

**UNIVERSIDAD DEL VALLE DE
GUATEMALA**
Facultad de Ingeniería
Departamento de Computación



**Interfaz móvil del módulo de servicios de un
manejador de relaciones de clientes**

Helder Ottoniel Gramajo López

**Guatemala
2010**

Interfaz móvil del módulo de servicios de un manejador de relaciones de clientes

**UNIVERSIDAD DEL VALLE DE
GUATEMALA**
Facultad de Ingeniería
Departamento de Computación



**Interfaz móvil del módulo de servicios de un
manejador de relaciones de clientes**

Modelo de trabajo de Graduación presentado por Helder Ottoniel Gramajo López para optar al grado académico de Licenciado en Ingeniería de Ciencias de la Computación.

Guatemala
2010

Vo. Bo. :

(f) _____
(Ing. Heberth Campos)

Tribunal Examinador:

(f) _____
(Ing. Bidkar Pojoy)

(f) _____
(Ing. Jenniffer Guzmán)

(f) _____
(Ing. Heberth Campos)

Fecha de aprobación: Guatemala, 9 de diciembre de 2010.

PREFACIO

En febrero de 2010 se inició el proyecto de una aplicación móvil como prácticas profesionales requeridas por la Universidad del Valle de Guatemala. El propósito de dichas prácticas era emplear el conocimiento adquirido durante la carrera para solucionar una necesidad particular. Además ayudaría al estudiante a ampliar su visión de cómo desarrollar un sistema en el mundo laboral. Las prácticas se realizaron en 1800 Software, una empresa que se dedica al comercio electrónico y a proveer soluciones de Cloud Computing, lo cual permitió que el proyecto estuviera relacionado con este tema.

Como se mencionó anteriormente, el proyecto consistió en el desarrollo de una aplicación móvil; este tipo de temas está tomando mucha importancia en el mercado actual gracias a la aparición de nuevas tecnologías más accesibles.

A través del proyecto se adquirieron nuevos conocimientos en temas poco explorados y una visión más amplia del desarrollo de aplicaciones y nuevas formas para compartir la información. Es importante mencionar que este proyecto forma parte de uno aún más grande destinado a construir una solución enfocada a mejorar el servicio al cliente de las empresas. La otra parte fue el desarrollo de un sistema para un servidor con el fin de ayudar a mejorar las relaciones entre los clientes y las empresas. Como referencia se puede consultar el trabajo de graduación: “Portal y servicio web del módulo de servicios de un sistema para el manejo de relaciones con los clientes” de Dennis Aldana, estudiante de la carrera de Ciencias de la Computación en la Universidad del Valle de Guatemala.

Quedo profundamente agradecido con: Dios, por ser la fuente de vida. Mis padres, por ser mis primeros maestros que me enseñaron la senda correcta. Mis hermanos, por su apoyo incondicional. Fundación Juan Bautista Gutiérrez, por proveer los medios para desarrollar mi carrera universitaria. 1800 Software, por permitirme poner en práctica mis conocimientos. Mis compañeros de clases, por su amistad y apoyo.

ÍNDICE

Prefacio	V
Índice	VI
Lista de tablas	XI
Lista de ilustraciones.....	XII
Resumen.....	XIV
I. Introducción	1
A. Antecedentes	1
B. Descripción general del trabajo.....	1
C. Definición del problema.....	2
D. Propuesta de solución	2
E. Alcance y límites.....	3
F. Limitaciones.....	3
G. Justificación	3
H. Objetivos	4
1. General.....	4
2. Específico.....	4
II. Marco teórico	5
A. CRM.....	5
1. Definición de un CRM.....	5
2. Lugares donde se puede utilizar CRM.....	5
3. La importancia de utilizar un CRM en una empresa.....	5
B. Servicio web.....	6
1. Definición de un servicio web	6
2. Arquitectura	6
3. XML (eXtensible Markup Language).....	7

4. WSDL (Web Services Description Language)	7
5. SOAP (Simple Object Access Protocol).....	8
6. UDDI (Universal Description, Discovery, and Integration)	8
7. Plataformas	9
C. Interfaz móvil de usuario	10
1. La tendencia actual de uso de dispositivos móviles.....	10
2. La importancia de uso de aplicaciones móviles en el área de trabajo.....	10
3. Principios para el desarrollo de aplicaciones móviles.....	10
D. Comparación entre el hardware de un Smartphone y de una computadora	11
E. Sistema de localización.....	11
1. GPS (Global Positioning System).....	11
2. Geocoding y Reverse Geocoding.....	12
F. Desarrollo de software	12
1. Modelo de desarrollo de procesos.....	12
2. Modelo cascada.....	12
3. Modelo incremental	13
4. Modelo RAD.....	14
G. Cifrado	15
1. Definición	15
2. Algoritmos de cifrados.....	15
3. DES (Data Encryption Standard).....	15
4. Triple-DES	16
H. Plataforma y herramientas de desarrollo.....	16
1. Framework .NET	16
2. Framework .NET Compact.....	16
3. Visual Studio.....	17
4. Windows Mobile.....	17

5. Versión de Windows Mobile	17
6. Microsoft SharePoint	18
III. Metodología	19
IV. Análisis	21
A. Introducción	21
B. Descripción general	21
1. Perspectiva del sistema	21
2. Características del sistema	22
3. Descripción de los usuarios del sistema.....	22
4. Ambiente de ejecución.....	22
5. Restricciones de diseño e implementación	22
6. Documentación de usuario.....	23
7. Supuestos y dependencias	24
C. Características del sistema	24
1. Autenticación del usuario con el CRM.....	24
2. Consulta de la lista asignada al usuario en el CRM	24
3. Desplegar la información detallada de una tarea	25
4. Editar la información relacionada con una tarea.....	26
5. Mostrar la lista de sub-tareas de una tarea seleccionada.....	26
6. Asignar una sub-tarea a otro usuario.....	27
7. Localización del usuario	27
D. Requerimientos de interfaces externas.....	28
1. Interfaz de usuario.....	28
2. Interfaz de hardware.....	30
3. Interfaz con software.....	30
4. Interfaces de comunicación.....	31
E. Otros requerimientos no funcionales	35

1. Requerimientos de rendimiento	35
2. Requerimientos de seguridad del hardware	35
3. Requerimiento de seguridad de información	36
4. Cualidades del software	36
V. Diseño	38
A. Diseño arquitectónico	38
B. Diseño de componentes	38
1. Introducción	38
2. Autenticación y conexión	38
3. Lista de tareas	46
4. Visor de contenidos.....	49
5. Editor de contenidos.....	52
6. Configuración del visor y editor	55
7. Localización.....	58
C. Interfaz de usuario.....	60
1. Introducción	60
2. Pantalla de autenticación	62
2. Pantalla de configuración.....	63
4. Visor de contenidos.....	66
5. Editor de contenido	67
6. Pantalla de configuración del visor.....	68
7. Pantalla de localización.....	69
7. Archivos de configuración	70
VI. Resultados y discusión	76
A. La aplicación móvil.....	76
B. La documentación del proyecto	77
C. La documentación de usuario	77

D.	Consumo promedio en Internet.....	77
1.	Medición del consumo en el teléfono	77
2.	Consumo mensual.....	80
3.	Valor de MB descargado.....	80
VII.	Conclusiones	82
VIII.	Recomendaciones	83
IX.	Bibliografía	84
X.	Apéndices.....	87
A.	Algoritmos de cifrado para envío de credenciales	87
2.	Componentes.....	87
3.	Pasos para el cifrado	87
4.	Pasos para el descifrado	88
B.	Manual del usuario.....	89
1.	Primeros Pasos	89
C.	Diagramas UML	94
1.	Diagrama de casos de uso	94
2.	Diagramas de actividades.....	95
3.	Diagrama de componentes	102
D.	Plantilla utilizada para realizar el análisis	103
XI.	Glosario.....	110

LISTA DE TABLAS

Tabla 1 Versiones de Windows Mobile	17
Tabla 2 Tipo de respuestas del servidor en archivo XML	32
Tabla 3 Parámetros de petición de mapas	34
Tabla 4 Tipos de respuesta del servicio de geocoding	35
Tabla 5 Posibles acciones en la clase SourceAdaptor	39
Tabla 6 Posibles resultados de proceso de parsear el archivo XML	40
Tabla 7 Posibles modos de la pantalla de configuración	43
Tabla 8 Posible vista de la lista de tareas	48
Tabla 9 Posibles vista del visor de contenido	50
Tabla 10 Posibles vistas del editor de contenido	53
Tabla 11 Tipos de campos de un perfil	55
Tabla 12 Campos de un perfil	72
Tabla 13 Tipos de campos	73
Tabla 14 Campos del archivo de configuración de la aplicación	74
Tabla 15 Información del archivo de configuración de mapas	75
Tabla 16 Tamaño de la petición de lista de tareas	78
Tabla 17 Tamaño de otras peticiones	79
Tabla 18 Otras mediciones	79
Tabla 19 Consumo diario de MB	80
Tabla 20 Planes de Internet en Claro	81
Tabla 21 Planes de Internet en Movistar	81
Tabla 22 La tabla para el algoritmo de codificación	87

LISTA DE ILUSTRACIONES

Ilustración 1 Componentes de un servicio web	7
Ilustración 2 Arquitectura de un servicio web	9
Ilustración 3 Modelo cascada.....	13
Ilustración 4 Modelo incremental	13
Ilustración 5 Modelo RAD.....	14
Ilustración 6 Una captura de pantalla de Microsoft SharePoint.....	18
Ilustración 7 Los incrementos del sistema	20
Ilustración 8 Arquitectura de ejecución	21
Ilustración 9 Esquema de navegación en CUI	61
Ilustración 10 Una captura de la pantalla de autenticación	62
Ilustración 11 Una captura de la pantalla de configuración	63
Ilustración 12 Una captura de la pantalla de tareas.....	64
Ilustración 13 Una captura de la pantalla de sub-tareas.....	65
Ilustración 14 Una captura de la pantalla de visor de contenido.....	66
Ilustración 15 Una captura de la pantalla de editor de contenido	67
Ilustración 16 Una captura de la pantalla de configuración del visor	68
Ilustración 17 Una captura de la pantalla de localización.....	69
Ilustración 18 Tamaño de petición de lista de tareas	79
Ilustración 19 Pantalla de inicio de sesión.....	89
Ilustración 20 Pantalla de configuración.....	90
Ilustración 21 Lista de tareas	90
Ilustración 22 Pantalla de localización.....	91
Ilustración 23 Visor de tareas.....	91
Ilustración 24 Lista de sub-tareas.....	92
Ilustración 25 Pantalla de nuevas sub-tareas.....	92
Ilustración 26 Pantalla de configuración del visor.....	93
Ilustración 27 Editor de tareas	93
Ilustración 28 Diagrama de casos de uso	94
Ilustración 29 Diagrama de actividades - Autenticación con el CRM	95
Ilustración 30 Diagrama de actividades - Consulta de lista de tareas	96
Ilustración 31 Diagrama de actividades - Mostrar una tarea.....	97
Ilustración 32 Diagrama de actividades - Editar tareas.....	98
Ilustración 33 Diagrama de actividades - Consulta de sub-tareas.....	99

Ilustración 34 Diagrama de actividades - Crear una sub-tarea	100
Ilustración 35 Diagrama de actividades - Localización	101
Ilustración 36 Diagrama de componentes	102

RESUMEN

Este trabajo presenta el desarrollo de una aplicación móvil que se utiliza para consulta y actualización de la información de las tareas que almacena el módulo de servicios de un manejador de relaciones con el cliente. El proyecto consistió en definir el propósito para desarrollar la aplicación, la toma de requerimientos y continuar con el ciclo de vida de un proyecto de software: análisis, diseño, implementación y pruebas.

Al final del trabajo se obtuvo una aplicación desarrollada en Framework .NET Compact 3.0 para teléfonos celulares con sistema operativo Windows Mobile 6, que tiene la capacidad de conectar y autenticar al usuario con el servidor de CRM, obtener la lista de tareas asignadas, consultar y actualizar la información de una tarea específica, crear y asignar una tarea a otro usuario, y obtener la dirección y el mapa de la posición del usuario utilizando el GPS conjuntamente con los servicios de Google Maps.

I. INTRODUCCIÓN

A. Antecedentes

En los últimos años de la carrera, la Universidad de Valle requiere que los estudiantes realicen 400 horas de prácticas en una empresa relacionada con un tema de estudio de la carrera, con el fin de que el estudiante pueda poner en práctica todos los conocimientos adquiridos en su proceso de aprendizaje.

En este caso, se escogió realizar dichas prácticas en la empresa 1800 Software, una empresa que trabaja en el comercio electrónico; siendo un punto intermedio entre los desarrolladores de software y las empresas que necesitan soluciones informáticas. Actualmente trabaja el área de Cloud Computing, que consiste en proveer los paquetes de software como servicios, lo cual permite a las empresas evitar los gastos que implica la adquisición y mantenimiento de la infraestructura.

Las prácticas en esta empresa consistieron en llevar a cabo el módulo de servicios de un manejador de relaciones con el cliente. Éste se dividió en dos proyectos: el de la aplicación servidor y el del portal web por el estudiante de Ingeniería de Ciencias de la Computación, Dennis Moritz Aldana Moscoso; y la interfaz móvil desarrollada en este trabajo.

B. Descripción general del trabajo

Este trabajo presenta el desarrollo de una aplicación móvil para la consulta y la realimentación de la información que está almacenada en el módulo de servicios de un CRM (por sus siglas en inglés) Manejador de Relaciones con el Cliente. Este módulo es el resultado de un proyecto que fue desarrollado paralelamente para construirlo sobre Microsoft SharePoint lo cual permitió establecer los lineamientos necesarios para que los trabajos fueran compatibles.

El resultado final de este proyecto fue el código fuente de la aplicación desarrollada en lenguaje de programación C# para la plataforma .NET Mobile, el manual de usuario (el cual está disponible en el apéndice) y un conjunto de documentación la cual se tratará a lo largo de este documento.

La aplicación se desarrolló pensando en que se utilice en una empresa de servicios, la cual dispone de un grupo de técnicos que tienen a su cargo realizar las tareas que cumplen los requerimientos establecidos por los clientes. Generalmente muchas de estas tareas se deberán realizar fuera de la empresa caminando

en la calle por ejemplo, por lo que a cada uno de estos técnicos se les instalará dicho software en su teléfono móvil con el fin de que puedan estar al tanto de la tarea que deben realizar e informar a la central de sus avances. La asignación de estas tareas se realizará en el CRM por el coordinador, que es la persona que tiene a su cargo supervisar el trabajo de los técnicos y velar por el cumplimiento de los requerimientos de los clientes.

C. Definición del problema

Actualmente los clientes tienen la posibilidad de escoger entre varias empresas para satisfacer una necesidad particular, lo cual obliga a las empresas a mejorar sus procesos o sus servicios para crear lealtad entre sus clientes y de esta manera competir en el mercado. Una manera de mejorar el servicio con los clientes es el uso de CRM que permite a las empresas mantenerlos al tanto de la información. El problema que se tiene con estos sistemas es que, tanto el personal del servicio al cliente, como los técnicos, no tienen una forma fácil e intuitiva de acceder a la información necesaria para realizar su trabajo en el momento justo. Esto se debe a que los sistemas que regularmente se utilizan para acceder a dicha información son muy caros y difíciles de transportar. Por otro lado, la realimentación que los técnicos deben proveer a la central no ocurre de forma inmediata, ya que tienen que llegar al lugar donde se encuentra la central o utilizar un medio de comunicación que incurre en gastos extras. Debido a estos problemas, el rendimiento del servicio es afectado en gran manera, lo cual puede crear insatisfacción en los clientes.

D. Propuesta de solución

Para solucionar el problema anteriormente descrito se consideró la implementación de una aplicación móvil para realizar consultas y actualizaciones de la información que almacena el servidor de CRM, ya que es fácil de transportar, el precio es accesible y se puede realizar una aplicación que sea fácil y sencilla de utilizar. Para realizar esta aplicación se consideraron los siguientes requerimientos:

- Desarrollar una aplicación móvil
- Establecer una conexión con el servidor del sistema CRM autenticando al usuario con el sistema
- Obtener la lista de tareas que el usuario tiene asignada
- Obtener la información detallada de una tarea
- Especificar los campos que se pueden mostrar al momento de desplegar una tarea
- Actualizar la información de una tarea
- Escoger los campos que se podrá modificar al momento de realizar la actualización.
- Consultar las sub-tareas de una tarea

- Asignar una nueva sub-tarea a otro usuario
- Consultar el GPS del dispositivo para obtener la localización del usuario y enviar dicha información al servidor
- Buscar una lista de tareas dado un criterio de búsqueda
- Tomar una fotografía del trabajo y enviarla a la central

E. Alcance y límites

- De parte de la empresa se requiere desarrollar una versión para Windows Mobile con posibilidad de ser desarrollada para otras plataformas, por lo que este proyecto sólo abarca el desarrollo de la versión para Windows Mobile 6
- No se implementará una pantalla de búsqueda de tareas, ya que se supone que esta aplicación manejará tareas muy pequeñas y puntuales que necesitarán poco tiempo para realizarse
- No se implementará el requerimiento de tomar una fotografía y enviarla a la central, ya que no es necesario en la entrega de este proyecto, pero se desea trabajar después

F. Limitaciones

- Se dispuso de 400 horas para el desarrollo del proyecto, trabajando 5 horas diarias de lunes a viernes.
- Las limitaciones que tiene una aplicación móvil se describirán en el capítulo de análisis.

G. Justificación

La implementación de una aplicación móvil soluciona el problema de la disponibilidad de la información del módulo de servicio del CRM a los técnicos, ya que tiene la ventaja de ser portable, lo que le permitiría a ellos estar al tanto de la tareas que deben realizar y realimentar a la central del progreso en los lugares que tienen cobertura de Internet. Además, es importante mencionar que el costo de estos dispositivos es accesible, permitiendo a la empresa de servicio implementar este sistema, de tal forma que cada técnico tenga acceso a la información. De esta manera se lograría que el servicio de la empresa mejore y se desarrolle en menos tiempo.

Por otro lado, al realizar este proyecto, se adquirió nuevos conocimientos en el área de programación de aplicaciones móviles, tales como los principios para el diseño de interfaz de usuario o las herramientas que se puede utilizar para facilitar el desarrollo. También se adquirió los conocimientos básicos para

establecer una conexión entre una aplicación móvil y un CRM de tal manera que se considere los aspectos necesarios de seguridad.

H. Objetivos

1. General

- La implementación de una interfaz móvil que tenga acceso a la información del módulo de servicios de un manejador de relaciones con el cliente.

2. Específico

- Establecer una forma para realizar el intercambio de información entre el servidor del CRM y un teléfono celular.
- Desarrollar una interfaz gráfica de usuario de la aplicación, que cumpla con los principios de desarrollo de aplicaciones móviles que se utilizan actualmente.
- Guardar los aspectos necesarios de seguridad que necesita la aplicación para realizar un mejor manejo de la información.

II. MARCO TEÓRICO

A. CRM

Customer Relationship Management. Manejador de Relaciones con el Cliente.

1. **Definición de un CRM.** Un CRM es un conjunto de estrategias que una empresa utiliza para mejorar la relación con sus clientes, con el objetivo de aumentar sus ingresos a través de la lealtad de ellos. Para implementar estas estrategias se utiliza un sistema que se encarga de facilitar la administración de la información de los clientes de tal manera que el cliente perciba que se está recibiendo un servicio personalizado.(Rodgers & Howlett, 2000)

Un CRM trabaja con los siguientes principios:

- Tratar a cada cliente de forma individual
- Adquirir y mantener la lealtad de los clientes a través de trato personal
- Proveer a las empresas de los mejores clientes. (Gray & Byun, 2001)

2. **Lugares donde se puede utilizar CRM.** En las organizaciones con un mercado pequeño, donde es fácil reconocer a cada cliente, o con poca competencia, se puede decidir no utilizar un CRM. En cambio las empresas que participan en un mercado grande con mucha competencia, necesitan utilizar un sistema de este tipo para lograr la satisfacción del cliente con el objetivo de lograr una alta demanda de él.(Gray & Byun, 2001)

3. **La importancia de utilizar un CRM en una empresa.** En las décadas pasadas, la principal estrategia para aumentar las ganancias era la reducción de los costos de producción o hacer más eficientes los procesos. Sin embargo, actualmente esta estrategia no es suficiente para mantener a una empresa compitiendo en el mercado. Otro factor que es importante considerar es aumentar los ingresos con una alta demanda de los clientes.(Rodgers & Howlett, 2000)

Para mantener la demanda de los clientes, se puede considerar dos vías: ganando nuevos clientes o manteniendo los existentes. Se ha comprobado que es más conveniente mantener los clientes que tratar de adquirir nuevos. La primera opción exige que la empresa mantenga buenas relaciones con sus clientes actuales ya que de esta manera logrará mantenerlos satisfechos y aumentar su fidelidad a la empresa. Una CRM es una buena opción para lograr buenas relaciones con el cliente, ya que permite que cada cliente

sea tratado de forma individual y recibir un buen servicio, dado que el CRM utiliza la tecnología para lograrlo, que de otra manera sería imposible. (Gray & Byun, 2001)

Por otro lado, un CRM ayuda a reducir costos en las ventas y distribución, ya que permite determinar a qué grupo de personas debe estar dirigida la publicidad de la empresa, de tal manera que la probabilidad que acepte la oferta sea alta. Además le permite dar otro enfoque a la empresa, trabajar por la satisfacción de los clientes y no por el producto. (Gray & Byun, 2001)

B. Servicio web

Web Services. Servicios Web.

