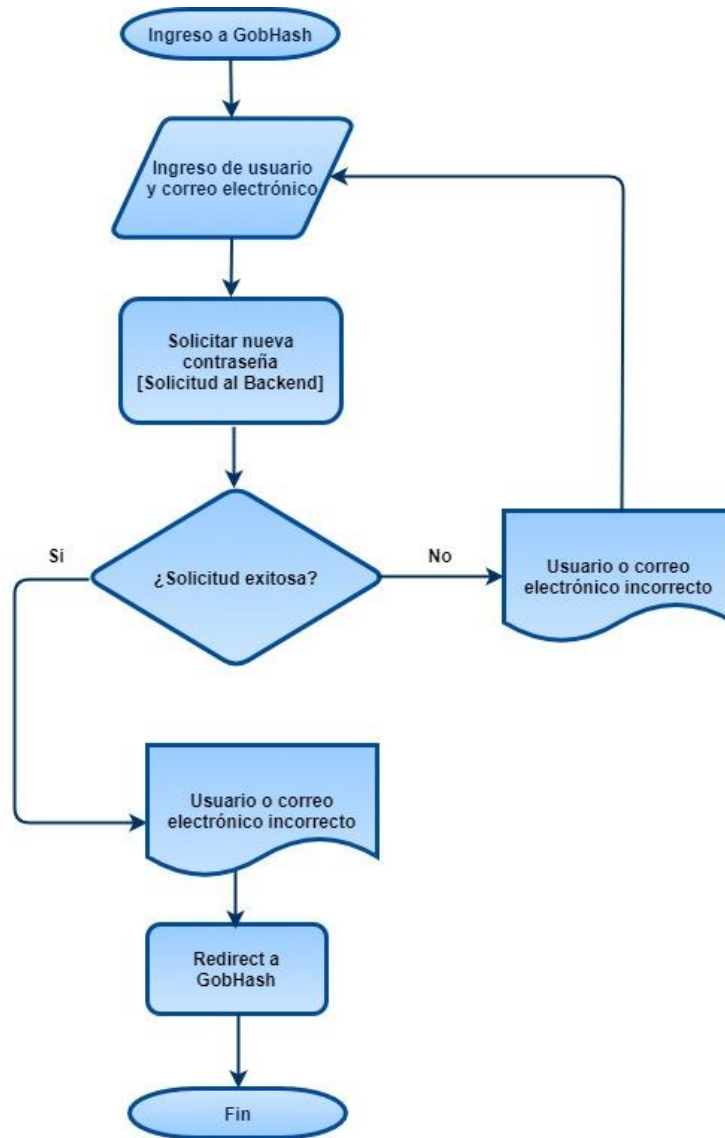


Figura 126. Diagrama de flujo: Recuperación de contraseña por medio de token

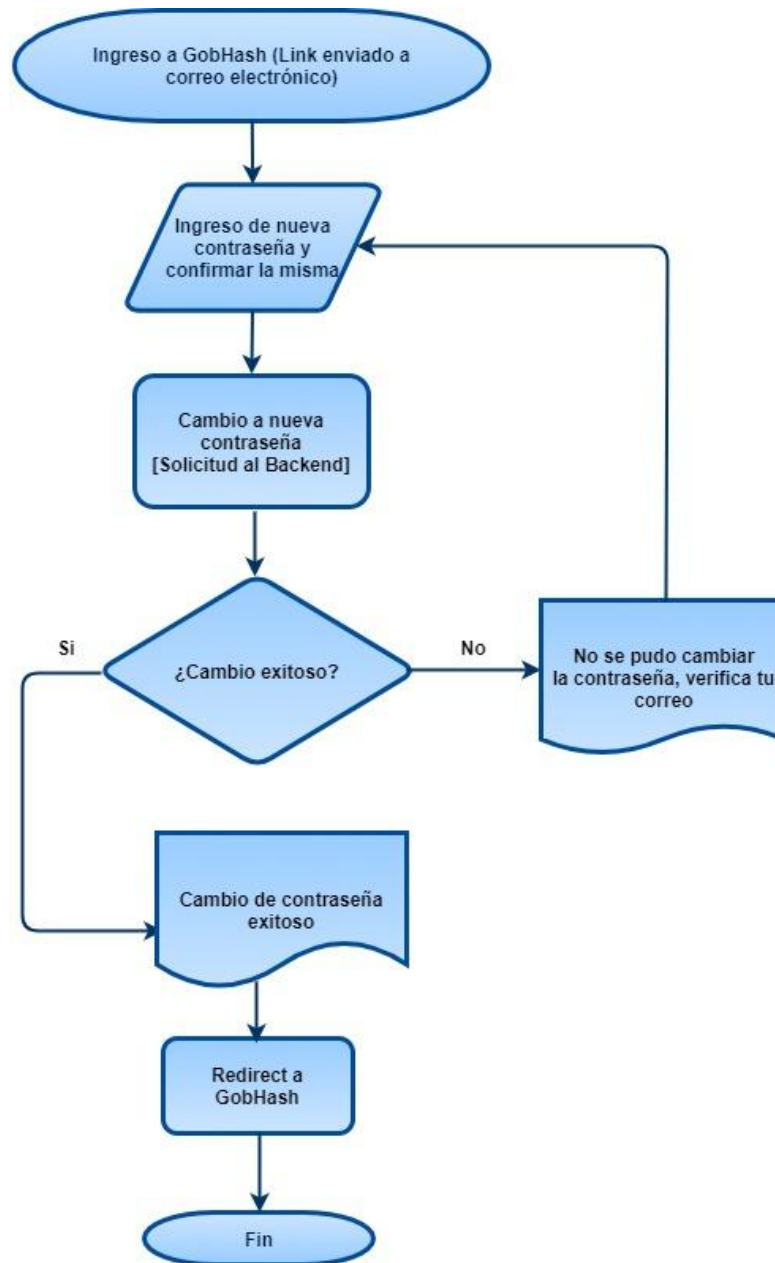


Figura 127. Diagrama de flujo: Crear una publicación "Post"

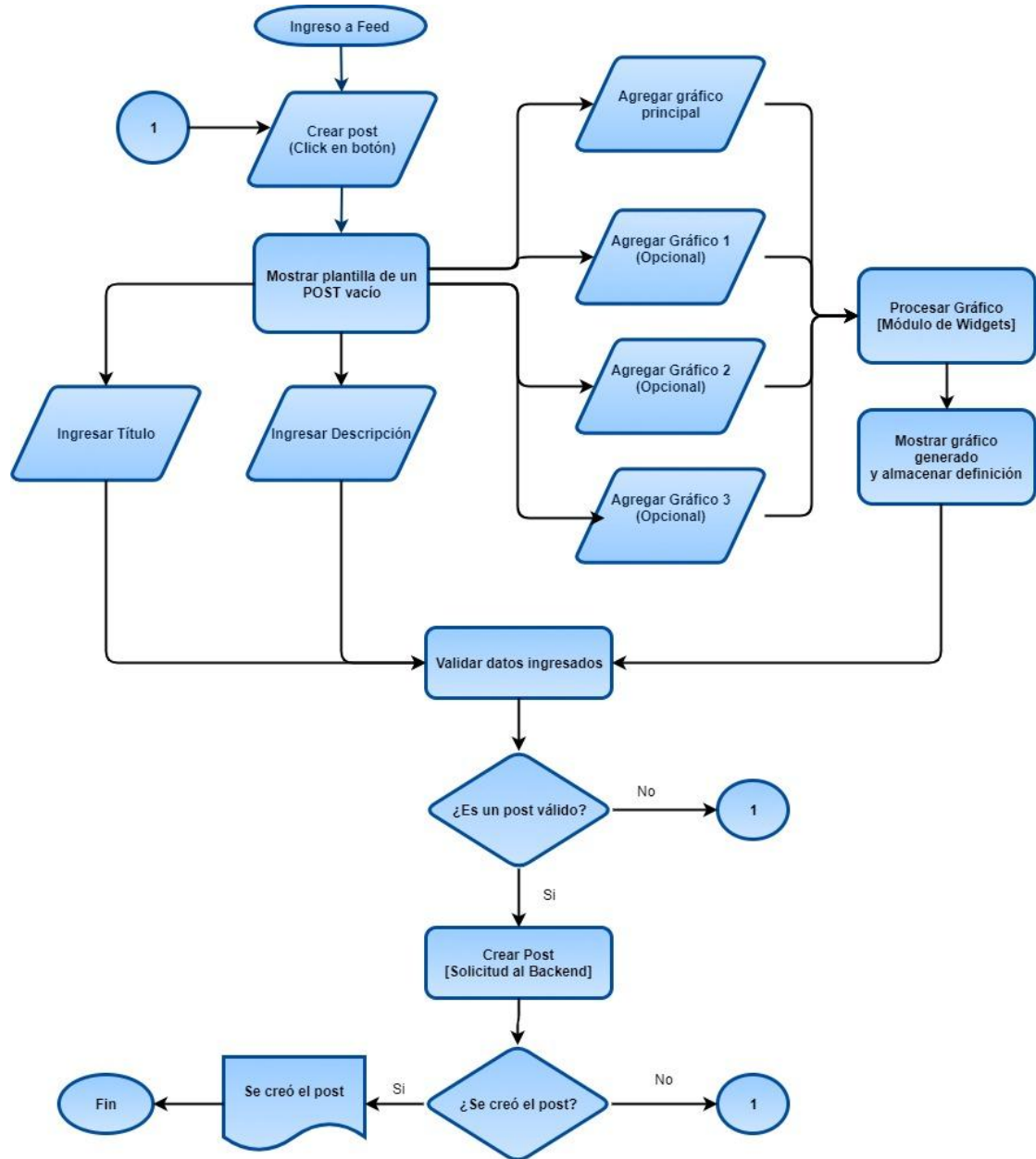


Figura 128. Diagrama de flujo: Feed Inicial

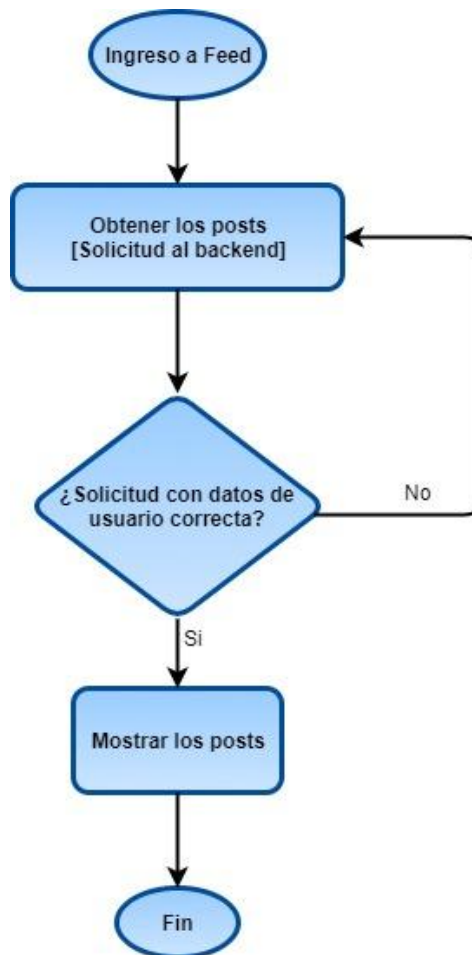
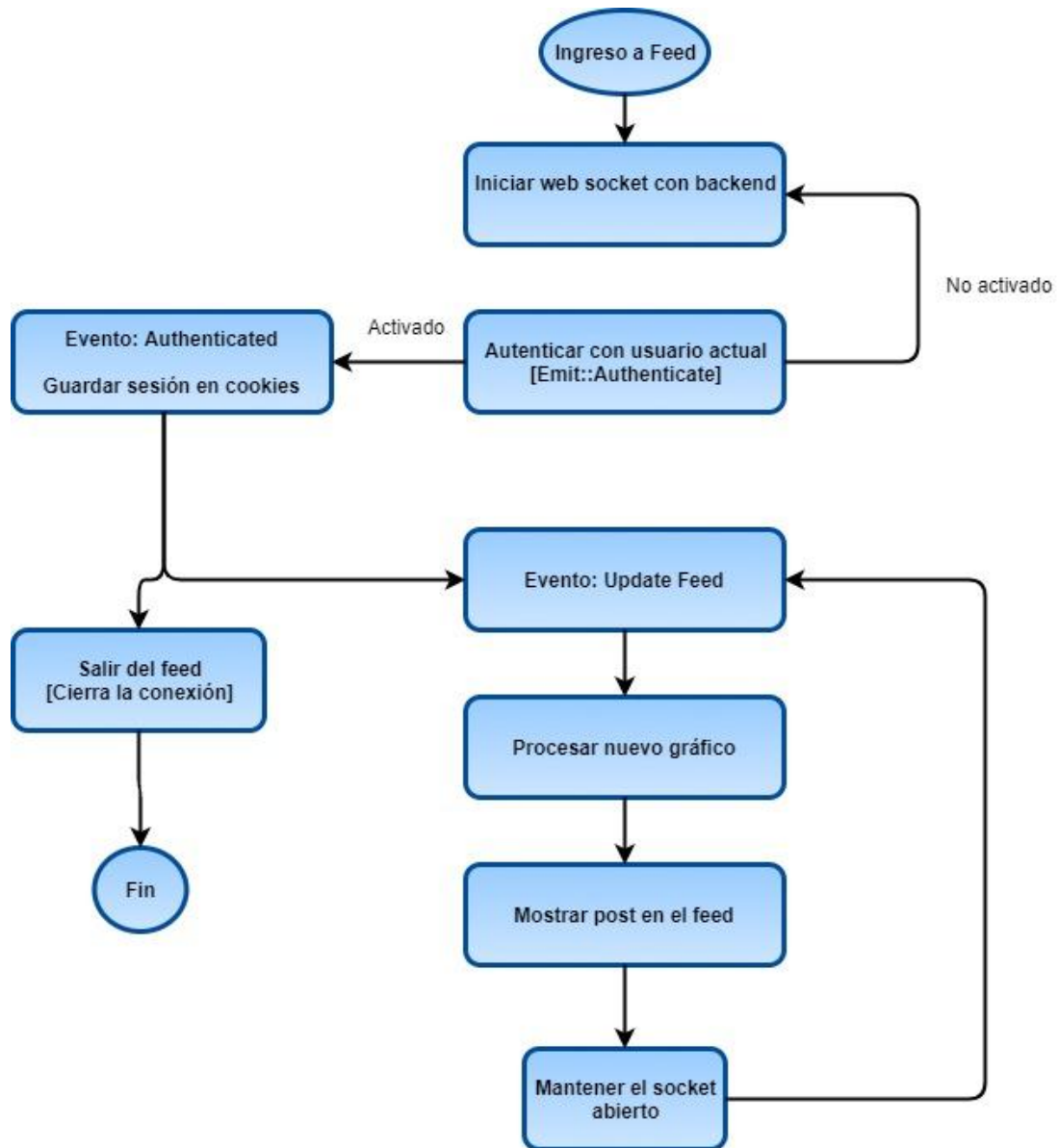


Figura 129. Diagrama de flujo: Publicaciones en tiempo real



c. Análisis de requerimientos no funcionales. Tomando como referencia los requisitos definidos se realizó una investigación exhaustiva sobre tecnologías disponibles en el mercado para decidir cuál se adaptaba de mejor manera a las necesidades del proyecto, las cuales son:

- Alta interactividad web
- Soporte de cambios frecuentes de modelos
- *Single Page Application*

Esta investigación se llevó a cabo en dos etapas y cada una de ellas con criterios diferentes.

1) Primera etapa. Para esta etapa se utilizaron los siguientes criterios:

- Seguridad informática
- Escalabilidad
- Mantenimiento
- Compatibilidad

Tomando en cuenta dichos criterios se encontraron las siguientes tecnologías:

- *AngularJS*
- *EmberJS*
- *ReactJS*

Para seguridad informática se tomaron en cuenta los siguientes criterios:

- *CSRF protection*
- *DOM-XSS protection*
- *HTML Sanitizer*

Para escalabilidad se tomaron en cuenta los siguientes criterios:

- Programación funcional
- Funciones primitivas
- *Single Responsibility*
- Horizontal y vertical

Para mantenimiento se tomaron en cuenta los siguientes criterios:

- Autor: Sea un autor reconocido y que tenga responsabilidad sobre el *framework*
- Contribuidores: Una alta comunidad de desarrolladores aportando continuamente a mejoras o parches del *framework*
- Control de versiones: Cuenta con una serie de versiones mostrando un precedente sobre el mantenimiento y mejoras realizadas de versión en versión
- Código abierto: Cualquier persona pueda contribuir
- Libertad de uso: Se pueda desarrollar una red social con esa tecnología

Para compatibilidad de utilizaron los siguientes criterios:

- Compatibilidad con plantillas
- Compatibilidad con los distintos navegadores web
- Compatibilidad con librerías

Tabla 24. Primera comparación sobre tecnologías para desarrollo de Frontend

Tecnología	Seguridad informática	Escalabilidad	Mantenimiento	Compatibilidad
AngularJS	Cumple con los 3 criterios	Cumple con los 4 criterios	Cumple con los 5 criterios	Cumple con los 3 criterios
EmberJS	Cumple con los 3 criterios	Cumple con los 4 criterios	Cumple con los 5 criterios	Cumple con los 3 criterios
ReactJS	Cumple con 2, no cuenta con HTML <i>Sanitizer</i> dentro del <i>framework</i>	Cumple con los 4 criterios	Cumple con 4 criterios. No se puede utilizar para desarrollar una aplicación que compita contra Facebook (Red Social)	Cumple con los 3 criterios

2) Segunda etapa. Para esta etapa se utilizaron los siguientes criterios:

- Comunidad
- Soporte
- Curva de aprendizaje

Tomando en cuenta estos criterios se decidió utilizar *AngularJS*.

Para comunidad se utilizaron los siguientes criterios:

- Reporte y solución de errores: Muchos foros y publicaciones sobre problemas encontrados por mala comprensión o mal uso del framework y soluciones.
- Documentación: Que explique bien sus conceptos base, caso de uso, ejemplos de cómo se utiliza cada uno de sus componentes y una definición de estructura.

Para soporte se utilizaron los siguientes criterios:

- Continuidad: Que el *framework* se encuentre en continuo desarrollo y que su última versión no sobrepase 6 meses de antigüedad.
- Nuevas versiones: Que el *framework* cuente con planes de nuevas versiones que no rompan lo actual.

Para curva de aprendizaje se utilizaron los siguientes criterios:

- Tendencia: Que tenga una alta tendencia lo cual implica que se utiliza mucho
- Facilidad de uso: Qué tan difícil es utilizar el *framework* en cuanto a conceptos que utiliza y cómo se utilizan
- Tiempo: Esto se refiere al tiempo que toma aprender a utilizarlo tomando en cuenta que se tiene conocimiento sobre *HTTP*, *HTML*, *CSS* y *JS*. Como criterio se utilizó que se tome menos o exactamente 4 meses.

Tabla 25. Segunda comparación sobre tecnologías para desarrollo de Frontend

Tecnología	Comunidad	Soporte	Curva de aprendizaje
AngularJS	Cumple con los 2 criterios	Cumple con los 2 criterios	Cumple con los 3 criterios
EmberJS	Cumple con los 2 criterios	Cumple con los 2 criterios	Cumple con la tendencia, sin embargo, no tiene facilidad de uso y toma aproximadamente 6 meses en utilizarlo
ReactJS	Cumple con los 2 criterios	Cumple con los 2 criterios, aunque la documentación es limitada	No cumple con ninguno de los 3 criterios

2. **DISEÑO.** Durante esta etapa de la metodología se definió la forma en la cual la plataforma GobHash fue desarrollada. En esta etapa se definió toda la estructura, librerías, versiones, conceptos, entornos de desarrollo/producción, buenas prácticas, entre otras cosas que se utilizaron a fin de cumplir con los requisitos de la plataforma GobHash.

El diseño se dividió en dos etapas:

- Diseño para cumplimiento de requerimientos funcionales
- Diseño para cumplimiento de requerimientos no funcionales

a. **Diseño para cumplimiento de requerimientos funcionales.** Una vez realizada la definición y análisis de los requerimientos funcionales se procedió a definir prioridad de funcionalidades, con esto nos referimos a la importancia que tiene cada una de ellas y el orden en que deben ser desarrollados según la función que cumplen dentro de la plataforma.

1) Orden de prioridad sobre las funcionalidades definida

- Usuario:
 - Registro de usuario
 - Inicio de sesión de usuario
 - Solicitud de cambio de contraseña (Olvidó su contraseña)
 - Recuperación de contraseña (Cambio de contraseña luego de la solicitud realizada)
- Post:
 - Crear un post
- *Feed:*

- Mostrar los posts creados
- *Feed* en vivo: Mostrar en tiempo real los posts creados por los usuarios a los que sigo
- Usuario:
 - Mostrar el perfil de un usuario
 - Editar perfil del usuario (Sesión activa)
 - Mostrar perfil del usuario al cual le pertenece un post
 - Permitir “Seguir” al usuario que creó un post
 - Cierre de sesión

Se escogió este orden de prioridad de desarrollo debido a que la base de todo son usuarios y posts. Antes de poder crear un post necesitamos los usuarios dado que son ellos quienes los crean, luego siguen los posts debido a que son la fuente de información principal del proyecto para realizar el análisis descriptivo, luego se decidió priorizar el *Feed* debido a que es en donde se le hace publicidad a cada uno de los posts y sus autores y finalmente las funcionalidades secundarias de un usuario como lo son su perfil, seguir otros usuarios y cierre de sesión.

2) Definición de la estructura de la plataforma. Durante esta etapa se definió la estructura de los estados y sus *payloads* a utilizar para poder representar cada uno de los requisitos funcionales.

Se decidió diseñar un tipo de máquina de estado debido a que de ese modo es más fácil encapsular o abstraer una funcionalidad.

3) Definición de modelos. Luego de haber realizado el análisis y diseño de funcionalidades se diseñaron los modelos necesarios, su estructura y su formato.

Se decidió utilizar JSON para definir los modelos debido a que es un formato de texto para intercambio de datos universal.

Los modelos que se definieron fueron:

- Usuario
- Post

Esos fueron los únicos modelos identificados como necesarios ya que son la información base. Por ejemplo, el *Feed* es un conjunto de posts de los que el usuario ha creado y de los que han creado los usuarios a los que sigue.