1. Definición de un servicio web. Un servicio web es un conjunto de funciones a los cuales se accede a través de protocolos estándares de internet con las siguientes características:

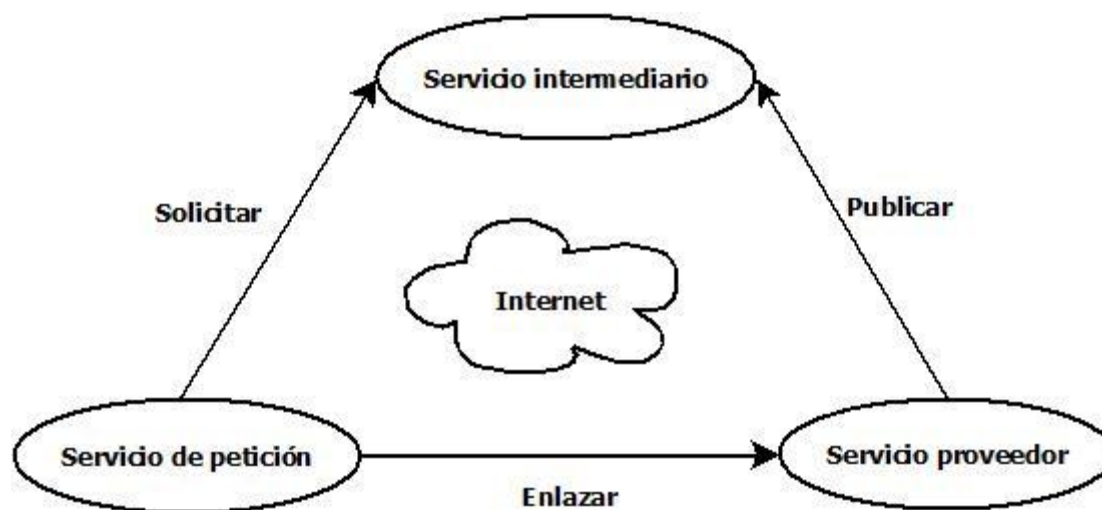
- **Encapsulada:** No se puede tener acceso a la codificación de las funciones.
- **Dependiente sólo de la definición de la función:** Significa que algún cambio en la codificación de las funciones no requiere un cambio en la forma de invocar a la función.
- **Bien documentada:** Indica que se dispone de la información del comportamiento de las funciones y de los parámetros que recibe y devuelve. (Gunzer, 2002)

2. Arquitectura. Para lograr que un servicio web funcione entre varios ambientes heterogéneos es necesaria la participación de varios componentes en el proceso de comunicación. Generalmente los componentes que se utilizan para que funcione un servicio web son los siguientes:

- **Un servicio intermediario:** éste tiene a su cargo indicar en donde encontrar el servicio pedido.
- **Un servicio proveedor:** se encarga de proveer los servicios ofrecidos por el *Servicio web*.
- **Un servicio de petición:** solicita un servicio a un intermediario, quien le indica cuál es el proveedor, para luego conectarse con éste y obtener la petición (Gunzer, 2002)

La siguiente imagen indica cómo interactúan estos componentes:

Ilustración 1 Componentes de un servicio web



Otros aspectos que ayudan a los servicios web a funcionar entre diferentes ambientes es el uso de estándares que se describirán en las siguientes secciones:

3. XML (eXtensible Markup Language). XML es la base de los servicios web, ya que indica la manera cómo se presentarán los mensajes que se intercambiarán entre los componentes. XML fue establecida por W3C (World Wide Web Consortium) con el propósito de proveer un lenguaje para la descripción de la información. Se utilizan etiquetas para organizar la información de tal manera que se pueda construir una estructura jerárquica de los datos, y se codifica de forma que sea legible a los seres humanos. Esta información puede ser transmitida entre los servicios utilizando protocolos de transporte como HTTP o FTP. Las demás tecnologías que se utilizan en un servicio web están basadas en mensajes XML con el fin de lograr interoperabilidad entre los sistemas. (Cavanaugh, 2006)

4. WSDL (Web Services Description Language). WSDL es un lenguaje que se utiliza para la descripción de los servicios web. Actualmente es mantenido por la W3C. Para definir los archivos WSDL se utiliza XML. A través del archivo WSDL de un servicio web, el cliente puede determinar la información necesaria para interactuar con éste. Generalmente este tipo de archivos contiene los siguientes elementos:

- **Tipo de puerto:** es una descripción de las funciones que se ofrece en el servicio.
- **Puerto:** indica la dirección para la conexión.
- **Mensajes:** describe los nombres y el formato de los mensajes que soporta el servicio.
- **Tipos:** define los tipos de datos utilizados en el servicio.

- **Binding:** especifica los protocolos que se están utilizando.
- **Servicio:** indica la dirección URL del servicio. (Cavanaugh, 2006)

5. SOAP (Simple Object Access Protocol). SOAP es un protocolo que se utiliza para indicar la manera de intercambiar la información entre los clientes y el servicio web. Los mensajes que se envían en los servicios se codifican en XML con las especificaciones establecidas por este protocolo, de tal manera que los dos ambientes puedan compartir información aun si son heterogéneos. (Gunzer, 2002)

Los mensajes SOAP pueden contener los siguientes elementos:

- **Sobre:** se encarga de encerrar todo el contenido del documento con una etiqueta de XML especificada por el protocolo.
- **Encabezado:** contiene la información importante del mensaje, como la fecha o los datos utilizados para autenticación de la información. Este elemento es opcional.
- **Cuerpo:** contiene la información del archivo, como petición a una función o el resultado.
- **Falla:** en caso que exista algún error en cliente o servidor, dicha información es colocada en este elemento. Es opcional. (Cavanaugh, 2006)

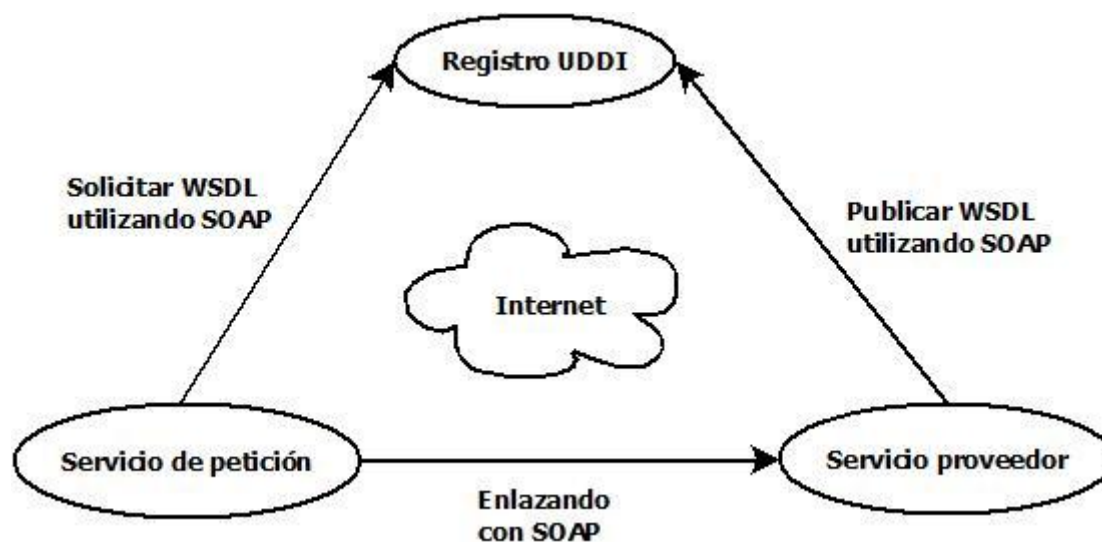
6. UDDI (Universal Description, Discovery, and Integration). Es un estándar basado en XML para tener un registro de los servicios que se ofrecen. Se encuentra en el servicio intermediario para cuando un cliente necesite la información de un servicio web, y puede proveer el archivo WSDL que contiene la información del servicio. Este estándar está patrocinado por OASIS (*Organization for the Advancement of Structured Information Standards*). (Cavanaugh, 2006)

Generalmente un UDDI está diseñado como un directorio telefónico de la siguiente manera:

- **Páginas amarillas:** contiene la información de los servicios de una industria, categoría de producto o posición geográfica determinada.
- **Páginas blancas:** almacena la información acerca del servicio proveedor como contacto, dirección o identificadores conocidos.
- **Páginas verdes:** contiene la información técnica del servicio web. Además indica la forma para conectarse a éste. (Gunzer, 2002)

Después de analizar estas tecnologías, se puede decir que la arquitectura de un servicio web tiene la siguiente forma:

Ilustración 2 Arquitectura de un servicio web



7. Plataformas. A pesar de que se ha buscado que las implementaciones de servicio web sean homogéneas entre ellas, algunas difieren en algunos aspectos. La teoría que se maneja es muy completa, pero algunos fabricantes han considerado agregar algunos detalles que traen algunos beneficios. A continuación se presentan las especificaciones para servicio web que se manejan actualmente. Como se mencionó anteriormente algunas tienen modificaciones.

- WebServices.Org
- IBM Web Services
- W3C Web Services Workshop
- Microsoft Web Services
- Sun Open Net Environment (Sun ONE)
- Oracle9i Web Services framework
- Hewlett-Packard
- BEA Systems
- Borland(Myerson, 2006)

C. Interfaz móvil de usuario

1. La tendencia actual de uso de dispositivos móviles. En la actualidad el uso de dispositivos móviles, especialmente los teléfonos celulares, se ha popularizado entre las personas hasta el punto que ya no se considera un lujo, sino un producto de primera necesidad. Esto se refleja en el siguiente dato: la suscripción de teléfonos celulares en Latinoamérica aumentó en un promedio de 26% al año entre 2000 y 2009 (Las comunicaciones móviles serán la base de crecimiento del sector en América Latina, 2010). En el ámbito nacional se observa una tendencia similar, ya que se reportó que en 2009 se otorgaron 17307459 líneas celulares, lo cual representa un 15.78% de crecimiento respecto al año anterior (Contratos por líneas móviles superan a las residenciales en Guatemala, 2010).

Aparte de esta gran tendencia de crecimiento en los mercados de celulares, se puede agregar que también las personas están más interesadas en comprar un teléfono celular más sofisticado, es decir que no se limiten sólo a realizar y recibir llamadas, sino que se pueda utilizar para navegar por internet, consumir multimedia o recibir correos. (Batz, 2010)

2. La importancia de uso de aplicaciones móviles en el área de trabajo. El aumento en popularidad y mejoras a la tecnología de los teléfonos móviles, ha permitido que crezca una demanda de desarrollo de aplicaciones que aumente la funcionalidad del aparato, permitiendo que sea más útil en el área de trabajo. Lo cual hace que las empresas los utilicen para ayudar a mejorar sus procesos. (Rodríguez Ventosa & Secilla)

Los teléfonos celulares se han convertido en una de las herramientas de comunicación más importantes de las empresas, ya que les permite estar al tanto de la situación, ya que las nuevas funcionalidades que traen los teléfonos facilitan mucho las actividades de las personas. (Batz, 2010)

3. Principios para el desarrollo de aplicaciones móviles. El desarrollo de una aplicación móvil requiere de un gran esfuerzo para lograr hacer una aplicación fácil de utilizar a pesar de las limitaciones que tiene en cuanto al hardware. Por lo que es necesario recordar que el desarrollo de una aplicación móvil es muy diferente al de una aplicación para una computadora de escritorio. La empresa UserCentric propone los siguientes consejos para mejorar la experiencia del usuario en el uso de la aplicación.

- Indicar al usuario cuál es el objeto que tiene el enfoque del sistema, para evitar que se pierda en la interfaz.
- Facilitar al usuario la búsqueda y acceso de funciones avanzadas.
- Uso de etiquetas consistentes y fácil de entender.
- Proveer a los usuarios iconos fácil de comprender sin que afecte el espacio de la pantalla.

- Permitir a los usuarios configurar y personalizar la aplicación.
- Hacer uso de pantalla de mayor longitud con un buen sistema de *scrolling* en vez de una gran multitud de pantallas. (Vaidya, 2009)

D. Comparación entre el hardware de un Smartphone y de una computadora

En esta sección se presenta una comparación de las capacidades de un Smartphone con una computadora. Un Smartphone es un teléfono celular que aparte de realizar llamadas telefónicas, incluye características similares a una computadora, como navegación por internet, envío de correos electrónicos o consulta de documento. (Cassavoy, 2009)

El propósito de esta comparación es establecer las diferencias que existen entre las capacidades de un teléfono Smartphone y una computadora; las cuales se deben considerar al momento de realizar una aplicación móvil.

- Velocidad de procesador
 - Teléfono: 300 a 620 MHz (Woyke, 2009)
 - Computadora: 1.66 GHz a 3.33 GHz (Software, 2010)
- Tamaño de pantalla
 - Teléfono: 240x320 pixeles (Hjerde, 2008)
 - Computadora: 1024x768 (W3Schools, 2010)
- Memoria interna para aplicaciones
 - Teléfono: 4 MB a 256 MB (Litchfield, 2010)
 - Computadora: 512 MB a 2 GB (Sklavos, 2005)

E. Sistema de localización

1. GPS (Global Positioning System). El sistema global de posicionamiento (GPS) es una herramienta para la navegación y determinación de las coordenadas geográficas de un punto. Fue desarrollado por el departamento de Defensa de los Estados Unidos en 1973. Los primeros usos que se le dieron fueron para operaciones militares con el fin de determinar la ubicación de las unidades de ataque, como aviones, tanques o barcos. En la actualidad se utiliza en varias áreas, por ejemplo en la comercial se utiliza para navegación y posicionamiento de cualquier medio de transporte. También se puede utilizar para asistir en el desarrollo de deportes como el alpinismo o pesca. En el área científica, se utiliza para predicciones de clima o medición de terremotos. (Glasscoe, 1998)

2. Geocoding y Reverse Geocoding. Geocoding es el proceso de tomar la ubicación de un lugar, generalmente las coordenadas geográficas, y traducir a una dirección que los seres humanos puedan utilizar como referencia. Reverse Geocoding, como su mismo nombre lo indica, es el proceso inverso de traducir una dirección a coordenadas geográficas.

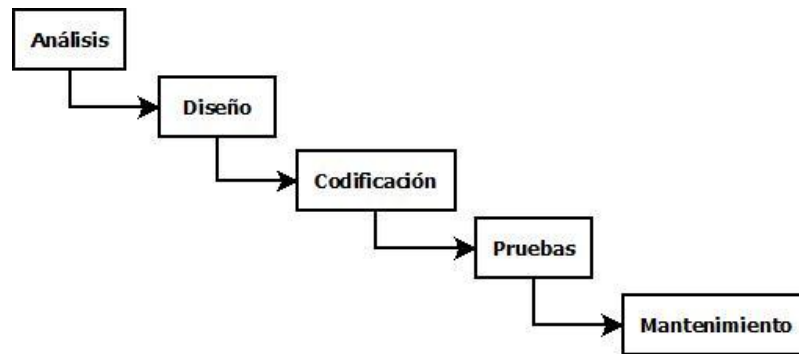
F. Desarrollo de software

1. Modelo de desarrollo de procesos. Es un conjunto de estrategias que indica las tareas y forma de distribuirlas para que se logre desarrollar la aplicación. Existen varios tipos, entre ellos se pueden mencionar los prototipos, que se enfocan en buscar e innovar ideas para el desarrollo de aplicaciones para crear el proyecto con el fin de determinar qué desea el cliente. Están los secuenciales, donde todas las tareas siguen una línea de desarrollo y es necesario completar una tarea para continuar con la siguiente. La evolutiva, aquella donde el cliente recibe varias entregas del software hasta obtener la versión completa. (Pressman, 2001)

2. Modelo cascada. Fue uno de los primeros modelos que se utilizó para desarrollar los proyectos de software, se caracteriza por tener etapas bien definidas y secuenciales. Se dice que es lineal, ya que todo el proyecto se desarrolla sobre una misma línea de tareas, es necesario completar una tarea para pasar a la siguiente, además es necesario iniciar el proyecto desde cero para obtener al final el resultado. Las etapas de modelo de cascada son:

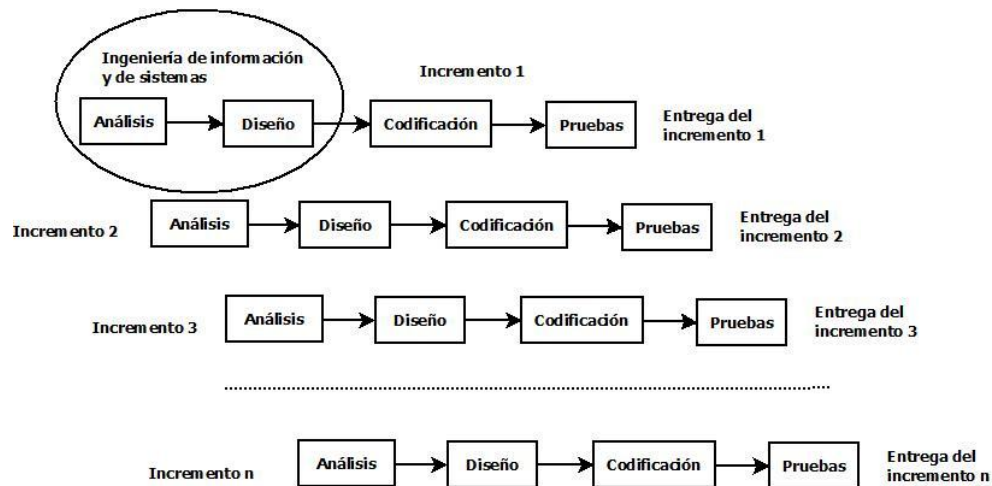
- **Análisis:** se utiliza esta fase para determinar en qué consistirá el proyecto, y qué funcionalidades deberá tener el resultado al final. Se establecen los requerimientos del cliente y se obtiene una visión general del proyecto.
- **Diseño:** en esta fase, se toman los requerimientos y la documentación generada en el análisis para determinar la manera de cómo se desarrollará el proyecto, qué herramientas se utilizarán, qué estructura de datos y la forma de cómo guardar la información.
- **Codificación:** se toma el diseño establecido en la fase anterior y se inicia el proceso de codificar e implementar la solución con las herramientas escogidas.
- **Pruebas:** se realizan las pruebas necesarias para determinar que la solución cumpla con todos los requerimientos establecidos en la fase de análisis. En caso de encontrar errores, se corrigen en esta etapa.
- **Entrega:** se entrega la solución al cliente y se instala en el ambiente de ejecución.
- **Mantenimiento:** son los trabajos que se deben realizar para permitir que la aplicación pueda seguir funcionando, tales como mejoras, correcciones, actualizaciones o nuevos desarrollos. (Pressman, 2001)

Ilustración 3 Modelo cascada



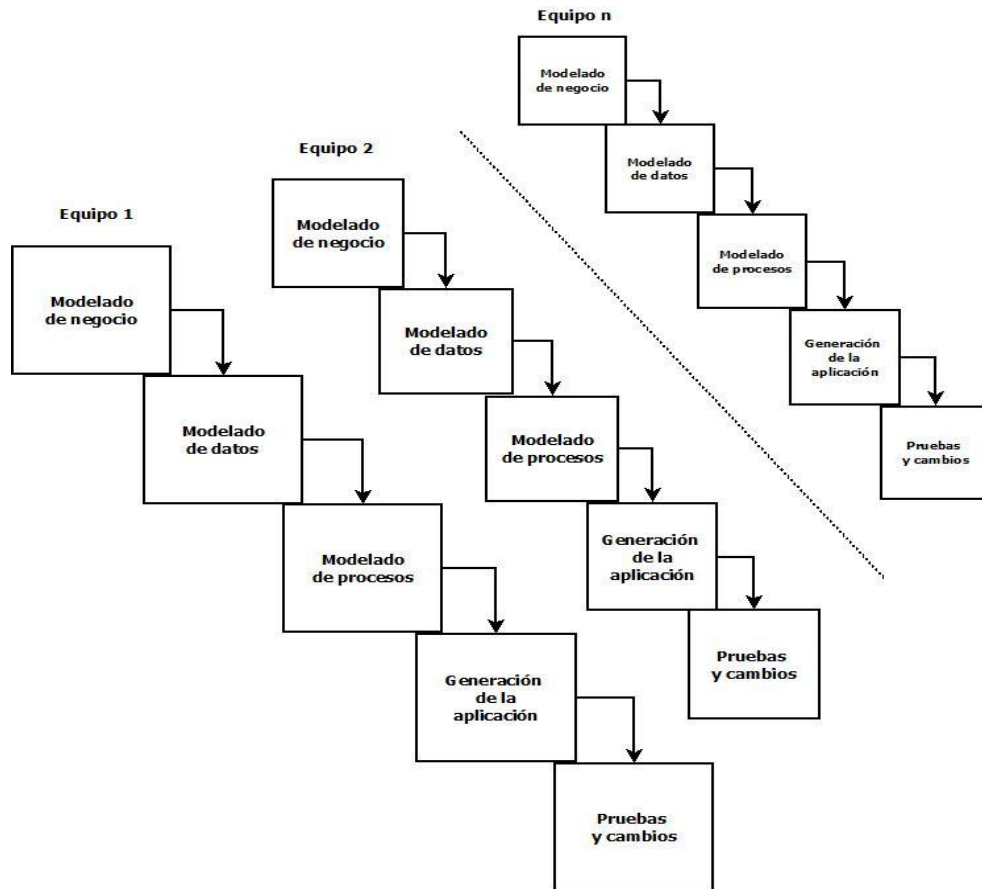
3. Modelo incremental. Este modelo consiste en realizar varias iteraciones del modelo en cascada para obtener un proyecto que cumpla más con los requerimientos del cliente. Las iteraciones reciben el nombre de incremento. En el primer incremento se define las características principales que se debe desarrollar del proyecto, cuando se termina éste se obtiene una versión básica del proyecto y se procede a realizar otro incremento para agregarle más características y funcionalidades. Al final de cada incremento se debe obtener una versión funcional del proyecto. (Pressman, 2001)

Ilustración 4 Modelo incremental



4. Modelo RAD. El modelo de desarrollo rápido de aplicaciones (RAD por sus siglas en inglés) es un modelo de procesos de desarrollo de software que se enfoca en ciclos cortos de desarrollo. Este modelo trata de lograr más resultados utilizando una construcción de aplicación basado en el uso de componentes prefabricados para integrarlos de tal manera que la aplicación se construya en poco tiempo. Está diseñado para que varios equipos de trabajo desarrollen el mismo proyecto y aumente su velocidad de desarrollo.