Figura 130. Diagrama de estados sobre la aplicación de GobHash

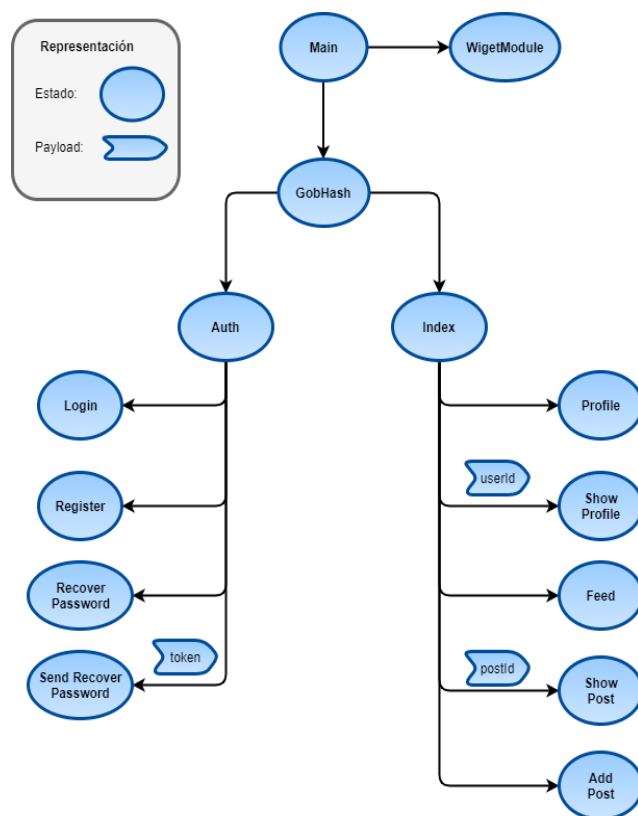
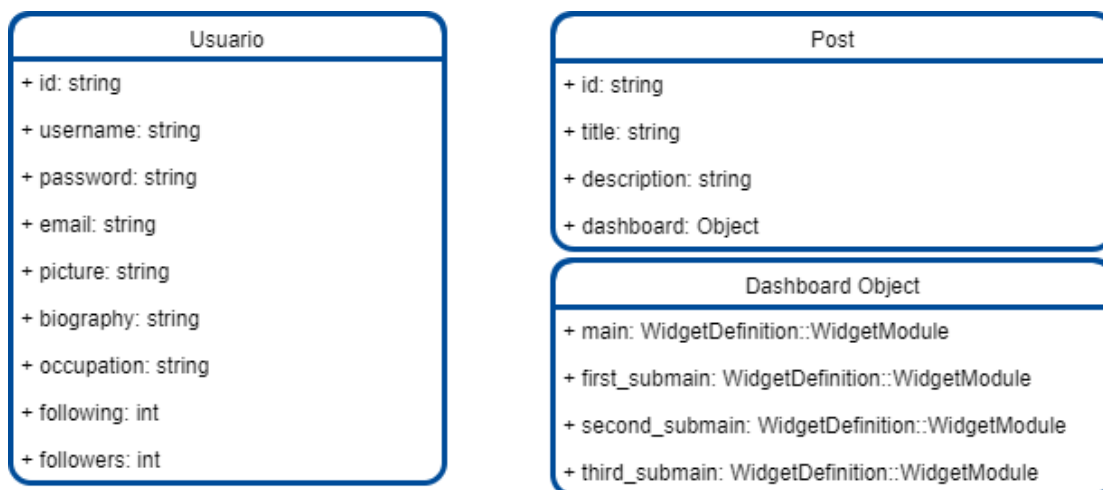


Figura 131. Modelo de datos definido para Frontend



4). Definición de Endpoints necesarios. Para poder identificar los *endpoints* necesarios del *backend* para que el *frontend* funcionara correctamente fue necesario volver a estudiar los requisitos funcionales dado que dependiendo de lo que se iba a permitir eso es lo que se debía soportar.

Luego de un análisis sobre los requerimientos funcionales y un análisis sobre los diagramas de flujo de cada uno de ellos se llegó a definir que los *endpoint* se debían separar en cuatro módulos:

- *Auth*: Este módulo se encargaría de toda la parte de autenticación para mantener un nivel de seguridad en cuanto a la comunicación *backend-frontend*, por lo que aquí se maneja:
 - Inicio de sesión
 - Solicitud para reiniciar contraseña
 - Reiniciar contraseña por medio de token
- *User*: Este módulo se encargaría de todo el manejo de usuarios (*CRUD: create, read, update and delete*), por lo que aquí se maneja:
 - Listado de usuarios
 - Crear usuario
 - Obtener información de un usuario
 - Eliminar un usuario
 - Manejo de información de biografía y foto de perfil
- *Post*: Este módulo se encargaría de todo el manejo de posts (*CRUD: create, read, update and delete*), por lo que aquí se maneja:
 - Crear post
 - Eliminar post
 - *Feed*: Listar los posts de uno o más usuarios
- *Stats*: Este módulo se encargaría de la estadística de un usuario y los posts más populares
 - Cantidad de usuarios que siguen a un usuario
 - Cantidad de usuarios que sigue el usuario
 - Cantidad de posts que ha realizado el usuario
 - El listado de los posts más populares

5). Definición de la estructura de la plantilla. Para la definición de la plantilla y su estructura se tomó como base los diseños realizados por el módulo de *Diseño y desarrollo de usabilidad para la interacción Humano-Computador*, en donde se realizó un análisis sobre las dependencias y funcionamiento de las vistas.

Para esto se hizo una investigación sobre la plantilla que se podría utilizar para desarrollar las vistas, que permitiera el desarrollo de la estructura necesaria y que soportara el *framework* de *frontend* escogido durante la etapa de análisis de requerimientos no funcionales (Angular JS). Se buscaron varios *templates* y se encontraron los siguientes:

- Monarch
- Triangular
- Avenxo
- Metronic

Luego de investigar los posibles templates se realizó una comparación de ellos en donde se utilizaron los siguientes criterios:

- Soporte: Tenga o provea un buen soporte por cualquier tipo de problema en la implementación del mismo.
- Revisiones: Que tenga buenas críticas por parte de los usuarios respecto a su uso con AngularJS
- Mantenimiento: Que tenga una bitácora de cambios realizados y que no haya sido discontinuado
- Estructura: Que tenga una buena maquetación y estructura a nivel de HTML y que permita su uso de forma dinámica y sencilla.
- Adaptación: Que se adapte a lo que se necesita, que no es una página web o un panel de administración sino una red social.

Tabla 26. Comparación sobre plantillas para desarrollo de vistas

Tecnología	Soporte	Revisiones	Mantenimiento	Estructura	Adaptación	Ventas
Monarch	Soporte por 6 meses	3.5 / 5	Si cumple. Última actualización el 01/10/2017	Si cumple	No se adapta	173
Triangular	Soporte por 6 meses	4.82 / 5	Si cumple. Última actualización el 03/05/2017	Si cumple	Si se adapta	2,243
Avenxo	Soporte por 6 meses	4.17 / 5	No cumple. Última actualización el 16/06/2015	Si cumple	No se adapta	569
Metronic	Soporte por 6 meses	4.89 / 5	Si cumple. Última actualización el 06/02/2017	Si cumple	Si se adapta	56,400

Se escogió Metronic por ser un template muy dinámico, con buen soporte técnico por parte de Keenthemes y porque cuenta con la mayor cantidad de ventas, sus revisiones han sido buenas lo cual le da un peso de calidad y porque se adapta a las necesidades del proyecto.

Ya con la plantilla se procedió a estudiar su estructura más a fondo y determinar cómo adaptarla a las necesidades del proyecto.

Figura 132. Plantilla con estructura base utilizando Metronic

```

1 <body class="page-container-bg-solid">
2   <div class="page-wrapper">
3     <div class="page-wrapper-row">
4       <div class="page-wrapper-top">
5         <div class="page-header">
6           <div class="page-header-top">
7             <div class="container">
8               <div class="page-logo">
9                 </div>
10              <div class="top-menu">
11                </div>
12              </div>
13            </div>
14          </div>
15        </div>
16      </div>
17      <div class="page-wrapper-row full-height">
18        <div class="page-wrapper-middle">
19          <div class="page-container">
20            <div class="page-content-wrapper">
21              <div class="page-head">
22                <div class="container">
23                  Toda la información del encabezado
24                </div>
25              </div>
26              <div class="page-content">
27                <div class="container">
28                  Todo la información del cuerpo
29                </div>
30              </div>
31            </div>
32          </div>
33        </div>
34      </div>
35      <div class="page-wrapper-row">
36        <div class="page-wrapper-bottom">
37          <div class="page-footer">
38            <div class="container">
39              2017 &copy; GobHash - UVG
40            </div>
41          </div>
42          <div class="scroll-to-top">
43            <i class="icon-arrow-up"></i>
44          </div>
45        </div>
46      </div>
47    </div>
48  </body>

```

Figura 133. Plantilla con estructura base para un post utilizando Metronic

```

1 <div class="portlet light portlet-fit">
2   <div class="portlet-title">
3     <div class="caption">
4       <span class="caption-subject font-dark sbold uppercase">
5         Título de publicación
6       </span>
7     </div>
8   </div>
9   <div class="portlet-body">
10    <div class="row">
11      <div class="col-md-8">
12        Gráfico
13      </div>
14      <div class="col-md-4">
15        <span align="justify" class="caption-subject font-dark sbold">
16          Descripción de la publicación
17        </span>
18      </div>
19    </div>
20    <div class="row">
21      <div class="col-md-8">
22        <div class="col-md-6 text-center">
23          <label>
24            <i class="fa fa-comment fa-lg" aria-hidden="true">
25              Comentarios
26            </i>
27          </label>
28        </div>
29        <div class="col-md-6 text-center">
30          <label>
31            <i class="fa fa-hashtag fa-lg" aria-hidden="true">
32              Número de hashes ("Me gusta")
33            </i>
34          </label>
35        </div>
36      </div>
37      <div class="col-md-4">
38        <a href="/feed" class="btn btn-menu pull-right">Ver más</a>
39      </div>
40    </div>
41  </div>
42 </div>

```

Con esta estructura se logró una plantilla genérica sobre Metronic con el fin de poder desarrollar sobre ella el prototipo y la aplicación de AngularJS.

a) Diseño para el cumplimiento de los requisitos no funcionales. La metodología de desarrollo a utilizar será *CI&CD*, debido a que permite el control de versiones y también entorno de pruebas y desarrollo automatizado. Para esto se utilizó *GitHub*, *Heroku*, *NPM*, *NodeJS*.

Primero se creó el repositorio en *GitHub*⁶ y luego se levantó y configuró una instancia de *Heroku*⁷.

⁶ <https://github.com/GobHash/GobHash>

⁷ <https://gobhashv1.herokuapp.com/>

6) Infraestructura del entorno de desarrollo del proyecto. A fin de poder tener una buena integración con *Heroku* se decidió utilizar NPM, que es una tecnología que facilita el desarrollo para JS, ya que cuenta con un manejador de paquetes JS, también permite compartir y reutilizar código.

Por otro lado, permite el uso de *ExpressJS* que es una herramienta que permite una infraestructura web rápida, minimalista y flexible para NodeJS.

7) Dependencias y configuración de infraestructura web del proyecto. Para iniciar con el desarrollo del proyecto antes fue necesario definir la versión a utilizar de *AngularJS*, las librerías necesarias y configuraciones necesarias.

Se decidió utilizar *AngularJS* v1.6.0 porque es una de las versiones más estables y porque el *template Metronic* cuenta con soporte para esa versión.

Las librerías definidas como dependencias fueron las siguientes:

- Angular: v1.6.0
- Angular animate: v1.6.0
- Angular blockUI: v0.2.2
- Angular chartJs: v1.1.1
- Angular cookies: v1.6.0
- Angular flash alert: v2.5.0
- Angular modal service: v0.12.1
- Angular sanitize: v1.6.0
- Angular touch: v1.6.0
- Angular ui bootstrap: v1.3.3
- Angular ui router: v1.0.3
- Bootstrap switch: v3.3.4
- ChartJs: v2.6.0
- Express: v4.16.2
- jQuery: v3.2.1
- Ng file upload: v12.2.13
- Oclazyload: v1.0.9
- Serve favicon: v2.4.5
- Socket.io: v2.0.3
- Ui cropper: v1.0.6
- WidgetsAngular: github:GobHash/Widgets-Angular

Como podemos ver una de las ventajas de utilizar NPM fue el uso de repositorios de GitHub como dependencias, como dependencia principal se utilizó el módulo de *Desarrollo e implementación de indicadores estadísticos (Gráficos/Widgets)* con base en la información extraída del portal de *GuateCompras*.

Una vez definidas las dependencias y sus versiones se creó el archivo de configuración de NPM, el cual fue el siguiente, para efectos de claridad en el documento se dividió la configuración en tres partes:

- Configuración del proyecto
- Archivo de configuración para servir aplicación
- Dependencias y versiones

Figura 134. Configuración inicial del proyecto de Frontend

```

1 {
2   .."name":.."gobhash",
3   .."version":.."1.0.0",
4   .."description":.."Megaproyecto UVG 2017",
5   .."main":.."index.js",
6   .."engines":..{
7     ..."node":..">=4.8.0",
8     ..."npm":..">=2.15.11"
9   },
10  .."scripts":..{
11    ..."start":.."node server.js"
12  },
13  .."repository":..{
14    ..."type":.."git",
15    ..."url":.."git+https://github.com/GobHash/GobHash.git"
16  },
17  .."keywords":..[
18    ..."AngularJS",
19    ..."OpenData",
20    ..."UVG"
21  ],
22  .."author":.."Adolfo Morales",
23  .."license":.."MIT",
24  .."bugs":..{
25    ..."url":.."https://github.com/GobHash/GobHash/issues"
26  },
27  .."homepage":.."https://github.com/GobHash/GobHash#readme",
28  .."devDependencies":..{
29    ..."angular-http-server":.."^1.3.1"
30  },

```

Figura 135. Configuración de *ExpressJS* y *NodeJs*

```

1 var express = require('express');
2 var favicon = require('serve-favicon')
3 var path = require('path')
4
5 var app = express();
6
7 app.use(express.static(__dirname));
8 app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')))
9
10 app.get('/', function (req, res) {
11   ...res.redirect('/');
12 });
13
14 // use port 8080 unless there exists a preconfigured port
15 var port = process.env.PORT || 8080;
16
17 app.listen(port);

```

Figura 136. Dependencias y versiones del proyecto de Frontend

```

31  ··· "dependencies": {
32  ···   "angular": "^1.6.0",
33  ···   "angular-animate": "^1.6.0",
34  ···   "angular-block-ui": "^0.2.2",
35  ···   "angular-chart.js": "^1.1.1",
36  ···   "angular-cookies": "^1.6.0",
37  ···   "angular-flash-alert": "^2.5.0",
38  ···   "angular-modal-service": "^0.12.1",
39  ···   "angular-route": "^1.6.0",
40  ···   "angular-sanitize": "^1.6.0",
41  ···   "angular-touch": "^1.6.0",
42  ···   "angular-ui-bootstrap": "^1.3.3",
43  ···   "angular-ui-router": "^1.0.3",
44  ···   "block-ui": "^2.70.1",
45  ···   "bootstrap": "^3.3.7",
46  ···   "bootstrap-switch": "^3.3.4",
47  ···   "chart.js": "^2.6.0",
48  ···   "express": "^4.16.2",
49  ···   "jquery": "^3.2.1",
50  ···   "ng-file-upload": "^12.2.13",
51  ···   "oclazyload": "^1.0.9",
52  ···   "serve-favicon": "^2.4.5",
53  ···   "socket.io": "^2.0.3",
54  ···   "ui-cropper": "^1.0.6",
55  ···   "widgets-angular": "github:GobHash/Widgets-Angular"
56  ··· }

```

8) Buenas prácticas de AngularJS. Una vez definida las versiones de las tecnologías a utilizar, se hizo énfasis en la calidad del código, por lo cual se buscó un manual de buenas prácticas para *AngularJS*, para esto se realizó una investigación y se encontró no solo una buena documentación con conceptos, casos de uso, ejemplos sino también la forma correcta de utilizar cada componente de *AngularJS*, su sintaxis y su correcta implementación.

La documentación encontrada es de John Papa, un desarrollador de *AngularJS* muy conocido y que se ha dedicado a explotar la tecnología y ayudar a las personas con buenas prácticas. Las buenas prácticas se pueden encontrar en <https://github.com/johnpapa/angular-styleguide/blob/master/a1/README.md>.

Se realizó la lectura completa del documento para ser implementadas en el proyecto de GobHash.

9) Definición de la estructura del proyecto. Luego de la lectura de buenas prácticas y cómo debe ser la estructura de un proyecto. La estructura por utilizar es la siguiente:

Primero se decidió dividir el proyecto en 2 ramas principales *html* y *js*, en donde *html* contiene todas las plantillas para el despliegue y solicitud de información y *js* contiene toda la programación necesaria para que funcione gobhash, cabe mencionar que dentro de cada una de estas ramas se planteó una subestructura:

- js
 - Controllers
 - Feed
 - Home
 - Post
 - User
 - Directives
 - PasswordChange
 - Profile
 - ShowPost
 - Services
 - Auth
 - Feed
 - Post
 - User
 - Utils
- html
 - Feed
 - Home
 - Post
 - User
 - assets

10 Definición de patrón de diseño a utilizar. Se decidió utilizar como principal patrón de diseño a Single Responsibility, ya que establece que cada módulo o clases debe tener una única responsabilidad sobre una parte de la funcionalidad. Lo que implica que una clase solo puede tener una función y que solo puede tener una razón para poder cambiar su comportamiento.

3. PROTOTIPADO. Una vez definida la plantilla y los requisitos funcionales se desarrolló un prototipo a modo de tener una prueba de concepto sobre lo que sería GobHash.

Para el prototipo se utilizó la tecnología *Silex* que es una herramienta de PHP que permite hacer enrutamiento de HTML estático con rutas estáticas.

Para esto se utilizó la estructura genérica desarrollada en la definición de estructura de plantilla, sobre la estructura se utilizó TWIG que nos serviría para simular el *rendering* por rutas de *AngularJS*.

A modo de tener un control de versiones sobre los cambios y distintas versiones del prototipo se utilizó un repositorio de GitHub⁸, por otro lado, para poder visualizar los

⁸ <https://github.com/GobHash/GobHashDemo>

cambios en tiempo real según los *commits* hechos al repositorio se utilizó *Heroku*, ya que permite *CI&CD*. El link para poder ver el demo es el siguiente⁹.

Cabe mencionar que sobre esto se realizó una prueba piloto, misma que fue llevada a cabo por el módulo *Diseño y desarrollo de usabilidad para la interacción Humano-Computador*, luego de haber realizado las pruebas se dieron una serie de observaciones y cambios a realizar, que luego fueron implementados para poder repetir el proceso hasta llegar a cómo se vería la red social.

B. DESARROLLO

Una vez completada la etapa de prototipado, en donde la prueba de concepto sobre las funcionalidades de GobHash fueron aprobadas se inició con el desarrollo.

Durante la etapa de análisis se hizo un estudio sobre las librerías, tecnología y versión de *AngularJS* a utilizar, así mismo se definió la estructura del proyecto y se creó el entorno de desarrollo y producción a utilizar.

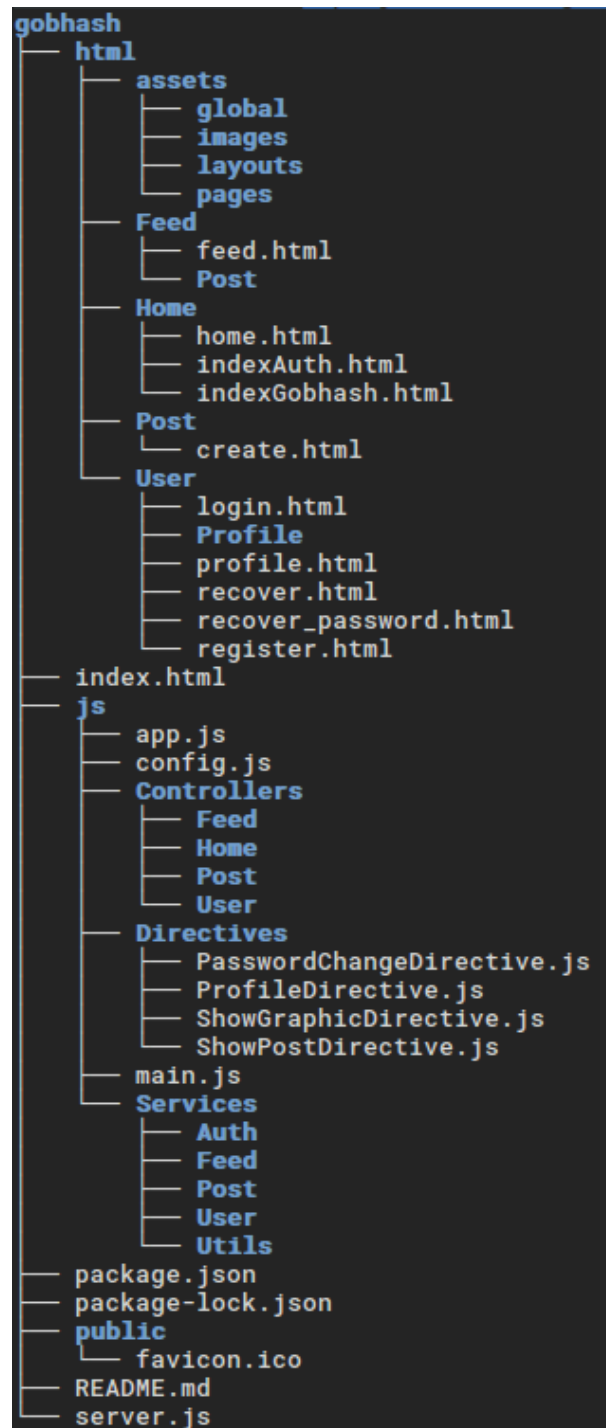
Como primer paso lo que se hizo fue implementar la estructura diseñada. Sin embargo, se tomó en cuenta que se trabajaría con el módulo de *Desarrollo e implementación de indicadores estadísticos (Gráficos/Widgets) con base en la información extraída del portal de GuateCompras*. Entonces se decidió utilizar una segunda estructura sobre la definida y esta es una estructura modular.

Para esto no se necesitó mucho ya que *AngularJS* permite utilizar otras aplicaciones como librerías, entonces lo único que se hizo fue crear una aplicación padre que importara GobHash y Widgets.

Una vez la estructura desarrollada se crearon todos los estados que se utilizarían para representar cada uno de los requisitos funcionales. Luego se adaptaron las plantillas utilizadas en el prototipado a fin de poder ser utilizadas con *AngularJS*.

Finalmente se desarrollaron los controladores, servicios y directivas, utilizando las buenas prácticas definidas durante el análisis.

⁹ <http://demo-gobhash.herokuapp.com/>

Figura 137. *Filesystem* del proyecto GobHash utilizado

C. RESULTADOS

Los resultados obtenidos fueron los siguientes:

- Control de versiones sobre el proyecto
 - Prototipo: <https://github.com/GobHash/GobHashDemo>
 - GobHash: <https://github.com/GobHash/GobHash>
- Un entorno de desarrollo y producción automatizado
 - Prototipo:

- <http://demo-gobhash.herokuapp.com/>
 - GobHash
 - Desarrollo: <https://gobhashv1.herokuapp.com>
 - Producción: <http://gobhash.herokuapp.com> o
<https://www.gobhash.com>
- Código fuente de alta calidad según los estándares de buenas prácticas encontradas y enfocado a escalabilidad
- Prototipo
- Red social para análisis descriptivo sobre la información de GuateCompras

Figura 138. Repositorio para control de versiones de prototipo den Github

This screenshot shows the GitHub interface for the repository 'GobHash / GobHashDemo'. At the top, there are navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are statistics: 3 Unwatch, 0 Star, and 1 Fork. The main content area shows the commit history for the 'master' branch. The latest commit is by user 'fitocs452' with the message 'Create README.md' and a commit hash of 'a44e933', made 13 minutes ago. The commit history table lists the following files and their commit messages:

File	Commit Message	Time Ago
.idea	Welcome page	2 months ago
assets	Se crearon los nuevos formularios de recuperar contraseña y registro...	a month ago
views	Se crearon los nuevos formularios de recuperar contraseña y registro...	a month ago
.gitignore	Welcome page	2 months ago
.htaccess	Preparing for heroku	a month ago
Application.php	Add files to repo	2 months ago
README.md	Create README.md	13 minutes ago
composer.json	Preparing for heroku	a month ago
composer.lock	Preparing for heroku	a month ago
index.php	Add files to repo	2 months ago

Figura 139. Repositorio para control de versiones de GobHash en Github

This screenshot shows the GitHub interface for the repository 'GobHash / GobHash'. At the top, there are navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are statistics: 1 Unwatch, 0 Star, and 0 Fork. The main content area shows the commit history for the 'master' branch. The latest commit is by user 'fitocs452' with the message 'Add package-lock to avoid version problems' and a commit hash of '3e4586b', made 3 days ago. The commit history table lists the following files and their commit messages:

File	Commit Message	Time Ago
html	Like post is working! but not as expected	3 days ago
js	Remove js/WidgetModule use node_modules/widgets-angular instead	3 days ago
public	Add npm start and favicon	11 days ago
.gitignore	Add package-lock to avoid version problems	3 days ago
README.md	Initial commit	a month ago
index.html	Remove js/WidgetModule use node_modules/widgets-angular instead	3 days ago
package-lock.json	Add package-lock to avoid version problems	3 days ago
package.json	Add WidgetAngular as dependency not as git submodule	3 days ago
server.js	Add npm start and favicon	11 days ago

Figura 140- Servidor automático para producción de GobHash

HEROKU

Personal apps > gobhash

GITHub [gadolformalesmartinez](#) [@wester](#)

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons **\$0.00/month** Configure Add-ons @

There are no add-ons for this app. You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation **\$0.00/month** Configure Dynos @

This app is using free dynos.

web: one start ON

Collaborator activity [Manage Access](#) @

[pablo.diaz@gobhash.com](#) 34 deploys

[gadolformalesmartinez@gmail.com](#) 6 deploys

Latest activity [All Activity](#) @

- [gadolformalesmartinez@gmail.com: Deployed \[3e15820a\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[358a61c4\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[358a61c4\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[f56ec276\]](#) 8 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 8 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[81127bd\]](#) 8 days ago - [View build log](#) - [Compare diff](#)

Figura 141. Servidor automático para desarrollo de GobHash

HEROKU

Personal apps > gobhashv1

GITHub [gadolformalesmartinez](#) [@wester](#)

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons **\$0.00/month** Configure Add-ons @

There are no add-ons for this app. You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation **\$0.00/month** Configure Dynos @

This app is using free dynos.

web: one start ON

Collaborator activity [Manage Access](#) @

[gadolformalesmartinez@gmail.com](#) 20 deploys

Latest activity [All Activity](#) @

- [gadolformalesmartinez@gmail.com: Deployed \[e45a3a2e\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[358a61c4\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[358a61c4\]](#) 3 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 3 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[f56ec276\]](#) 8 days ago - [View build log](#) - [Compare diff](#)
- [gadolformalesmartinez@gmail.com: Build succeeded](#) 8 days ago - [View build log](#)
- [gadolformalesmartinez@gmail.com: Deployed \[81127bd\]](#) 8 days ago - [View build log](#) - [Compare diff](#)

Figura 142. Prototipo: Página inicial

GOBHAS#

Plataforma de Analisis Estadístico de las Aplicaciones del Estado de Guatemala

GOBHAS# te permite interpretar datos como nunca antes!!

Inicio de Sesión

Usuario

Contraseña

[Iniciar Sesión](#)

[¿Olvidaste tu contraseña?](#)

[Haz click aquí](#)

[Regístrate ahora](#)

#MasPopulares

Los 5 proveedores con mas ganancia en el Ministerio de Educacion(2016) #Educacion #2016 [Ver Mas](#)

Ejecucion del presupuesto de MINFIN 2016 #Finanzas #2016 [Ver Mas](#)

¿Quieres un servicio personalizado?

[Comunicate con nosotros](#)

Figura 143. Prototipo: Inicio de sesión

Este prototipo de inicio de sesión presenta un fondo naranja con un título centralizado "Inicio de Sesión" en un color más oscuro. Debajo del título, se encuentran dos campos de entrada blancos con bordes naranjas, etiquetados como "Usuario" y "Contraseña". En la parte inferior del formulario, hay tres elementos de texto: un botón "Iniciar Sesión", un enlace "¿Olvidaste tu contraseña" con el texto "Haz click aquí" debajo, y un enlace "Regístrate ahora".

Figura 144. Prototipo: Registro de usuario

Este prototipo de registro de usuario tiene un fondo naranja y un título "Registro de usuario" en un color más oscuro. Incluye cuatro campos de entrada blancos con bordes naranjas, etiquetados como "Usuario", "Correo", "Contraseña" y "Confirme contraseña". En la parte inferior, se muestran dos elementos de texto: un botón "Regístrate ahora" y un enlace "Iniciar sesión".

Figura 145. Prototipo: Solicitar cambio de contraseña

Este prototipo de recuperación de contraseña presenta un fondo naranja y un título "Recuperar contraseña" en un color más oscuro. Cuenta con dos campos de entrada blancos con bordes naranjas, etiquetados como "Usuario" y "Correo". En la parte inferior, se encuentran dos elementos de texto: un botón "Recuperar contraseña" y un enlace "Iniciar sesión".

Figura 146. Prototipo: Página inicial luego de haber iniciado sesión

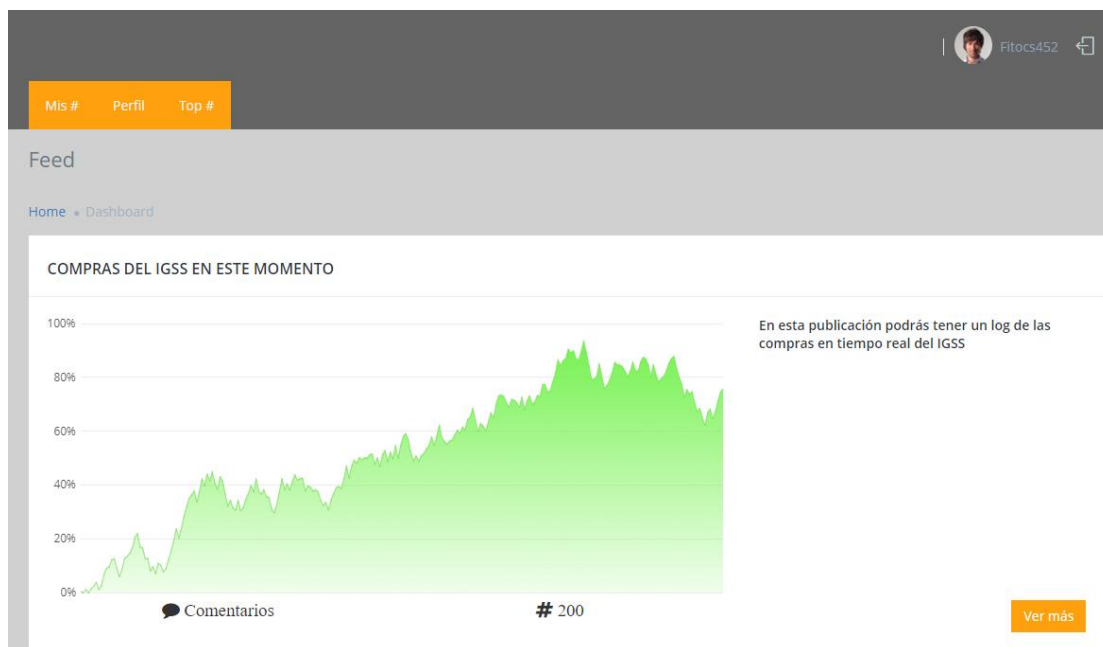


Figura 147. Prototipo: Perfil del usuario

Mis # Perfil Top #

Feed

Home • Dashboard

Fitocs452

FOLLOW

37 POSTS 51 LIKES

PERFIL Información personal Contraseña

Nombre de usuario
Fitocs452

Ocupación
Analista de datos

Acerca de ti
Soy un gran analista!

Guardar cambios Cancelar

Figura 148. Prototipo: Ver posts y agregar comentarios

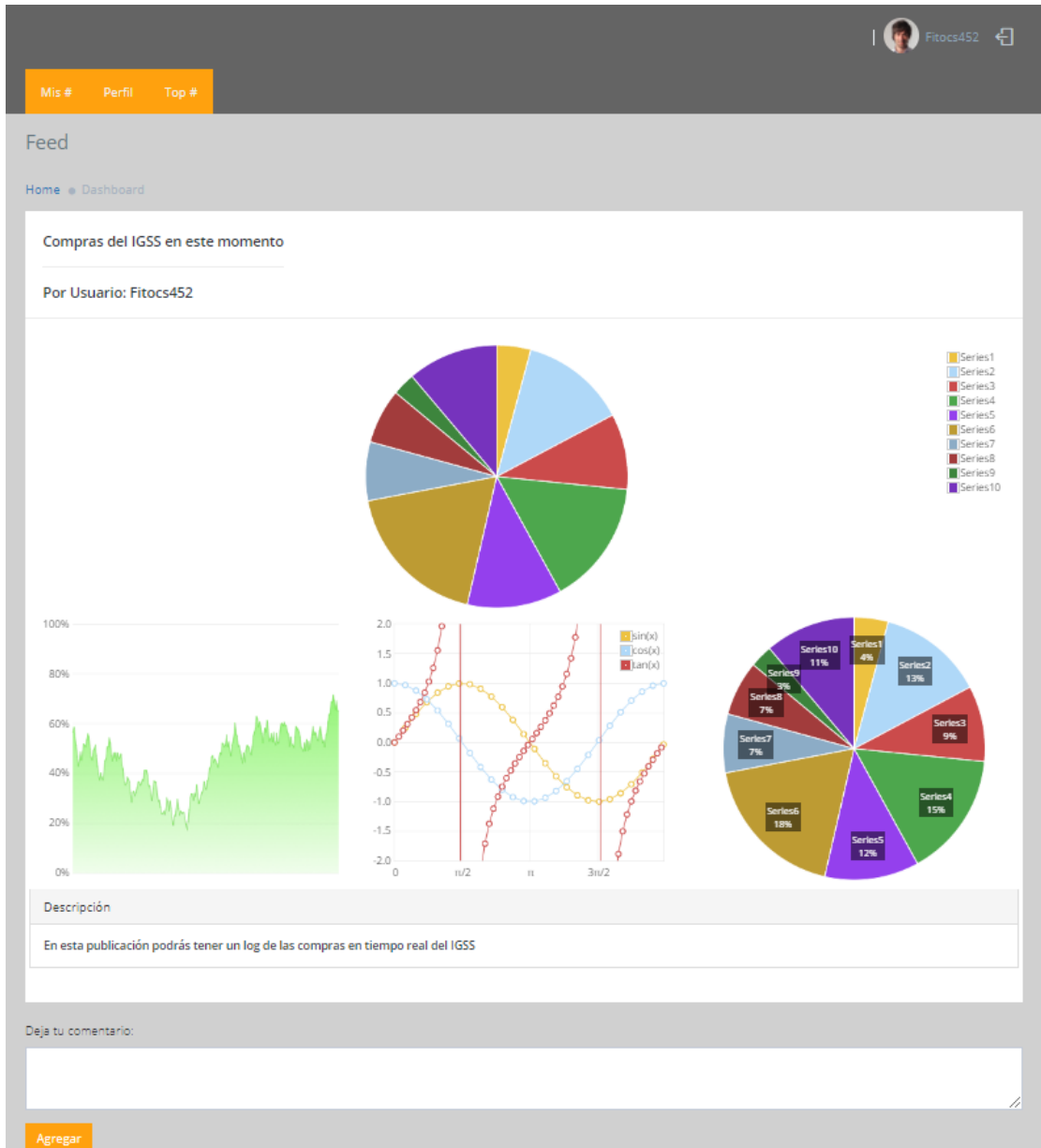
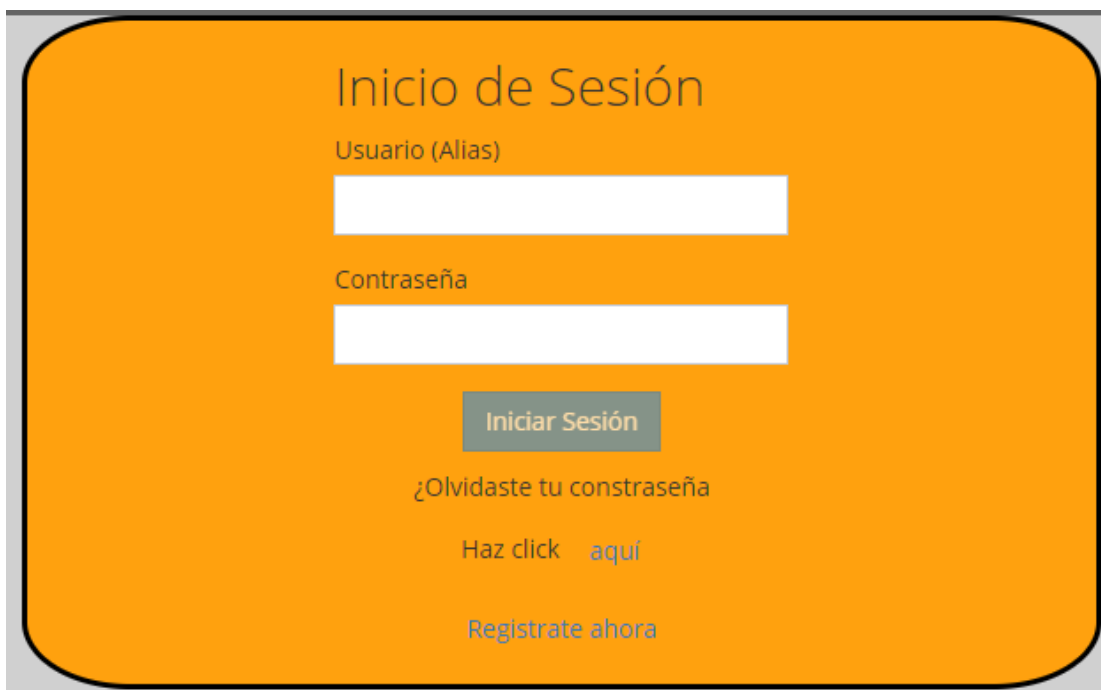


Figura 149. GobHash: Página inicial



Figura 150. GobHash: Inicio de sesión



Inicio de Sesión

Usuario (Alias)

Contraseña

Iniciar Sesión

¿Olvidaste tu contraseña

Haz click [aquí](#)

[Regístrate ahora](#)

Figura 151. GobHash: Registro de usuario



Registro

Email

Usuario (Alias)

Contraseña

Registrar Cancelar

Figura 152. GobHash: Solicitud de cambio de contraseña



Recupera tu contraseña

Usuario (Alias)

Email

Recuperar Cancelar

Figura 153. GobHash: Recuperación de contraseña mediante token



Recupera tu contraseña

Nueva contraseña

Escriba de nuevo la contraseña

Recuperar Cancelar

Figura 154. GobHash: Feed inicial

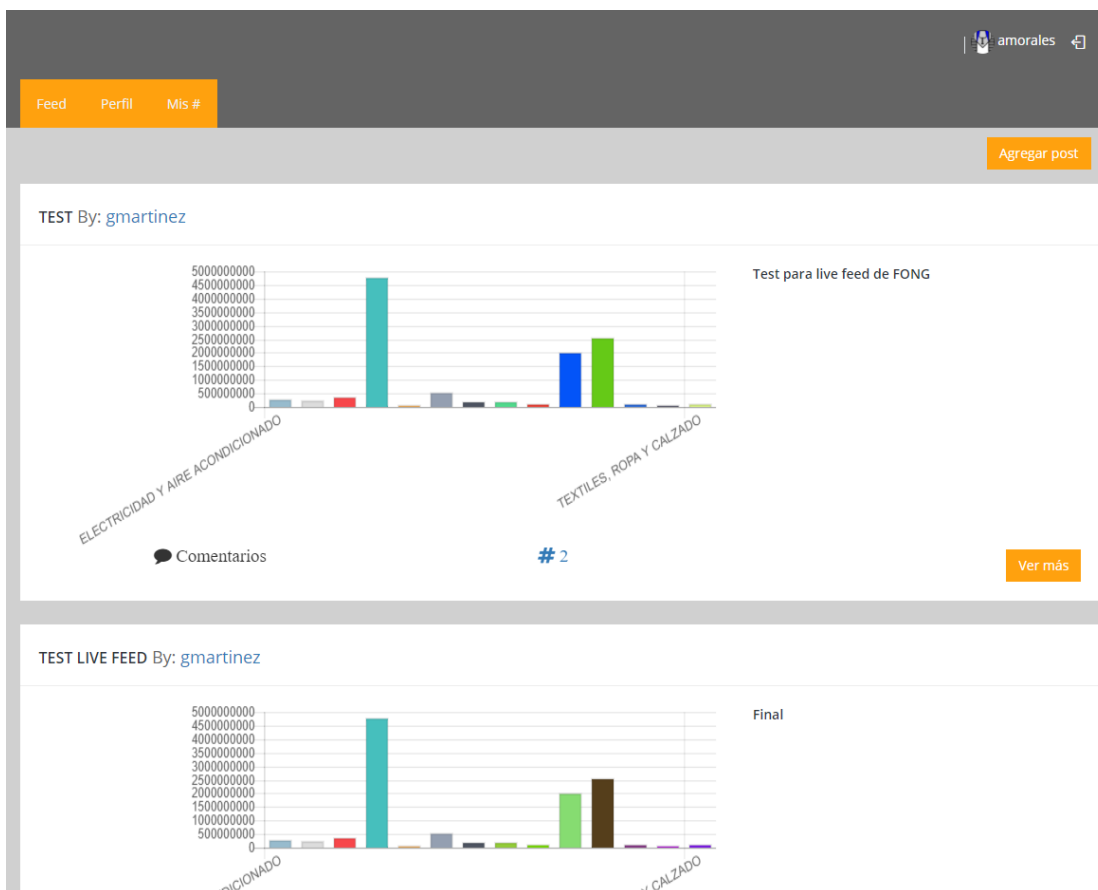


Figura 155. GobHash: Perfil del usuario

The screenshot shows the user profile for "amorales". The profile picture is a circular logo with a stylized figure. The "PERFIL" section includes the following information:

- Nombre de usuario:** amorales
- Ocupación:** programador
- Acerca de ti:** Escribiendo sobre mi x 2

At the bottom of the profile, the user's statistics are displayed: 12 POSTS, 4 FOLLOWERS, and 4 FOLLOWING.