Ilustración 5 Modelo RAD



Sigue el siguiente ciclo de desarrollo:

- **Modelado del negocio:** se encarga de identificar cuáles son los procesos a implementar y cuáles son las entidades encargadas de producirlas y consumirlas. Es decir que define el flujo de la información.
- **Modelado de la información:** se diseña un conjunto de objetos de información para soportar el flujo de negocio definido en la fase anterior. Se define las relaciones entre los datos.
- **Modelado de procesos:** se crean los procesos necesarios para manipular los objetos definidos en la fase anterior.

- **Generación de módulo o aplicación:** se utiliza lenguaje especializado para ciertos ámbitos de computación o componentes prefabricados para construir el módulo o la aplicación.
- **Pruebas y cambio:** se realizan las pruebas sobre desarrollado y cuando se haya terminado el trabajo se procede a cambiar a otro módulo. Se enfoca mucho la reutilización de código. (Pressman, 2001)

G. Cifrado

1. **Definición.** El cifrado es el proceso de tomar información, llamada texto plano, y aplicarle una secuencia de operaciones para generar una forma diferente de ésta de tal forma que sea difícil realizar el proceso inverso. El resultado de este proceso recibe el nombre de texto cifrado y a la secuencia de operaciones, algoritmo. Para lograr que el proceso inverso, llamado descifrado, sea difícil se puede esconder el algoritmo o agregar otro componente, llamado llave que sólo con el conocimiento de éste y otro algoritmo se pueda obtener el texto plano. (Cook, 2000)

Generalmente el cifrado se utiliza para:

- **Autenticación:** identificar a una persona.
- **Validación:** para comprobar la identidad de una persona.
- **Protección de información:** para evitar que una persona no autorizada tenga acceso a los datos.
- Protección de la comunicación: para evitar el robo de información. (Cook, 2000)

2. **Algoritmos de cifrados.** En el cifrado, existen dos componentes muy importantes que se deben proteger, la llave y el algoritmo. Actualmente la tendencia ha sido mantener en secreto la llave y dar a conocer cómo funciona el algoritmo. Hay dos tipos de algoritmos, el primero se llama Algoritmo simétrico, ya que maneja una sola llave para el proceso de cifrado, lo que es importante para proteger dicha llave, ya que con el conocimiento de ésta, se puede tener acceso a la información. El segundo tipo se llama Algoritmo Asimétrico, donde se manejan dos llaves, una de conocimiento general, llamada llave pública y la otra de conocimiento individual, llamada llave privada. Con el uso de dos llaves se hace más efectivo el trabajo del cifrado.(Harris, 2008)

3. **DES (Data Encryption Standard).** Es un estándar de algoritmos de cifrados de datos desarrollado por el NIST (National Institute of Standards and Technology) en 1974, basado sobre un algoritmo llamado Lucifer desarrollado por IBM. Este algoritmo utiliza una llave de 64 bits, de las cuales utiliza ocho para comprobar la validez de la misma llave (se llaman bits de paridad), por lo que realmente el tamaño de la llave es de 56 bits. Debido al aumento de la capacidad de las computadoras, se puede

encontrar la llave de un texto cifrado por una algoritmo que implemente DES, por lo que la información está vulnerable. Actualmente se utiliza sólo para fines educativos.(Harris, 2008)

4. Triple-DES. Este algoritmo se basaba sobre el estándar establecido por DES, sólo que realiza tres veces el proceso de cifrado sobre el texto con tres diferentes llaves para evitar que el texto sea fácil de descifrar por una persona ajena. Se utiliza uno de los siguientes modos para operar este algoritmo:

- **DES-EEE3:** la información es cifrada tres veces con tres llaves diferentes.
- **DES-EDE3:** los datos son cifrados, descifrados, cifrados por tres llaves diferentes.
- **DES-EEE2:** la información es cifrada tres veces con dos llaves diferentes, donde el primer y último proceso utilizan la misma llave.
- **DES-EDE2:** los datos son cifrados, descifrados, cifrados por dos llaves diferentes, donde el primer y último proceso utilizan la misma llave.(Harris, 2008)

H. Plataforma y herramientas de desarrollo

1. Framework .NET. es una plataforma de Microsoft para el desarrollo y ejecución de aplicaciones que provee lo siguiente:

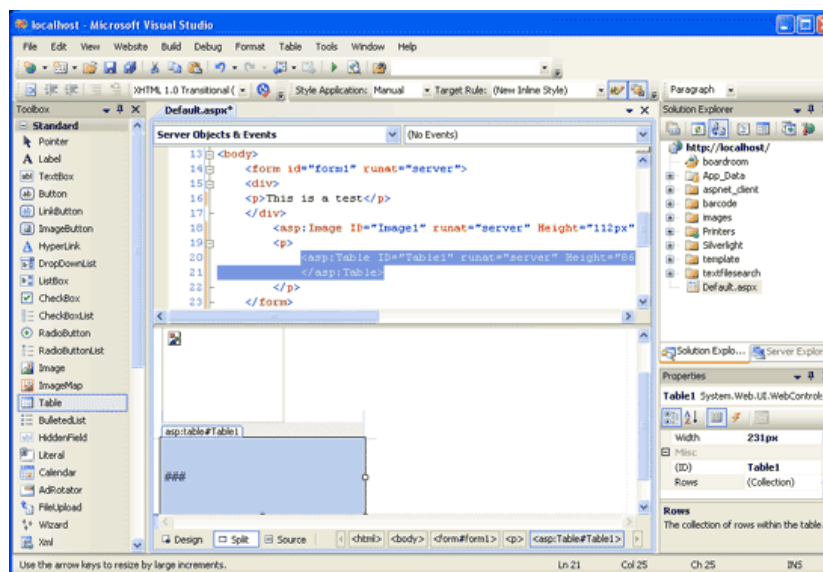
- Una forma fácil de acceder a los servicios del sistema operativo a través de un conjunto de librerías.
- Una infraestructura de gestión de ejecución de las aplicaciones.
- Un conjunto de herramientas para desarrollo de aplicaciones web.
- Una arquitectura enfocada en sistemas distribuidos. (Grundgeiger, 2002)

El Framework .NET está compuesto por los siguientes elementos:

- CLR (*Common Language Runtime*): que provee una capa de abstracción sobre el sistema operativo para ejecutar aplicaciones.
- Librerías base: librerías para tarea de programación de bajo nivel.
- Una conjunto de herramientas y tecnologías para aumentar la productividad de los desarrolladores. (Microsoft, .NET Framework Overview, 2009)

2. Framework .NET Compact. Es una versión de .NET especialmente diseñada para dispositivos portables. Las librerías provistas por este Framework son muy similares a las de la contraparte en la versión en computadoras, ya que se han reducido ciertas funcionalidades con el fin de proveer un mejor soporte a los dispositivos móviles que tienen limitaciones en cuanto a hardware. (Makofsky, 2003)

3. Visual Studio. Es ambiente integrado de desarrollo (IDE por sus siglas en inglés) que contiene un conjunto de herramientas para desarrollo de aplicaciones web, servicios web, aplicaciones de escritorios y aplicaciones para dispositivos móviles. Por lo que permite realizar el proceso de codificar, compilar, probar y ejecutar aplicaciones. Permite, también, programar en una gran variedad de lenguajes de programación soportados por Framework .NET como Visual C# y utilizar todas las librerías que éste pone a disposición. (Spillmann & Shmedding, 2009).



4. Windows Mobile. Es una plataforma diseñada para dispositivos móviles basada sobre Windows CE, el núcleo del sistema. Actualmente existe una gran variedad de teléfonos modernos soportados por Windows Mobile. Para facilitar el desarrollo de aplicaciones móviles para esta plataforma, está disponible Windows Mobile SDK que le permite a los desarrolladores utilizar varias herramientas que facilitan el proceso de desarrollo. (Microsoft, Windows Mobile, 2010)

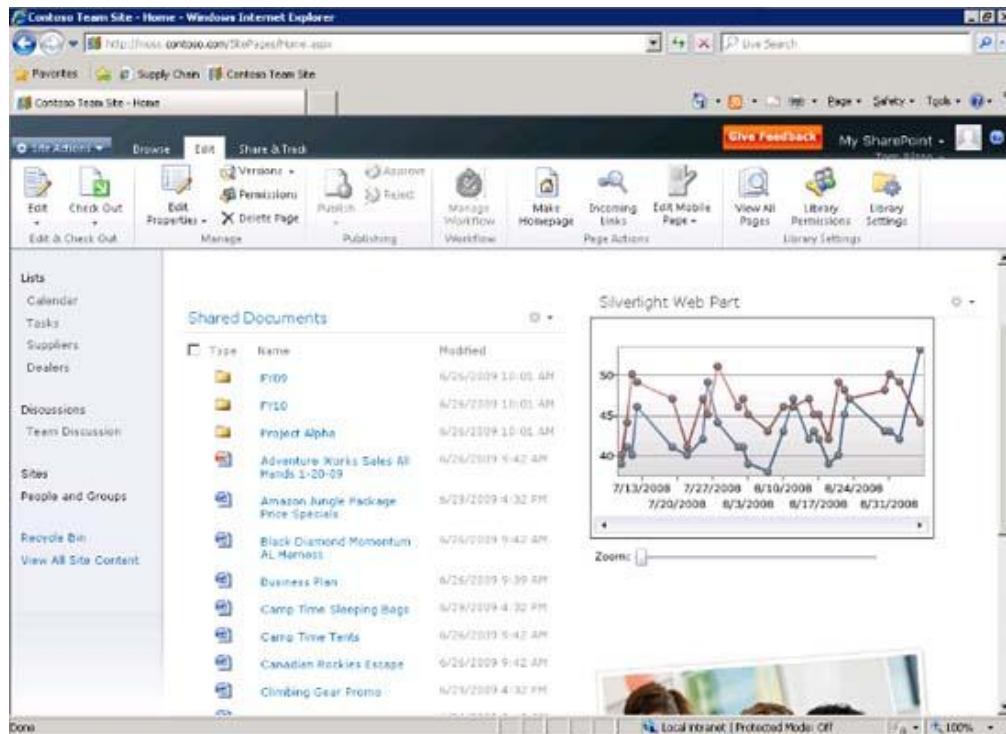
5. Versión de Windows Mobile. A continuación se presenta las diferentes versiones de Windows Mobile. (De Herrera, 2009)

Tabla1 Versiones de Windows Mobile

Fecha	Versión	Kernel
2000	Pocket PC	Windows CE 3.0
2003	Windows Mobile 2003 for Pocket PC	Windows CE 4.2
2004	Windows Mobile 2003 Second Edition	Windows CE 4.2
2005	Windows Mobile 5	Windows CE 5.1
2007	Windows Mobile 6	Windows CE 5.2

6. Microsoft SharePoint. Es una plataforma para crear sitios de colaboración donde las personas pueden compartir archivos e información y administrar los procesos de negocios. Microsoft SharePoint se encarga de abstraer el manejo de la base de datos y proveer el concepto de lista para facilitar la administración de la información. También provee una plataforma de desarrollo para que los desarrolladores puedan extender las funcionalidades de los sitios creados. Se pueden utilizar los servicios web de SharePoint para que las aplicaciones externas tengan acceso a la información que éste almacena. (Bates & Smith, 2010)

Ilustración 6 Una captura de pantalla de Microsoft SharePoint



III. METODOLOGÍA

Para desarrollar este proyecto era necesario determinar la metodología que más se adaptaba a las necesidades del sistema y para seleccionarla se tomaron en cuenta varias variables. Lo primero que se consideró fue los requerimientos del cliente, los cuales fueron especificados y no era necesario realizar algún tipo de metodología relacionada con prototipos para que el cliente completara su definición; tampoco fue necesario utilizar una metodología donde los requerimientos pudieran variar en la ejecución del proyecto. Es importante mencionar que se tenía buena comunicación con el cliente para afinar algunos detalles de este proyecto, lo cual era de mucha ayuda para el desarrollo. Lo segundo que se consideró fue la existencia de un proyecto previo para continuar trabajando sobre éste. Se determinó que era necesario comenzar desde cero, ya que no se contaba con un proyecto de tal naturaleza. Otro detalle importante era utilizar componentes prefabricados, pero se pensó que no era necesario ya que se necesitaba desarrollar componentes que aprovecharan de la mejor forma el ambiente de ejecución. En cuanto al tiempo, sólo se disponía de 400 horas para desarrollarlo, por lo que era necesario buscar una metodología con resultados rápidos.

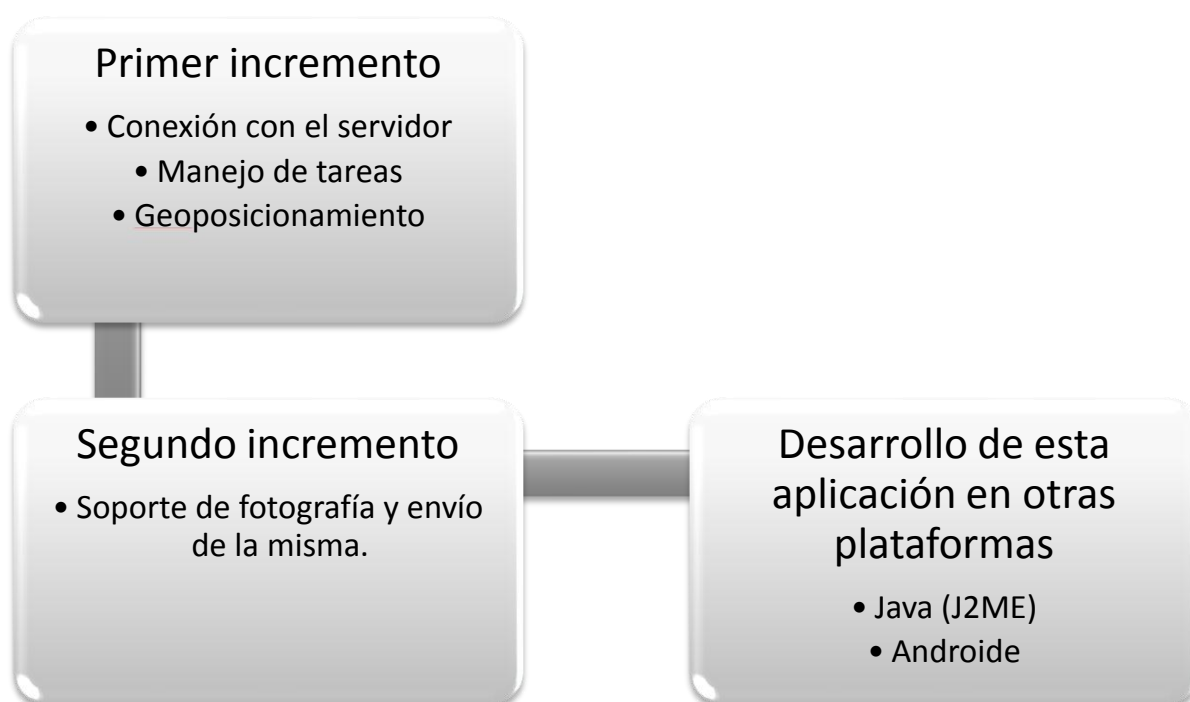
La característica que más influyó en la decisión de seleccionar la metodología fue la necesidad de dejar un proyecto que pudiera continuar en el futuro con nuevas entregas o nuevas versiones. Ya que en esta primera fase era necesario entregar una versión incompleta del sistema porque no se implementarían todos los requerimientos establecidos por el cliente, (búsqueda y el soporte de cámara), debido a que estas características se implementarían en futuras versiones de este sistema que están fuera del alcance de este proyecto. Es importante mencionar que la empresa desea implementar este sistema en otras plataformas móviles en otros proyectos, por lo que era necesario desarrollar este sistema de tal manera que sirviera de base para estas versiones.

El primer incremento de este proyecto fue realizar la versión de Windows Mobile de esta aplicación, la cual comenzó primero con un fase de análisis, por lo que primero se elaboró un documento de especificación de requerimiento de software (SRS por sus siglas en inglés) siguiendo una plantilla elaborada por Karl E. Wiegers que está disponible en el sitio www.processimpact.com. Luego se realizó la etapa de diseño donde se dividió el proyecto en varios módulos en los cuales se determinó las clases que se debían realizar para implementarlo. En esta fase se tomó un tiempo para realizar los diseños de la pantallas que compondrían la interfaz gráfica de la aplicación. Mucha de la documentación mencionada está desarrollada en los capítulos de análisis y diseño. Ya hecho el diseño, se procedió a la fase de codificación donde se utilizó como herramienta de desarrollo Visual Studio 2008 y para realizar las pruebas se utilizó un emulador de Windows Mobile 6. Se verificó que cumpliera los requerimientos y se

realizó una presentación final de trabajo con las personas de 1800 Software para mostrar los resultados de las prácticas.

El segundo incremento, el cual ya está afuera del alcance de este proyecto y será realizado por la empresa 1800 Software, es agregar el soporte de tomar fotos y enviarlo a la central. El requerimiento de búsqueda de tareas aun no es una prioridad en el desarrollo de esta aplicación, por lo que se espera incluirlo en futuros incrementos de este sistema. Además, la empresa desea que al momento que se esté realizando el segundo incremento, también se planifique el desarrollo en otras plataformas. La siguiente gráfica es útil para mostrar estos detalles.

Ilustración 7 Los incrementos del sistema



IV. ANÁLISIS

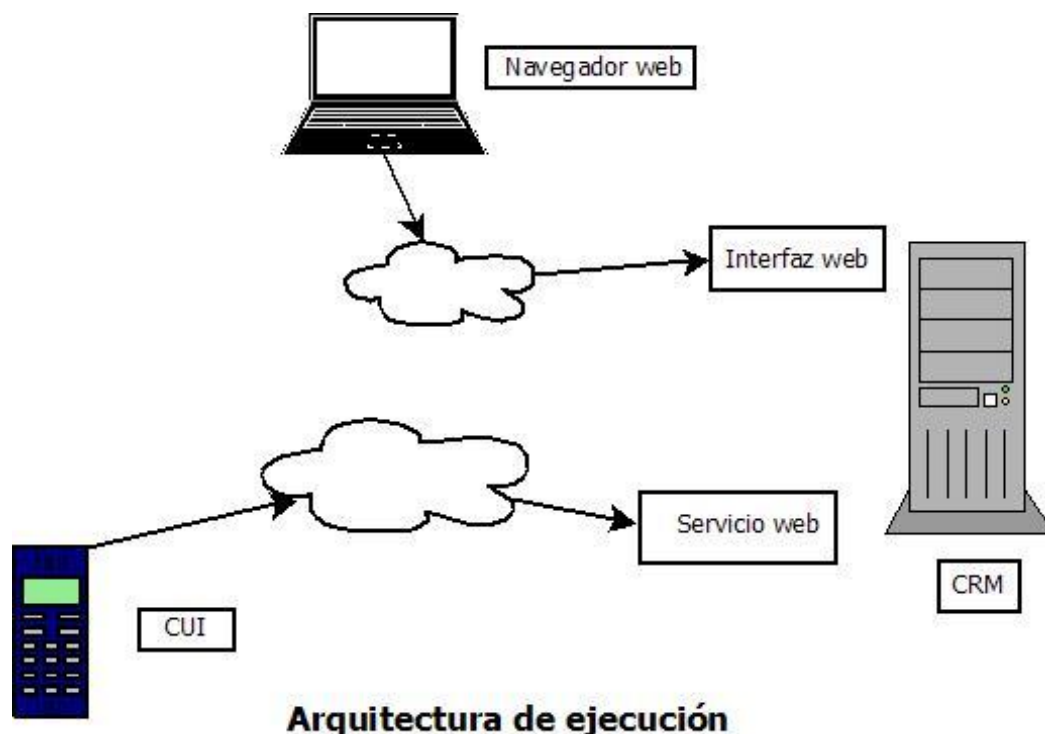
A. Introducción

El propósito de este capítulo es dar una descripción detallada de los requerimientos del sistema desarrollado en este proyecto. Se establecerán las bases para demás secciones del desarrollo del sistema, estableciendo sus características y limitaciones. Además, sus capacidades y los requerimientos de seguridad e interacciones del cliente que tiene la aplicación. Para facilidad de descripción de este proyecto, se decidió colocarle un nombre a la aplicación, el cual es *Celular User Interface* que de aquí en adelante será mencionado como CUI, ya que es la interface usuario en un celular para la consulta de la información del módulo de servicio de un CRM.

B. Descripción general

1. Perspectiva del sistema. CUI es una interfaz de usuario para un dispositivo móvil que provee el acceso a la información del módulo de servicio de un CRM desarrollado paralelamente. El desarrollo de esta aplicación inició desde cero, ya que no se contaba con una aplicación de tal naturaleza dado que en el desarrollo del otro proyecto no se consideró desarrollar una aplicación móvil.

Ilustración 8 Arquitectura de ejecución



El servidor de CRM proveerá dos servicios: una interfaz web para que los coordinadores puedan acceder la información del módulo de servicios y realizar las actividades de mantenimiento; un servicio web donde CUI podrá conectarse para realizar las peticiones.

2. Características del sistema. En esta sección, se listará las características principales de CUI donde muchos de los términos utilizados serán detallados en las siguientes secciones.

- Una aplicación diseñada para ejecutar en un teléfono móvil con Windows Mobile 6.
- Puede establecer una conexión con el servidor que aloja el CRM autenticando al usuario con el sistema.
- Obtiene la lista de tareas que el usuario tiene asignado.
- Obtiene la información detallada de una tarea.
- Permite especificar los campos que se deberán mostrar al momento de acceder a la información de una tarea.
- Permite actualizar la información de una tarea.
- Tiene la opción de indicar qué campos se podrán modificar al momento de realizar una actualización.
- Obtiene la lista de sub-tareas de una tarea.
- Permite crear y asignar una nueva sub-tarea a otro usuario.
- Utilizando el GPS del dispositivo y los servicios de Google Maps, permite obtener la dirección y un mapa de la ubicación del usuario y enviar dicha información al servidor.