Figura 156. GobHash: Cambio de foto de perfil

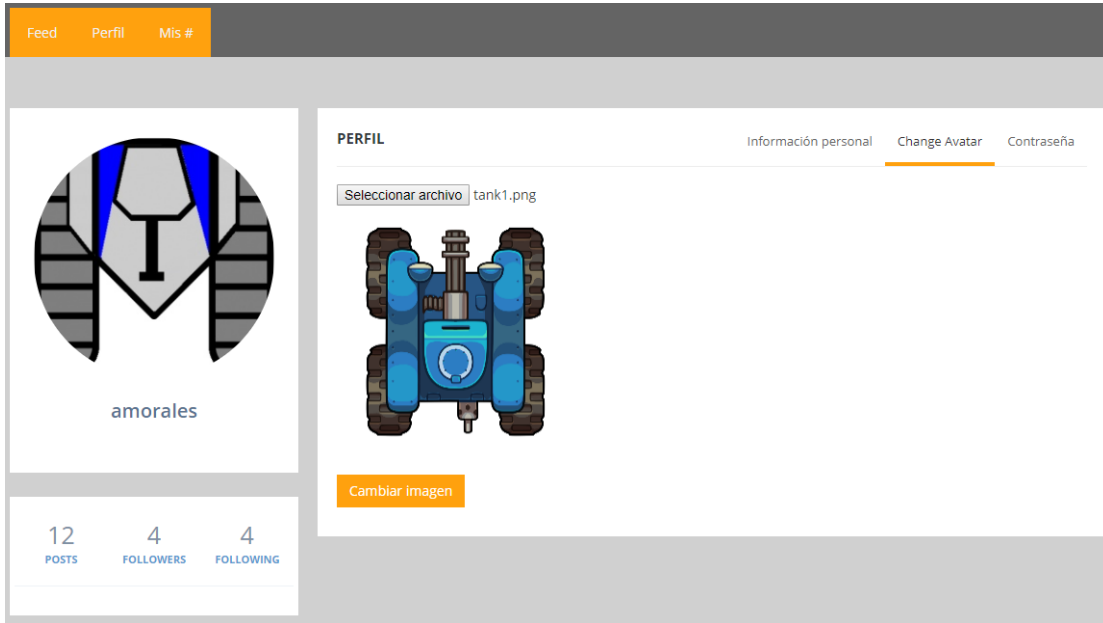


Figura 157. GobHash: Cambio de contraseña mediante plataforma

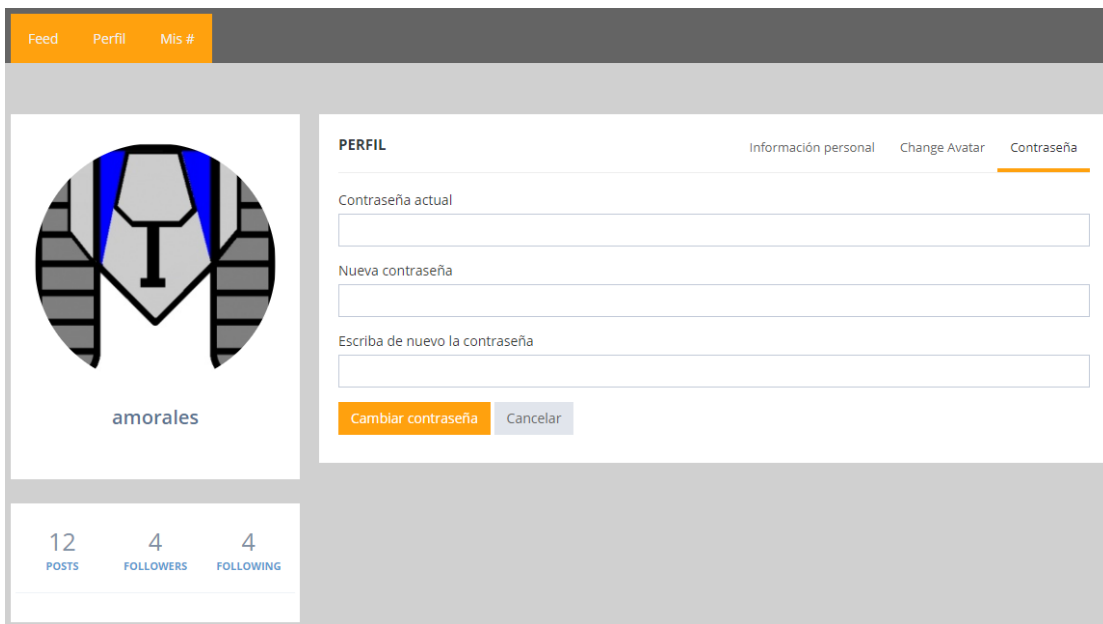


Figura 158. GobHash: Estadísticas generales del usuario

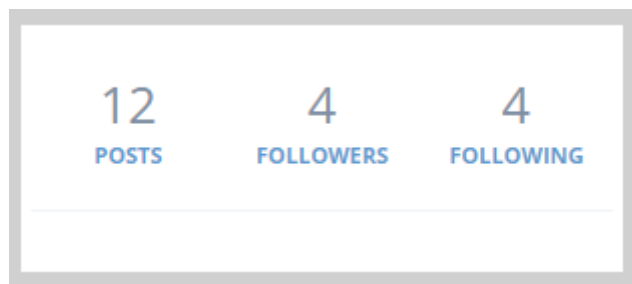


Figura 159. GobHash: Vista de las publicaciones creadas por el usuario actual

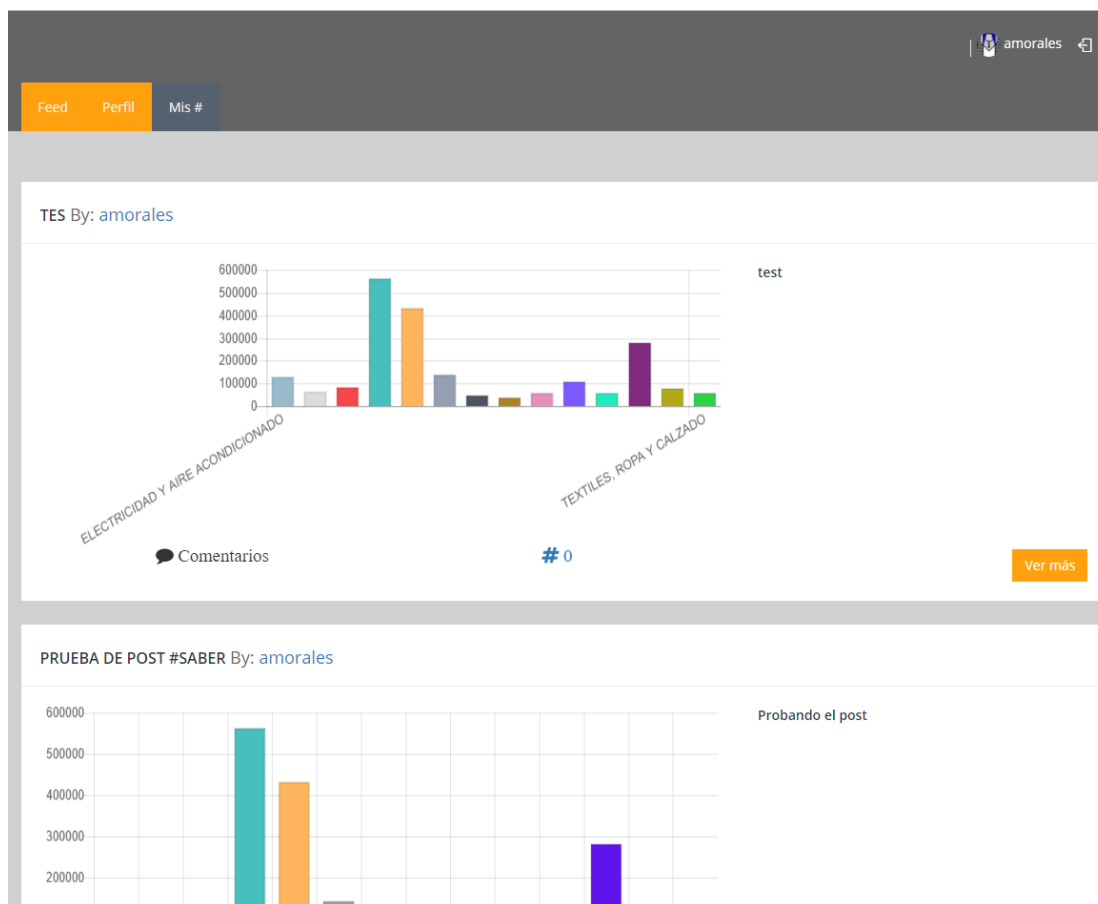


Figura 160. GobHash: Vista individual de una publicación

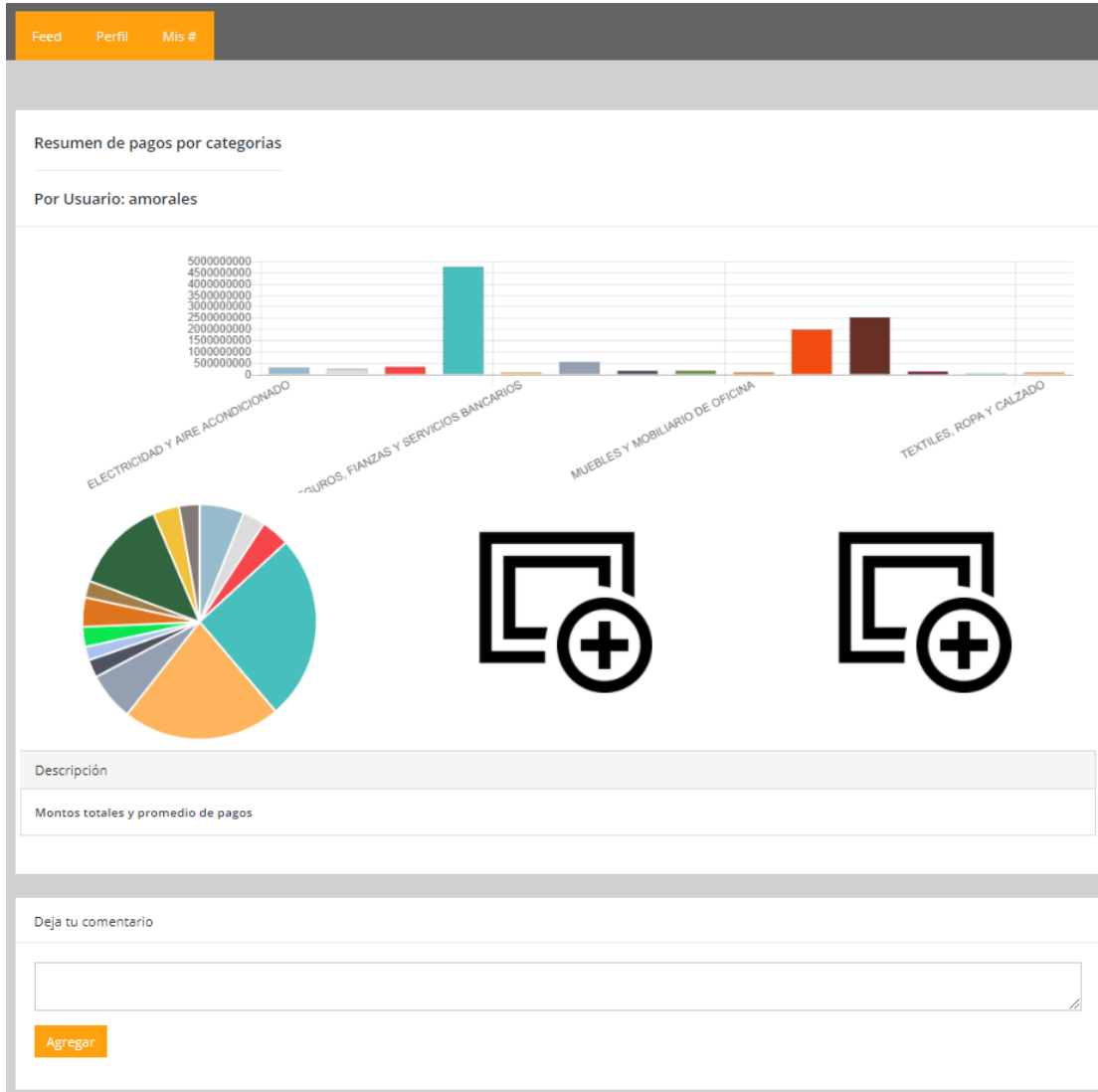


Figura 161. GobHash: Comentarios sobre una publicación

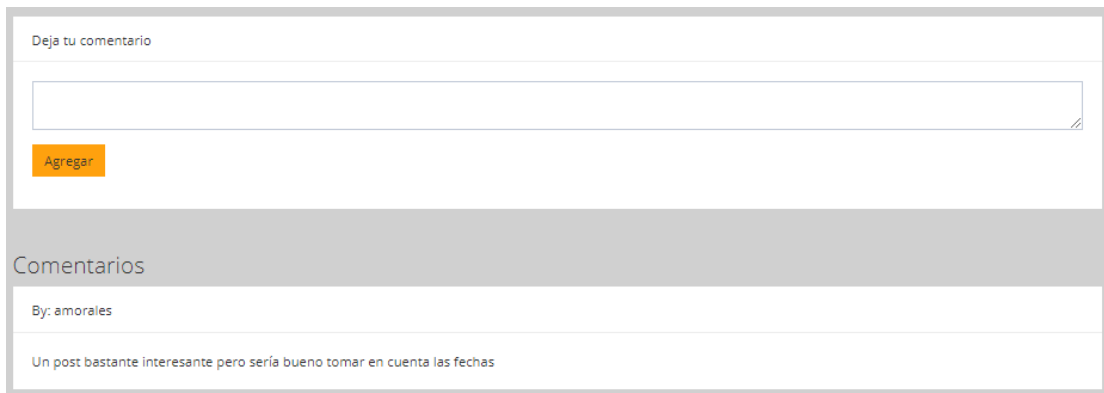


Figura 162. GobHash: Agregar una nueva publicación

Feed Perfil Mis #

amorales

Título y Descripción Crear

Título

Descripción

Gráfica Principal

Gráfica 1





Figura 163. Integración con el módulo desarrollo e implementación de indicadores estadísticos

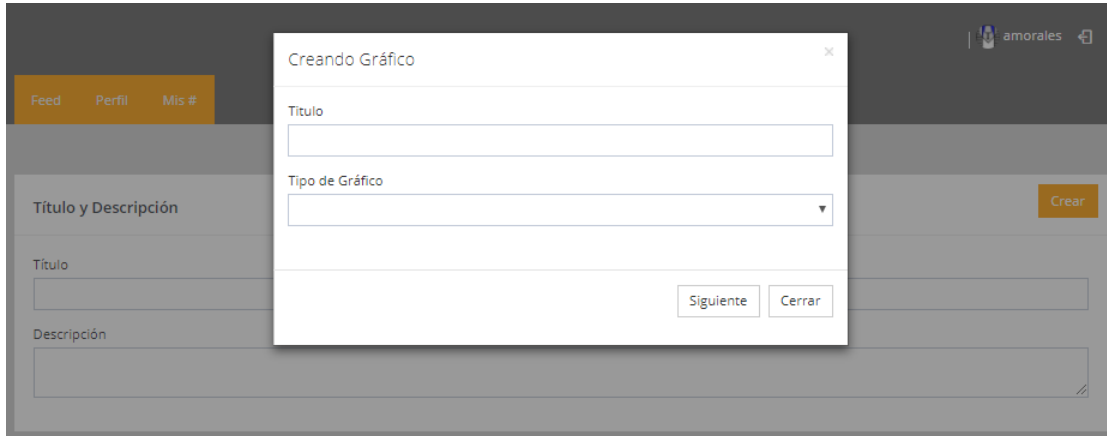
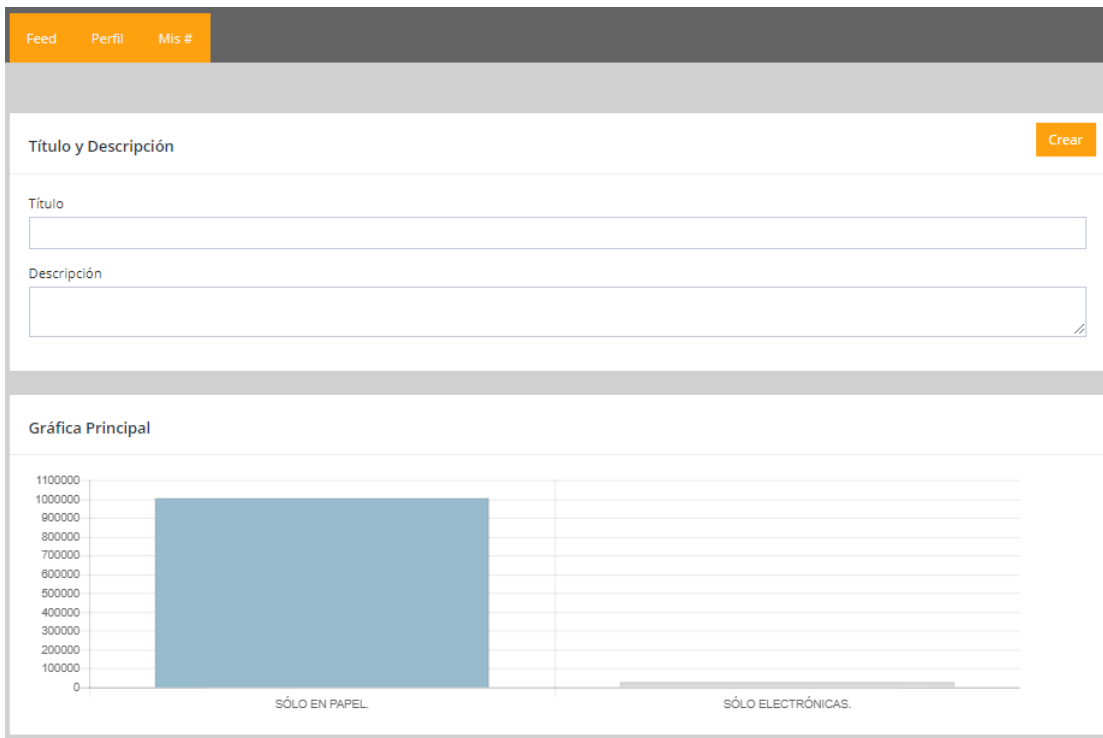


Figura 164. Integración de gráficos por el módulo desarrollo e implementación de indicadores.



D. DISCUSIÓN

1. Tecnologías para infraestructura. Las tecnologías utilizadas para la infraestructura fueron:

- *Heroku*
- *GitHub*
- *NPM*
- *NodeJS*
- *ExpressJS*

Heroku es una plataforma como servicio de computación en la nube que soporta muchos lenguajes de programación, lo cual es una ventaja. Se escogió esta tecnología dado que apoya al cumplimiento de algunos requisitos no funcionales como lo son la escalabilidad y la implementación de *CI&CD*.

Y esto se debe a que *Heroku* está basado en *Dynos* que son pequeños contenedores de cómputo basados en Linux, lo cual no solo permite alta capacidad de procesamiento sino crecimiento vertical y horizontal.

Las ventajas encontradas al utilizar *Heroku* fueron las siguientes:

- Elasticidad y crecimiento: Permite agregar más nodos a una aplicación y también más recursos a un nodo en específico en cualquier momento.
- Tamaño/Capacidad: Cuenta con varias versiones sobre capacidades a distintas tarifas, pero lo que es mejor es que cuenta con instancias gratis capaces de soportar GobHash, las capacidades de un *Dyno* gratuito es de 512MB de RAM, 1 CPU compartido.
- Enrutamiento
- Seguimiento: Monitoreo en vivo sobre los recursos utilizados por la instancia
- *CI&CD*: *Heroku* ya cuenta con esta integración

GitHub es una plataforma como servicio que permite el alojamiento de proyectos que permite el uso de un sistema de control de versiones. Se escogió esta tecnología debido a que tiene integración con muchas de las plataformas como servicios, por ejemplo, *Heroku* tiene integración al 100% con *GitHub* y de hecho se recomienda *GitHub* para poder controlar el *CI&CD* por medio *branches*.

NPM es un manejador de paquetes de JS, lo que permite reutilizar código y librerías que otras personas han hecho y aprovechar las bondades de lo que ofrecen. Se decidió utilizar esta tecnología porque cuenta con un control de dependencias a instalar en donde se le especifican las versiones que se desean utilizar, lo cual es una gran ventaja ya permite la mitigación de fallas en el desarrollo y despliegue por incongruencia en las versiones de las dependencias. Por otro lado, NPM permite especificar repositorios de *GitHub* como dependencias y eso fue una gran ventaja debido a que el módulo de *Desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de GuateCompras*, fue desarrollado como un proyecto independiente a fin de aprovechar la modularización que permite *AngularJS*.

Las ventajas encontradas al utilizar *Heroku* fueron las siguientes:

- Integración con *NodeJS*
- Control de versiones de dependencias
- Integración *GitHub* como dependencias
- Integración con *GitHub* como alojamiento de proyecto
- Permite la ejecución de *scripts*: Esto se utilizó para poder iniciar el servidor con *NodeJS*

NodeJS es un entorno de ejecución multiplataforma de código abierto. En este proyecto se utilizó como un entorno de JS del lado del servidor. NodeJS ejecuta JS utilizando el motor V8, mismo que fue desarrollado por Google como principal motor del navegador Google Chrome. Utilizar el motor V8 de Google permite a NodeJS compilar y ejecutar código de JS con un alto rendimiento. La razón por la que provee un alto rendimiento es que V8 no interpreta el código JS y luego lo ejecuta sino primero lo compila en código de máquina nativo y luego ejecuta.

Las ventajas encontradas al utilizar NodeJS fueron las siguientes:

- Concurrencia: Esto soporta la escalabilidad
- V8: Compila código de JS en código de máquina para evitar interpretarlo en tiempo real.

ExpressJS es una *framework* de aplicaciones web para *NodeJS*, tiene como objetivo principal el construir aplicaciones web y *APIs*. Express ha sido conocido como el *framework* estándar para servidores basados en *NodeJS*.

2. Tecnología para *frontend*

- *AngularJS*
- *Metronic*

Se decidió utilizar *AngularJS* debido a que ayuda a cumplir los requisitos no funcionales establecidos durante la etapa de análisis:

- Seguridad informática
- Escalabilidad
- Compatibilidad
- Mantenimiento
- Comunidad
- Soporte
- Curva de aprendizaje
- Control de versiones

Durante la etapa de desarrollo se encontraron algunas bondades de *AngularJS* que no se tomaron en cuenta y es su alta capacidad para reutilizar código abierto, permitió la integración con el *Backend* sin problema alguno y permite programación funcional lo cual ayuda mucho a crear funciones primitivas, las cuales se encargan de tomar responsabilidad sobre pequeñas funcionalidades.

Se decidió utilizar *Metronic* debido a que cumple con los criterios utilizados durante la etapa de análisis. El uso de esta tecnología permitió generar una plantilla totalmente genérica, lo cual permite realizar modificaciones sin preocuparse de quebrar alguna

funcionalidad. Por otra parte, cuenta con una alta integración con *AngularJS*, lo cual fue una gran ventaja durante la etapa de desarrollo. Cabe mencionar, que durante la etapa de integración con el módulo *Desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de GuateCompras*, no hubo problema alguno respecto a compatibilidad de HTML y CSS.

3. Uso de *Single Responsibility* como patrón de diseño. Lo que este principio obliga a hacer es que cada clase en nuestro caso controlador o directiva tenga una única responsabilidad, por lo que si un controlador tiene dos responsabilidades es mejor realizar dos controladores, por ejemplo, es por esto que se dividió *MyFeed* y *Feed* en dos controladores diferentes, debido a que *MyFeed* es el encargado de mostrarle al usuario actual todas las publicaciones que ha hecho, por otro lado, *Feed* muestra todas las publicaciones de los usuarios a los que sigue.

Al final lo que se logra conseguir utilizando este patrón es una mayor mantenibilidad y eso nos ayuda a cumplir el requisito no funcional de escalabilidad, compatibilidad y mantenimiento.

4. Controladores, servicios y directivas utilizadas. La estructura definida durante la etapa de análisis tenía como propósito no solo apoyar en el cumplimiento de *Single Responsibility* sino también desarrollar código que cumpla con los estándares de calidad definidos por las buenas prácticas de *AngularJS*.

Al iniciar el proyecto se decidió registrar un proyecto llamado *Main*, y esto se debe a que de alguna manera necesitábamos poder soportar la modularización planteada. El uso de este proyecto permitió el uso del *GobHash* y *Widgets* como submódulos, y como consecuencia de esto fue fácil la integración de ambos módulos.

En cuanto a los controladores definidos, según la estructura propuesta. Nos podremos dar cuenta que se separó no solo según responsabilidades sino también por conceptos, esto se hizo de esta manera ya que *AngularJS* utiliza MVC lo cual es un patrón de diseño de desarrollo comúnmente utilizado para separar la responsabilidad por capas. Es por ello que los controladores no pueden realizar llamadas al *backend*, debido a que solo deberían manejar información y ejecutar lógica de negocio, y aquí es en donde cobra sentido el uso de servicios, los servicios tienen como única responsabilidad la interacción con el *Backend*, lo cual permitió agregar una capa de abstracción que nos ayuda con la escalabilidad. Para los servicios se utilizó programación funcional. Una vez un servicio realiza una petición al *Backend* tiene 2 opciones, petición exitosa lo cual regresa información y petición fallida lo cual regresa un error, y todo este manejo de posibles respuestas también se manejó en los servicios. Sin embargo, al recibir nosotros una respuesta a una petición realizada y luego manejar el tipo de respuesta, el servicio no puede hacer nada con esa información más que

devolverla de forma asíncrona. Es aquí en donde entra la programación funcional, se utilizó el concepto de *Callback*, lo cual permitió el poder ejecutar acciones sobre la información una vez esta haya sido recibida y procesada por medio del servicio.

Las directivas se utilizaron para poder cumplir con el patrón de *Single Responsibility*, y esto se debe a que *AngularJS* nos permite el manejo de HTML dinámico, las directivas se utilizaron para dos cosas en específico:

- Perfil
 - Mostrar información
 - Cambio de foto de perfil
 - Cambio de contraseña
- Posts
 - Despliegue de vista preliminar de un post

Para perfil solo contamos con un controlador y se decidió no darle toda la responsabilidad sobre las cinco funcionalidades (mostrar estadísticas, mostrar foto, mostrar información, cambio de foto de perfil y cambio de contraseña). Dado que las primeras dos funcionalidades de perfil son puramente de consulta se decidió que podrían estar en el controlador, sin embargo, las otras tres tiene manejo de información, vistas y peticiones al *Backend*, por lo que se dividieron en tres directivas.

En cuanto a la directiva del Post se consideró totalmente innecesario el colocar código repetido en el HTML del *Feed*, por esto se le delegó la responsabilidad de despliegue a una directiva.

XI. CONCLUSIONES

A. Diseño e implementación de modelo de datos de análisis descriptivo y desarrollo de modelo de red social básico

1. La técnica de modelación dimensional diseñado conserva dentro de la estructura del modelo el proceso de negocio de adjudicación de concursos por parte del estado de Guatemala.
2. El proceso de validación de datos no puede ser efectuado por una computadora, ya que es necesario hacer inferencias sobre la información y tomar decisiones.
3. El modelo desarrollado en MongoDB cumplió los requisitos de almacenamiento de datos de la plataforma GobHash.
4. Durante el desarrollo del proyecto se pudieron poner en práctica conceptos y habilidades adquiridos a lo largo de la carrera.

B. Desarrollo e implementación de indicadores estadísticos representados a través de gráficos

1. Se creó una interfaz dentro de la plataforma que permite generar gráficos para hacer análisis.
2. La interfaz de usuario para crear gráficas, al solo incluir tres pasos, se vuelve un proceso simple, y que los usuarios pueden rehacer varias veces.
3. Para que el usuario pueda sacar el mayor provecho de los datos recopilados del portal de Guatecompras es necesario que puede crear más de tres diferentes tipos de gráficos
4. Es necesario que los usuarios que vayan a crear gráficas tengas un conocimiento básico sobre los datos que se manejan en Guatecompras
5. Las gráficas no bastan como representación de indicadores estadísticos, el uso de tablas o alguna otra representación de un indicador estadístico en ocasiones es una mejor forma de representar los datos.
6. La elección de AngularJS como framework, fue acertada por el hecho de que la integración con el módulo de "Diseño, Análisis y Desarrollo de Frontend" fue exitosa y sencilla de ejecutar.
7. Se logró hacer una integración exitosa con los módulos de la plataforma "Diseño, Análisis y Desarrollo de Frontend" y "Desarrollo e Implementación de Backend".
8. El usuario puede crear tres distintos tipos de gráficas.

C. Extracción de datos de adjudicaciones, proveedores y compradores del sistema Guatecompras

1. El programa generado es capaz de obtener la información de proveedores, compradores y adjudicaciones, contenidos en Guatecompras, de manera automatizada para cualquier día, mes o año.
2. Tener tiempo de espera después de cada solicitud de información a Guatecompras no garantiza evitar ser bloqueado por parte del sitio.
3. Limpiar la información de la caché, de adjudicaciones, de manera periódica contribuye a disminuir el consumo de recursos del ambiente de ejecución.
4. Los reportes de Guatecompras son vitales para la validación de los datos obtenidos y de esta manera asegurar su fiabilidad.
5. El rendimiento del programa no es proporcional la cantidad de recursos disponibles en el ambiente de ejecución, a pesar de contar con 8 veces más memoria RAM y el doble de núcleos en el CPU, la diferencia en rendimiento con el Server no fue mayor al 30%.
6. La versión del algoritmo de conexión a Guatecompras sin tiempos de espera obligatorios tiene un mejor desempeño que la que sí los tiene, sin embargo, este no es proporcional a la cantidad de consultas que se hacen.
7. La elección de *Python* como lenguaje de programación fue acertada, la duración promedio en la extracción se vio afectada principalmente por la espera de una respuesta del servidor de Guatecompras.
8. La obtención de datos de manera diaria permite reducir la carga de trabajo y así como disminuir posibilidad de ser sujeto a bloqueos por parte del sitio, siendo este el mejor enfoque para la obtención de datos.

D. Diseño y desarrollo de estudio de usabilidad para la interacción humano-computador

1. Se llevó a cabo un estudio de estándar y usabilidad que ayudo a que el proyecto definiera aspectos claves para su desarrollo y correcto diseño, tales como la proposición de valor, el diagrama de flujo de aplicación, el grupo de usuarios objetivo, la definición de métricas y el análisis de competencia, entre otros. De esta manera el estudio de estándar y usabilidad satisface las expectativas respecto de su función y ayudo al correcto desarrollo del proyecto desde sus inicios.
2. Se desarrolló un prototipo en papel, orientado al mapeo, flujo de aplicación y diseño simple de la plataforma, este prototipo luego se sometió a un proceso de validación por medio del cual se generaron cambios que luego se implementaron en el prototipo diseñado. El prototipo en papel demostró ser útil por su simpleza y versatilidad.

3. Se desarrolló un prototipo diseñado haciendo uso de una herramienta de manejo de imágenes, este prototipo demostró una primera versión de cómo se vería el producto final e introdujo una paleta de colores que demostró cumplir con su objetivo, además de ser agradable al usuario. Luego de su validación el prototipo diseñado fue la referencia para la conclusión de un prototipo interactivo.
4. En conjunto con el módulo de Diseño, Análisis y Desarrollo de Frontón, se desarrolló un prototipo interactivo haciendo uso de una herramienta de maquetación llamada Silex, esto demostró ser de mucho valor para el desarrollo del proyecto ya que en vez de transferir un prototipo interactivo al módulo de Frontend y esperar que este lo replicara de la mejor manera posible, se desarrolló en conjunto el prototipo definitivo que tendría el mismo aspecto y comportamiento que la primera versión funcional del proyecto.
5. Se realizaron pruebas de usabilidad que cumplieron con los objetivos de cumplimiento de métricas, descubrimiento de errores de funcionamiento críticos y no críticos y retroalimentación sobre la primera versión funcional de la plataforma, estas pruebas de usabilidad descubrieron errores que fueron documentados e informados a las partes interesadas e identificaron cambios en diseño que fueron informados mediante una solicitud de cambios a los módulos respectivos.
6. Los resultados de las pruebas de usabilidad cumplieron con los requisitos de mejora definidos en el estudio de estándar y usabilidad respecto de las métricas definidas, relación de éxito, relación de error, tiempo de finalización y medidas subjetivas, satisfaciendo con éxito las expectativas del estudio de usabilidad en el proyecto GobHash.

E. Diseño e implementación del Backend

1. Se diseñó y se implementó un servicio para proveer y guardar información de la red social GobHash cumpliendo los requisitos funcionales y no funcionales definidos.
2. Se implementó un protocolo de comunicación para transmitir en tiempo real.
3. Se integró con el módulo de análisis, diseño e implementación de frontend aunque no se usaron todas las funcionalidades disponibles.
4. El API se diseñó de tal manera que fuera lo más comprensible posible, se documentó y se usaron buenas prácticas para un mantenimiento sencillo.
5. Las pruebas de estrés muestran que el API soporta una carga de al menos 1000 usuarios en total y 100 concurrentemente.

F. Análisis, diseño y desarrollo de frontend

1. Se creó una red social que permite realizar análisis descriptivo sobre las adquisiciones del Estado de Guatemala.
2. Se creó una red social capaz de soportar una discusión sobre información polémica resultante de un análisis realizado por un usuario.
3. Se permitió la integración con los módulos Desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de GuateCompras y Desarrollo e implementación de Backend.

4. El uso de una infraestructura automatizada permitió reducir los tiempos de entrega y fácil actualización y mantenimiento.
5. El análisis y diseño realizado cumplió con los requisitos funcionales y no funcionales planteados.
6. El desarrollo de la red social cumple con los requisitos funcionales y no funcionales planteados.

XII. RECOMENDACIONES

A. Diseño e implementación de modelo de datos de análisis descriptivo

1. Agregar más información al modelo de datos como las ofertas y concursos publicados en Guatecompras.
2. Si en algún momento se libera este mismo conjunto de datos por parte del Gobierno de Guatemala, compararlo con el conjunto desarrollado en este proyecto.
3. Expandir las funcionalidades de la plataforma GobHash para que se asemeje a otros medios sociales más maduros.

B. Desarrollo e implementación de indicadores estadísticos representados a través de gráficos

1. Desarrollar un módulo de HCI sobre este módulo para validar que el nombre de los campos que el usuario llenar para crear gráficos se entiendan y que los pasos para crearlos sean claros.
2. Crear pruebas de usabilidad para el proceso de crear gráficos.
3. Agregar la posibilidad de generar más tipos de gráficos para aumentar el valor para el usuario.
4. Agregar la posibilidad de crear otros tipos de filtros sobre los gráficos.
5. Agregar la posibilidad de descargar el conjunto de datos que se utiliza para formar el gráfico en formato Excel o CSV.

C. Extracción de datos de adjudicaciones, proveedores y compradores del sistema Guatecompras

1. Adaptar el programa para la extracción de datos en el nuevo formato de Guatecompras.
2. Ejecutar pruebas de rendimiento exhaustivas, para detectar posibles puntos de mejora del programa.
3. Analizar las bitácoras generadas durante la ejecución para encontrar la cantidad de bloqueos sufridos.
4. Adaptar el programa para reutilizar la información almacenada en la base de datos y de esta manera obtener solamente información nueva de Guatecompras.
5. Obtener los datos correspondientes a los años 2004-2015 para dotar a la plataforma de información histórica para análisis.
6. Obtener los datos correspondientes al año 2017 con el objetivo de tener datos actualizados.
7. Ejecutar el proceso de obtención de datos de manera diaria

8. Crear un API, basado en el programa, para conexiones con Guatecompras.gt con el objetivo permitir la extracción de datos de una manera más sencilla.
9. Adaptar el formato de fechas obtenidas de Guatecompras.gt hacia uno estándar para facilitar el proceso de carga de los archivos CSV hacia la base de datos.
10. Automatizar el proceso de validación de cantidad y monto de las adjudicaciones obtenidas.
11. Obtener los datos adicionales que se encuentran disponibles para los compradores, proveedores y adjudicaciones con el objetivo de enriquecer la variedad de análisis que se puedan llevar a cabo en GobHash.
12. Encontrar un método de prevención de bloqueos con base a la reacción de los dispositivos de seguridad establecidos por Guatecompras
13. Aprovechar las características de paralelismo que presenta *Go* para mejorar el rendimiento del programa al desarrollarlo en este lenguaje de programación.

D. Diseño y desarrollo de estudio de usabilidad para la interacción humano-computador

1. Llevar a cabo los módulos de usabilidad y diseño lo más cerca posible del equipo de desarrollo, transmitir las ideas y diseños previos para que el equipo de desarrollo se haga a la imagen de lo que se desea en términos de diseño y usabilidad, de esta manera el desarrollo inicia y termina enfocado en las metas de usabilidad.
2. De ser posible, emplear el uso de herramientas de prototipos que sean en lenguajes de maquetación o que posean la opción de exportar código para la construcción de la primera versión funcional, de esta manera, el desarrollo se agiliza y se maximiza la consistencia entre prototipo y producto final.

E. Diseño e implementación del Backend

1. Se recomienda guardar una imagen por parte del módulo desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de GuateCompras para reducir el tamaño de una solicitud ya que actualmente se envía toda la definición de un gráfico para mostrarla en el feed. Esto reduciría el tiempo de respuesta del API y también se tendría un posible mejor rendimiento en el frontend.
2. Se recomienda aumentar la cantidad de conexiones disponibles por parte del ORM en el módulo desarrollo e implementación de indicadores estadísticos representados a través de gráficos con base en la información extraída del portal de GuateCompras para mejorar el tiempo de respuesta del endpoint de los gráficos.

3. Cuando se trabaje con infraestructura basada en clúster de nodos se recomienda tener en cuenta la posibilidad de que no se mantenga la sesión en memoria a través de los nodos por lo que puede ser necesario un lugar aparte donde guardar las sesiones.
4. Se recomienda utilizar las versiones más recientes de JavaScript (ES6 y mayor) para aprovechar las buenas prácticas que se pueden obtener de usar las versiones más actualizadas, como *async/await* lo cual puede reducir el tiempo del desarrollo.
5. El protocolo de tiempo real se podría mejorar utilizando un micro servicio y usando un lenguaje de programación funcional como Elixir para tener un mejor rendimiento

F. Diseño e implementación del frontend

1. Se recomienda tomar en cuenta que en Guatemala no solo se habla el idioma español, por lo cual es necesario el soporte de varias lenguas nativas sobre la red social a modo de ser una plataforma inclusiva.
2. Se recomienda utilizar una infraestructura más robusta como *AWS*. *Heroku* es una plataforma basada en *AWS*, sin embargo, no cuenta con las mismas capacidades.
3. Se recomienda agregar una sala de conversación para poder tener una red social completa. Esto ayudaría a que la comunicación en la red social no sea totalmente vertical.

XI. BIBLIOGRAFIA

- (n.d.).
- Banks, C., & Cao, J. (n.d.). *The Guide to Usability Testing*. US: UxPin.
- Berkun, S. (2007, Marzo 12). *The art of usability benchmarking*. Retrieved from scottberkun.com: <http://scottberkun.com/essays/27-the-art-of-usability-benchmarking>
- Bourn, J. (2011, Febrero 16). *Bourn Creative*. Retrieved from <https://www.bourncreative.com/meaning-of-the-color-orange/>
- Bourne, J. (2010, Diciembre 27). *Bourn Creative*. Retrieved from www.bourncreative.com: <https://bourncreative.com/meaning-of-the-color-grey/>
- Creativyst*. (n.d.). Retrieved septiembre 25, 2017, from How To: The Comma Separated Value (CSV) File Format: <http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm>
- de Ville, B. (2001). *Microsoft Data Mining Integrated Business Intelligence for e-commerce and Knowledge Management*. Digital Press.
- Dix, A. (2009). *Human-computer interaction*. US.
- Dumas, J. S., & Redish, J. (1999). *A practical guide to usability testing*. Intellect books.
- Gehani, N. (2006). *The Database Book Principle and Practice Using MySQL* (1a ed.). Silicon Press.
- Gupta, S. (2015). *Practical MongoDB: Architecting, Developing and Administering MongoDB*. Apress.
- HTML Introduction*. (n.d.). Retrieved from W3Schools: https://www.w3schools.com/html/html_intro.asp
- Lenguaje de marcado*. (s.f.). Recuperado el 25 de septiembre de 2017, de Wikipedia: https://es.wikipedia.org/wiki/Lenguaje_de_marcado
- Mitchell, R. (2015). *Web Scraping with Python: Collecting Data from the Modern Web*. O'Reilly Media, Inc.
- Müller, A. (2009). *ASJP World Language Tree of Lexical Similarity: Version 2*. Retrieved 10 20, 2017, from <https://web.archive.org/web/20120208154949/http://email.eva.mpg.de/~wichmann/WorldLanguageTree-002.pdf>
- Nielsen, J. (1994). *Usability engineering*. Elsevier.
- Nielsen, J. (2000, Marzo 19). *www.nngroup.com*. Retrieved from <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- Nisbet, R., Elder, J., & Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press.
- OECD. (30 de enero de 2004). *Science, Technology and Innovation for the 21st Century. Meeting of the OECD Committee for Scientific and Technological Policy at Ministerial Level, 29-30 January 2004 - Final Communiqué*. Recuperado el 4 de abril de 2017, de OECD: <http://www.oecd.org/science/sci-tech/sciencetechnologyandinnovationforthe21stcenturymeetingoftheoecdcommitteeofscientificandtechnologicalpolicyatministeriallevel29-30january2004-finalcommunique.htm>
- Open Definition. (s.f.). *Open Definiton 2.1*. Recuperado el 27 de abril de 2017, de Open Definition: <http://opendefinition.org/od/2.1/en/>
- Open Government Partnership. (2011). *Open Government Declaration*. Recuperado el 5 de mayo de 2017, de <https://www.opengovpartnership.org/about/open-government-declaration>
- Open Government Partnership. (s.f.). *Guatemala*. Recuperado el 4 de abril de 2017, de Open Government Partnership: <https://www.opengovpartnership.org/countries/guatemala>
- PHP. (n.d.). *What is PHP?* Retrieved septiembre 25, 2017, from PHP: <http://php.net/manual/en/intro-what-is.php>
- Ramakrishnan, R. (2002). *Database Management System* (3a ed.). McGraw Hill.

- Richardson, L. (2017, agosto 11). *Beautiful Soup*. Retrieved abril 14, 2017, from Beautiful Soup: <https://www.crummy.com/software/BeautifulSoup/>
- Rob, P., & Coronel, C. (2009). *Databases Systems: Design, Implementation, and Management*. Thomson Course Technology.
- Ross, M., & Kimball, R. (2011). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons.
- Scrapy. (n.d.). *Scrapy at a glance*. Retrieved abril 22, 2017, from Scrapy docs: <https://docs.scrapy.org/en/latest/intro/overview.html>
- Shafraiovich, Y. (2017, 10 25). *Common Format and MIME type for Comma-Separated Values (CSV) Files*. Retrieved from <https://tools.ietf.org/html/rfc4180>
- Smart Vision Europe Ltd. (2015). *CRISP-DM Cross Industry Standard Process for Data Mining*. Retrieved 05 22, 2017, from <http://crisp-dm.eu/home/crisp-dm-methodology/>
- Stone, T. L., Adams, S., & Morioka, N. (2008). *Color Design Workbook: A Real World Guide to Using Color in Graphic Design*. Rockport Pub.
- Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press.
- Transparente GT. (s.f.). *¿Qué es transparente.gt?* Recuperado el 2 de mayo de 2017, de Transparente GT: <https://transparente.gt/about/>
- Urritia, A. (2017, 03 22). *Sistema Guatecompras*. Retrieved from http://sitios.usac.edu.gt/wp_auditoria/wp-content/uploads/2014/10/SISTEMA-GUATECOMPRAS.pdf
- Van Rossum, G., & Drake, F. (2000). *An introduction to Python*. New York: Network Theory LTD.
- Wichmann, M. (2017, septiembre 10). *Should I use Python 2 or Python 3 for my development activity?* Retrieved septiembre 25, 2017, from Python: <https://wiki.python.org/moin/Python2orPython3>

XII. ANEXOS

A. Proceso de comprensión de datos

Como parte de la metodología CRISP-DM se realizó un proceso de navegación del portal Guatecompras, con el objetivo de identificar y comprender la información que se despliega en él.

Empezamos la exploración de la información en la página principal de www.guatecompras.gt. En esta nos dirigimos a la sección de *Búsquedas* y se seleccionó la opción de *Adjudicaciones*.

Figura 165. Página principal www.guatecompras.gt

Bienvenido al servidor 2

DNCAE 10-2017
Medicamentos paquete 4

AVISO IMPORTANTE:
De conformidad con el subnumeral 6.4 de las Bases del Concurso Nacional de Oferta de Precios DNCAE No. 10-2017 para el suministro y adquisición de "PRODUCTOS MEDICINALES Y FARMACÉUTICOS PAQUETE 4", requerido por el Ministerio de Salud Pública y Asistencia Social y el Instituto Guatemalteco de Seguridad Social, con Número de Operación Guatecompras 6630055, la Dirección Normativa de Contrataciones y Adquisiciones del Estado invita al público en general a participar como observador, al acto público de apertura de pliegos de dicho concurso, el cual se llevará a cabo el 25 de Octubre de 2017, a las 10:00 horas, en el Salón de Capacitaciones de Recursos Humanos, nivel 10 del Edificio del Ministerio de Finanzas Públicas.

Bienvenido a GUATECOMPRAS

Autenticarse
Contrato Abierto
Proveedores
Compradores
Inconformidades
Plan Anual de Compras
Búsquedas
Base Legal
Información y Asistencia
Estadísticas
MIPYME

Concursos Publicados

Concursos Categoría	Vigentes		Adjudicados	
	Hoy	Últimos 7 días	Todos	Hoy
Alimentos y semillas	22	77	87	26
Computación y telecomunicaciones	32	138	185	24
Construcción y materiales afines	54	259	551	63
Electricidad y aire acondicionado	33	86	113	20
Limpieza, fumigación y artículos afines	32	61	74	28
Muebles y mobiliario de oficina	33	83	96	19
Papelaría y artículos de librería	49	95	99	53
Publicidad, campañas y vallas	3	27	33	8
Salud e insumos hospitalarios	322	970	1091	376
Seguridad y armamento	2	7	20	2
Seguros, fianzas y servicios bancarios	4	11	18	5
Textiles, ropa y calzado	10	40	53	14
Transporte, repuestos y combustibles	24	119	168	35
Otros tipos de bienes o servicios	186	535	728	177
Todos	574	1730	2329	668

Guatecompras DNCAE Contáctenos Boletines GC

www.guatecompras.gt pertenece al Estado de la República de Guatemala
Compatible con Internet Explorer 9 en adelante, Mozilla Firefox y Chrome.

Esto nos llevó a la página de búsqueda de adjudicaciones. Se seleccionó la opción de *Buscar TODAS las adjudicaciones* para que el sistema nos desplegará un listado de todos los concursos que han sido adjudicados hasta el día de hoy. Al obtener esta lista se accedió a la primera adjudicación del listado haciendo click sobre el link en el NOG (Número de Operación Guatecompras) del concurso.

Figura 166. Sección de búsqueda de adjudicaciones de www.guatecompras.gt

www.guatecompras.gt/proveedores/consultaadyprovee.aspx

GUATECOMPRAS.gt
Sistema de Adjudicaciones y Contrataciones del Estado de Guatemala

Bienvenido al servidor 2

Bienvenido a la sección de Búsqueda

Inicio > Buscar adjudicaciones

Buscar Adjudicaciones

GUATECOMPRAS contiene todas las adjudicaciones por concursos efectuadas, tanto a proveedores con NIT como a proveedores extranjeros sin NIT, desde el 19.Jul.2004, fecha en que comenzó a funcionar el registro de adjudicaciones.

Por favor elija una opción:

- Opción 1:** Buscar TODAS las adjudicaciones
- Opción 2: Buscar adjudicaciones a PROVEEDORES con NIT
- Opción 3: Buscar adjudicaciones a EXTRANJEROS sin NIT
- Opción 4: Buscar adjudicaciones a proveedores INHABILITADOS
- Opción 5: Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor
- Opción 6: Buscar por NIT del proveedor
- Opción 7: Buscar por NOG del concurso
- Opción 8: Buscar adjudicaciones con contratos de infraestructura con recursos públicos
- Opción 9: Buscar adjudicaciones por Entidad compradora, segmentadas por modalidad **nuevo**

www.guatecompras.gt pertenece al Estado de la República de Guatemala
Compatible con Internet Explorer 9 en adelante, Mozilla Firefox y Chrome.

Figura 167. Selección de adjudicación por NOG

www.guatecompras.gt/proveedores/consultaadyprovee.aspx

GUATECOMPRAS.gt
Sistema de Adjudicaciones y Contrataciones del Estado de Guatemala

Bienvenido al servidor 2

Bienvenido a la sección de Búsqueda

Inicio > Buscar adjudicaciones

Buscar Adjudicaciones

GUATECOMPRAS contiene todas las adjudicaciones por concursos efectuadas, tanto a proveedores con NIT como a proveedores extranjeros sin NIT, desde el 19.Jul.2004, fecha en que comenzó a funcionar el registro de adjudicaciones.

Por favor elija una opción:

- Opción 1:** Buscar TODAS las adjudicaciones
- Opción 2: Buscar adjudicaciones a PROVEEDORES con NIT
- Opción 3: Buscar adjudicaciones a EXTRANJEROS sin NIT
- Opción 4: Buscar adjudicaciones a proveedores INHABILITADOS
- Opción 5: Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor
- Opción 6: Buscar por NIT del proveedor
- Opción 7:** Buscar por NOG del concurso
- Opción 8: Buscar adjudicaciones con contratos de infraestructura con recursos públicos
- Opción 9: Buscar adjudicaciones por Entidad compradora, segmentadas por modalidad **nuevo**

1 al 50 de 537,314 adjudicaciones encontradas.
Si desea saber ¿Qué incluye la columna "Monto"? [presione aquí](#)

Fecha de adjudicación ▼	Nombre o razón social	NIT o país	Monto (Q.)	NOG
18.oct.2017	IMPORTADORA FARMACEUTICA Y PRODUCTOS MEDICOS, SOCIEDAD ANONIMA			6912710
18.oct.2017	BOJORQUEZ,MEJIA, MARIA,CRISTINA	5506433	3,807.14	6922753
18.oct.2017	REPRESENTACIONES Y SERVICIOS ELECTROMEDICOS INDUSTRIALES Y DE COMUNICACION S.A.	4066154	14,303.57	6944744
18.oct.2017	MEDEXA DE CENTRO AMERICA, SOCIEDAD ANONIMA	74867784	5,257.14	6946054
18.oct.2017	SUPLIDORA HOSPITALARIA, SOCIEDAD ANONIMA	2475546K	3,026.79	6923798
18.oct.2017	SUPLIDORA HOSPITALARIA, SOCIEDAD ANONIMA	2475546K	5,000.00	6923674
18.oct.2017	JCP ASOCIADOS, SOCIEDAD ANONIMA	60524510	12,966,336.00	6469361
18.oct.2017	ABBOTT LABORATORIOS SOCIEDAD ANONIMA	671517	861,000.00	6502024
18.oct.2017	CARLOS ERNESTO ANTILLON SUCEORES	3389421	56,538.59	7019033
18.oct.2017	OFFYMARKET, SOCIEDAD ANONIMA	29010438	603,377.00	6654878
18.oct.2017	GEOGRAPHICAL INFO SYSTEMS, SOCIEDAD ANONIMA	36104973	61,200.00	6888555
18.oct.2017	GRUPO SOLID (GUATEMALA), SOCIEDAD ANONIMA	5492343	16,555.00	7044127

Este link nos lleva a la página de detalle de concurso del concurso con el NOG que se seleccionó (En este caso 6912710). En esta página se pudo empezar a analizar la información que se presenta en Guatecompras relativa a concursos que han sido adjudicados.

Figura 168. Despliegue de datos del concurso con NOG 6912710 en www.guatecompras.com

Detalle de concurso	
Datos del Concurso	
NOG:	6912710
Categoría:	Salud e insumos hospitalarios
Descripción:	COMPRA DE MATERIAL MEDICO QUIRURGICO EVENTO NO. 483-2017 CODIGO 6332
Modalidad:	Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)
Tipo de concurso:	Público
Entidad:	INSTITUTO GUATEMALTECO DE SEGURIDAD SOCIAL -IGSS-
Tipo de entidad:	Entidades Descentralizadas, Autónomas y de Seguridad Social
Unidad compradora:	HOSPITAL DR. JUAN JOSE AREVALO BERMEJO
Fecha de publicación:	26.septiembre.2017 Hora:10:13:33 a. m.
Fecha de presentación de ofertas:	28.septiembre.2017 Hora:8:00:00 a. m.
Fecha de cierre de recepción de ofertas:	28.septiembre.2017 Hora:8:30:00 a. m.
Recepción de ofertas:	Sólo electrónicas. Todas las ofertas deben recibirse en forma electrónica a través de Guatecompras y no se permite recibirlas en papel. Vea: ¿Qué es y como puedo ofertar electrónicamente?
Fecha de finalización:	18.octubre.2017 Hora:7:27:48 p. m.
Estatus:	Terminado adjudicado
Ofertas Presentadas:	Ver ofertas presentadas(4)

Estos datos se encuentran en el encabezado de la página llamado *Datos del Concurso*. Luego encontramos tablas que nos presentan otro tipo de información Relacionada a la adjudicación. Encontramos la tabla de anexos que contiene las bases, especificaciones generales o términos de referencia del concurso, los cuales dependen de la entidad compradora. Estos se encuentran como un archivo descargable de word.