3. Descripción de los usuarios del sistema. El usuario final de CUI es un técnico del área de servicio de una empresa. Esta persona tiene a su cargo realizar las tareas que solucionen las solicitudes del cliente. Generalmente estas tareas serán trabajos manuales o técnicos, por lo que se espera que el usuario final tenga al menos estudios secundarios con los conocimientos mínimos para utilizar celulares modernos, lo cual indica que el diseño de la aplicación debe ser simple y fácil de utilizar. El usuario tendrá permiso de ver la información de las tareas asignadas y modificar la información de progreso. Pero dependerá del coordinador de la aplicación de CRM cuáles otros campos podrán ser modificados por el usuario.

4. Ambiente de ejecución. El CUI está diseñado para ejecutarse en un teléfono celular con sistema operativo Windows Mobile 6. Se espera que la aplicación no afecte la configuración de la otra aplicación que tiene instalado en el dispositivo.

5. Restricciones de diseño e implementación. A continuación se listará las restricciones que se deben considerar en el desarrollo de esta aplicación:

- Dado que esta aplicación se debe ejecutar en un teléfono celular, se tienen las siguientes limitaciones en hardware del ambiente de ejecución (para más detalle consulte la sección D del marco teórico):
 - El tamaño de la pantalla es reducido, por lo que las pantallas de la aplicación deben ser simples.
 - El tamaño de memoria de ejecución (RAM) es limitado, por lo que la aplicación debe hacer uso considerado de la memoria.
 - La cantidad de tareas que se pueden ejecutar en el procesador del teléfono es limitado, por lo que no se deben realizar tareas muy complejas que podría afectar el tiempo de respuesta de la aplicación.
 - La forma de ingresar la información de datos no es muy cómoda, por lo que se debe buscar la mejor forma para pedir la información para facilitar el uso al usuario.
- La aplicación debe ejecutarse en el sistema operativo Windows Mobile 6 que implica considerar las siguientes restricciones en uso de lenguajes o librerías:
 - El lenguaje de desarrollo debe ser uno de los soportados por el Framework .NET 3.0 en el caso de una aplicación nativa.
 - En caso que sea un aplicación Web, debe estar soportado para ejecutar Internet Explorer Mobile.
 - En caso que la aplicación sea una aplicación para la máquina virtual de Java, debe estar programada en Java o un lenguaje que soporte la máquina virtual de Java.
- La aplicación tiene que tener la capacidad de conectarse al CRM, por lo que debe cumplir las siguientes restricciones:
 - Soportar la conexión a través del servicio web proveído por el servidor que aloja el CRM.
 - Proveer los datos necesarios para autenticación del usuario con el sistema.
 - Tener la capacidad de interpretar los archivos de respuesta del servidor que está dado en formato de XML con la especificación establecida por el servicio web del CRM.
 - La capacidad de cifrar y descifrar la información utilizando el método de cifrado TripleDES, ya que es un requerimiento de seguridad.
- 1. Para que la aplicación tenga soporte para mostrar la dirección del usuario y un mapa de su ubicación es necesario:
 - El soporte de GPS.
 - La capacidad de conectarse a los servicios de Google Maps.
 - La pantalla debe soportar el desplegar una imagen.

6. Documentación de usuario. El usuario final podrá obtener una copia del manual del usuario de la aplicación en formato PDF que estará disponible en el sitio del CRM. Este archivo contendrá una descripción general de la aplicación con el fin de que él pueda conocer para qué fue diseñada la aplicación y luego en cada sección se describirá cada una de las pantallas de la interfaz gráfica con unas capturas de pantallas de la misma. En cada una de estas secciones podrá conocer qué puede hacer en dicha pantalla y a qué otras pantallas el usuario podrá mover. Este documento está escrito de tal forma que el usuario pueda aprender más rápido cómo utilizar la aplicación.

7. Supuestos y dependencias. Para el desarrollo de esta aplicación es necesario considerar los siguientes supuestos y dependencias.

- Se supone que el usuario debe tener una cuenta en el CRM, ya que este sistema no provee un sistema para registrarse.
- Se espera que el usuario tenga todos los permisos necesarios para acceder a la información del CRM, esto depende del coordinador que tiene a cargo el sistema.
- Se espera que el usuario tenga conexión a internet para establecer la conexión con el CRM.
- El éxito de este proyecto depende en gran parte del éxito del proyecto que tiene a su cargo el desarrollo del CRM, ya que ésta será la fuente de información.

C. Características del sistema

1. Autenticación del usuario con el CRM.

a. Descripción y prioridad. El sistema tiene la capacidad de proveer al usuario una forma de autenticarse con el CRM con el fin de que pueda tener acceso a la información que le corresponde, además protegerá que otra persona intente hacer mal uso de su información. Tiene alta prioridad, ya que sin ésta no podrá realizar las otras características del sistema.

b. Secuencia de acciones.

- El sistema presenta una pantalla de autenticación o login al usuario.
- El usuario ingresa su dominio, nombre y contraseña de su cuenta en el CRM.
- Se comprobará con el CRM si la información de usuario ingresada es correcta.
 - En caso que sea correcta, el usuario podrá ingresar al sistema
 - En caso contrario se desplegará un mensaje de error

c. Requerimientos funcionales.

- Crear una pantalla de autenticación o login para el usuario
- Tener una forma de comprobar con el CRM que la información ingresada por el usuario es correcta.

2. Consulta de la lista asignada al usuario en el CRM.

a. Descripción y prioridad. Esta característica es muy importante, ya que es uno de los propósitos por el cual se desarrolló este proyecto. Consiste que el usuario podrá obtener el listado de tareas que tiene asignado. Esta información se obtendrá del servidor que aloja el CRM.

b. Secuencia de acciones.

- El usuario se autentifica con el sistema.
- El sistema consultará al CRM la información acerca de la lista de tarea que tiene asignado el usuario.
- El CRM deberá responder al sistema con la información solicitada al sistema.
- El sistema desplegará la información al usuario.

c. Requerimiento funcional.

- El sistema deberá tener la capacidad de obtener la información de la tarea de un usuario.
- Es necesario que el sistema tenga una forma simple y sencilla de presentar la información de la lista.
- El sistema deberá estar programado para manejar cualquier error de conexión y notificar al usuario.

3. Desplegar la información detallada de una tarea.

a. Descripción y prioridad. El usuario podrá obtener la información relacionada con una tarea y además podrá configurar qué campos se desplegarán en la pantalla. En caso que existan algunos campos que tengan más información, estos deberán presentarse como enlaces donde el usuario podrá consultar en otra pantalla la información restante. Esta característica también tiene una prioridad alta ya que sin ésta el usuario no tendría toda la información que necesita.

b. Secuencias de acciones.

- En la lista de tareas, el usuario selecciona una tarea.
- El programa solicita de la lista de tareas la información relacionada con la tarea seleccionada.
- Determina qué datos debe presentar al usuario según la configuración.
- Presenta la información al usuario.
- En caso que existan campos visibles con más información, estos se presentan como enlaces.
- Si el usuario selecciona un enlace se le presenta la información restante en otra pantalla.

c. Requerimientos funcionales.

- Que el sistema tenga la capacidad de presentar la información relacionada con una tarea.
- Que el usuario pueda configurar los campos que muestra en la pantalla.
- Que el usuario pueda acceder la información restante de un campo, en caso que existiera.

4. Editar la información relacionada con una tarea.

a. Descripción y prioridad. A través de esta característica, el usuario podrá proveer retroalimentación de la información de una tarea, ya que podrá editar esta información y los cambios serán enviados al servidor de CRM. Esta característica también tiene alta prioridad ya que complementa a la característica anteriormente descrita.

b. Secuencias de acciones.

- El usuario consulta la información de una tarea.
- Selecciona la opción de editar la información de la tarea.
- El sistema despliega un formulario con los campos que él puede modificar y que la configuración indica que se debe modificar.
- El usuario ingresa la información solicitada.
- El sistema verifica qué datos se deben enviar ya que se puede dar el caso que el usuario dejó un campo sin modificar.
- Envía los cambios al servidor CRM.
- El CRM guarda los cambios en sus bases de datos.
- Notifica al sistema que ya realizó el cambio.
- El sistema notifica que se realizaron los cambios.
- Despliega la información de la tarea modificada.

c. Requerimientos funcionales.

- El sistema debe proveer la capacidad de modificar la información de la tarea en los campos que se tiene permiso.
- El usuario puede configurar qué datos se le presentará en el formulario para editar las tareas.
- El sistema debe manejar cualquier error que ocurra cuando se esté enviando los cambios al CRM y notificar al usuario cualquier error.

5. Mostrar la lista de sub-tareas de una tarea seleccionada.

a. Descripción y prioridad. Esta característica amplía la funcionalidad de este sistema, ya que le permite al usuario consultar las sub-tareas que son necesarias para realizar una tarea. Se entiende que una sub-tarea es una tarea menor que ayuda a realizar un progreso en la ejecución de la tarea superior. Esto permite a los técnicos tener más organizadas las tareas que tienen que realizar y delegar a otros técnicos la responsabilidad del trabajo. Su desarrollo la complementa la anterior.

b. Secuencias de acciones.

- El usuario consulta la información de una tarea.

- Selecciona la opción de ver la lista de sub-tareas.
- El sistema consulta esta información al servidor de CRM.
- El CRM procesa la información y la envía al sistema.
- El sistema despliega la información al usuario de forma similar como si fuera la lista de tareas.

c. Requerimientos funcionales.

- El sistema tendrá la capacidad de desplegar la lista de sub-tareas de una tarea.
- El sistema deberá manejar los posibles problemas de conexión y notificarlo al usuario.

6. Asignar una sub-tarea a otro usuario.

a. Descripción y prioridad. Como se mencionó anteriormente, se consideró el tema de las sub-tareas, ya que permite a los técnicos tener un mejor orden de las tareas asignadas, por lo que es muy importante incluir esta característica, ya que le permite al usuario delegar la responsabilidad de trabajo de una parte de la tarea a otro usuario. La prioridad de esta característica es la misma que la anterior, ya que sólo aumenta la capacidad del sistema.

b. Secuencias de acciones.

- El usuario consulta la información de una tarea.
- Selecciona la opción de crear una nueva sub-tarea.
- El sistema presenta un formulario para ingresar la información de la nueva tarea.
- El usuario ingresa la información al formulario.
- El sistema envía esta información al servidor de CRM.
- El CRM procesa la información y la almacena.
- El CRM reporta si la operación tuvo éxito, en caso contrario, envía un mensaje de error.
- El sistema reporta el resultado al usuario.
- Si la operación tuvo éxito regresa a la pantalla donde se muestra la información de la tarea, en caso contrario regresa al formulario para que el usuario realice los cambios.

c. Requerimientos funcionales.

- El sistema debe tener la capacidad de crear sub-tareas.
- El sistema debe manejar cualquier error que ocurra cuando éste envía la información de la nueva sub-tarea.

7. Localización del usuario.

a. Descripción y prioridad. Esta opción le permite al usuario obtener la dirección y un mapa de su posición a través del GPS de su dispositivo, pero esta característica está deshabilitada por defecto, puesto que se necesita una licencia de Google para utilizar los servicios de Google Maps para propósitos

comerciales. Para las próximas versiones se considerará habilitar dicha característica. Cuando esté habilitada se podrá acceder desde la pantalla de lista de tareas. Dado su naturaleza, esta característica tiene baja prioridad con respecto a las demás.

b. Secuencia de pasos.

- El usuario ingresa a la pantalla de localización.
- Si se ha consultado con anterioridad su ubicación, esta pantalla mostrará la información actual.
- En caso contrario sólo mostrará en pantalla un mensaje que debe actualizar la información de ubicación.
- Si el usuario selecciona la opción de actualizar su ubicación, se realizarán los siguientes pasos:
 - La aplicación consulta al GPS la ubicación del usuario.
 - Con esta información, realiza una consulta al servicio de *geocoding* de Google Maps para obtener la dirección del usuario.
 - Además realiza una petición al servicio de mapas estático de Google Maps para obtener un mapa de la ubicación.
 - Despliega y guarda la información consultada.

D. Requerimientos de interfaces externas

1. Interfaz de usuario. La interfaz del usuario del sistema está compuesta por varias pantallas donde el usuario podrá ingresar los datos que se le soliciten y ver la información que le interesa. Las pantallas deberán ser sencillas para aprovechar el espacio y no sobrecargar el procesamiento con gráficas complicadas, además le permitirá al usuario tener una mejor interacción con el sistema y agilizar el proceso de aprendizaje de la interfaz.

La paleta de colores que se utilizará en las pantallas será la siguiente: fondo de color celeste para los elementos de la lista que utilicen una pantalla, fondo blanco para desplegar información de contenido y fondo color naranja para los mensajes de error. En cuanto al color de las letras que se desplegarán, deberán ser de color negro.

Para construir cada una de las pantallas se utilizarán varios componentes de software como, *textfield* para ingreso de información, *combobox* para seleccionar entre varias opciones, *label* para desplegar el contenido, los menús para navegación entre las pantallas y *checkbox* para los datos que sólo pueden tener dos valores. Los componentes para desplegar la lista se deberán hacer a partir de los componentes básicos para asegurar que no se requiera mucho procesamiento y se manejará un sencillo sistema de accesos directos con las teclas de los números para agilizar el ingreso a los elementos de dicha lista.

A continuación se presenta una descripción de las pantallas que se incluyen en el sistema:

a. Pantallas de autenticación. Será la primera pantalla que se le presentará al usuario cuando ejecute por primera vez el sistema. En esta pantalla se ingresará la información necesaria para autenticar el usuario con el servidor de CRM. Esta información es el dominio, el nombre y la contraseña de la cuenta del usuario en el CRM. Además, contendrá un apartado para indicar al sistema que el usuario desea guardar sus credenciales para futuras sesiones. El menú contendrá la opción de ir a la pantalla de configuraciones de la aplicación.

b. Pantalla de configuraciones. En esta pantalla el usuario podrá configurar la dirección del servidor de CRM y seleccionar el idioma de la aplicación. También se indicará la cantidad máxima de tareas que podrá solicitar la aplicación, ya con esto se evitará que en un futuro las peticiones del sistema sean tan grandes que afecte el desempeño de la aplicación.

c. Pantalla de lista de tareas. Después que el usuario se haya autenticado se le presentará esta pantalla donde se desplegará el listado de tareas que tiene asignado. Se espera que en esta lista se despliegue la información más importante de cada tarea para evitar sobrecargar la pantalla que puede afectar su usabilidad. Además debe contener un menú para tener la opción de configurar la aplicación, refrescar la información de la lista o salirse de la aplicación. Es importante mencionar que esta misma pantalla se debe utilizar para mostrar las sub-tareas, ya que esencialmente es la misma idea.

d. Visor de contenido. Esta pantalla se utilizará para desplegar la información detallada de un contenido, que puede ser una tarea o un dato relacionado con éste. La pantalla desplegará los campos que indiquen la configuración del visor que se podrá modificar en otra pantalla. Además contendrá un menú donde el usuario podrá navegar hacia las siguientes pantallas:

- Listado de sub-tareas.
- Crear sub-tareas.
- Editar la información de este contenido.
- Configuración del visor.

e. Editor de contenido. En esta pantalla se le presenta al usuario un formulario donde podrá modificar la información de un contenido. Los campos que podrá modificar están indicados en la configuración del visor, pero antes la pantalla de configuración comprobará si tiene permiso de modificar ciertos campos. Es importante mencionar que esta misma pantalla se utilizará para ingresar la información de una nueva tarea.

f. Pantalla de configuración del visor. Esta pantalla se utilizará para que el usuario indique qué campos desea que se muestren cuando vea la información de un contenido en el visor y qué campos podrá modificar en el editor. Es importante mencionar que esta pantalla comprueba cuáles son los campos que el usuario tiene permiso de modificar.

g. Pantalla de localización. En esta pantalla se desplegará un mapa y la dirección del usuario. Si la información que se despliega está desactualizada, el usuario tendrá la opción de actualizarlo utilizando el GPS de su dispositivo. Es importante notar que esta pantalla se desactivará al momento de utilizarla como una aplicación comercial. En las próximas versiones se considerará activarla.

2. Interfaz de hardware. Como se mencionó anteriormente, esta aplicación se diseñó para ejecutarse en un teléfono móvil que soporte el sistema operativo Windows Mobile 6, por lo que es necesario que utilice un conjunto de librerías que maneje la interacción con el hardware del teléfono con el fin de que la aplicación sea compatible en varios modelos de celulares. Además, si se desea que la aplicación tenga la opción de localización del usuario, se necesita que tenga acceso al GPS del sistema y una forma fácil de obtener la ubicación del usuario. La mejor opción para manejar estos aspectos es utilizar las librerías de Framework .NET, ya que éstas permiten manejar la entrada y despliegue de datos a través del uso de los componentes de interfaz gráfica.

Permiten, además, abstraer el manejo de comunicación con el servidor, ya que se utilizará una librería que maneja la conexión a través de Internet y el usuario sólo tendrá que preocuparse de configurar el acceso a Internet en las configuraciones generales de su dispositivo. También permite acceder de forma fácil a los dispositivos de GPS para obtener la ubicación actual del usuario.

3. Interfaz con software. Una de las interfaces que se debe manejar con otros paquetes de software es la del CRM. Como se mencionó anteriormente, el CRM es un proyecto que se desarrolló paralelamente. Utiliza como base Microsoft SharePoint 2010 que es un sistema diseñado para facilitar la colaboración de un grupo de trabajo a través de una plataforma para organizar actividades, almacenar y compartir documentos y tener un manejo de la información de los colaboradores. Sobre este sistema se construyó el CRM utilizando la gran variedad de librerías que proveen una abstracción de la base de datos a través de listas, lo cual facilita el desarrollo, ya que tiene a disposición un gran conjunto de componentes que manejan el mantenimiento de las listas. Para comunicarse con este sistema, se utilizó un servicio web que se describirá en la siguiente sección.

La otra es manejar las solicitudes para la información de la pantalla de localización. Google ha implementado un conjunto de servicios de *geocoding* y de mapas estáticos que permiten obtener

información de geoposicionamiento. A través de estos servicios se puede obtener direcciones legibles al ser humano proporcionando las posiciones geográficas, longitud y latitud, así como también un mapa del lugar. Para utilizar estos servicios es necesario seguir algunos lineamientos que Google ha establecido. Los cuales se describirán en las siguientes secciones.

4. Interfaces de comunicación. Para abstraer la comunicación con el CRM se utilizó un servicio web que se encarga de proveer un conjunto de funciones que facilitan la consulta y actualización de la información con el servidor. Este servicio web internamente se conecta con otros servicios *web* que proporciona Microsoft SharePoint 2010, que provee acceso a la información que está almacenada en las listas. Esta información es obtenida en formato de archivo de SOAP y luego el servicio web se encarga de formatear la información en un archivo de XML que CUI entiende.

El beneficio de utilizar un servicio web es que facilita la interacción con el servidor y permite aumentar la portabilidad de la aplicación a otros modelos. Es importante mencionar que el servicio web está desarrollado para ejecutarse sobre un servidor IIS, por lo que es importante que la base de CUI soporte este tipo de conexión. En las siguientes secciones se describe un conjunto de aspectos a considerar en la conexión con el servicio web.

Además al final de esta sección se describirán los lineamientos que una aplicación debe seguir para utilizar los servicios de geoposicionamiento de Google Maps. Fueron muy importantes al momento que la aplicación deseaba utilizar las capacidades del GPS y proporcionar al usuario una dirección que pueda entender.

a. Las peticiones a servicio web. Cada petición que se realiza al servidor es de forma transaccional, lo cual permite que en caso de que ocurra un error en la ejecución de la aplicación y deba cerrar ésta, todos los cambios que se realizaron en el servidor permanezcan. Cada petición debe incluir las credenciales del usuario para que el sistema determine si tiene permiso en realizar la solicitud o simplemente denegar el servicio. Estas credenciales están cifradas en algoritmo que los dos proyectos establecieron. Este algoritmo se describe en el apéndice 1 de este trabajo.

b. Las respuestas del servicio web. La respuesta que proporciona el servicio web son archivos de XML que están encriptados con el algoritmo de TripleDES. Una parte de la llave para descifrar el mensaje está incluida en los primeros 16 caracteres de la respuesta y la otra está establecida de antemano. Al unir estas dos partes se obtiene la llave que se utilizará para descifrar el resto de la respuesta para obtener el mensaje.

El mensaje tiene la siguiente estructura:

```
<? xml versión='1.0' encoding='utf-8'>
<root>
  <tipo> TIPO_MENSAJE</tipo>
  <data> LOS_DATOS </data>
</root>
```

Las primeras líneas indican la versión y la codificación de XML que se está utilizando. La siguiente sección indica el tipo de mensaje que se está recibiendo, el cual se identifica por un número dentro de la etiqueta tipo.

Tabla 2 Tipo de respuestas del servidor en archivo XML

Tipo	Numero	Descripción
Datos	0	Contiene una lista de elementos de información
Credenciales incorrectas	1	Indica que la petición incluía algún dato incorrecto de las credenciales
Falta algún dato de credencial	2	En la petición falta algún dato de las credenciales del usuario
Error al actualizar	3	Cuando el usuario realizó una petición de actualización y ésta tenga algún error, a través de esta respuesta se informa a la aplicación.
Éxito al actualizar	4	Es la respuesta que recibe la aplicación que la petición de actualización se realizó con éxito.
Lista de usuario	5	Indica que la respuesta es una lista de usuario del sistema.