Luego encontramos una descripción más detallada de los productos o servicios que se desean adquirir. Esta tabla cuenta con los campos Nombre, Cantidad y Unidad de medida. Al realizar revisiones de diferentes adjudicaciones se pudo determinar que no existe un estándar o formato definido por el que los compradores tengan que llenar esta tabla al momento de publicar el concurso. Por lo que queda al criterio del comprador la forma en la que describió los productos, la cuantificación de las cantidades y la determinación de las unidades de medida que conforman un producto dentro del concurso.

Al ser un concurso adjudicado, después podemos encontrar la tabla *Proveedor Adjudicado*. Esta cuenta con los campos *NIT o País, Nombre o razón social, Contrato y Monto*. Cada proveedor se identifica por su NIT (Número de Identificación Tributaria) si es un proveedor nacional o con su país de origen si es un proveedor extranjero. Esta información se encuentra en una tabla ya que una sola adjudicación puede ser adjudicada a varios proveedores.

Figura 169. Visualización de tablas de tipos de anexos, tipo de Producto, proveedor adjudicado

Tipo de Anexos			
+ Bases, especificaciones generales o t...			
Tipo de Producto			
1 al 1 de 1 tipos de productos			
Nombre	Cantidad	Unidad de medida	
RODILLERA ELÁSTICA ALGODÓN Ó POLIESTER CON FILAMENTO DE LATEX, CON ORILLA REFORZADA, TAMAÑO MEDIANO, EMPAQUE INDIVIDUAL	50	RODILLERA	
Ir a la página: 1			
Proveedor Adjudicado			
NIT o país	Nombre o razón social	Contrato	Monto (O.) ▼
87780011	IMPORTADORA FARMACEUTICA Y PRODUCTOS MEDICOS, SOCIEDAD ANONIMA		Q937.50
		Total	Q937.50
Operaciones sobre el concurso			
Operaciones Disponibles			
1	Efectuar preguntas (aquí puede hacer preguntas sobre el concurso)		
2	Dar seguimiento (aquí puede suscribirse al boletín de seguimiento del concurso)		
3	Presentar inconformidad (aquí puede ingresar quejas relacionadas con el concurso)		
4	Imprimir (genera archivo en formato PDF para imprimir)		
5	Exportar (toma los Tipos de Producto y los copia en un archivo de formato XLS)		
Historial de Acciones			

Por último, encontramos las tablas de *Operaciones sobre el concurso*, donde se dan links hacia las diferentes acciones que un usuario puede hacer sobre el concurso como efectuar preguntas, dar seguimiento, presentar inconformidades o imprimir los datos. Y la tabla de Historial de acciones que nos muestra los eventos por los que ha pasado el concurso desde el momento que es publicado hasta que el evento se adjudica.

Figura 170. Visualización de tabla: Historial de Acciones

No	Acción	Descripción
3	FINALIZACIÓN: se publicó la finalización del concurso La adjudicación debe incluir el detalle de la evaluación	Estatus anterior: En evaluación . Estatus actual: Terminado adjudicado . Proveedores adjudicados: 1) (NIT: 87780011) IMPORTADORA FARMACEUTICA Y PRODUCTOS MEDICOS, SOCIEDAD ANONIMA - Q937.50 6912710@483.xls (31KB) Por: ROJAS, CASTRO, GABRIELA, YAZMIN. 18.octubre.2017 19:27:47
2	ESTATUS (AUTOMÁTICO): hubo un cambio automático del estatus del concurso porque se alcanzó la fecha máxima para presentar ofertas Los precios ofertados deben publicarse a más tardar al día siguiente La adjudicación debe efectuarse dentro de los 5 días	Estatus anterior: Vigente . Estatus actual: En evaluación Por: Sistema GUATECOMPRAS. 26.septiembre.2017 08:30:25
1	CREACIÓN: se creó el concurso en el sistema	El concurso público con NOG '6912710' fue publicado exitosamente. La creación con origen desde bases fue exitosa. Se crearon 1 productos Se agregaron 1 documentos como anexos de las bases. Bases, especificaciones generales o términos de referencia 6912710@483.doc (185KB) Por: Aracely González Méndez. 26.septiembre.2017 10:13:33
Ir a la página: 1		

Luego, utilizando el link en el dato de unidad compradora encontramos el perfil de la unidad compradora encargada del concurso. En esta encontramos los siguientes datos en el encabezado.

Figura 171. Despliegue de datos de la unidad compradora Hospital Dr. Juan José Arévalo

Detalle de entidad compradora

Datos registrados actuales	
Entidad:	INSTITUTO GUATEMALTECO DE SEGURIDAD SOCIAL -IGSS-
NIT:	2342855
Unidad Compradora:	HOSPITAL DR. JUAN JOSE AREVALO BERMEJO
Departamento:	GUATEMALA
Municipio:	GUATEMALA
Dirección:	19 avenida 7-14 zona 6
Teléfonos:	2282431-39
Números de Fax:	2280157-2280163
Apartado Postal:	[--No Especificado--]
Páginas Web:	[--No Especificado--]
Direcciones de Correo:	compras.ijab@igssqt.org
Información de SAT:	Para consultar haga clic aquí

Regresando a los datos encontrados en los detalles del concurso adjudicado llegamos al perfil de la entidad compradora. En esta se encontraron los siguientes datos en el encabezado.

Figura 172. Despliegue de datos de la entidad Instituto Guatemalteco de Seguridad

Detalle de entidad compradora

Datos registrados actuales	
Entidad:	INSTITUTO GUATEMALTECO DE SEGURIDAD SOCIAL -IGSS-
NIT:	2342855
Tipo:	Entidades Descentralizadas, Autónomas y de Seguridad Social
Entidad origen de los fondos:	11400068 - INST. GUAT. DE SEGURIDAD SOCIAL -IGSS-
Departamento:	GUATEMALA
Municipio:	GUATEMALA
Dirección:	7a. Avenida 22-72, Centro cívico, Zona 1
Teléfonos:	24121224 ext. 1233-39
Números de Fax:	24121224 ext. 1235 1238
Apartado Postal:	[--No Especificado--]
Páginas Web:	www.igssqt.org
Direcciones de Correo:	quatecompras.igss@igssqt.org
Información de SAT:	Para consultar haga clic aquí

Por último, se encuentra una tabla con los nombre y links hacia los perfiles de las unidades compradoras que forman parte de esta entidad. De estas observaciones pudimos determinar que las unidades compradoras de una entidad son las encargadas de ejecutar concursos en Guatecompras.

Regresando a la tabla de *Proveedores Adjudicados* del concurso se llegó al perfil del proveedor por medio del link en el campo de Nombre de esta tabla. En este perfil se encuentran los siguientes datos del proveedor.

Figura 173. Visualización de la identificación en la página de registro de proveedores

Registro de Proveedores - [REDACTED]

Identificación
(Datos recibidos de la SAT el: 18.oct..2017 10:55:12)

CUI:
Nombre o razón social: [REDACTED]
Número de Identificación Tributaria (NIT): [REDACTED]
Nombre comercial 1: [REDACTED]

Estatus Actual del Proveedor
(lea antes ¿Qué significa estar inscrito, habilitado, adjudicado y con contraseña?)

Estatus en Guatecompras

Habilitado o Inhabilitado: HABILITADO
Adjudicado o No adjudicado: ADJUDICADO
Participa o no en Contrato Abierto: NO PARTICIPA (no tiene productos en el catálogo)
Con o Sin contraseña: CON CONTRASEÑA

Estatus en la SAT(18.oct..2017 10:55:12)

Estatus del RTU: ACTIVO
Solvencia del NIT en RTU: SOLVENTE

Domicilio Fiscal
(Datos recibidos de la SAT el: 18.oct..2017 10:55:12)

Departamento: GUATEMALA
Municipio: GUATEMALA
Dirección: [REDACTED]
Teléfonos: [REDACTED]
Números de fax: [REDACTED]

Representantes legales
(Datos recibidos de la SAT el: 18.oct..2017 10:55:12)

Proveedor	Plazo de Nombramiento
Representante Legal 1 [REDACTED]	08/09/2017

Datos de inscripción
(Datos recibidos de la SAT el: 18.oct..2017 10:55:12)

Tipo de organización: SOCIEDAD ANÓNIMA
Número de escritura de constitución: 41
Fecha de constitución: [REDACTED]
Actividad Económica: VENTA AL POR MENOR DE PRODUCTOS FARMACÉUTICOS Y MEDICINALES, COSMÉTICOS Y ARTÍCULOS DE TOCADOR
Nit Notario: [REDACTED]

Otro de los campos que se pueden encontrar en el perfil de los proveedores son el NIT y nombre de los representantes legales. Pueden ser una o más personas. El NIT también funciona como link que redirecciona al perfil del representante legal dentro de Guatecompras. Este perfil cuenta con la siguiente información.

Figura 174. Visualización de la identificación en la página registro de proveedores

Registro de Proveedores - [REDACTED]

Identificación
(Datos recibidos de la SAT el: 10.feb..2017 19:57:47)

CUI: [REDACTED]

Nombre o razón social: [REDACTED]

Número de Identificación Tributaria (NIT): [REDACTED]

— Estatus Actual del Proveedor
(lea antes ¿Qué significa estar inscrito, habilitado, adjudicado y con contraseña?)

Estatus en Guatecompras

Habilitado o Inhabilitado: HABILITADO

Adjudicado o No adjudicado: NO ADJUDICADO

Participa o no en Contrato Abierto: NO PARTICIPA (no tiene productos en el catálogo)

Con o Sin contraseña: SIN CONTRASEÑA

Estatus en la SAT(10.feb..2017 19:57:47)

Estatus del RTU ACTIVO

Solvencia del NIT en RTU SOLVENTE

— Datos de inscripción
(Datos recibidos de la SAT el: 10.feb..2017 19:57:47)

Tipo de organización: INDIVIDUAL

Actividad Económica: EMPLEADO EN RELACIÓN DE DEPENDENCIA (SECTOR PRIVADO)

— Inconformidades e Inhabilitaciones

Cantidad de INCONFORMIDADES presentadas: [--NINGUNA--]

Cantidad de INHABILITACIONES recibidas: [--NINGUNA--]

De esta navegación por las diferentes entidades que forman parte de una adjudicación se sacaron las siguientes observaciones:

- No hay estándar o regla para ingresar la información de los productos o servicios adquiridos por medio del concurso adjudicado.
- Existe error humano en el ingreso de los nombres de departamentos y municipios.
- Las fechas están en un formato propio de Guatecompras, que mezcla números, letras y signos de puntuación. Este formato no sigue siempre exactamente la misma estructura, pero sí podría ser expresado por una expresión regular.
- Varios de los campos contienen comas, por lo que será necesario definir un delimitador diferente para los CSV que genere el módulo de extracción de datos.

B. Creación de modelo de datos

El código utilizado para crear el modelo de datos para análisis sobre las adjudicaciones del estado se puede encontrar en el repositorio https://github.com/JJ-Fong/MG_ETL.git. Este modelo se creó utilizando la versión del lenguaje SQL implementada en PostgreSQL 9.5

C. Reporte de validación de datos

Para medir la compleción y precisión de los datos se utilizaron reportes generados en la sección de estadísticas de Guatecompras de las adjudicaciones hechas por modalidad de comprar. La validación se hizo generando reportes de periodos mensuales de la información, para que fuera más fácil encontrar incongruencias entre estos. Después de una validación inicial fue necesario hacer ajustes manuales a los reportes por dos motivos diferentes:

- La información del modelo no contiene adjudicaciones hechas a proveedores extranjeros.
- Los reportes no contienen algunos valores por razones puntuales.

Los ajustes que se hicieron fueron los siguientes.

Tabla 27. Ajustes para validación de datos

Nog	Mes	Monto	Modalidad de Compra	Causa de Ajuste
3424014	4	Q 15,213,028.88	Convenios y Tratados Internacionales (Art. 1 LCE)	Adjudicacion a proveedor extranjero
4679490	6	Q 149,420.00	Adquisición Directa por Ausencia de Oferta (Art. 32 LCE)	Adjudicacion a proveedor extranjero
4855868	6	Q 645,000.00	Cotización (Art. 38 LCE)	Adjudicacion a proveedor extranjero
5058864	7	Q 5,785.00	Convenios y Tratados Internacionales (Art. 1 LCE)	Adjudicacion a proveedor extranjero
3518957	9	Q 4,995,166.11	Convenios y Tratados Internacionales (Art. 1 LCE)	Adjudicacion a proveedor extranjero
5361788	10	Q 380,172.70	Adquisición Directa por Ausencia de Oferta (Art. 32 LCE)	Adjudicacion a proveedor extranjero
5522838	10	Q 25,300.00	Adquisición Directa por Ausencia de Oferta (Art. 32 LCE)	Adjudicacion a proveedor extranjero
5529964	10	Q 8,779.25	Adquisición Directa por Ausencia de Oferta (Art. 32 LCE)	Adjudicacion a proveedor extranjero
5544793	10	Q 912.60	Bienes y Suministros Importados (Art. 5 LCE)	Adjudicacion a proveedor extranjero
5109280	10	Q 71.18	Convenios y Tratados Internacionales (Art. 1 LCE)	Adjudicacion a proveedor extranjero
5225442	10	Q 264,107.92	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5391164	10	Q 18,196.67	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5391172	10	Q 18,184.04	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5531098	10	Q 16,093.73	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5391024	10	Q 22,730.04	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5530768	10	Q 8,843.22	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5528577	11	Q 57,720.69	Adquisición Directa por Ausencia de Oferta (Art. 32 LCE)	Adjudicacion a proveedor extranjero
5113814	11	Q 767,935.00	Convenios y Tratados Internacionales (Art. 1 LCE)	Adjudicacion a proveedor extranjero
5668603	11	Q 10,785,000.00	Procedimientos regulados por el artículo 54 LCE	Adjudicacion a proveedor extranjero
5583055	12	Q 869,600.00	Cotización (Art. 38 LCE)	Adjudicacion a proveedor extranjero
5773857	12	Q 13,280.01	Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)	Adjudicacion a proveedor extranjero
5156807	7	Q 23,943.00	Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)	Readjudicacion
5247438	8	Q 32,564.82	Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)	Readjudicacion
5319153	9	Q 62,400.00	Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)	Correccion monto original incorrecto
5169232	9	Q 896,987.00	COTIZACIÓN (ART. 38 LCE)	Concurso Prorrogado
5505941	10	Q 45,000.00	Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)	Concurso anulado en 2017

Luego de haber hecho estos ajustes los resultados de la validación mensual fueron los siguientes.

Tabla 28. Validación por modalidad de compra de enero 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	1	1	0	0.00%	Q 778,400.00	Q 778,400.00	Q -	0.00%
Compra Directa con Oferta Electrónica	3,006	3006	0	0.00%	Q 92,198,241.35	Q 92,198,241.35	Q -	0.00%
Contrato Abierto (Art. 46 LCE)	1	1	0	0.00%	Q 16,896,286.15	Q 16,896,286.15	Q -	0.00%
Convenios y Tratados Internacionales	3	3	0	0.00%	Q 941,743,904.95	Q 941,743,904.95	Q -	0.00%
Cotización (Art. 38 LCE)	99	99	0	0.00%	Q 41,568,099.67	Q 41,568,099.67	Q -	0.00%
Licitación Pública (Art. 17 LCE)	21	21	0	0.00%	Q 244,619,990.71	Q 244,619,990.71	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	2	2	0	0.00%	Q 236,666.66	Q 236,666.66	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	12	12	0	0.00%	Q 16,181,647.07	Q 16,181,647.07	Q -	0.00%
Ajustes	0	0			Q -	Q -		
Total	3145	3145	0	0.00%	Q 1,354,223,236.56	Q 1,354,223,236.56	Q -	0.00%

Tabla 29. Validación por modalidad de compra de febrero 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	4	4	0	0.00%	Q 158,389.09	Q 158,389.09	Q -	0.00%
Compra Directa con Oferta Electrónica	6380	6380	0	0.00%	Q 204,547,345.87	Q 204,547,345.87	Q -	0.00%
Convenios y Tratados Internacionales	6	6	0	0.00%	Q 1,803,748.54	Q 1,803,748.54	Q -	0.00%
Cotización (Art. 38 LCE)	50	50	0	0.00%	Q 19,326,993.53	Q 19,326,993.53	Q (0.00)	0.00%
Donaciones (Art. 1 LCE)	1	1	0	0.00%	Q 168,000.00	Q 168,000.00	Q -	0.00%
Licitación Pública (Art. 17 LCE)	11	11	0	0.00%	Q 34,023,516.95	Q 34,023,516.95	Q (0.00)	0.00%
Procedimientos Regulados por el artículo 44 LCE	19	19	0	0.00%	Q 3,050,213.42	Q 3,050,213.42	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	13	13	0	0.00%	Q 23,359,539.46	Q 23,359,539.46	Q -	0.00%
Ajustes	0	0			Q -	Q -		
Total	6484	6484	0	0.00%	Q 286,437,746.86	Q 286,437,746.86	Q -	0.00%

Tabla 30. Validación por modalidad de compra de marzo 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	10	10	0	0.00%	Q 3,791,910.23	Q 3,791,910.23	Q -	0.00%
Compra Directa con Oferta Electrónica	6188	6188	0	0.00%	Q 245,253,497.07	Q 245,253,497.07	Q (0.00)	0.00%
Convenios y Tratados Internacionales	7	7	0	0.00%	Q 5,668,727.34	Q 5,668,727.34	Q -	0.00%
Cotización (Art. 38 LCE)	121	121	0	0.00%	Q 52,350,579.61	Q 52,350,579.61	Q (0.00)	0.00%
Donaciones (Art. 1 LCE)	1	1	0	0.00%	Q 3,140,235.00	Q 3,140,235.00	Q -	0.00%
Licitación Pública (Art. 17 LCE)	6	6	0	0.00%	Q 13,310,639.95	Q 13,310,639.95	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	45	45	0	0.00%	Q 3,069,162.01	Q 3,069,162.01	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	15	15	0	0.00%	Q 12,753,748.96	Q 12,753,748.96	Q (0.00)	0.00%
Ajustes	0	0			Q -	Q -		
Total	6393	6393	0	0.00%	Q 339,338,500.17	Q 339,338,500.17	Q (0.00)	0.00%

Tabla 31. Validación por modalidad de compra de abril 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	22	22	0	0.00%	Q 4,582,741.02	Q 4,582,741.02	Q -	0.00%
Compra Directa con Oferta Electrónica	5607	5607	0	0.00%	Q 213,180,144.78	Q 213,180,144.78	Q (0.00)	0.00%
Contrato Abierto (Art. 46 LCE)	1	1	0	0.00%	Q 6,066,008.48	Q 6,066,008.48	Q -	0.00%
Convenios y Tratados Internacionales	7	6	-1	-16.67%	Q 15,935,722.00	Q 722,696.12	#####	-2105.04%
Cotización (Art. 38 LCE)	180	180	0	0.00%	Q 89,191,207.02	Q 89,191,207.02	Q -	0.00%
Licitación Pública (Art. 17 LCE)	13	13	0	0.00%	Q 33,127,720.19	Q 33,127,720.19	Q (0.00)	0.00%
Procedimientos Regulados por el artículo 44 LCE	50	50	0	0.00%	Q 18,299,823.96	Q 18,299,823.96	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	13	13	0	0.00%	Q 49,513,728.40	Q 49,513,728.40	Q -	0.00%
Ajustes	0	1			Q -	Q 15,213,025.88		
Total	5893	5893	0	0.00%	Q 429,897,095.85	Q 429,897,095.85	Q (0.00)	0.00%

Tabla 32. Validación por modalidad de compra de mayo 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	28	28	0	0.00%	Q 5,861,676.08	Q 5,861,676.08	Q -	0.00%
Compra Directa con Oferta Electrónica	7339	7339	0	0.00%	Q 256,260,236.37	Q 256,260,236.37	Q (0.00)	0.00%
Contrato Abierto (Art. 46 LCE)	9	9	0	0.00%	Q 259,575,768.40	Q 259,575,768.40	Q -	0.00%
Convenios y Tratados Internacionales	3	3	0	0.00%	Q 939,362.50	Q 939,362.50	Q -	0.00%
Cotización (Art. 38 LCE)	261	261	0	0.00%	Q 130,406,459.48	Q 130,406,459.48	Q -	0.00%
Licitación Pública (Art. 17 LCE)	27	27	0	0.00%	Q 60,695,034.58	Q 60,695,034.58	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	16	16	0	0.00%	Q 37,848,013.38	Q 37,848,013.38	Q (0.00)	0.00%
Procedimientos regulados por el artículo 54 LCE	16	16	0	0.00%	Q 35,756,327.10	Q 35,756,327.10	Q -	0.00%
Sin Modalidad	2	2	0	0.00%	Q 0.58	Q 0.58	Q -	0.00%
Ajustes	0	0			Q -	Q -		
Total	7701	7701	0	0.00%	Q 787,342,878.47	Q 787,342,878.47	Q (0.00)	0.00%

Tabla 33. Validación por modalidad de compra de junio 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	44	43	-1	-2.33%	Q 12,007,705.10	Q 11,858,285.10	Q (149,420.00)	-1.26%
Compra Directa con Oferta Electrónica	7855	7855	0	0.00%	Q 283,866,660.95	Q 283,866,660.95	Q (0.00)	0.00%
Convenios y Tratados Internacionales	10	10	0	0.00%	Q 8,981,949.54	Q 8,981,949.54	Q -	0.00%
Cotización (Art. 38 LCE)	419	418	-1	-0.24%	Q 181,970,650.45	Q 181,325,650.45	Q (645,000.00)	-0.36%
Donaciones (Art. 1 LCE)	1	1	0	0.00%	Q 89,004.00	Q 89,004.00	Q -	0.00%
Licitación Pública (Art. 17 LCE)	72	72	0	0.00%	Q 178,258,738.65	Q 178,258,738.65	Q -	0.00%
Negociaciones entre Entidades Públicas (Art. 2 LCE)	1	1	0	0.00%	Q 173,499.20	Q 173,499.20	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	10	10	0	0.00%	Q 7,504,699.90	Q 7,504,699.90	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	17	17	0	0.00%	Q 36,704,529.17	Q 36,704,529.17	Q -	0.00%
Ajustes	0	2			Q -	Q 794,420.00		
Total	8429	8429	0	0.00%	Q 709,557,436.96	Q 709,557,436.96	Q (0.00)	0.00%

Tabla 34. Validación por modalidad de compra de julio 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	26	26	0	0.00%	Q 4,389,596.89	Q 4,389,596.89	Q -	0.00%
Compra Directa con Oferta Electrónica	5974	5975	1	0.02%	Q 215,605,490.90	Q 215,629,433.90	Q 23,943.00	0.01%
Convenios y Tratados Internacionales	9	8	-1	-12.50%	Q 3,444,801.10	Q 3,439,016.10	Q (5,785.00)	-0.17%
Cotización (Art. 38 LCE)	343	343	0	0.00%	Q 156,745,619.61	Q 156,745,619.61	Q -	0.00%
Donaciones (Art. 1 LCE)	2	2	0	0.00%	Q 281,000.00	Q 281,000.00	Q -	0.00%
Licitación Pública (Art. 17 LCE)	111	111	0	0.00%	Q 224,453,905.48	Q 224,453,905.48	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	16	16	0	0.00%	Q 128,344,225.55	Q 128,344,225.55	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	12	12	0	0.00%	Q 73,777,420.61	Q 73,777,420.61	Q -	0.00%
Ajustes	1	1			Q 23,943.00	Q 5,785.00		
Total	6494	6494	0	0.00%	Q 807,066,003.14	Q 807,066,003.14	Q (0.00)	0.00%