La última parte del mensaje es la información principal y dependiendo del tipo, puede cambiar, pero en general se manejan dos tipos, los mensajes para el usuario y una lista de datos. El primer tipo se utiliza para las respuestas de error. Puede tener la siguiente estructura:

```
<data>
  <msgmessage="EL_MENSAJE">
    //Detalle del mensaje
  </msg>
</data>
```

Si es una lista de datos, se sigue la siguiente estructura:

```
<data>
  <listaName="NOMBRE_DE_LISTA_1" DisplayName="NOMBRE_PARA_DESPLEGAR">
    <ítem>
      <campoName="NOMBRE_CAMPO_1" DisplayName="Nombre a
desplegar">
        // valor del campo
      <campo>
```

```

        </item>
    </lista>
</data>

```

Como se puede ver en el ejemplo anterior, una lista de datos contiene en la etiqueta data con la información de varias listas del servidor, cada una con sus respectivos elementos. En cada elemento se indica los campos con su respectiva información.

c. Las funciones del servicio web. A continuación se presenta la lista de funciones que ofrece el servicio web:

- **getAllLists:** Esta función se utiliza para solicitar todos los datos que contiene la lista que el usuario tiene permiso a acceder en el sistema. Recibe como parámetro las credenciales de usuario, el dominio, el nombre y la contraseña.
- **getList:** Se utiliza para obtener cierta parte de una lista que el servidor almacena. Recibe los siguientes parámetros:
 - Dominio del usuario
 - Nombre del usuario
 - Contraseña del usuario
 - Nombre de lista que se está solicitando
 - Los campos o las columnas de lista que la respuesta debe incluir
 - El ID del primer elemento que incluyera la lista.
 - El ID del último elemento. Es importante mencionar que en el caso que estos dos parámetros tengan valor 0, se enviarán todos los elementos de la lista.
 - La lista de campos que deben cumplir una condición. Estos campos se envían como lista de elementos separados por coma.
 - La lista de valores que los campos del anterior parámetro deben cumplir. Se envía de forma similar como el parámetro anteriormente mencionado.
- **updateItem:** Esta función se utiliza para realizar ciertas tareas de manipulación con la información almacenada en el servidor. Recibe los siguientes parámetros, pero algunos pueden ir vacíos dependiendo de la acción que se desea realizar:
 - Dominio del usuario
 - Nombre del usuario
 - Contraseña del usuario
 - Nombre de la lista a modificar
 - El id del elemento que será modificado
 - Los campos que serán afectados
 - Los valores con que se realizará la modificación
 - Un parámetro que indica las acciones que se deben realizar. Las cuales pueden ser:
 - Update: actualizar un registro de la lista.
 - Delete: borrar un registro de la lista.
 - New: insertar un nuevo registro a la lista.

- **changeUserPos:** se encarga de actualizar la posición del usuario. Recibe los siguientes parámetros:
 - Dominio del usuario
 - Nombre del usuario
 - Contraseña del usuario
 - La longitud de la nueva posición
 - La latitud de la nueva posición
- **getUsers:** se utiliza para obtener el listado de los usuarios que están registrados en el sistema de CRM. Recibe como parámetro, las credenciales del usuario.

d. La solicitud de un mapa. Google ofrece un servicio para obtener una imagen del mapa de un lugar proporcionando sus coordenadas. Para realizar este proceso, sólo se necesita una conexión por HTTP con la siguiente información:

- La dirección del servicio: <http://maps.google.com/maps/api/staticmap>
- Reciben como parámetros en una petición de HTTP GET

Tabla 3 Parámetros de petición de mapas

Parámetro	Variables de GET	Descripción
Centro	center	El centro del mapa, que está dado por un par de números separado por comas que representa la longitud y la latitud
Tamaño	size	Indica el tamaño de la imagen
Formato	format	El formato de la imagen
Tipo de mapa	maptype	El tipo de mapa que se debe solicitar
Sensor	sensor	Indica si el sensor está activado
Móvil	mobile	Indica si el cliente es un dispositivo móvil

Como resultado será un archivo de imagen del mapa solicitado con las especificaciones dadas en la petición.

e. El servicio de geocoding. Como se menciona anteriormente, un servicio de *geocoding* permite obtener una dirección legible a los seres humanos dada la posición geográfica. Para realizarlo sólo se necesita establecer una conexión por HTTP con la siguiente información:

- La dirección del servicio: <http://maps.google.com/maps/api/geocode/xml>
- Recibe la longitud y latitud de la posición como un parámetro de GET llamado latlng.
- Además recibe un parámetro que indica si el sensor está activado.

Por ejemplo, si se desea saber la dirección de la posición 10,20 (longitud, latitud) la petición sería de la siguiente manera:

<http://maps.google.com/maps/api/geocode/xml?latln=20.10&sensor=false>

La respuesta de la petición es un archivo XML que contiene la siguiente estructura:

```
<? xml versión='1.0' encoding='utf-8'>
<data>
  <status>TIPO_RESPUESTA</status>
  <result>
    <formatted_address>DIRECCIÓN</formatted_address>
    // OTROS RESULTADOS
  </result>
</data>
```

En la etiqueta status se indica el tipo de resultado que se obtuvo de la petición:

Tabla 4 Tipos de respuesta del servicio de geocoding

Tipo de resultado	En el archivo	Descripción
Petición rechazada	REQUEST_DENIED	La petición realizada ha sido rechazada
Petición inválida	INVALID_REQUEST	La petición no sigue los lineamientos establecidos por Google
Ningún resultado	ZERO_RESULTS	A la posición dada no se obtuvo ningún resultado
Petición exitosa	OK	Se obtuvo un resultado

Si se obtuvo un resultado, se encontrará la respuesta entre las etiquetas de formatted_address, las siguientes líneas se utilizan para otros resultados.

E. Otros requerimientos no funcionales

1. Requerimientos de rendimiento. Dado que es una aplicación para la consulta y actualización que será utilizada por usuarios que desean aumentar su eficiencia, se necesita que el tiempo de respuesta de la aplicación sea muy alto. De esa manera se evitará que la aplicación sea una molestia para el usuario y lo encamine a utilizar más la aplicación. Por eso, es importante que la aplicación tenga pantallas que sean sencillas y no muy cargadas de componentes. Se espera que el tiempo que tarde en cargar una pantalla dependa de la velocidad de la conexión de internet.

2. Requerimientos de seguridad del hardware. No se permitirá que la aplicación guarde en caché información valiosa para evitar que personas no autorizadas tengan acceso a ella. En caso que el teléfono sea robado, el usuario deberá solicitar de forma inmediata al coordinador del servidor del CRM que bloquee la cuenta de usuario para evitar algún robo de información.

3. Requerimiento de seguridad de información. Dado que esta aplicación manejará información vital para la empresa que esté utilizando el sistema de CRM, es necesario que la aplicación móvil maneje esta información para evitar que sea un punto débil donde personas no autorizadas puedan tener acceso a la información y manipularla de manera indebida. Para describir cómo se manejó la seguridad informática en esta aplicación, se consideraron los siguientes cinco aspectos, que son los pilares de la seguridad informática explicado en el artículo de Larry Westfall. (Larry, 2005)

- **Confidencialidad:** indica la manera de asegurar que sólo personas autorizadas tengan acceso a la información. Para aspecto se decidió utilizar el manejo de cuentas de usuario que ya tiene implementada la base del proyecto del CRM, SharePoint; por lo que es necesario que cada usuario de esta aplicación disponga de una cuenta en el sistema CRM para tener acceso a la información.
- **Autenticación:** se refiere la manera de identificar correctamente al usuario con el sistema. Para cubrir este aspecto fue necesario implementar una pantalla de login o autenticación donde el usuario ingresaba sus credenciales, dominio, nombre y contraseña y esta información era guardada temporalmente en el sistema y enviada al servidor al momento de realizar cada petición para que el sistema se encargara de verificar si las credenciales eran correctas. Si eran correctas se devolvía la respuesta de su petición, en caso contrario sólo se devolvía un mensaje de error.
- **Integridad:** indica la manera como se asegura que la información sea modificada por personas autorizadas. Para manejar este aspecto, se dejó al lado del servidor, ya que al momento de crear al usuario en el servidor, el coordinador se encargaba de asignar a un grupo de técnicos donde estaba establecido cuáles son las tareas que pueden realizar en el sistema. Al momento de realizar una petición de actualización, después que el sistema haya autenticado al usuario, deberá verificar si éste tiene permiso de modificar dicha información, en caso contrario devolverá un mensaje de error.
- **No repudio:** no implementado, ya que necesita el uso de una tecnología como firmas digitales para evitar que el usuario niegue la responsabilidad de algo que realizó y se tiene comprobante de ese hecho.
- **Disponibilidad:** se refiere a la capacidad que tiene el usuario de tener acceso a la información en el momento que lo necesita. Este aspecto depende primero de la disponibilidad del servicio de Internet, que es un aspecto que debe velar el usuario; y segundo de la disponibilidad de servidor de CRM, ya que toda la información que maneja el sistema se almacena en dicho servidor y no se guarda ningún dato de las tareas del usuario en teléfono por motivo de memoria y seguridad.

4. Cualidades del software. Este software se diseñó con el fin de que sea sencillo y fácil de utilizar. Además, se deseaba que tuviera la capacidad de aprovechar algunas de las características de los teléfonos celulares modernos, como el GPS; por lo que se necesita que se acople correctamente al ambiente de ejecución. Además tiene flexibilidad en las pantallas, ya que el usuario puede seleccionar entre los dos idiomas que dispone la interfaz y determinar qué información podrá consultar de las tareas u otros contenidos. Para asegurar su robustez y correcto funcionamiento, se ha realizado una serie de pruebas para asegurar que cumple con los requerimientos establecidos. Se diseñó de forma modular con el fin de

que se pueda continuar el trabajo y agregar nuevas funcionalidades o trabajar una versión de este software en otra plataforma. Es importante mencionar que esta versión únicamente funciona en el sistema operativo Windows Mobile 6.

V. DISEÑO

A. Diseño arquitectónico

Para obtener un mejor resultado en el desarrollo de esta aplicación se decidió trabajar por módulos, donde cada módulo tiene su función específica dentro de la aplicación y debe cubrir uno o más de los requerimientos del sistema. A continuación se listan los módulos con una breve explicación:

- **Autenticación y conexión:** se encarga de obtener las credenciales del usuario y de manejar la conexión de la aplicación con el servidor. También maneja las peticiones a los servicios de Google Maps.
- **Lista de tareas:** se encarga de proveer una interfaz para listar las tareas que tiene asignado el usuario.
- **Visor de contenido:** implementa la pantalla del visor para mostrar la información que le interesa al usuario.
- **Editor de contenido:** se utiliza para manejar los formularios que el usuario usará para actualizar la información de un contenido o crear uno nuevo.
- **Configuración de visor y editor:** se encarga de manejar los archivos perfiles, que se utilizarán para indicar qué información manejará el visor y editor de un contenido.
- **Localización:** maneja la pantalla de localización y gestiona las consultas al GPS. Para realizar las peticiones a los servicios de Google utiliza las funciones del módulo autenticación y conexión.

B. Diseño de componentes

1. Introducción. Como se mencionó en el diseño arquitectónico, fue necesario establecer una división del desarrollo de la aplicación para facilitar el manejo del código. Se establecieron seis módulos con el fin de proveer a la aplicación de otras funciones y facilitar el mantenimiento. En esta sección se describirá detalladamente cada módulo mencionando cada una de las clases que lo conforman y las tareas principales que implementa.

2. Autenticación y conexión. Se puede mencionar que este módulo se encarga de establecer la conexión con el servidor de CRM y proveer una gran variedad de funciones que le permite a los demás módulos obtener la información.

A continuación se presenta una descripción de las clases y otras estructuras que componen este módulo.

a. WSAction.

1. Descripción. Es una enumeración que se utiliza para representar las posibles acciones que se pueden realizar con los métodos de la clase SourceAdaptor. A continuación se presentan las opciones:

Tabla 5 Posibles acciones en la clase SourceAdaptor

Valor	Descripción
VIEW	Se solicitará información de la lista que almacena el servidor
UPDATE	Se actualizará la información de un elemento de la lista
NEW	Se insertará un nuevo valor a una lista
GET_USERS	Se solicita la información de los usuarios del sistema
REVERSE_GEOCODING	Se solicita hacer una traducción de unas coordenadas geográficas a una dirección legible al ser humano

b. SourceAdaptor.

1. Descripción de la clase. Esta clase se encarga de ser un interfaz con el servicio web y proveer un conjunto de funciones que facilitan la consulta y actualización de la información.

2. Atributos.

- Una referencia a la clase principal de proyecto, para tener acceso a los demás módulos
- Una referencia a la configuración de la aplicación
- Una referencia al servicio web
- Una variable que indica el éxito de la operaciones

3. Métodos.

- **ChangeWBSPath:** se encarga de cambiar la dirección del servicio web.
- **RequestWebServices:** se encarga de proveer una interfaz para consulta y actualización de la información, requiere de algunos de los siguiente datos:
 - La acción que se realizará representado con un elemento de WSAction (obligatorio)
 - Las credenciales de usuario (obligatorio)
 - La lista de trabajo
 - La identificación del elemento
 - Los campos

- Los valores
- El rango de elementos a trabajar
- GetErrorMessage: se encarga de proveer un mensaje apropiado a una excepción dada
- GetTaskList: se encarga de realizar los pasos necesarios para obtener la lista de tareas

c. XMLParserResult.

1. Descripción. Es una enumeración que utiliza para representar los posibles resultados de la clase XMLParse. A continuación se presentan los posibles valores:

Tabla 6 Posibles resultados de proceso de parsear el archivo XML

Valor	Descripción
TASKS_LIST_XML	El archivo analizado contiene el resultado de una consulta a una lista
INCORRECT_USER_PSWD_XML	El servidor rechazó la consulta porque algún dato de las credenciales de usuario está incorrecta
NO_USER_PSWD_XML	Se rechazó la consulta porque faltaba algún dato de las credenciales del usuario
UPDATE_ERROR	Al momento de realizar una actualización ocurrió un error
UPDATE_SUCCESS	Indica que se completó una solicitud de actualización
USER_LIST_XML	El archivo analizado contiene información de usuario del CRM
REVERSE_GEOCODING	El archivo analizado contiene información de geocoding

d. XMLParser.

1. Descripción de la clase. Se encarga de analizar la respuesta obtenida del servicio web o los servicios de Google Maps, el cual viene en formato de un archivo XML y realiza las transformaciones necesarias para que sea útil a los otros módulos.

2. Atributos.

- Una referencia a la configuración de la aplicación
- El tipo de resultado que se obtuvo del análisis
- Una variable para guardar el resultado
- Algún mensaje que se obtuvo al final del análisis

3. Métodos.

- Parse: se encarga de realizar el análisis correspondiente a la información dada
- GetMessage: se encarga de obtener los mensajes que contiene la respuesta
- ParseReverseGeocoding: si la respuesta contiene información de geocoding, este método se encarga de extraer la dirección
- MakeList: cuando el resultado contiene una consulta de datos, se encarga de realizar una lista
- MakeNode: crea un elemento de la lista anteriormente mencionada
- MakeUserList: se utiliza para crear una lista que contiene información de usuarios
- MakeUserNode: crea un elemento de una lista de usuario
- SearchNode: busca un nodo en el árbol que se genera del archivo XML

e. MapWebRequest.

1. Descripción de la clase. Esta clase se utiliza como interfaz para realizar las peticiones a los servicios de Google Maps para obtener mapas estáticos de lugares o la dirección de un punto de la tierra, dadas sus coordenadas geográficas.

2. Atributos.

- Una referencia a la configuración de la aplicación
- Una configuración base para realizar las peticiones
- Una variable que indica si la operación tuvo éxito
- Los mensajes que se pueden obtener de las peticiones

3. Métodos.

- GetImage: se encarga de solicitar un archivo de imagen que contiene el mapa de un lugar dadas sus coordenadas geográficas y las configuraciones del archivo.
- ReverseGeocoding: dadas las coordenadas geográficas de un lugar, solicita una dirección legible a los seres humanos.
- GetFormattedAdress: cuando se realiza una petición de *Reverse Geocoding*, éste se encarga de extraer la dirección solicitada de la respuesta o un mensaje de error en caso ocurra un problema.
- StaticMapFormatUrl: dada una configuración, construye la dirección de URL para solicitar un mapa a los servicios de Google Maps.

- ReverseGeocodingFormat: proporciona la dirección de URL para solicitar una petición de *Reverse Geocoding*, dada una configuración.

f. Setting.

1. Descripción de la clase. Se utiliza para almacenar las configuraciones de la aplicación en el momento de la ejecución y además se encarga de cargar dicha información de los archivos XML. Es importante mencionar que hay otros archivos que guardan un dato en específico de la aplicación, como el idioma o la dirección del servidor, que la aplicación se encarga de gestionarlo.

2. Atributos.

- La lista de los mensajes para desplegar en la interfaz con sus debidas traducciones.
- La lista del nombre de los archivos perfil que contiene la configuración de los contenidos.
- El perfil que se está utilizando.
- El perfil que se utilizó anteriormente.
- Las configuraciones internas de la aplicación, como las constantes y la dirección de los archivos de configuración.
- El idioma de la interfaz.
- Los idiomas disponibles en la aplicación.
- La cantidad máxima de tareas que se pueden solicitar al servidor.

3. Métodos.

- Load: carga toda la configuración de la aplicación de los archivos XML.
- LoadProfiles: carga los archivos perfiles cuando inicia la aplicación, donde se almacena la configuración del editor y visor.
- GiveProfileXmlText: obtiene el contenido de un archivo perfil dado el nombre del contenido.
- LoadNewProfiles: carga un nuevo perfil para el editor y visor.
- RestoreProfile: regresa al perfil que anteriormente se estaba utilizando.
- GoBackProfile: se encarga de ejecutar las acciones necesarias para cargar el perfil anterior.
- LoadSelectLanguage: carga de un archivo la información de cuál es el idioma que está utilizando la interfaz.
- LoadNewSelectLanguage: cambia el idioma de la interfaz y lo guarda en una archivo.
- LoadWBSUrl: carga de un archivo la dirección del servicio web.
- SaveNewPath: cambia la dirección del servicio web y lo guarda en un archivo.

- LoadMessage: carga todos los mensajes de la interfaz del archivo XML de mensajes dependiendo del idioma seleccionado.
- LoadAColor: carga la paleta de color de la interfaz.
- LoadStringVar: carga la configuración interna de la aplicación.
- ReLoadLanguage: cuando hay un cambio de idioma, carga todos los mensajes de la interfaz.

g. SettingFormMode.

1. Descripción. Es una enumeración que se utiliza para indicar el modo en que se levantará la pantalla de configuración. Los posibles valores son los siguientes:

Tabla 7 Posibles modos de la pantalla de configuración

Valor	Descripción
LOGIN	A la pantalla de configuración se accede desde la pantalla de autenticación y debe permitir cambiar las opciones de servidor, idioma y cantidad de tareas.
TASKSLIST	Se accede a esta pantalla desde la lista de tareas, por lo que el usuario ya está autenticado y sólo debe permitir cambiar el idioma y la cantidad de tareas a solicitar.

h. SettingForm.

1. Descripción de la clase. Esta clase implementa la pantalla para las configuraciones de la aplicación. Se encarga de llamar al objeto apropiado para realizar los cambios necesarios.

2. Atributos.

- Un nombre para diferenciarlo de las demás pantallas.
- Una referencia a la clase principal.
- Una referencia a la configuración de la aplicación.

3. Componentes gráficos.

- El menú de la pantalla:
 - Guardar
 - Cancelar
- Un área de texto para cambiar la dirección del servidor.
- Un área para seleccionar el idioma de la aplicación.
- Una sección para cambiar la cantidad máxima de tareas a solicitar.

- Un área para mostrar los mensajes de error.

4. Métodos.

- Setup: se encarga de inicializar la pantalla y todos sus componentes.
- LoadLanguagesItem: carga en la sección de idiomas los idiomas disponibles.
- SaveLanguage: cuando guarda los cambios, este método se encarga de actualizar el idioma de la interfaz.
- SaveWebServices: actualiza la dirección del servicio web cuando el usuario haya guardado los cambios.
- SaveAmountOfTask: guarda los cambios que se realizaron a la cantidad de tareas a cargar.
- GoOut: sale de esta pantalla y regresa a la pantalla anterior.
- SaveMI_Click: maneja los eventos de la opción Guardar de la pantalla. Guarda los cambios realizados y regresa a la pantalla anterior.
- CancelMI_Click: maneja los eventos de la opción Cancelar de la pantalla. No guarda los cambios y sólo regresa a la pantalla anterior.
- PrintMSG: muestra el área de mensajes, cuando hay un error.

i. Login.

1. Descripción de la clase. Se encarga de implementar la pantalla de autenticación del usuario, y obtener las credenciales del usuario y guardarlas en la clase principal para luego utilizarlas en las peticiones que se realicen al servicio web.

2. Atributos.

- Un nombre para diferenciarlo de las demás pantallas.
- Una referencia a la clase principal.
- Una referencia a la configuración de la aplicación.

3. Componentes gráficos.

- El menú de la pantalla
 - Acceder
 - Menú:
 - Salir
 - Configuraciones

- Despliegue de mensajes
- Ingreso del dominio del usuario
- Área para ingresar el nombre del usuario
- Ingreso de la contraseña
- Checkbox para guardar las credenciales
- Un área para desplegar una barra de progreso
- *Timers* para controlar la ejecución de las peticiones

4. Métodos.

- Setup: se encarga de inicializar todos los componentes de la pantalla.
- SettingMI_Click: maneja los eventos del menú configuraciones. Cambia a la pantalla de configuraciones.
- LoginMI_Click: se encarga del menú Acceder. Para iniciar sesión.
- ExiMI_Click: cierra la aplicación.
- Wbtimer_Tick: se encarga de los eventos de timer para iniciar una petición para iniciar sesión.
- Guitimer_Tick: maneja los eventos de timer para levantar la pantalla de progreso.
- DomainTbx_KeyUp: maneja los eventos del teclado en la entrada para el dominio. Activa la opción de enter para iniciar sesión.
- UserTbx_KeyUp: se encarga de los eventos de teclado en la entrada del nombre de usuario. Activa la opción del enter.
- PswdTbx_KeyUp: se encarga de los eventos de teclado en la entrada de contraseña y activa el uso de enter en el campo de texto.
- SetupTitle: configura una etiqueta para utilizar como título.
- SetupWarning: configura una etiqueta para desplegar mensaje de error.
- SetupInput: configura un campo de texto para seguir el estilo de la pantalla.
- Authenticate: inicia la sesión de la aplicación, dependiendo si el usuario guardó sus credenciales, inicia automáticamente la sesión o levanta la pantalla para solicitar las credenciales.
- AuthenticateUsingFile: cuando se ha guardado las credenciales, este método se encarga de iniciar sesión.
- MakeLogin: obtiene la información ingresado por el usuario para iniciar sesión.
- LoadPosgressBar: levanta la pantalla de progreso.
- StopPosgresBar: esconde la pantalla de progreso.
- MakeRequest: inicia el proceso de iniciar sesión y carga la pantalla de progreso.