Tabla 35. Validación por modalidad de compra de agosto 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	42	42	0	0.00%	Q 5,010,171.27	Q 5,010,171.27	Q -	0.00%
Compra Directa con Oferta Electrónica	6270	6271	1	0.02%	Q 213,218,609.33	Q 213,251,174.15	Q 32,564.82	0.02%
Convenios y Tratados Internacionales	7	7	0	0.00%	Q 6,083,645.55	Q 6,083,645.55	Q -	0.00%
Cotización (Art. 38 LCE)	694	694	0	0.00%	Q 332,387,266.40	Q 332,387,266.40	Q -	0.00%
Licitación Pública (Art. 17 LCE)	77	77	0	0.00%	Q 207,804,869.52	Q 207,804,869.52	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	20	20	0	0.00%	Q 5,801,940.01	Q 5,801,940.01	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	9	9	0	0.00%	Q 27,431,740.57	Q 27,431,740.57	Q -	0.00%
Sin Modalidad	2	2	0	0.00%	Q 20,285,852.27	Q 20,285,852.27	Q -	0.00%
Ajustes	1	0			Q 32,564.82	Q -		
Total	7122	7122	0	0.00%	Q 818,056,659.74	Q 818,056,659.74	Q -	0.00%

Tabla 36. Validación por modalidad de compra de septiembre 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	40	40	0	0.00%	Q 4,949,490.12	Q 4,949,490.12	Q (0.00)	-0.00%
Compra Directa con Oferta Electrónica	9426	9427	1	0.01%	Q 308,543,577.56	Q 308,605,977.56	Q 62,400.00	0.02%
Contrato Abierto (Art. 46 LCE)	1	1	0	0.00%	Q 144,408,763.32	Q 144,408,763.32	Q -	0.00%
Convenios y Tratados Internacionales	24	23	-1	-4.35%	Q 36,226,853.45	Q 31,231,687.34	Q (4,995,166.11)	-15.99%
Cotización (Art. 38 LCE)	866	867	1	0.12%	Q 421,320,703.65	Q 422,217,690.65	Q 896,987.00	0.21%
Licitación Pública (Art. 17 LCE)	159	159	0	0.00%	Q 436,860,739.24	Q 436,860,739.24	Q -	0.00%
Procedimiento Regulado por el Art. 54Bis	1	1	0	0.00%	Q 194,647.80	Q 194,647.80	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	8	8	0	0.00%	Q 2,679,009.60	Q 2,679,009.60	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	7	7	0	0.00%	Q 47,389,506.35	Q 47,389,506.35	Q -	0.00%
Ajustes	2	1			Q 959,387.00	Q 4,995,166.11		
Total	10534	10534	0	0.00%	Q 1,403,532,678.09	Q 1,403,532,678.09	Q (0.00)	0.00%

Tabla 37. Validación por modalidad de compra de octubre 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	56	53	-3	-5.66%	Q 5,165,229.05	Q 4,750,977.10	Q (414,251.95)	-8.72%
Bienes y Suministros Importados (Art. 5 LCE)	1	0	-1	-100.00%	Q 912.60	Q -	Q (912.60)	-100.00%
Compra Directa con Oferta Electrónica	7856	7857	1	0.01%	Q 290,725,897.04	Q 290,770,897.04	Q 45,000.00	0.02%
Convenios y Tratados Internacionales	15	14	-1	-7.14%	Q 19,320,866.85	Q 19,320,795.67	Q (71.18)	0.00%
Cotización (Art. 38 LCE)	588	588	0	0.00%	Q 273,416,529.29	Q 273,416,529.29	Q -	0.00%
Licitación Pública (Art. 17 LCE)	221	221	0	0.00%	Q 616,754,700.11	Q 616,754,700.11	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	27	21	-6	-28.57%	Q 7,420,063.62	Q 7,071,908.00	Q (348,155.62)	-4.92%
Procedimientos regulados por el artículo 54 LCE	24	24	0	0.00%	Q 14,060,054.13	Q 14,060,054.13	Q (0.00)	0.00%
Ajustes	1	11			Q 4,500.00	Q 722,891.35		
Total	8789	8789	0	0.00%	Q 1,226,868,752.69	Q 1,226,868,752.69	Q (0.00)	0.00%

Tabla 38. Validación por modalidad de compra de noviembre 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	80	79	-1	-1.27%	Q 24,464,463.40	Q 24,406,742.71	Q (57,720.69)	-0.24%
Compra Directa con Oferta Electrónica	7055	7055	0	0.00%	Q 277,466,014.85	Q 277,466,014.85	Q -	0.00%
Contrato Abierto (Art. 46 LCE)	1	1	0	0.00%	Q 41,245,519.57	Q 41,245,519.57	Q -	0.00%
Convenios y Tratados Internacionales	30	29	-1	-3.45%	Q 23,985,402.79	Q 23,217,467.79	Q (767,935.00)	-3.31%
Cotización (Art. 38 LCE)	492	492	0	0.00%	Q 215,454,424.45	Q 215,454,424.45	Q -	0.00%
Licitación Pública (Art. 17 LCE)	95	95	0	0.00%	Q 211,560,130.42	Q 211,560,130.42	Q -	0.00%
Negociaciones entre Entidades Públicas (Art. 2 LCE)	1	1	0	0.00%	Q 84,719.26	Q 84,719.26	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	18	18	0	0.00%	Q 5,466,450.55	Q 5,466,450.55	Q -	0.00%
Procedimientos regulados por el artículo 54 LCE	34	33	-1	-3.03%	Q 44,893,231.66	Q 34,108,231.66	Q (10,785,000.00)	-31.62%
Ajustes	0	3	3	100.00%	Q -	Q 11,610,654.83	Q 11,610,654.83	100.00%
Total	7806	7806	0	0.00%	Q 844,620,356.95	Q 844,620,356.95	Q (0.86)	0.00%

Tabla 39. Validación por modalidad de compra de diciembre 2016

Modalidad de Compra	Cantidad de Adjudicaciones				Monto total			
	Guatecompras	GobHash	Diferencia	%Error	Guatecompras	GobHash	Diferencia	%Error
Adquisición Directa por Ausencia de Oferta	58	58	0	0.00%	Q 10,062,182.98	Q 10,062,182.98	Q -	0.00%
Compra Directa con Oferta Electrónica	3821	3823	2	0.05%	Q 166,509,399.49	Q 166,518,193.15	Q 8,793.66	0.01%
Convenios y Tratados Internacionales	28	29	1	3.45%	Q 23,994,821.33	Q 24,013,885.13	Q 19,063.80	0.08%
Cotización (Art. 38 LCE)	383	382	-1	-0.26%	Q 162,942,488.85	Q 162,045,888.85	Q (896,600.00)	-0.55%
Donaciones (Art. 1 LCE)	1	1	0	0.00%	Q 51,720.00	Q 51,720.00	Q -	0.00%
Licitación Pública (Art. 17 LCE)	68	68	0	0.00%	Q 282,075,789.16	Q 282,075,789.16	Q -	0.00%
Procedimientos Regulados por el artículo 44 LCE	20	19	-1	-5.26%	Q 8,554,446.04	Q 8,541,166.03	Q (13,280.01)	-0.16%
Procedimientos regulados por el artículo 54 LCE	26	26	0	0.00%	Q 38,653,369.37	Q 38,653,369.37	Q -	0.00%
Ajustes	1	2	1	50.00%	Q 19,063.80	Q 909,880.01	Q 890,816.21	97.90%
Total	4406	4408	2	0.05%	Q 692,863,281.02	Q 692,872,074.68	Q 8,793.66	0.00%

D. Código de automatización para la preparación de datos

El código utilizado para automatizar las tareas de formateo e integración de datos para cargas en el modelo de análisis de datos de adquisiciones del estado se puede encontrar en el repositorio https://github.com/JJ-Fong/MG_ETL.git en los archivos *main.py*, *sanitizer.py* y *classifier.py*.

E. Protocolo de mantenimiento de los datos

El proceso de mantenimiento de datos debe realizarse bajo una de las siguientes de las premisas

- Se desea agregar nueva información al sistema
- Se desea validar de nuevo o actualizar la información que actualmente se encuentra en el sistema

Si se presenta alguno de estos dos casos, los pasos a seguir son los siguientes:

1. Configuración de los archivos *schemas.json* y *files.json*
2. Integración de las fuentes utilizando el programa *main.py*
3. Carga de la data cruda en la base datos *ghanalysis*
4. Reconstrucción del esquema en la base de datos *ghanalysis*
5. Validación de los datos utilizando los reportes de Guatecompras de adjudicaciones por modalidad de compra en el periodo de tiempo que se desee.

1. Configuración de los archivos *schemas.json* y *files.json*. Estos archivos son los responsables de manejar la configuración bajo la cual se llevarán a cabo el formateo de datos y la integración de las diferentes fuentes. El archivo *schemas.json* define los diferentes esquemas de consolidación que existen. En este archivo se definen el nombre y tipo de los campos que conforman el esquema. Es importante tener cuidado con el tipo de los campos, ya que es este valor el que el programa *sanitizer.py* utiliza para formatear correctamente el valor del campo. La estructura del json que conforma este archivo es la siguiente.

Figura 175. Estructura JSON schemas.json

```
{
  "tablename": "raw_proveedores",
  "fields": [
    {
      "name": "DEPARTAMENTO",
      "type": "GEO"
    },
    {
      "name": "REP_LEGAL",
      "type": "TXT"
    },
    {
      "name": "MUNICIPIO",
      "type": "GEO"
    },
    {
      "name": "TIPO",
      "type": "TXT"
    },
    {
      "name": "INSCRIPCION_RM",
      "type": "DATE"
    },
    {
      "name": "NOMBRE",
      "type": "TXT"
    },
    {
      "name": "ACTIV_ECONOMICA",
      "type": "TXT"
    },
    {
      "name": "NIT",
      "type": "TXT"
    }
  ]
},
```

Por otro lado tenemos el archivo *files.json*. En este archivo se definen las direcciones de los archivos que se desea unificar bajo que esquema definido en el archivo *schemas.json*. La estructura de este archivo es la siguiente.

Figura 176. Estructura JSON files.json

```
{
  "schema": "raw_proveedores",
  "files": [
    "Data/proveedores.csv",
    "Data/2016/05/proveedores.csv",
    "Data/2016/06/proveedores.csv"
  ]
},
```

2. Integración de las fuentes utilizando el programa *main.py*

Luego de haber configurado los archivos *schemas.json* y *files.json*, se ejecuta el programa *main.py* desde la consola. Este programa lleva a cabo las tareas de formateo de datos e integración de los datos. Como resultado genera archivos con un identificador unico en la carpeta ConsolidatedData. El nombre de estos archivos depende del esquema bajo el que se unifique. Los archivos que se integran bajo el esquema “raw_adjudicaciones” empiezan con ra_*, los archivos que lo hacen con el esquema “raw_proveedores” empiezan con rp_* y los archivos que lo hacen con “raw_compradore” con rc_*.

3. Carga de la data cruda en la base datos *ghanalysis*

El paso luego de haber consolidado la información de las diferentes fuentes es cargarla en el manejador de datos con el que se esté trabajando. En este caso fue en la base de datos de PostgreSQL *ghanalysis*. Para eso se utiliza el comando COPY de PostgreSQL. El query de carga se ve de la siguiente manera.

```
COPY <tabla de la base de datos a la que se desea agregar>
FROM <path del archivo que se desea agregar>
WITH DELIMITER '|' NULL AS 'null'
```

4. Reconstrucción del esquema en la base de datos *ghanalysis*

Debido a que el modelo de análisis de datos se construyó utilizando la función materialized views solo es necesario refrescar estas vistas. Para esto se corren los siguientes comandos sobre la base de datos *ghanalysis*.

```
REFRESH MATERIALIZED VIEW fact_adjudicacion;
REFRESH MATERIALIZED VIEW dim_proveedor;
REFRESH MATERIALIZED VIEW dim_comprador;
REFRESH MATERIALIZED VIEW dim_fecha;
```

```
REFRESH MATERIALIZED VIEW dim_ubicacion;  
REFRESH MATERIALIZED VIEW consolidated_table;
```

5. Validar la información Ya que se agregó nueva información al sistema, es necesario evaluar la calidad de esta. Para ello es necesario repetir el proceso de validación e integridad de la información. Para ello se utilizan los queries bajo la carpeta queries que comienzan con “validación_*”. Los resultados del query *validación_adjmodalidadxmes.sql* se comparan con los reportes obtenidos de Guatecompras en la sección de estadística en la opción de Adjudicaciones segmentadas por mes, año y modalidad de compra. El objetivo es obtener reportes de validación similares al obtenido en el anexo C.

F. Código desarrollo e implementación de indicadores estadísticos

Link al repositorio de GitHub donde se encuentra el código que fue implementado para crear el servicio que permite crear gráficos. <https://github.com/GobHash/Widgets-Angular.git>

G. Análisis para extracción de datos del nuevo formato de la página de guatecompras.gt

Debido al cambio en la estructura del sitio guatecompras.gt, el programa que se encarga de obtener la información, para luego ser utilizada ingresada a la base de datos de GobHash, ya no funciona de manera adecuada. Es por esto que este análisis tiene como objetivo proponer la serie de pasos que permitan obtener la información tomando en cuenta el nuevo diseño de la página.

El proceso de búsqueda en Guatecompras se encuentra dividido en varias etapas de acuerdo con el proceso de obtención actual, funcional con el esquema anterior hasta antes del 10 de agosto de 2017.

1. Búsqueda de las adjudicaciones. El proceso comienza con acceder a la siguiente página de internet, Figura 177: <http://guatecompras.gt/proveedores/consultaadvprovee.aspx>

Figura 177. Página inicial de búsquedas en Guatecompras

GUATECOMPRAS. *gt*
Sistema de Adquisiciones y Contrataciones del Estado de Guatemala.

Bienvenido a la sección de Búsqueda

Inicio > [Buscar adjudicaciones](#)

Buscar Adjudicaciones

GUATECOMPRAS contiene todas las adjudicaciones por concursos efectuadas, tanto a proveedores con NIT como a proveedores extranjeros sin NIT, desde el 19.Jul.2004, fecha en que comenzó a funcionar el registro de adjudicaciones.

Por favor elija una opción:

- Opción 1: Buscar TODAS las adjudicaciones
- Opción 2: Buscar adjudicaciones a PROVEEDORES con NIT
- Opción 3: Buscar adjudicaciones a EXTRANJEROS sin NIT
- Opción 4: Buscar adjudicaciones a proveedores INHABILITADOS
- Opción 5: Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor
- Opción 6: Buscar por NIT del proveedor
- Opción 7: Buscar por NOG del concurso
- Opción 8: Buscar adjudicaciones con contratos de infraestructura con recursos públicos
- Opción 9: Buscar adjudicaciones por Entidad compradora, segmentadas por modalidad **Nuevo**

www.guatecompras.gt pertenece al Estado de la República de Guatemala
Compatible con Internet Explorer 9 en adelante, Mozilla Firefox y Chrome.

El siguiente paso consiste en seleccionar el tipo de búsqueda que se va a llevar a cabo, en este caso se escoge la opción #5, “*Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor*”. En este punto se nos presentan los diferentes parámetros que se usan para buscar.

Figura 178. Solicitud de parámetros de búsqueda por parte de Guatecompras

GUATECOMPRAS. *gt*
Sistema de Adquisiciones y Contrataciones del Estado de Guatemala.

Bienvenido a la sección de Búsqueda

Inicio > [Buscar adjudicaciones](#)

Buscar Adjudicaciones

GUATECOMPRAS contiene todas las adjudicaciones por concursos efectuadas, tanto a proveedores con NIT como a proveedores extranjeros sin NIT, desde el 19.Jul.2004, fecha en que comenzó a funcionar el registro de adjudicaciones.

Por favor elija una opción:

- Opción 1: Buscar TODAS las adjudicaciones
- Opción 2: Buscar adjudicaciones a PROVEEDORES con NIT
- Opción 3: Buscar adjudicaciones a EXTRANJEROS sin NIT
- Opción 4: Buscar adjudicaciones a proveedores INHABILITADOS
- Opción 5: Buscar por fecha de adjudicación, monto adjudicado o tipo de proveedor
- Opción 6: Buscar por NIT del proveedor
- Opción 7: Buscar por NOG del concurso
- Opción 8: Buscar adjudicaciones con contratos de infraestructura con recursos públicos
- Opción 9: Buscar adjudicaciones por Entidad compradora, segmentadas por modalidad **Nuevo**

Opción 5: Fecha de adjudicación, monto adjudicado o tipo de proveedor (seleccione una o más opciones)

(Fecha) Del Al

(Monto) De A

Escrito sin comas, puede usar símbolo decimal.(Ejemplo: 12500.50)

Tipo de proveedor

www.guatecompras.gt pertenece al Estado de la República de Guatemala
Compatible con Internet Explorer 9 en adelante, Mozilla Firefox y Chrome.

A continuación se escoge el periodo de tiempo del cual queremos obtener las adjudicaciones; el rango de los montos del valor de las adjudicaciones que deseamos consultar; finalmente escogemos el tipo de proveedor que deseamos consultar.

Primero el manejo de las fechas se hace de obtiene en periodos diarios, de esta manera la información es más fácil de manejar, el campo de monto se deja en blanco porque nos interesa consultar cualquier adjudicación sin importar la cantidad de dinero que fue desembolsado. El último parámetro de tipo de proveedor se establece como “Con NIT”, porque se necesita un identificador único para los proveedores y el NIT es eso.

Figura 179. Ejemplo de parámetros de búsqueda

Opción 5: Fecha de adjudicación, monto adjudicado o tipo de proveedor (seleccione una o más opciones)

(Fecha) Del: AI:

(Monto) De: A:

Escrito sin comas, puede usar símbolo decimal.(Ejemplo: 12500.50)

Tipo de proveedor:

Después se hace la consulta a la página y se obtienen los resultados con las adjudicaciones hechas el día solicitado y ya sobre esta lista es que se comienza a obtener la información.

Figura 180. Resultado de la búsqueda con los parámetros de la Figura 179

1 al 50 de 459 adjudicaciones encontradas.
Si desea saber ¿Qué incluye la columna "Monto"? [presione aquí](#)

Fecha de adjudicación ▼	Nombre o razón social	NIT o país	Monto (Q.)	NOG
17.ago..2017	BICSA, SOCIEDAD ANONIMA	32315767	74,017,509.21	5789184
17.ago..2017	DATAFLEX, SOCIEDAD ANONIMA	7127170	433,630.02	6479030
17.ago..2017	COMPAÑIA INTERNACIONAL DE PRODUCTOS Y SERVICIOS SOCIEDAD ANONIMA	4863461	55,000.00	6479030
17.ago..2017	TRITECH GUATEMALA, SOCIEDAD ANONIMA	1049232	203,610.00	6562477
17.ago..2017	PERFECT SYSTEMS SOCIEDAD ANONIMA	2409868K	703.13	6564232
17.ago..2017	VILLATORO,CHINCHILLA,,MARLON,RICARDO	7280122	398,440.80	6526322
17.ago..2017	LLANTAS Y REENCAUCHES SOCIEDAD ANONIMA	5040701	693,603.00	6294499
17.ago..2017	RIVERA,RODAS,,WALTER,ORLANDO	2872996K	877,999.49	6629393
17.ago..2017	MALDONADO,RODAS,,HERNAN,ARTURO	27981681	701,000.00	6053966
17.ago..2017	MARTINEZ,ARDAVIN,,JORGE,ANIBAL	25264079	89,950.00	6695558
17.ago..2017	LIMA,,,GUSTAVO,ADOLFO	60704888	104,400.00	6633552
17.ago..2017	SANCHEZ,TEJEDA,,RAUL,ENRIQUE	5445744	15,500.00	6633552
17.ago..2017	CASTILLO,VASQUEZ,,CARLOS,ALBERTO	829657K	116,000.00	6633552
17.ago..2017	CORONADO,GOMEZ,,MARVIN,ISAIAS	5426359	22,500.00	6633552
17.ago..2017	XUYA,PUNAY,,ROLANDO,	25247492	136,200.00	6633552
17.ago..2017	BOR,GARCIA,,JOSE,DAVID	34376542	15,000.00	6633552
17.ago..2017	RAMIREZ,BARILLAS,,EDDY,ESTUARDO	7408692	20,000.00	6633552
17.ago..2017	RAMOS,PALMA,,JULMAN,ARNULFO	8511632	30,000.00	6633552
17.ago..2017	INVERSIONES CONSTRUCCIONES Y NEGOCIOS EL EXITO, SOCIEDAD ANONIMA	59579706	916,840.94	6614995
17.ago..2017	CALMO,MARTINEZ,,JUAN,DE DIOS	1343319	462,985.00	6554016
17.ago..2017	MERIDA,DE LEON,,ARNOLDO,RENE	2721279	449,800.00	6551319
17.ago..2017	PEREZ,GUTIERREZ,,HELBIN,DANILO	26400839	1,038,000.00	6390552
17.ago..2017	CASTILLO,BARRIOS,,ADRIAN,GUALBERTH	1772996	1,899,000.00	6418201
17.ago..2017	OLIVEROS,RODAS,,JORGE,ARNULFO	26503190	1,258,350.00	6194672
17.ago..2017	SOLUCIONES DE INGENIERIA Y ARQUITECTURA, SOCIEDAD ANONIMA	78994179	3,037,366.49	6194664
17.ago..2017	CABISA, SOCIEDAD ANONIMA	53982746	325,000.00	6705014
17.ago..2017	TELNET SOCIEDAD ANONIMA	22325700	14,995.00	6710131
17.ago..2017	LEIVA,TEZO,,HENRY,OTONIEL	31698891	42,180.00	6710107
17.ago..2017	XICARA,MEJIA,,ERICK,GUMERCINDO	18191851	45,040.00	6664202
17.ago..2017	SUZUKI SOCIEDAD ANONIMA	1198416	84,000.00	6616801
17.ago..2017	CORPORACION NODUM, SOCIEDAD ANONIMA	38504847	5,167.50	6668941
17.ago..2017	KEY CAPITAL INVESTMENTS, SOCIEDAD ANONIMA	82297509	89,975.00	6695701
17.ago..2017	ROBLERO,DIAZ,,ARMANDO,GUSTAVO	4382811	144,000.00	6519563
17.ago..2017	PAPELES ECOLOGICOS SOCIEDAD ANONIMA	1251800K	1,182,000.00	6207375
17.ago..2017	SOCOP,VALIENTE,IC,MILDRED,YESENIA	12434809	67,100.00	6688764
17.ago..2017	FERRO AGRO SANTIAGO, SOCIEDAD ANONIMA	43265553	43,875.00	6695272
17.ago..2017	COFIÑO STAHL Y COMPAÑIA SOCIEDAD ANONIMA	332917	834,290.00	6343406
17.ago..2017	SUMINISTROS INFORMATICOS, SOCIEDAD ANONIMA	89771125	25,600.00	6673643
17.ago..2017	P.V.C. GERFOR GUATEMALA, SOCIEDAD ANONIMA	16897560	505,909.08	4233891
17.ago..2017	KAMIL SOCIEDAD ANONIMA	19554648	37,137.90	6692575
17.ago..2017	JAYOR MEDICAL GUATEMALA, SOCIEDAD ANONIMA	87537931	23,230.00	6692575
17.ago..2017	RIVERA,ZUÑIGA,,JUAN,HERMEZ	11367121	7,189.00	6692575
17.ago..2017	PROYECTOS Y SUMINISTROS INDUSTRIALES SOCIEDAD ANONIMA	5247969	182,000.00	6617042
17.ago..2017	OFFYMARKET, SOCIEDAD ANONIMA	29010438	58,296.00	6683401
17.ago..2017	E. CORP, SOCIEDAD ANONIMA	25240447	513,100.00	6259685
17.ago..2017	FORMULARIOS STANDARD SOCIEDAD ANONIMA	1532227	44,000.00	6665772
17.ago..2017	DROGUERIA AMERICANA SOCIEDAD ANONIMA	5974488	10,625.00	6192637
17.ago..2017	REDES Y TECNOLOGIA SOCIEDAD ANONIMA	12770388	85,120.00	6709885
17.ago..2017	DE LEON,MOGOLLON,,GABRIEL,AMILCAR	5965330	368,696.00	6267998
17.ago..2017	MARTINEZ,ARDAVIN,,JORGE,ANIBAL	25264079	80,000.00	6695817

Ir a la página: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

Una vez tenemos la lista de adjudicaciones el siguiente paso es obtener la información necesaria acerca de estas. Eso le corresponde a la siguiente etapa del proceso.