- **SendRequest:** sólo realiza la petición al servidor.
- **SaveData:** guarda la credenciales de usuario en un archivo.
- **ValidateCUI:** verifica si el usuario ya ingresó sus credenciales.
- **Print:** muestra un mensaje de error en el área de mensaje.
- **TurnOnMSM:** muestra el área de mensajes.

3. Lista de tareas. Este módulo se encarga de desplegar la lista de tareas solicitadas por el módulo de autenticación y conexión. A continuación se presenta una descripción de las clases que lo componen.

a. TasksListItem.

1. Descripción de clase. Se encarga de implementar el despliegue de un elemento de la lista. Dado que es un elemento para despliegue, hereda de la clase Panel todas estas funciones.

2. Atributos.

- Una referencia de la configuración de la aplicación.
- El número de elementos.
- La información a desplegar.

3. Componentes gráficos.

- Un panel para borde.
- Una etiqueta para la fecha.
- Una etiqueta para la descripción de la tarea.
- Una etiqueta para el id de la tarea

4. Métodos.

- **Constructor:** se encarga de inicializar este componente y llamar el constructor de la clase base.
- **PanelGotFocus:** cuando esta tarea gana enfoque en la lista, se encarga de modificarlo para resaltarlo.
- **PanelLostFocus:** cuando la tarea pierde enfoque, regresa a la normalidad.
- **TasksListItem_GotFocus:** se encarga de manejar el evento cuando el elemento gana focus.
- **TasksListItem_LostFocus:** se encarga de manejar el evento cuando el elemento pierde focus.

b. TasksList.

1. Descripción de la clase. Se encarga de implementar un componente para desplegar un conjunto de tareas. Está compuesto por una lista de objetos de la clase `TasksListItem`, donde cada uno almacena una tarea. Esta clase también se encarga de eventos y la interacción con la lista.

2. Atributos.

- Una referencia de la configuración de la aplicación.
- Una referencia a la clase principal.
- Una referencia al objeto que implementa el visor.
- Las dimensiones del área de despliegue.
- La posición inicial en la pantalla.
- Número máximo de tareas que se puede mostrar en la pantalla.
- La tarea seleccionada.
- La información a mostrar.

3. Componentes gráficos.

- La lista de objetos `TasksListItem`.

4. Métodos.

- `Setup`: se encarga de iniciar el componente, iniciando los eventos de teclado si es necesario. Determina si la lista está vacía para desplegar un mensaje a usuario.
- `CreateMessage`: se encarga de crear un área para desplegar mensajes en caso que no se vaya a desplegar la lista.
- `CreatePaneList`: se encarga de inicializar la lista de tarea, asignando a cada elemento la información que debe desplegar.
- `Parent_KeyUp`: maneja los eventos de teclado de la aplicación.
- `LoadTaskViewer`: en caso que se desea ver una tarea, este método se encarga de pasar la información al componente y llamarlo.
- `KeyPressE`: dado un evento del teclado, determina la tarea que se debe realizar.
- `MoveFocusAt`: cambia el enfoque de la lista a un elemento determinado.
- `MoveFocusUp`: mueve el enfoque de la lista a un elemento arriba.
- `MoveFocusDown`: mueve el enfoque de la lista a un elemento abajo.

c. *TaskListViewMode*.

1. Descripción. Es una enumeración que se utiliza para determinar el modo de vista de la lista de tareas. A continuación se presentan los posibles valores.

Tabla 8 Posible vista de la lista de tareas

Valor	Descripción
TASK	Muestra las tareas asignadas
SUBTASK	Muestra las sub-tareas de una tarea

d. *ShowTasksList*.

1. Descripción de la clase. Implementa la pantalla para desplegar la lista de tareas. Utiliza un elemento `TasksList` para crear la lista y en esta clase se agrega el menú de la pantalla.

2. Atributo.

- Un nombre para diferenciarlo de las demás pantallas.
- Una referencia a la clase principal.
- Una referencia a la configuración de la aplicación.
- Una referencia para el visor de contenido.
- Una lista para guardar las tareas.
- Una lista para guardar las sub-tareas.

3. Componentes gráficos.

- Los elementos del menú:
 - Ver tarea
 - Menú
 - Refrescar
 - Configuraciones
 - Localización
 - Cerrar sesión (en modo tareas)
 - Regresar (en modo sub-tareas)
 - Cerrar la aplicación
- El componente para mostrar la lista de tareas

- Timers para manejar la pantalla de progreso

4. Métodos.

- Setup: inicializa todos los componentes de la pantalla.
- RefreshMI_Click: maneja los eventos de la opción refrescar de menú y realiza un actualización de la información.
- SettingMI_Click: cuando se selecciona la opción configuración, se encarga de cambiar a la pantalla de configuración.
- GpsMI_Click: si se seleccionó la opción de localización, cambia a la pantalla de localización
- BackMI_Click: regresa a la pantalla anterior.
- LogoutMI_Click: se cerró la sesión y regresa a la pantalla de Login.
- ExitMI_Click: cierra la ejecución de la aplicación.
- ShowTaskMI_Click: se ha seleccionado ver el detalle de una tarea, por lo que carga el visor.
- LoadData: carga la información a la lista de tareas.

4. Visor de contenidos. Este módulo se encarga de mostrar detalles de los contenidos, los cuales son tareas o información relacionada con ellas, como cuentas, contratista y otras entidades. La forma de mostrar la información está descrita en un archivo de perfil, que contiene los campos, sus tipos y si se deben mostrar. De acuerdo a esta información se crea la pantalla.

a. ViewerItem.

1. Descripción de la clase. Implementa un componente para mostrar un campo de un contenido. Se encarga de formatear el valor y mostrarlo de la forma más requerida a su tipo.

2. Atributos.

- Una referencia a la configuración de la aplicación.
- Una referencia a la clase principal.
- Dato a mostrar.
- El ProfileItem que indica la manera como despegar el dato.

3. Componentes gráficos.

- Una etiqueta para mostrar el nombre del campo.
- Una etiqueta para mostrar el dato del campo.

- En caso que sea un enlace, se utilizará en vez del componente anterior un enlace.

4. Métodos.

- Setup: inicializa y configura el componente.
- linkLbl_Click: maneja el evento de link para mostrar el detalle relacionado con este campo.
- GetLabelFittingHeight: se encarga de ajustar la altura necesaria para mostrar correctamente el campo.

b. TaskViewMode.

1. Descripción. Es una enumeración para indicar qué modo se utilizará en el visor, puesto que esto depende del tipo de contenido que se cargará. Los posibles valores son los siguientes:

Tabla 9 Posibles vista del visor de contenido

Valor	Descripción
NORMAL	El visor desplegará contenido de tareas o sub-tareas
PLAIN	El visor trabajará con otro tipo de contenido

c. Viewer.

1. Descripción de la clase. Se encarga de implementar la pantalla para desplegar contenido. También se encarga de la navegación entre contenidos, es decir, cuando el usuario haya seguido una secuencia de enlaces.

2. Atributos.

- Un nombre para diferenciarlo de las demás pantallas.
- Una referencia a la clase principal.
- Una referencia a la configuración de la aplicación.
- Una pila para manejar los contenidos que se han visto, después de seguir una secuencia de enlaces.

3. Componentes.

- El menú de la pantalla
 - Regresar
 - Menú

- Ver sub-tareas (Modo normal)
 - Crear una sub-tarea (Modo normal)
 - Configurar visor
 - Editar contenido
- La lista de componentes para mostrar los campos
- Un área para desplegar mensajes
- Una barra de progreso
- Timers para manejar las peticiones

4. Métodos.

- Setup: inicializa y configura todos los componentes de la pantalla.
- LoadData: carga los componentes de las campos que se debe mostrar.
- SetupData: configura los componentes para desplegar los campos.
- SendRequest: se encarga de pedir la información para cargar la lista de sub-tareas.
- GetDataShow: cuando el usuario selecciona un enlace, este método se encarga de solicitar la nueva información al servidor y cargar el nuevo perfil para desplegar esta información.
- MoveNext: se encarga de iniciar el proceso de cambiar de contenido después de seleccionar un enlace.
- LoadEditForm: carga la información al editor y cambia a dicha pantalla.
- PrintMSG: muestra un mensaje en el área de mensajes.
- LoadingPanelOn: activa la barra de progreso.
- LoadingPanelOff: desactiva la barra de progreso.
- GoOut: se encarga de regresar a la pantalla anterior, que puede ser de contenido o lista de tareas.
- BackMI_Click: maneja las acciones necesarias después de seleccionar la opción del menú regresar.
- ShowSubtaskMI_Click: se encarga de cargar la lista de sub-tareas.
- CrearSubtaskMI_Click: se encarga de cargar el editor preparado para crear una nueva sub-tarea.
- SettingMI_Click: carga la pantalla para la configuración del visor.
- EditMI_Click: llama el método para cargar el editor de contenido.
- Lptimer_Tick: se encarga de activar la barra de progreso.
- Wbtimer_Tick: realiza la petición para cargar la lista de sub-tareas.
- Wbtimer2_Tick: realiza la petición para cargar un nuevo contenido.

5. Editor de contenidos. En este módulo se implementa la pantalla para editar contenido, especialmente tareas, además se utiliza para acceder a la pantalla para nuevas sub-tareas.

a. EditFormItem.

1. Descripción de la clase. Implementa un elemento que se utiliza para editar la información de un campo de un contenido. De acuerdo al tipo de campo, coloca la entrada apropiada.

2. Atributo.

- Una referencia a la configuración de la aplicación.
- Una referencia a la clase principal.
- Antiguo valor del campo.
- El ProfileItem del campo para indicar la forma de trabajarlo.

3. Componentes gráficos.

- Una etiqueta para mostrar el nombre del campo.
- Un campo de texto para los campos infinidad de datos posibles.
- Un combo box para los campos que tienen una lista posible de valores.

4. Métodos.

- Setup: inicializa y configura el componente.
- GetValue: obtiene el valor ingresado y lo proporciona en el formato más conveniente.
- GiveFocus: se encarga del enfoque del componente.
- Validate: valida que la información ingresada esté correcta.
- ValidatePercentage: valida en el caso de que sea de tipo porcentaje.
- ValidateNumber: valida si el valor ingresado es un número. Se utiliza para campos de tipo numérico.

b. EditTaskMode.

1. Descripción. Es una enumeración que indica el modo como operará el editor de contenido, lo cual puede afectar el modo de trabajar la información. Los posibles valores son:

Tabla 10 Posibles vistas del editor de contenido

Valor	Descripción
EDITING	Se está editando un contenido
CREATING	Se está creando una nueva sub-tarea

c. EditForm.

1. Descripción de la clase. Esta clase se encarga de implementar la pantalla de editor, la cual se utiliza para editar la información de un contenido o la creación de un nueva sub-tarea. Para editar la información utiliza unos archivos de tipo profile que contiene la información necesaria para manejar los campos de un contenido.

2. Atributos.

- Un nombre para diferenciarlo de las demás pantallas.
- Una referencia a la clase principal.
- Una referencia a la configuración de la aplicación.
- La información a modificar.
- Una referencia del contenido padre, en caso de las sub-tareas es la tarea principal.
- Información que sólo se dispone en el servidor.
- El modo de vista del editor.

3. Componentes gráficos.

- El menú de la pantalla
 - Guardar
 - Cancelar
- La lista de componentes para editar los campos del contenido
- El área para desplegar mensajes
- La barra de progreso

4. Métodos.

- Setup: se encarga de iniciar y configurar los componentes de la pantalla.
- SetupWithoutLoading: es cuando se carga la pantalla sin la necesidad de hacer una consulta al servidor para obtener información extra.

- SetupWithLoading: antes de cargar la pantalla, realiza una consulta al servidor para obtener información necesaria, por ejemplo las opciones que se tienen para una cuenta.
- RequestServerDataForLinkType: se encarga de realizar la petición de la información extra del servidor.
- RequestALinkTypeItem: se encarga de realizar una petición por un campo de la información extra.
- RequestANormalLinkTypeItem: solicita información para los campos normales.
- RequestAUserTypeItems: se encarga de los campos que contienen información del usuario.
- PrintLoadingOperationMessage: cambia los mensajes de la barra de progreso.
- SetupLoadingPanel: configura la barra de progreso.
- SetupMenu: configura el menú de la pantalla.
- LoadComponentesList: carga la información de los componentes.
- LoadAcomponent: carga la información de un componente.
- SetupComponents: se encarga de configurar los demás componentes de la pantalla, como el área de mensaje.
- SetupTimers: se encarga de configurar los timers de la pantalla
- LoadComponentsList: levanta la información de los componentes para editar.
- LoadAComponente: levanta la información de acuerdo al perfil.
- SetupComponetsList: configura todos los componentes para editar.
- LoadingPanelOn: levanta la barra de progreso.
- LoadingPanelOff: esconde la barra de progreso.
- SendRequest: realiza las peticiones al servidor para actualizar la información.
- GetParameters: al momento de realizar una actualización al servidor, este método se encarga de obtener todos los cambios realizados en los componentes de edición.
- Validate: verifica que todos los datos estén correctos.
- GoOut: regresa a la pantalla del visor.
- KeepAndGo: actualiza la información y regresa a la pantalla del visor del contenido.
- PrintMSG: imprime un mensaje en el área de mensajes.
- Lptimer_Tick: levanta la barra de progreso cuando se está realizando una actualización.
- Wbtimer_Tick: se encarga de realizar una solicitud de actualización.
- Lp2timer_Tick: se encarga de levantar la barra de progreso cuando inicia esta pantalla.
- Wb2timer_Tick: solicita la información extra para editor.
- SaveMI_Click: guarda los cambios realizados en la pantalla.

- CancelMI_Click: no realiza ningún cambio y regresa a la pantalla del visor.

6. Configuración del visor y editor. Para permitir que el usuario pueda configurar qué campos se deben mostrar en el visor o que se pueda modificar al momento de actualizar la información de un contenido, se creó el concepto de profile. Son archivos de XML que guardan la información de los contenidos que se pueden trabajar en la interfaz y en cada archivo se describe la información general del contenido y los campos. Además indica cómo se debe manejar dichos campos al momento de utilizar el visor o el editor.

a. ProfileItemType.

1. Descripción. Es una enumeración que utiliza para representar los tipos de ProfileItem que se dispone. Esta información es muy importante al momento de trabajar con un campo que esté relacionado con el ProfileItem. Los posibles valores son los siguientes:

Tabla 11 Tipos de campos de un perfil

Valor	Descripción
TEXT	El campo es de tipo texto
FORMATTING_TEXT	El valor del campo necesita un tipo de formateo antes de utilizarse
TEXT_AS_NUMBER	El campo es texto que se debe tratar como un número
NUMBER	El campo es un número real
PERCENTAGE	El campo guarda un valor de porcentaje
CHOICES	El campo contiene varias opciones predefinidas
DATE	Se guarda una fecha en el campo
LINK	Es campo que está relacionado con un registro de una lista y se muestra como un link en el visor y cuando se edita, se solita las opciones al servidor
EDITING_LINK	Es similar al campo link, pero no se puede editar
USER	El campo contiene información de usuario

b. ProfileItem.

1. Descripción de la clase. Esta clase se utiliza para guardar la información de cómo trabajar un campo de un contenido. Esta información se obtiene de un archivo XML de configuración del contenido.

2. Atributos.

- Una referencia a la configuración de la aplicación
- Las características del campo:

- Nombre a desplegar
- Nombre en la lista
- Tipo
- Es necesario solicitarlo
- Se puede editar
- Se puede mostrar
- Se debe pedir al momento de crear un nuevo elemento
- Las posibles valores que puede tomar, si es del tipo CHOICES.

3. Métodos.

- LoadXmlNode: carga la información del campo de un nodo de un archivo XML.
- LoadChoices: carga la sección de opciones, si el campo lo requiere.
- FormatData: formatea un valor dado de acuerdo al tipo del campo.
- FormatChoices: formatea un valor si el campo es del tipo CHOICES.
- FormatLink: formatea un valor si el campo es del tipo LINK.
- FormatDate: formatea un valor si el campo es una fecha.
- FormatList: formatea un valor de una lista.
- FormatPorcentaje: formatea un valor si el campo es un porcentaje.
- GetLinkID: si el campo es un LINK, obtiene su ID en la lista.
- GetDefault: provee el valor por defecto dependiendo del tipo del campo.
- GetUserInfoFields: formatea la información si el campo es de tipo usuario.

c. Profile.

1. Descripción de la clase. Esta clase provee objetos que se utilizan para manipular y trabajar los archivo perfiles, que son los que contiene la información de cómo la interfaz trabajará los contenidos en el visor y el editor.

2. Atributos.

- Una referencia de la configuración de la aplicación.
- Los títulos que se utilizarán en el visor o editor.
- La información de configuración del contenido.
- Los objetos que representan los campos del contenido.
- El archivo XML que contiene la información.

3. Métodos.

- LoadXmlString: carga la información del archivo XML.
- LoadMessages: carga los mensajes de un nodo XML, de acuerdo al idioma del sistema.
- LoadItem: carga la información de un campo de un nodo de XML.
- LoadDictionary: carga la información de un nodo a un diccionario.
- GetRequestingItemList: obtiene la lista de campos que se deben solicitar en una petición al servidor.
- FormatData: formatea toda la información del contenido para desplegar en una pantalla.
- WhoHasYesGivenType: este método se utiliza para determinar qué campos se debe solicitar dado el tipo.
- SaveChanges: guarda las modificaciones realizadas al archivo de XML.
- SearchItem: busca un ProfileItem Objeto dado el nombre del campo.
- GetDefaultDataDic: construye una estructura de datos con los valores por defecto de un contenido.

d. ProfileEditItem.

1. Descripción de la clase. Esta clase implementa los componentes que se utilizan para editar la información de un campo de un perfil para indicar si el campo se debe mostrar o modificar en el editor.

2. Atributos.

- Una referencia a la configuración de la aplicación.
- Una referencia a la clase principal.
- La referencia de ProfileItem que utilizará para modificar la información del campo.

3. Componentes gráficos.

- Etiqueta para nombre del campo.
- Una checkbox para indicar si el campo se muestra en el visor.
- Una checkbox para indicar si el campo se puede editar en el editor.

4. Métodos.

- Setup: inicializa y configura los elementos de este componente.
- GetDefaultSizeOfItem: indica el tamaño por defecto de una etiqueta.

e. ProfileEdit.

1. Descripción de la clase. Se encarga de implementar la pantalla de configuración de visor y editor. Esta clase se encarga de proveer una interfaz gráfica para manipular la información del archivo perfil para indicar qué campos se podrán mostrar en el visor o modificar en el editor.

2. Atributos.

- Un nombre para diferenciarla de las demás pantallas
- Una referencia a la configuración de la aplicación
- Una referencia a la clase principal
- La referencia del objeto profile que se utilizará para modificar la información del archivo perfil.

3. Componentes gráficos.

- El menú de la pantalla
 - Guardar
 - Cancelar
- La lista de componentes ProfileEditItem para editar cada campo.

4. Métodos.

- Setup: inicializa y configura los componentes de la pantalla.
- LoadItems: carga los componentes para editar campos.
- SetupItems: configura los componentes para editar campos.
- GoBack: regresa a la pantalla del visor del contenido.
- SaveMI_Click: guarda los cambios y regresa a la pantalla de visor.
- Cancel_Click: no guarda los cambios y regresa a la pantalla de visor.

7. Localización. Este módulo se utiliza para manejar la pantalla de localización. Utiliza un conjunto de librerías que facilitan el acceso a GPS para obtener las coordenadas geográficas del dispositivo. Las librerías están incluidas en los ejemplos del SDK para desarrollar aplicaciones en Windows Mobile 6. Para manejar las peticiones de geocodificación se utilizaron algunas clases que se implementaron en el módulo de autenticación y conexión. MapWebRequest para las peticiones y XMLParser para interpretar los resultados.

a. LocationForm.

1. Descripción de la clase. Esta clase se encarga de implementar la pantalla de localización para mostrar la ubicación del usuario. También se encarga de interactuar con GPS utilizando librerías de ejemplo de Microsoft.

2. Atributos.

- Un nombre para diferenciarla de las demás pantallas.
- Una referencia a la configuración de la aplicación.
- Una referencia a la clase principal.
- Las referencias de los objetos necesarios para interactuar con el GPS.

3. Componentes gráficos.

- El menú de la pantalla
 - Regresar
 - Actualizar
- Un área para desplegar las coordenadas del dispositivo
 - Longitud
 - Latitud
- Una etiqueta para desplegar la dirección de las coordenadas guardadas
- Un área para desplegar el mapa generado por Google Maps
- La barra de progreso
- Un área para desplegar mensajes
- Timers para controlar las peticiones

4. Métodos.

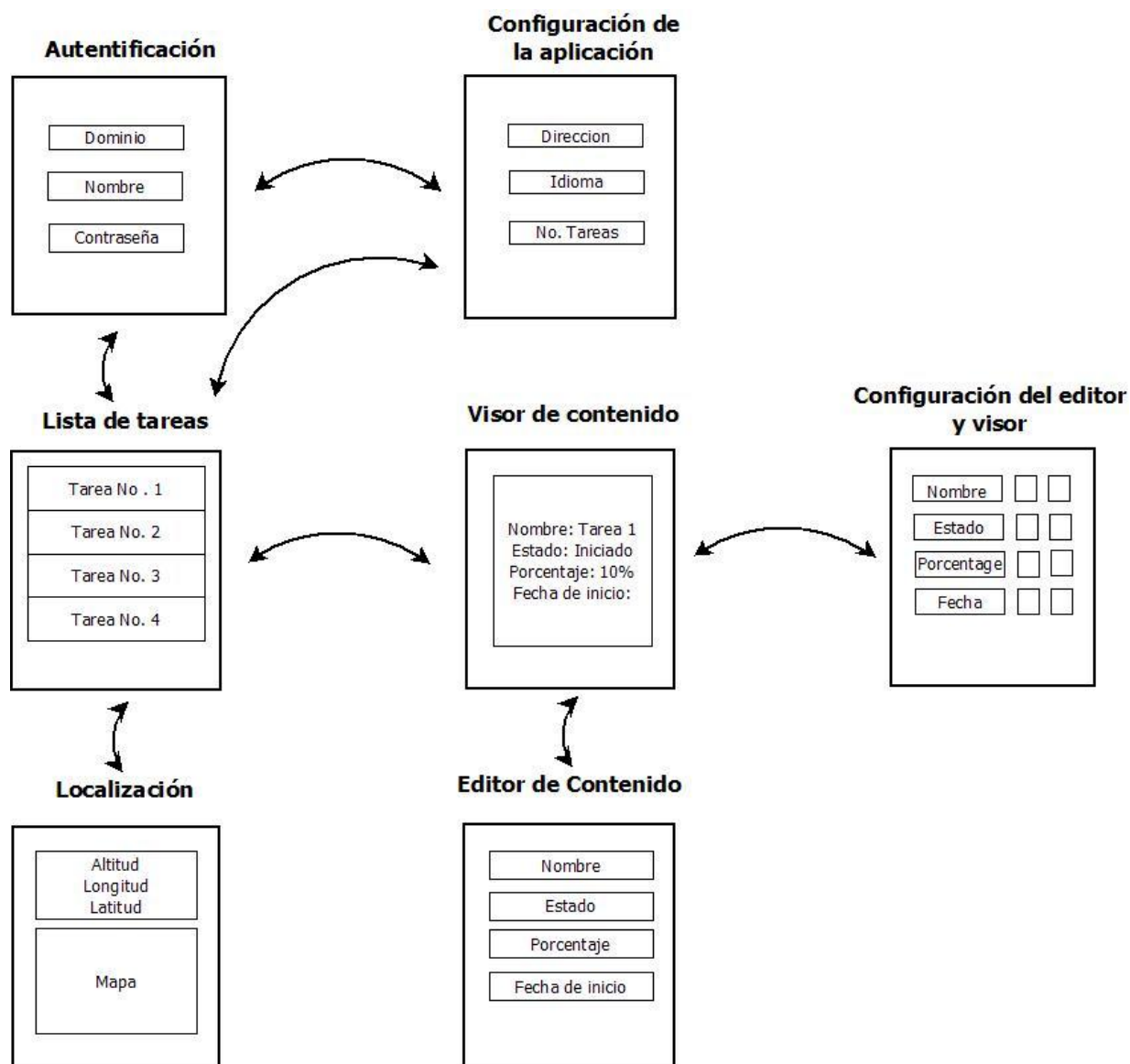
- Setup: inicializa y configura los componentes de esta pantalla.
- Setup: configura el menú de la pantalla.
- SetupMessagePanel: configura el panel para mensajes.
- SetupMap: configura el área para desplegar mapas.
- SetupLabel: configura el área para desplegar la información de la ubicación.
- SetupLoadingBar: configura la barra de progreso.

- InitializeGPS: inicializa los objetos que maneja el GPS.
- LoadGPS: carga el servicio de GPS.
- Gps_LocationChange: método que se encarga de actualizar los campos de posición del GPS.
- UpdateData: actualiza la información del GPS, tanto en la pantalla como los archivos para guardar la información de GPS para próximas sesiones.
- LoadData: carga la información del GPS guardada en las sesiones anteriores.
- Setup: inicializa y configura los timers para las peticiones.
- maploadingTimer_Tick: carga la barra de progreso para peticiones del mapa.
- Maptimer_Tick: realiza la petición del mapa.
- Changelptimer_Tick: realiza un cambio en la barra de progreso.
- Loadlptimer_Tick: carga la barra de progreso.
- Wbstimer_Tick: realiza peticiones a los servicios de google para geocodificación.
- Gpstimer_Tick: realiza la petición al gps para actualizar las coordenadas del usuario.
- SendRequest: se encarga de realizar la petición para geocodificación.
- SendRequest2: actualiza las coordenadas del usuario al servidor de CRM.
- UpdateDataDisplay: actualiza la información de la pantalla.
- PrintMSG: muestra un mensaje en el área de mensajes.
- MapLoadingOn: se utiliza para indicar que se está cargando un mapa en la pantalla.
- MapLoadingOff: en caso que no haya mapa, esconde el componente.
- FormatData: formatea los datos obtenidos del GPS.
- RefreshMI_Click: actualiza la información de la pantalla.
- BackMI_Click: regresa a la lista de tareas.

C. Interfaz de usuario

1. Introducción. En esta sección se describe en forma detallada la interfaz del usuario que se utilizó en CUI. Como se especificó en la sección de requerimientos, la interfaz se dividió en varias pantallas, donde cada una tiene su propia función en la aplicación, por lo que a continuación se presenta cada una de estas pantallas con una descripción breve de su función, los componentes de la interfaz y los eventos de los menús. Además se describirá la navegación general de la aplicación, es decir, cómo las pantallas están relacionadas entre sí. Esta parte se representa de una mejor forma en la siguiente imagen.

Ilustración 9 Esquema de navegación en CUI



Esquema de la navegación en la aplicación

Después se presentan los archivos de configuración que se utilizaron en la aplicación para permitir que la aplicación pudiera modificarse. Los archivos están en formato XML, ya que en Framework .NET se dispone de una gran cantidad de librerías que facilitan la lectura y manipulación de estos archivos, lo cual permitió que la aplicación se desarrollara más rápido.

2. Pantalla de autenticación.

Ilustración 10 Una captura de la pantalla de autenticación



a. Descripción. Esta pantalla también recibe el nombre de pantalla de Login, ya que le permite al usuario ingresar su información de usuario para iniciar la sesión en la aplicación para consultar y actualizar la información que le interese. Cuando la aplicación se ejecuta la primera vez, se presenta en pantalla para ingresar sus credenciales para el servidor de CRM.

b Componentes gráficos. El espacio principal de la pantalla está compuesto por los siguientes elementos:

- **Ingreso del dominio:** está compuesto con una etiqueta y un campo de texto para ingresar el dominio del usuario en el CRM.
- **Ingreso del usuario:** en esta parte se ingresa el nombre de la cuenta de usuario a través de un campo de texto.
- **Ingreso de la contraseña:** contiene un campo de texto donde el usuario podrá ingresar su contraseña; es importante mencionar que los caracteres que se ingresen serán sustituidos por caracteres asterisco para evitar que otra persona conozca la contraseña.
- **Checkbox para guardar las credenciales:** con este componente el usuario le informará a la aplicación que guarde sus credenciales para futuras sesiones. Esta información se almacena en un archivo en la carpeta del ejecutable del sistema. Para evitar el problema de que la información sea vista por personas ajenas, se guarda cifrada utilizando un algoritmo diseñado en este proyecto que se describirá más adelante.

c. Eventos del menú. El menú principal está compuesto por los siguientes elementos:

- **Acceder:** Se encarga de indicar a la interfaz que inicie la sesión con el servidor. Éste, primero verifica que el usuario haya ingresado toda la información solicitada. En caso que falte algún campo, se detendrá el proceso de autenticación y se notificará al usuario que deberá ingresar la información faltante. Si él ya ingresó toda la información necesaria, se determina si seleccionó la opción de guardar sus credenciales para realizar este procedimientos antes de iniciar la petición. Cuando se esté realizando la petición se le presentará una pantalla de cargando mientras se realiza la autenticación. Si ocurre un error en el proceso se regresa a la pantalla de Login y se despliega el error. En caso que se logró la autenticación, se presenta la pantalla de tareas.
- **Menú:** Presenta un pequeño menú con las siguientes opciones:
 - Configuración: Se mueve a la pantalla de configuración de la aplicación.
 - Salir: Cierra la aplicación.

2. Pantalla de configuración.

Ilustración 11 Una captura de la pantalla de configuración



a. Descripción. Esta pantalla se utiliza para que el usuario configure las principales opciones de la aplicación. Además se indica que la pantalla cuenta con dos vistas, una antes de iniciar sesión y la otra cuando se ha iniciado la sesión. La única diferencia es que en la segunda vista se presenta el área para ingresar la dirección del servidor, ya que se necesita que el usuario no tenga abierta su sesión.

b. Componentes gráficos. A continuación se describirá los componentes que forman esta pantalla:

- **Área del servidor:** se utiliza para ingresar la dirección del servidor de CRM. Se utiliza un campo de texto para modificar dicha información. Este campo de texto es opcional, ya que no aparece cuando no se ha iniciado sesión.
- **Lenguaje:** es un combobox donde se selecciona el idioma de la aplicación. Actualmente se cuenta con una versión en español y su traducción al idioma inglés.
- **Número de tarea:** es un campo de texto donde el usuario ingresa la cantidad máxima de tareas que se puede desplegar en la pantalla de la lista de tareas.

c. Eventos del menú. El menú presenta las siguientes dos opciones:

- **Guardar:** se encarga de guardar los cambios en el archivo de configuración de la aplicación. Antes de realizar este proceso, comprueba si han ingresado los campos, si no despliega un mensaje de error para que el usuario ingrese la información faltante.
- **Cancelar:** no guarda los cambios y sólo se regresa a la pantalla de autenticación.

3. Lista de tareas.

Modo tarea

Ilustración 12 Una captura de la pantalla de tareas



Modo sub-tarea

Ilustración 13 Una captura de la pantalla de sub-tareas



a. Descripción. Se utiliza para desplegar el listado de las tareas asignadas con la información principal de cada una de ellas. También, para desplegar la información de las sub-tareas, ya que el procedimiento es muy similar.

b. Componentes gráficos.

- **Lista de tareas:** es un componente que se desarrolló en este proyecto, ya que las listas que provee el Framework .NET tiene limitaciones en cuanto a aspecto gráfico y forma de uso. Además se desarrolló de tal forma que fuera sencilla, fácil de usar y no necesitara de mucho procesamiento. En caso que la lista no cuente con ningún elemento para desplegar, se notificará al usuario con un mensaje.

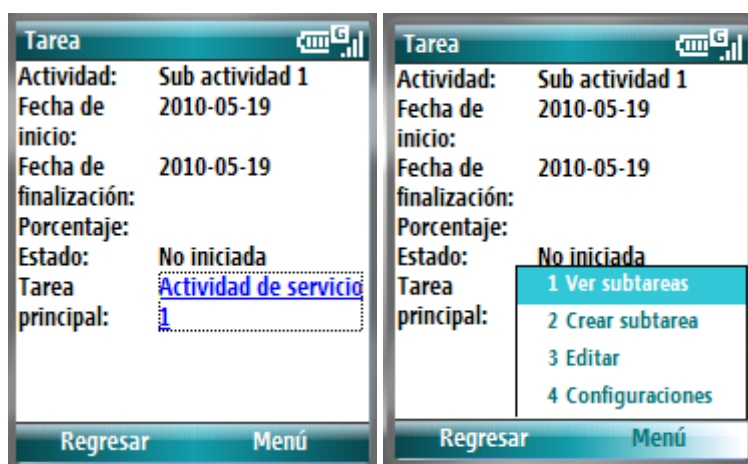
c. Eventos del menú. El menú cuenta con los siguientes elementos:

- **Ver tarea:** se mueve al visor de contenido para mostrar información detallada de la tarea seleccionada.
- **Menú:** es un sub-menú con las siguientes opciones:
 - **Refrescar:** realiza una petición al servidor para actualizar la información de las tareas.
 - **Configuración:** se mueve para la pantalla de configuración de la aplicación.
 - **Mi posición:** cambia a la pantalla para mostrar la información del GPS.
 - **Log out** (en la vista de tarea): cierra la sesión del usuario y se dirige a la pantalla de autenticación para que el usuario pueda iniciar otra sesión.

- **Regresar** (en la vista de sub-tarea): regresa al visor de contenido con la información detallada de la tarea principal.
- **Salir**: cierra la aplicación.

4. Visor de contenidos.

Ilustración 14 Una captura de la pantalla de visor de contenido



a. Descripción. En esta pantalla se desplegará la información detallada de un contenido, el cual puede ser una tarea, las cuentas de tarea o un incidente. El visor se desarrolló desde cero, ya que igual a las listas, no se disponía de un componente que cumpliera con requerimientos dados del visor, ya que la información que se despliega, está indicada en un archivo de configuración y además que fuera sencillo y fácil de utilizar. Este visor está compuesto con una lista de campos, los cuales se clasifican en dos tipos, normal y enlazado.

b. Componentes gráficos.

- **Campos normales:** se encargan de desplegar el valor de un campo, el cual no está relacionado con más información, por lo que se puede presentar en una etiqueta normal (label).
- **Campos enlazados:** presenta el valor de un campo que a su vez está relacionado con la información de una fila de otra tabla, por lo que se presenta como un link de tal manera que si el usuario escoge este link, se mueve a la información detallada de este campo.

c. Eventos de menú.

- **Regresar:** se mueve a la pantalla anterior que puede ser la lista de tareas u otro visor con la información de un dato relacionado.

- **Menú:** se presenta un sub-menú con las siguientes opciones:
 - **Ver sub-tareas:** es opcional, ya que sólo se despliega en los contenidos que sean tareas. Dirige al usuario a la lista de sub-tareas, que tiene un funcionamiento similar al de tareas.
 - **Crear una sub-tarea:** al igual que la anterior, también es opcional ya que sólo es necesaria para contenidos de tareas. Esto dirige al usuario a una pantalla con un formulario para ingresar la información de la nueva tarea.
 - **Editar:** mueve al editor de contenido que se utiliza para editar la información de un contenido.
 - **Configuraciones:** cambia a la configuración del visor y editor, donde el usuario podrá indicar qué campos se desplegarán en el visor y qué datos se podrán modificar en el editor.

5. Editor de contenido.

Ilustración 15 Una captura de la pantalla de editor de contenido

Nueva tarea	
Actividad	<input type="text"/>
Fecha de inicio	<input type="text" value="2010-05-19"/>
Fecha de finalización	<input type="text" value="2010-05-19"/>
Porcentaje	<input type="text" value="00.00"/>
Teléfono	<input type="text"/>
Estado	<input type="text"/>
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

a. Descripción. Esta pantalla se utiliza para editar la información de un contenido seleccionado. Los campos que se pueden modificar están indicados en el archivo de configuración del visor, por lo que esta pantalla se genera en tiempo de corrida. Los campos se clasifican en numéricos, texto y de opciones y dependiendo del tipo se determina el componente que se utilizará para recibir los cambios. Es importante indicar que esta pantalla también se utiliza para crear nuevas sub-tareas, sólo que se presentan todos los campos que deben llenar para guardar una nueva tarea.

b. Componentes gráficos.

- **Campos numéricos:** se modifican a través de un campo de texto que automáticamente configura el teclado del dispositivo para sólo ingresar números.

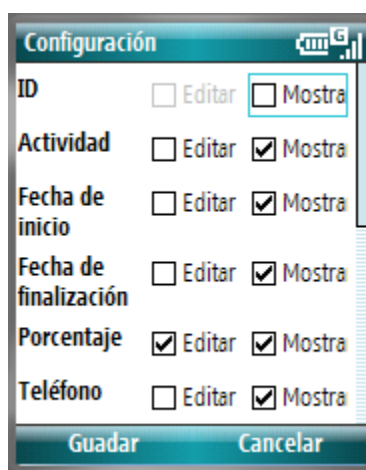
- **Campos de texto:** se utiliza un campo de texto para recibir la nueva información. Para facilitar el ingreso, la aplicación selecciona el modo texto.
1. **Con opciones:** son campos que tienen un cierto número de opción para tomar, por lo que se despliega en un ComboBox. Si no se tienen las opciones en la memoria, éstas se solicitan al servidor al inicio de la pantalla y se le presenta un pantalla de cargando mientras se completa la petición.

c. Eventos de ratón.

- **Guardar:** envía los cambios realizados en el formulario al servidor para realizar la actualización. Si el proceso tuvo éxito regresa a la pantalla del visor del contenido, en caso contrario permanece en esta pantalla y despliega un mensaje de error.
- **Cancelar:** no guarda ningún cambio realizado y sólo se regresa a la pantalla del visor del contenido que se estaba modificando.

6. Pantalla de configuración del visor.

Ilustración 16 Una captura de la pantalla de configuración del visor



a. Descripción. Esta pantalla se encarga de administrar el archivo de configuración del visor de un tipo de contenido, por lo que cada tipo de contenido tiene su propia configuración. En esta pantalla se colocan todos los posibles campos de un contenido y el usuario escoge cuáles serán visibles en el visor de contenido y modificables en el editor de contenido. La pantalla está organizada por filas, donde cada fila representa un campo de una lista.

b. Componentes gráficos. Los elementos de cada fila tienen los siguientes elementos:

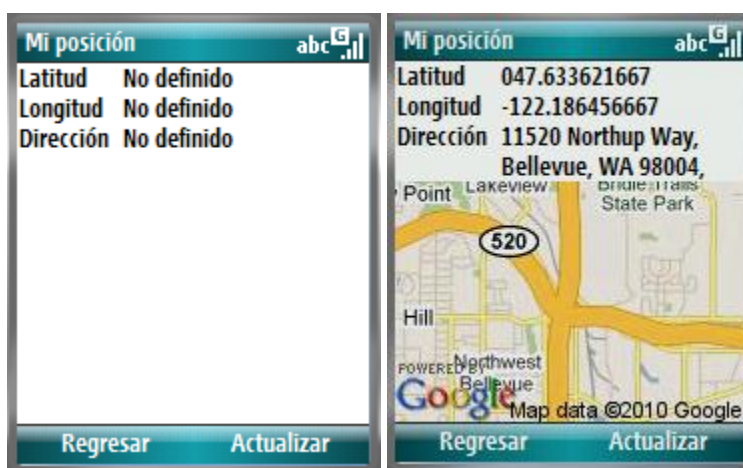
- **Etiqueta:** muestra el nombre del campo.
- **Editar:** indica que el campo se podrá modificar en el editor y si está deshabilitado es porque el usuario no tiene permiso de modificar este campo.
- **Mostrar:** Indica si el campo será visible en el editor del usuario.

c. Eventos del menú.

- **Guardar:** guarda los cambios en el archivo de configuración y se dirige al visor.
- **Cancelar:** sólo regresa el visor, ya que el usuario decidió no guardar los cambios.

7. Pantalla de localización.

Ilustración 17 Una captura de la pantalla de localización



a. Descripción. En esta pantalla, el usuario podrá consultar y actualizar la información de su localización. Esta información incluye la longitud y latitud, una dirección legible a los seres humanos y un mapa del lugar. El proceso de actualización involucra consultar al GPS las nuevas coordenadas y a través de los servicios de Google Maps obtener la dirección y el mapa. La primera vez que se ejecuta esta página no se mostrará la información de localización, por lo que el usuario deberá actualizar dicha información para futuras consultas.

b. Componentes gráficos.

- **Etiqueta de latitud:** despliega la última latitud guardada en el sistema.
- **Etiqueta de longitud:** muestra la última longitud registrada.

- **Etiqueta de dirección:** indica la dirección correspondiente a las coordenadas anteriormente mencionadas.
- **Una imagen:** se utiliza para desplegar el mapa de la posición.

c. Eventos del menú.

- **Actualizar:** se encarga de hacer el proceso de actualización de la información.
- **Regresar:** regresa a la lista de tarea.

7. Archivos de configuración.

a. Introducción. Después de la presentación de la interfaz gráfica utilizada en CUI, ahora se tratarán los archivos de configuración. Se utilizarán para aumentar la funcionalidad de la aplicación y permitir que la aplicación tenga algunos aspecto que se puedan modificar. Como se mencionó anteriormente los archivos están en formato XML ya que las librerías de Framework .NET facilitan el manejo de estos archivos.

b. Archivos para los mensajes de la interfaz. Para lograr que la interfaz de la aplicación estuviera disponible en más de un idioma, se utilizan archivos para almacenar dicha información. Los principales archivos que contiene la información de los mensajes para mostrar en la aplicación son `languages_name.xml` y `languages.xml`. El primer archivo contiene la información de qué idiomas están disponibles en la interfaz y el segundo contiene la lista de mensajes.

c. Estructura del archivo `language_name.xml`. Este archivo contiene los lenguajes que están disponibles en la aplicación. Su estructura es muy sencilla, primero contiene el encabezado de archivo de XML con versión 1.0 y codificación UTF-8, luego entre las etiquetas de `<data>` (la raíz) contiene el listado de idiomas con su nombre para referir en la aplicación y la forma desplegarlo en la interfaz. A continuación se presenta el archivo que se utilizó.

```
<?xml version = '1.0' encoding = 'utf - 8'? >
< data >
< spanish > Español </spanish >
< english > English </english >
</data >
```

d. Estructura del archivo `language.xml`. Este archivo contiene el listado de todos los mensajes que se desplegarán en la interfaz de la aplicación. El archivo tiene la siguiente estructura, la cual es muy sencilla:

```

<ENCABEZADO_XML>
<DATA>
<ETIQUETA_MENSAJES_1>
  <ETIQUETA_IDIOMA_1> MENSAJE_IDIOMA_1_1</ETIQUETA_IDIOMA_1>
  <ETIQUETA_IDIOMA_2> MENSAJE_IDIOMA_1_2</ETIQUETA_IDIOMA_2>
</ETIQUETA_MENSAJES_1>
<ETIQUETA_MENSAJES_2>
  <ETIQUETA_IDIOMA_1> MENSAJE_IDIOMA_2_1</ETIQUETA_IDIOMA_1>
  <ETIQUETA_IDIOMA_2> MENSAJE_IDIOMA_2_2</ETIQUETA_IDIOMA_2>
</ETIQUETA_MENSAJES_2>
</DATA>

```

La primera parte es el encabezado del archivo, que sólo indica la versión y codificación del archivo. Luego entre la etiqueta raíz (<DATA>) se coloca el listado del mensaje, cada una con su traducción en todos los idiomas que se utilizan en la interfaz. La etiqueta del mensaje se utiliza para referenciar al mensaje en el código de la aplicación y entre estas etiquetas se coloca la traducción.

e. Archivos de perfil. Estos archivos se utilizan para manejar la configuración del visor y editor de un contenido, y la forma de realizar la petición de la información al servidor. El archivo está dividido principalmente en encabezado (<HEADER>), que contiene la información general del perfil; y el del cuerpo (<BODY>), que contiene el listado de campos que se manejarán del contenido. En el siguiente esquema se describirá el detalle de la información del encabezado, nótese que entre el paréntesis se coloca la etiqueta que se utiliza en el archivo.