2. Obtención de información

a. Información del concurso. Para esta etapa se comienza utilizando la lista con las adjudicaciones de determinado día (Figura 180), cada adjudicación en Guatecompras cuando con un identificador único y a este se le llama número de operación Guatecompras (NOG). Al hacer clic sobre cualquier fila de la columna NOG, el sitio nos dirige hacia la página donde se encuentra la información de esa adjudicación (Figura 181).

Figura 181. Información general de una adjudicación hecha en Guatecompras

Detalle de concurso

NOG: 5789184

Descripción: PROYECTO: MEJORAMIENTO CARRETERA RN-16, TRAMO: EL BOQUERÓN - CHIQUIMULLA LONGITUD APROXIMADA: 38.0 KM.

Modalidad: Licitación Pública (Art. 17 LCE)

Categoría: Construcción y materiales afines

Tipo de concurso: Público

Entidad: MINISTERIO DE COMUNICACIONES, INFRAESTRUCTURA Y VIVIENDA

Tipo de Entidad: Administración Central

Unidad compradora: COMPRAS DGC

Recepción de ofertas: **Sólo en papel.** Todas las ofertas deben recibirse en papel y no se permite recibirlas en forma electrónica.

Fecha de finalización: 17.agosto.2017 Hora:07:54:35 p.m.

Fecha de publicación: 09.junio.2017 Hora:08:54:38 p.m.

Fecha de presentación de ofertas: 24.julio.2017 Hora:11:00:00 a.m.

Fecha de cierre de recepción de ofertas: 24.julio.2017 Hora:11:30:00 a.m.

Estatus: Terminado adjudicado

No	Operaciones Disponibles
1	Dar seguimiento
2	Ver inconformidades (este concurso recibió 1 inconformidad)
3	Presentar inconformidad (aquí puede ingresar quejas relacionadas con el concurso)
4	Imprimir (genera archivo en formato PDF para imprimir)
5	Exportar (toma los Tipos de Producto y los copia en un archivo de formato XLS)

La información que se obtiene en esta vista inicial (Figura 181) es la general. En estos campos también se puede acceder a la información de la entidad que se encarga a la cual pertenece la unidad compradora, es decir la entidad más general y la unidad específica de dicha entidad que hace la adjudicación.

Figura 182. Información general de una adjudicación

Detalle de concurso

NOG: 5789184

Descripción: PROYECTO: MEJORAMIENTO CARRETERA RN-16, TRAMO: EL BOQUERÓN - CHIQUIMULLA LONGITUD APROXIMADA: 38.0 KM.

Modalidad: Licitación Pública (Art. 17 LCE)

Categoría: Construcción y materiales afines

Tipo de concurso: Público

Entidad: MINISTERIO DE COMUNICACIONES, INFRAESTRUCTURA Y VIVIENDA

Tipo de Entidad: Administración Central

Unidad compradora: COMPRAS DGC

Recepción de ofertas: **Sólo en papel.** Todas las ofertas deben recibirse en papel y no se permite recibirlas en forma electrónica.

Fecha de finalización: 17.agosto.2017 Hora:07:54:35 p.m.

Fecha de publicación: 09.junio.2017 Hora:08:54:38 p.m.

Fecha de presentación de ofertas: 24.julio.2017 Hora:11:00:00 a.m.

Fecha de cierre de recepción de ofertas: 24.julio.2017 Hora:11:30:00 a.m.

Estatus: Terminado adjudicado

Una vez ya se han recolectado los datos anteriores (Figura 182) se puede pasar a la siguiente pestaña, la de *Tipos de anexo* (Figura 181), cabe mencionar que en esta página la mayoría de documentos son PDF,

Word, Excel. Por lo cual no tiene mucho sentido obtenerlos debido a que se necesitan herramientas más sofisticadas para analizar este tipo de archivos y eso se encuentra fuera del alcance de este análisis. El campo que valdría la pena obtener es el de “Criterio de calificación”, esto porque adicional al documento adjunto se pueden obtener los valores directamente de la tabla presentada (Figura 183). Nuevamente se debe mencionar hay adjudicaciones que debido a su antigüedad no cuentan con este campo, por lo cual se debe considerar esto para evitar generar errores, por esta causa, al momento de ejecutar el programa.

Figura 183. Criterio de calificación de una adjudicación.

Tipo de documento(s)	Documento(s) adjunto(s) exitosamente
Criterio de calificación	5789184@CRITERIOS DE CALIFICACION.pdf

Listado de Criterios de Calificación	
Nombre de Criterio	Ponderación
COSTO OFERTADO	50%
EXPERIENCIA DEL OFERENTE	10%
EXPERIENCIA DEL SUPERINTENDENTE	10%
PERSONAL PROFESIONAL (ESPECIALISTAS) PARA EL ASEGURAMIENTO DEL CONTROL DE CALIDAD DE LA OBRA	10%
EQUIPO PARA EL ASEGURAMIENTO DEL CONTROL DE CALIDAD DE LA OBRA	5%
MAQUINARIA Y EQUIPO MINIMO	15%
Total:	100%

La siguiente pestaña es *Tipos de producto*, en esta detallan qué tipo de bienes o servicios se adquirieron, su descripción y la cantidad (Figura 184).

Figura 184. Tipos de producto de una adjudicación

Tipos de Producto			
Nombre	Cantidad	Unidad de Medida	Acciones
PARA LA EJECUCION DEL PROYECTO: MEJORAMIENTO CARRETERA RN-16, TRAMO: EL BOQUERÓN - CHIQUIMULILLA LONGITUD APROXIMADA: 38.0 KM.	1	MEJORAMIENTO	

En la columna de acciones al hacer clic se muestran más detalles de los productos en la tabla (Figura 185).

Figura 185. Detalle de un producto

Características del Producto	
Nombre	PARA LA EJECUCION DEL PROYECTO: MEJORAMIENTO CARRETERA RN-16, TRAMO: EL BOQUERÓN - CHIQUIMULILLA LONGITUD APROXIMADA: 38.0 KM.
Código del tipo de producto:	033100000000
Características:	PROYECTO: MEJORAMIENTO CARRETERA RN-16, TRAMO: EL BOQUERÓN - CHIQUIMULILLA LONGITUD APROXIMADA: 38.0 KM.
Unidad de medida:	MEJORAMIENTO
Tipo de obligatoriedad:	Obligatorio siempre.

A continuación se encuentra la pestaña de “*Proceso de adjudicación*”, aquí se puede encontrar la información de los integrantes de la junta calificadora (Figura 186), es decir las personas que se encargaron de analizar las ofertas presentadas y escoger la mejor.

Figura 186. Integrantes de la junta de calificación de una adjudicación

Proceso de Adjudicación: Integrantes de la Junta de Calificación		
NIT	Nombre o razón social	Entidad Responsable
5046130	FLORES,VALENZUELA,,OBDULIO,DE JESUS	MINISTERIO DE COMUNICACIONES INFRAESTRUCTURA Y VIVIENDA
18004660	ZAVALA,ESCARATE,,SANDRA,LETICIA	MINISTERIO DE COMUNICACIONES INFRAESTRUCTURA Y VIVIENDA
3750337	MIRANDA,TURUY,,JULIO,ROMEO	MINISTERIO DE COMUNICACIONES INFRAESTRUCTURA Y VIVIENDA

El listado de empresas que presentaron ofertas también aparece para fines de comparación (Figura 187).

Figura 187. Listado de oferentes de una adjudicación

Proceso de Adjudicación: Listado de Oferentes

NIT o país	Nombre o razón social	Entidad Afianzadora	Monto total ofertado	Monto Seguro de Caucción
5631858	DRAGADOS Y CONSTRUCCIONES SOCIEDAD ANONIMA	ASEGURADORA FIDELIS, S.A.	69,976,015.610	3,750,000.000
325996	CONSTRUCTORA NACIONAL, SOCIEDAD ANONIMA	SEGUROS PRIVANZA, S.A.	66,414,545.130	3,750,000.000
32315767	BICSA, SOCIEDAD ANONIMA	SEGUROS PRIVANZA, S.A.	74,017,509.210	4,250,000.000

Aunque, lo más importante de esta pestaña es la lista de empresas a las que fue adjudicado el concurso y el monto que se pagó a cada una (Figura 188).

Figura 188. Proveedores adjudicados de una adjudicación

Proceso de Adjudicación: Proveedor Adjudicado

NIT o país	Nombre o razón social	Contrato	Monto
32315767	BICSA, SOCIEDAD ANONIMA		74,017,509.210
		Total	74,017,509.210

La última pestaña es la de “Historial de acciones” y esta simplemente se presenta una tabla con el listado de acciones relevantes a la adjudicación, desde su creación hasta su finalización (Figura 189).

Figura 189. Historial de acciones de una adjudicación

Historial de acciones

No	Acción	Descripción
12	FINALIZACIÓN: se publicó la finalización del concurso	FINALIZACION DEL CONCURSO Estatus anterior: En evaluación . Estatus actual: Terminado adjudicado . Proveedores adjudicados: 1) (NIT: 32315767) BICSA, SOCIEDAD ANONIMA - Q74017509.21 5789184@RESOLUCION SA- 584-2017 DGC-032-2016-C.pdf (194KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 17 agosto 2017 19:54:35
11	DOCUMENTOS: se publicaron documentos sobre el concurso	Comentario: PUBLICACION DEL OFICIO POR LA JUNTA Y ACTA DE EVALUACION TECNICA Y ADJUDICACION No. 061-2017... Se agregaron varios documentos. al NOG 5789184 5789184@OFICIO POR LA JUNTA PUBLICACION ACTA No. 061-2017.pdf (29KB) 5789184@ACTA DE EVALUACION TECNICA Y ADJUDICACION No. 061-2017.pdf (215KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 04 agosto 2017 22:14:35
10	OFERENTES: se publicó el listado de oferentes del concurso. El pago debe efectuarse dentro de los 30 días hábiles	Publicación del listado de oferentes del NOG 5789184 Listado de Oferentes: - DRAGADOS Y CONSTRUCCIONES SOCIEDAD ANONIMA. - CONSTRUCTORA NACIONAL, SOCIEDAD ANONIMA. - BICSA, SOCIEDAD ANONIMA. Por: VEGA,MEJICANOS,,JULIO,ANTONIO 26 julio 2017 20:12:41
9	Se publicó el acta de apertura	Se publicó el acta de apertura del NOG 5789184 listado de integrantes de la Junta: - (5046130) FLORES,VALENZUELA,,ORIBULLO,DE JESUS. - (16004660) ZAVALA,ESCARATE,,SANDRA,LETICIA. - (3750337) MIRANDA,TURUY,,JULIO,ROMEO. 5789184@OFICIO POR LA JUNTA PUBLICACION ACTA 043-2017.pdf (34KB) 5789184@ACTA DE RECEPCION DE OFERTAS Y APERTURA DE PLICAS 043-2017.pdf (159KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 26 julio 2017 20:07:46
8	ESTATUS (AUTOMÁTICO): hubo un cambio automático del estatus del concurso porque se alcanzó la fecha máxima para presentar ofertas Los precios ofertados deben publicarse a más tardar al día siguiente La adjudicación debe efectuarse dentro de los 5 días	Estatus anterior: Vigente . Estatus actual: En evaluación Por: Sistema GUATECOMPRAS 24 julio 2017 11:30:01
7	DOCUMENTOS: se publicaron documentos sobre el concurso	Comentario: PUBLICACION DEL OFICIO No. 1603 PUBLICACION DEL BOLETIN No. 3 DE LICITACION No. DGC-032-2017-C.. Se agregaron varios documentos. al NOG 5789184 5789184@OFICIO No. 1603 PUBLICACION DEL BOLETIN No. 3 DE LICITACION No. DGC-032-2017-C.pdf (42KB) 5789184@BOLETIN No. 3 DE LICITACION No. DGC-032-2017-C.pdf (107KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 20 julio 2017 18:38:11
6	DOCUMENTOS: se publicaron documentos sobre el concurso	Comentario: Publicación del Oficio No. 1590 y del Boletín No. 2 Licitación No. DGC-032-2016-C.. Se agregaron varios documentos. al NOG 5789184 5789184@BOLETIN No. 2 DE LICITACION No. 032-2016-C.pdf (60KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 18 julio 2017 18:24:25
5	DOCUMENTOS: se publicaron documentos sobre el concurso	Comentario: Publicación del Oficio No. 1551 y del Boletín No. 1 Licitación No. DGC-032-2017-C.. Se agregaron varios documentos. al NOG 5789184 5789184@Oficio No. 1551 Publicación del Boletín No. 1 de la Licitación DGC-032-2017-C.pdf (41KB) 5789184@Boletín No. 1 de la Licitación DGC-032-2017-C.pdf (299KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 14 julio 2017 16:52:03
4	DOCUMENTOS: se publicaron documentos sobre el concurso	Comentario: Publicación del Oficio No. 1525 y del Apéndice No. 1 Licitación No. DGC-032-2017-C.. Se agregaron varios documentos. al NOG 5789184 5789184@Oficio No. 1525 Publicación de Apéndice No. 1 Licitación No. DGC-032-2017-C.pdf (37KB) 5789184@Apéndice No. 1 Licitación No. DGC-032-2017-C.pdf (210KB) Por: VEGA,MEJICANOS,,JULIO,ANTONIO 12 julio 2017 18:12:54
		El concurso público con NOG '5789184' fue publicado exitosamente despues de haber finalizado el tiempo del proyecto de base. La creación con origen desde Proyecto de bases fue exitosa. Se crearon 1 productos Se agregaron 18 documentos como anexos de las bases. Criterio de calificación 5789184@CRITERIOS DE CALIFICACION.pdf (329KB) Proyecto de contrato 5789184@ANEXO 8 MODELO DE CONTRATO.pdf (716KB) Modelo de oferta (formulario) 5789184@ANEXO 4 MODELO DE OFERTA.pdf (55KB) Modelo de oferta (formulario) 5789184@ANEXO 5 MODELO DE CUADRO DE OFERTA.pdf (28KB) Modelo de oferta (formulario) 5789184@ANEXO 6 MODELO DE INTEGRACION DE COSTO UNITARIO.pdf (47KB) Modelo de oferta (formulario) 5789184@ANEXO 7 MODELO DE CARTA DE ACEPTACION DEL CARGO.pdf (28KB) Estudios, diseños o planos 5789184@ANEXO 1 DESCRIPCION DEL PROYECTO.pdf (40KB)

b. Información del comprador

En este punto ya se tiene toda la información relevante de la adjudicación como tal, y se puede pasar a obtener la información de la entidad que hizo el concurso. En la Figura 181 se puede apreciar que los campos de Entidad y Unidad compradora tienen links, estos dirigen hacia la información detallada de cada una de estas.

En el detalle de la entidad compradora (Figura 190) se obtienen los siguientes datos: nombre, NIT, origen de los fondos, departamento y municipio. Al compararse con los de la unidad compradora (Figura 191) es evidente que los datos son los mismos, la información adicional que queremos obtener en este caso es el nombre.

Figura 190. Página con los detalles de la entidad compradora

GUATECOMPRAS.gt
Sistema de Adquisiciones y Contrataciones del Estado de Guatemala

Inicio > Registro Compradores > Detalle entidad

Detalle de entidad compradora

Datos registrados actuales

Entidad:	MINISTERIO DE COMUNICACIONES, INFRAESTRUCTURA Y VIVIENDA
NIT:	3440915
Tipo:	Administración Central
Entidad origen de los fondos:	11130013 - MINISTERIO DE COMUNICACIONES, INFRAESTRUCTURA Y VIVIENDA
Departamento:	GUATEMALA
Municipio:	GUATEMALA
Dirección:	8 Av. y 15 Calle Zona 13
Teléfonos:	22234000
Números de Fax:	22606254
Apartado Postal:	[--No Especificado--]
Páginas Web:	www.civ.gob.gt
Direcciones de Correo:	comunicaciones@comunicaciones.gob.gt , edgargomez@comunicaciones.gob.gt
Información de SAT:	Para consultar haga clic aquí

Figura 191. Página con los detalles de la unidad compradora

GUATECOMPRAS.gt
Sistema de Adquisiciones y Contrataciones del Estado de Guatemala

Inicio > Registro Compradores > Detalle entidad

Detalle de entidad compradora

Datos registrados actuales

Entidad:	MINISTERIO DE COMUNICACIONES, INFRAESTRUCTURA Y VIVIENDA
NIT:	3440915
Unidad Compradora:	COMPRAS DGC
Departamento:	GUATEMALA
Municipio:	GUATEMALA
Dirección:	Finca Nacional La Aurora Salon 7 Zona 13
Teléfonos:	24721019, 24720441 al 3
Números de Fax:	24720534
Apartado Postal:	[--No Especificado--]
Páginas Web:	www.caminos.gob.gt
Direcciones de Correo:	gamartin@caminos.gob.gt
Información de SAT:	Para consultar haga clic aquí

c. Información del proveedor

Es importante hacer la distinción entre los distintos tipos de proveedores que pueden encontrarse en Guatecompras, personas individuales o empresas. Hay que datos que en ambos casos se pueden obtener NIT y nombre.

En la información de la empresa (Figura 192) hay variaciones importantes, la primera es la información del domicilio comercial a veces se encuentra disponible y otras no. Es por esta situación que se ha tomado la decisión de utilizar los datos presentados en el apartado de Domicilio Fiscal, esto es el municipio y el departamento. El siguiente punto variable es la actividad económica, en este caso se ha optado por marcarla como “nula” cuando esté ausente. La inscripción en el registro mercantil es otro de los datos que varía, puede

ser provisional o definitiva, en este caso en particular si ambas están presentes se toma la definitiva, en caso contrario se toma la que esté disponible. Finalmente solo resta obtener el nombre de los representantes legales.

Figura 192. Información de proveedor (empresa)

Registro de Proveedores - BICSA, SOCIEDAD ANONIMA

Identificación	
(Datos recibidos de la SAT el: 19.jul..2017 16:30:49)	
CUI:	
Nombre o razón social:	BICSA, SOCIEDAD ANONIMA
Número de Identificación Tributaria (NIT):	32315767
Nombre comercial 1:	BICSA
Actividad principal y Principales trabajos realizados	
Actividad principal:	.
Estatus Actual del Proveedor (lea antes ¿Qué significa estar inscrito, habilitado, adjudicado y con contraseña?)	
Estatus en Guatecompras	
Habilitado o Inhabilitado:	HABILITADO
Adjudicado o No adjudicado:	ADJUDICADO
Participa o no en Contrato Abierto:	NO PARTICIPA (no tiene productos en el catálogo)
Con o Sin contraseña:	CON CONTRASEÑA
Estatus en la SAT(19.jul..2017 16:30:49)	
Estatus del RTU	ACTIVO
Solvencia del NIT en RTU	SOLVENTE
Domicilio Fiscal (Datos recibidos de la SAT el: 19.jul..2017 16:30:49)	
Departamento:	GUATEMALA
Municipio:	SANTA CATARINA PINULA
Dirección:	CARRETERA A EL SALVADOR Z.0 KILOMETRO 14.7 OFIBODEGA 303 CENTRO EMPRESARIAL GRAN PLAZA PUERTA PARADA
Teléfonos:	6661-7465
Números de fax:	[--No Especificado--]
Domicilio Comercial	
Páginas Web:	
Direcciones de Correo Electrónico:	CORPAPRISA@ITELGUA.COM
Departamento:	GUATEMALA
Municipio:	GUATEMALA
Dirección:	3A. CALLE "A" NO. 3-41 ZONA 10
Teléfonos:	2334-6205
Números de fax:	2331-5727

Figura 193. Información de proveedor (persona individual)

The screenshot shows the 'Detalle proveedor' page for Antonio Enrique Carrillo Batres. The page is titled 'Registro de Proveedores - BATRES,CARRILLO,,ENRIQUE,ANTONIO'. It contains several sections:

- Identificación:** CUI: 1677272490101; Nombre o razón social: BATRES,CARRILLO,,ENRIQUE,ANTONIO; Número de Identificación Tributaria (NIT): 130400; Nombre comercial 1: BATRES & BATRES, INGENIEROS CONSTRUCTORES; Nombre comercial 2: RESIDENCIALES DE SAN FRANCISCO.
- Estatus Actual del Proveedor:** Estatus en Guatecompras: HABILITADO; Adjudicado o No adjudicado: NO ADJUDICADO; Participa o no en Contrato Abierto: NO PARTICIPA (no tiene productos en el catálogo); Con o Sin contraseña: SIN CONTRASEÑA; Estatus en la SAT(26.ene..2017 03:55:42): Estatus del RTU: ACTIVO; Solvencia del NIT en RTU: SOLVENTE.
- Datos de inscripción:** Tipo de organización: INDIVIDUAL; Actividad Económica: INGENIERO (EN TODAS SUS RAMAS).
- Inconformidades e Inhabilitaciones:** Cantidad de INCONFORMIDADES presentadas: [--NINGUNA--]; Cantidad de INHABILITACIONES recibidas: [--NINGUNA--].

H. Ingeniería inversa de Facebook

1. Tecnologías utilizadas por Facebook

- Linux
- Apache
- MySQL
- PHP
- Memcache
- Thrift protocol
- Cassandra
- HipHop for PHP

2. Funcionalidades

- Página principal: Facebook carga publicaciones a la página inicial dependiendo de las capacidades del equipo en donde se está utilizando.
- Perfil: Se carga la información cada vez que se ingresa a la página
- Facebook compila el lenguaje de PHP para que sea rápido

Seguridad

- Facebook utiliza autenticación mediante JWT y OAuth.

XIII. GLOSARIO

1. **Materialized view:** En computación un Materialized view es un objeto de base de datos que contiene el resultado de una consulta, Por ejemplo, puede ser una copia local de los datos localizados de forma remota, o puede ser un subconjunto de las filas y / o columnas del resultado de unir una tabla con otra.
2. **Backend:** Es el conjunto de servicios que sirven para generar y/o guardar información a nivel de infraestructura en una aplicación para soportar los servicios de frontend. Generalmente, el usuario no puede ver estos servicios ya que incluyen servidores, bases de datos, seguridad, etc. Que sirven para comunicarse con el frontend.
3. **Commit:** En contexto de control de versiones de software, un commit es la acción de guardar los cambios a un conjunto de archivos.
4. **Solicitud (Request en inglés):** Se refiere al intercambio de información según el protocolo de comunicación web http realizada por lo general por un navegador web.
5. **Repositorio:** en contexto de control de versiones de software, es el lugar donde se almacena el código.
6. **Frontend:** Es la parte del desarrollo web que se dedica a la parte frontal de una plataforma web, esto incluye su diseño desde la estructura del sitio hasta los estilos como colores, fondos, tamaños, animaciones y efectos.
7. **Branch:** Es la idea de trabajar diferentes versiones de la raíz de un repositorio concurrentemente.
8. **Framework:** Es un conjunto de conceptos, prácticas y criterios para enfocar alguna problemática particular que sirve como referencia, para analizar y resolver problemas del mismo tipo o similar.
9. **Repositorio:** Es un lugar lógico en donde se almacena información estructurada o no estructurada.
10. **Rendering:** Proceso de generar una vista mediante HTML y JS.
11. **Endpoint:** Es una ruta expuesta que tiene como fin la representación de una funcionalidad.
12. **Single Page Application:** Es un sitio web en donde su ejecución es mediante una sola página con el proposito de dar una experiencia más fluida a los usuarios.
13. **Diagrama de flujo:** Es la representación gráfica de un algoritmo o proceso.
14. **Dashboard;** Es un conjunto de gráficos.
15. **Feed;** Es un conjunto de publicaciones.
16. **Post:** Es una publicación.
17. **Automatización:** Es el proceso de permitir la integración mediante comandos de ejecución, sin necesidad de configuración.