- **Títulos (<TITLES>):** contiene el nombre de las distintas pantallas que se utilizarán para manipular el contenido. Es importante mencionar que cada una de estas secciones, se coloca de forma similar como el archivo de mensaje, la traducción de cada uno de los idiomas.
 - **Visor (<SHOW>):** contiene el nombre que desplegará cuando se ejecute el visor del contenido.
 - **Editor modo creación (<CREATE>):** se guarda el nombre que se utiliza en el editor, cuando está en modo para crear contenido.
 - **Editor (<EDIT>):** el nombre de la pantalla de editor para estos contenidos.
- **Configuraciones (<SETTING>):** contiene algunas configuraciones muy importantes.
 - **Campo de nombre (<NAME_TAG>):** indica el campo que se utilizará para obtener el nombre del contenido.
 - **Fuente (<LIST>):** contiene el nombre de la lista en CRM, donde se guarda este tipo de contenido.

- **Vista (<VIEW>):** indica la forma de desplegar el contenido. Si el valor es *normal*, indica que este contenido es para tareas y *plain* para los demás contenidos. Si el contenido tiene vista normal, se incluirá los siguientes campos.
 - **Campo de resumen 1 (<BRIEF_1>):** el campo que se mostrará en el elemento de una lista en la parte superior derecha.
 - **Campo de resumen 2 (<BRIEF_2>):** también se muestra en el elemento de una lista, sólo que en la parte superior izquierda.
 - **Campo de resumen 3(<BRIEF_3>):** indica el campo que se mostrará en la parte inferior de un elemento de una lista.
 - **Nombre padre (<FATHER_FIELD>):** indica el nombre del campo que almacena el nombre del padre.
 - **Etiqueta de identificación del padre (<FATHER_ID_TAG>):** indica el campo ID del contenido.
 - **Etiqueta del título del padre: (<FATHER_TITLE_TAG>):** se utiliza para indicar el campo donde se almacena el título del contenido.

La otra sección de un archivo de perfil es el cuerpo (<BODY>) donde se describen todos los campos que componen al contenido y estos son las columnas de la lista que almacena el servidor. El cuerpo está compuesto por una lista de campos que son descritos de la siguiente manera:

Cada elemento de la lista está encerrado con la etiqueta <ítem> que contiene como atributo el nombre del campo, luego, dentro de esta etiqueta, se encuentra la información importante de campo, es decir, la configuración del campo al momento de utilizarla en el visor y editor.

Tabla12 Campos de un perfil

Campo	Etiqueta	Descripción
Nombre del campo	<DISPLAY_NAME >	Es el nombre que se desplegará en la pantalla, además aquí se guarda la traducción del nombre del campo.
Nombre en la tabla	<TAG>	Es el nombre del campo en las lista del CRM
Tipo	<TYPE>	Indica el tipo de dato del campo
Petición	<REQUEST>	Determina si es necesario pedir este campo al servidor
Mostrar	<SHOW>	Indica que el campo se debe mostrar en el visor
Editar	<EDIT>	Se utiliza para indicar que el componente se podrá editar en el visor
Crear	<CREATE>	Determina si este campo se debe mostrar en el visor
Editable	<EDITABLE>	Indica que es posible modificar la información de este campo al momento de crear o editar un campo.

Continuación de la tabla 12

Campo	Etiqueta	Descripción
Opciones	<CHOICES>	Es requerido para los campos del tipo opciones, ya que contiene una lista de las opciones de valores que podrá tomar el campo. Cada una de las opciones incluye la información de la etiqueta para diferenciarlo y una lista de nombres para los distintos idiomas que se tiene disponible en la interfaz.
Lista	<LIST>	En los campos del tipo enlace, este dato indica la lista donde se obtendrá la información.
Etiqueta de id	<ID_TAG>	En los campos de tipo enlace, éste indica cuál es el campo que se utilizará para diferenciar en el sistema los datos que se tomará.
Etiqueta de título	<TITLE_TAG>	En los campos de tipo enlace, indica cuál es el campo que se utilizará para obtener la información a desplegar.

A continuación se presenta los distintos tipos que un campo puede tomar:

Tabla13 Tipos de campos

Tipo	Etiqueta para los archivos	Descripción
Texto	TEXT	Utilizado para representar los campos tipo texto
Texto formateado	FORMATTING_TEXT	En el caso de algunos datos que se obtienen de una lista y necesitan algún tipo de formateado, pero no se utilizan como enlace.
Número	TEXT_AS_NUMBER	Es texto que se representa como dato numérico
Porcentaje	PERCENTAGE	Representa algún dato tipo de porcentaje
Opciones	CHOICES	Son para aquellos campos que tienen como posible valor un elemento de una lista definida de datos
Fecha	DATE	El campo representa una fecha
Enlace	LINK	Representa un campo que tiene más información relacionada con éste en otra lista, pero que este campo no puede ser editado por el usuario
Enlace con soporte de edición	EDITING_LINK	Es similar al enlace, pero tiene la opción que el usuario puede cambiar su información. Los posibles valores de este campo se obtendrán de la tabla con la que está relacionado
Usuario	USER	Es para campo que contiene información de usuarios.

f. Estructura del archivo stringvar.xml. Este archivo contiene una gran cantidad de información importante para el funcionamiento de la aplicación, las primeras etiquetas se utilizan para definir algunas constantes dentro de la aplicación, como las que se utilizan para representar el tipo de respuesta que se obtiene en una solicitud al servidor. Otras se utilizan para configurar la aplicación. A continuación se presenta una descripción de éstos.

Tabla 14 Campos del archivo de configuración de la aplicación

Campo	Etiqueta	Descripción
Archivo de idioma seleccionado	<STG_SELECT_LANGUAGE_FN >	La dirección de archivo que almacena el idioma seleccionado de la aplicación
Idioma defecto	<STG_DEFAULT_LANGUAGE>	El idioma que se utilizará por defecto en caso que no exista el archivo anteriormente mencionado
Archivo de servicio web	<STG_WBS_PATH_FILENAME>	Es la dirección del archivo que contiene la dirección del servicio web
URL por defecto del servicio web	<STG_DEFAULT_WBS_URL>	La dirección del servicio web en caso que no exista el archivo anterior
Archivo Cantidad de tareas	<STG_AMOUNT_OF_TASK_VAL UE>	La dirección del archivo que contiene la cantidad máxima de tareas que se podrá descargar
Cantidad por defecto de tareas	<STG_DEFAULT_AMOUNT_OF_TASK_VALUE>	Determina la cantidad por defecto de tareas que podrá descargar el archivo en caso que el archivo anteriormente mencionado no se encuentre
La dirección para obtener mapas	<MWR_GOOGLE_STATICMAPS_ADDRESS>	La dirección donde se obtendrá el mapa para mostrar en la pantalla de localización
La dirección para geocoding	<MWR_GOOGLE_REVERSE_GEOCODING>	La dirección del servicio para geocoding.
El archivo de configuración de login	<LGN_SETTING_FILENAME>	La dirección del archivo que contiene la información para la autenticación

g. Estructura del archivo staticmaps.xml. Este archivo se utiliza para guardar la información que se utilizará cuando el sistema esté solicitando información a los servicios de Google Maps para generar el mapa de ubicación del usuario y su posición como una dirección entendible por seres humanos, contiene los siguientes datos:

Tabla 15 Información del archivo de configuración de mapas

Campo	Etiqueta	Descripción
Formato	FORMAT	Indica el formato de archivo en que se descargará el mapa de los servicios de Google Maps.
Tipo de mapa	MAPTYPE	Indica el tipo de mapa que se descargará, por ejemplo: un mapa que incluya sólo las carreteras.
Sensor	SENSOR	Activa o desactiva la opción de sensor.
Zoom	ZOOM	Indica la cantidad de <i>zoom</i> que se aplicará al mapa.
Móvil	MOBILE	Indica si el cliente de la solicitud es un dispositivo móvil.

VI. RESULTADOS Y DISCUSIÓN

Al final del desarrollo de este proyecto se obtuvo los siguientes resultados:

- Una aplicación móvil de acuerdo a los requerimientos y alcance establecidos a un inicio.
- La documentación del proyecto presentada en el capítulo análisis y diseño de este trabajo.
- El manual de usuario.
- Medición del consumo promedio en conexión de Internet al mes del programa.

A. La aplicación móvil

La aplicación móvil es el principal producto del desarrollo del proyecto, por lo que era muy importante que cumpliera todos los requerimientos que se acordaron en el análisis de requerimientos considerando lo establecido en el alcance. Este resultado es el primer incremento de proyecto que continuará en la empresa 1800 Software. Los otros incrementos, que implementan los requerimientos que están afuera del alcance de este trabajo, estarán a cargo de dicha empresa. Se debe mencionar que la implementación de este sistema a otras plataformas móviles también está a cargo de 1800 Software, lo cual se realizará en otros proyectos.

Se puede decir que el uso de la metodología incremental ayudó a manejar el desarrollo de este sistema, ya que con el tiempo que se disponía para realizarlo era muy difícil entregar un sistema con todos los requerimientos establecidos por el cliente. Se establecieron las características esenciales del sistema y se dejaron otras para los siguientes incrementos, como la búsqueda de tareas y el soporte de cámara. Si se hubiera utilizado una metodología en cascada, sería imposible establecer la entrega del sistema, ya que ésta sólo contempla una entrega con todas las características. También hubiera sido difícil el desarrollo si se hubiera utilizado una metodología RAD, ya que no se utilizó componentes prefabricados o no se disponía de un proyecto anterior, además se deseaba que esta aplicación fuera lo más independiente posible a la plataforma de Windows Mobile para que se utilizara como base para el desarrollo de versiones en otras plataformas.

Un tema que es importante comentar es que al inicio se utilizaba sólo el emulador de Visual Studio para ejecutar la aplicación, pero después cuando se instaló la aplicación en un teléfono celular se encontraron algunos problemas en cuanto al manejo de hijos de ejecución, por lo que se consideró que era necesario aumentar las pruebas con un teléfono real, ya que el ambiente emulado no era suficiente para probar el funcionamiento completo de la aplicación. Pero se tiene el problema que es un proceso más

lento, ya que se requiere más tiempo para transferir el ejecutable de la computadora de desarrollo al ambiente de prueba, por lo que es necesario combinar los dos para aprovechar el tiempo que se dispone.

Otro tema que es importante mencionar, es verificar la codificación que se está utilizando con los archivos XML, ya que en algunos casos se generaba algunos archivos con codificación diferente y al momento de importarlo en la siguiente operación, la aplicación fallaba, por lo que es necesario establecer una codificación estándar entre el productor de archivo XML y el que los consume, ya que algunas veces el producto generaba los archivos de acuerdo a la codificación del ambiente de ejecución, por lo que es necesario indicarlo explícitamente para evitar futuros problemas.

B. La documentación del proyecto

Parte de la documentación utilizada para implementar el proyecto se incluyó en este documento. Elaborar la documentación en forma paralela facilitaba el desarrollo de la aplicación y permitía en cierta manera corregir algunos errores antes de implementar el código y evitar el problema de tener que corregirlos al momento de las pruebas. Fue necesaria la elaboración de toda esta documentación, ya que permite tener una mejor vista de todo el sistema para analizarlo tanto como un todo como por partes aisladas. Se trató de utilizar metodologías muy detalladas en el análisis, ya que era necesario determinar las características que se debían cumplir en el primer incremento y dejar una base para los siguientes incrementos.

C. La documentación de usuario

Además de entregar una aplicación, también es necesario entregar un documento que le sirve de referencia al usuario cuando esté utilizando la aplicación, ya que este documento le permite tener una visión más amplia de cómo utilizar el software y evitar ciertos problemas y mejorar su experiencia con la aplicación. El desarrollo de este documento se enfocó en explicar cuál es el propósito principal de la aplicación y las pantallas que el usuario podía encontrar y la posible pantalla que se podía mover de una a otra. Es importante este tipo de documento, pero se debe considerar que es muy probable que el usuario no lo lea, por lo que es necesario desarrollar la aplicación de manera que sea muy fácil de utilizar y muy intuitiva, para evitar que el usuario se frustre. Basado en ello, se utilizaron algunos principios que se presentan en el marco teórico.

D. Consumo promedio en Internet

1. Medición del consumo en el teléfono. Para el funcionamiento de este programa se necesita tener una conexión a Internet, pero muchos de los planes de Internet que ofrecen los operadores de telefonía móvil celular tienen restricciones en la cantidad de gigabytes que se pueden descargar al mes.

Por lo que se consideró oportuno incluir un análisis del consumo promedio que tiene el programa en el teléfono de manera que el usuario pueda buscar un plan que se adapte más a sus necesidades y no tenga un costo muy elevado.

Lo primero que se realizó fue medir la cantidad promedio de cada mensaje que recibe y solicita el programa al servidor. Para realizar dicha medición se utilizó el programa Fiddler2, un programa que se utiliza para registrar el tráfico HTTP entre una computadora y el Internet (Bogue, 2006). También fue necesario realizar algunas modificaciones del código de programa para que se pudiera ejecutar sobre una computadora de manera que fuera posible realizar la medición, ya que la versión de Fiddler que se utilizó, sólo registra peticiones de aplicaciones para Windows.

Se creó una cuenta nueva en el CRM para ir midiendo el peso de cada uno de las peticiones que se realizaba al servidor conforme se iba agregando nuevas tareas. Se obtuvo los siguientes resultados, a los cuales se les aplicó una regresión lineal para tener una función y calcular el tamaño de una petición de la lista de tareas de acuerdo a la cantidad de tareas.

Tabla 16 Tamaño de la petición de lista de tareas

Cantidad de tareas	Tamaño de la respuesta (Bytes)
1	1703
2	2875
3	3951
4	5031
5	6107
6	7419
7	8295
8	9703
9	11111
10	12315
11	13519
12	14951
13	16079
14	17484
15	18715

Ilustración 18 Tamaño de petición de lista de tareas



Tabla 17 Tamaño de otras peticiones

Intercepto	76.40
Pendiente	1230.97
R Cuadrado	0.99

La respuesta de una petición de la lista de tareas está conformada por la información propia del servicio web y los datos que se están solicitando. El intercepto representa la información propia de servicio web, ya que siempre está presente sin importar el tamaño de la lista. La pendiente indica la cantidad de MB que aumenta la respuesta por cada tarea que se añade. El R Cuadrado indica que tan bien se ajustaron los datos a la regresión lineal y dado que su valor está muy cerca de uno, indica que los datos se ajustaron muy bien a la regresión. Por lo que se puede decir que el tamaño de una petición se puede aproximar con la siguiente fórmula:

$$\text{Tamaño} = 1230.97 * \text{Cantidad} + 76.40.$$

También se midieron las respuestas relacionadas con las actualizaciones y los mensajes que el usuario puede recibir del servidor, por ejemplo acceso denegado o que la lista está vacía.

Tabla 18 Otras mediciones

Acción	Tamaño de la respuesta (Bytes)
Mensaje de error por error de autenticación	635
Solicitar un contenido extra de una tarea de 6 campos	497
Datos para una actualización	1115
Respuesta después de actualizar una tarea	4539

El error de autenticación ocurre cuando el usuario ha ingresado algún dato incorrecto de sus credenciales, como su nombre de usuario o la contraseña, lo cual se espera que sea baja, puesto que el usuario tiene la opción de guardar sus credenciales para que el programa inicie automáticamente la sesión. La solicitud de contenido extra es cuando el usuario desea ver información relacionada a una tarea por ejemplo una cuenta o contacto. Los datos de actualización son los datos que se necesita para una actualización de una tarea, ya que éstos se guardan en el servidor. La respuesta después de la actualización es cuando el usuario ha actualizado exitosamente una tarea.

2. Consumo mensual. Se realizó el cálculo aproximado de la cantidad de MB consumido por un técnico a la semana. Pero es importante mencionar que este cálculo se realizará en base de varias suposiciones, lo cual puede proporcionar un valor muy inexacto, pero puede proporcionar una idea acerca de este tema. Se podría realizar una estimación, pero el trabajo requerido para dicho proceso se sale del alcance de esta tesis.

Se trabaja con la suposición de que al técnico se le asigna quince tareas diarias, por lo que es muy probable que consulte diez veces la lista de tareas y actualice el estado de las quince tareas. Pueda ser que consulte cinco contenidos extras durante el día. Es importante recordar que se han supuesto varios datos.

Tabla 19 Consumo diario de MB

Actividad	Veces realizada	Bytes por actividad	Total (Bytes)
Consulta de la lista de tareas	10	$15 * 1230.97 + 76.40 = 18540.95$	185400.95
Actualizar tareas	15	$1115 + 4539 = 5654$	84410
Consulta información extra	5	497	2485
Total			272295.95

El técnico gastaría diariamente 272295.95 bytes = 0.26 MB y considerando que trabaje veinte días al mes, entonces su consumo puede ser de 5.2 MB.

3. Valor de MB descargado. Para obtener un valor aproximado del costo de MB de descarga en un plan de una compañía de teléfono se visitó los sitios de dos empresas que operan en Guatemala.

a. *Planes Claro.* A continuación se presentan los planes de internet que ofrece la empresa Claro. (Claro, 2010)

Tabla 20 Planes de Internet en Claro

Plan	Cantidad de descarga	Costo mensual	Costo por 1 MB
Light	2 GB (2048MB)	Q 110.00	Q0.05
Full	3 GB (3072 MB)	Q 190.00	Q 0.06
Avanzado	5 GB (5120 MB)	Q220.00	Q 0.04
Premium	10 GB (10240 MB)	Q280.00	Q 0.03

b. *Planes Movistar*. Esta tabla presenta los precios de los planes que ofrece Movistar. (Movistar, 2011)

Tabla 21 Planes de Internet en Movistar

Tiempo	Cantidad de descarga	Costos	Costo por 1 MB
1 día	100 MB	Q 5.00	Q 0.05
7 días	500 MB	Q50.00	Q 0.10
15 días	1024 MB	Q 75	Q 0.07

Como se puede observar en estas dos tablas, el plan más favorable es el que ofrece Claro, ya que los precios son más favorables y el plan de Movistar tiene restricciones en el tiempo de uso, lo cual implicaría considerar otros factores que se salen de esta medición. Por lo que se utilizó el plan de claro como referencia para el cálculo. En promedio el MB descargado vale Q 0.045.

4. **Valor total de consumo mensual.** Con base a lo que se presentó anteriormente, se constató que el consumo mensual en MB de la aplicación es de 5.2 MB aproximadamente y el valor de cada MB es 0.045, se puede decir que el consumo total es de Q 0.234, pero se debe mencionar que este dato depende del tipo de trabajo del técnico.

VII. CONCLUSIONES

- El uso de la metodología incremental se adaptó a las necesidades del proyecto ya que se consideró el hecho de iniciar un proyecto desde cero con tiempo limitado y dejar una base funcional para siguientes versiones de la aplicación.
- La elección de las herramientas de desarrollo que se utilizaron en el proyecto fue la mejor, ya que éstas permitieron integrarse con las tecnologías que estaba utilizando el servidor Web y facilitar el desarrollo con la gran cantidad de librerías que disponía.
- Se determinó que era necesario balancear el uso de un ambiente real de pruebas con un ambiente emulador para probar las funcionalidades de la aplicación, puesto que de esa manera se logró obtener una aplicación que funcionara bien con el ambiente real sin gastar mucho tiempo en la fase de pruebas.

VIII. RECOMENDACIONES

- Considerar el desarrollo de una aplicación WAP para aumentar la portabilidad de la aplicación, aunque no disponga de todas las características de la versión original.
- Configurar el ambiente emulado para que se acerque a la forma como funciona un ambiente real de tal manera que sólo logre realizar las pruebas principales en el teléfono celular.
- Considerar la compra de la licencia de Google Maps para utilizar los servicios de *geocoding* y de mapas estáticos con el fin de habilitar la funcionalidad de localización de esta aplicación.
- Desarrollar la versión de Java que sea compatible con las versiones actuales de la máquina virtual de Java con el fin de aumentar la portabilidad de la aplicación.
- Modificar el servicio web del CRM para que soporte varios tipos de clientes, ya que actualmente sólo se puede realizar con clientes programada en plataforma .NET.
- Agregar al módulo de localización el soporte de buscar las coordenadas del dispositivo por medio de celdas telefónicas.

IX. BIBLIOGRAFÍA

- Contratos por líneas móviles superan a las residenciales en Guatemala.* (5 de abril de 2010). Recuperado el 27 de Septiembre de 2010, de Revista Summa: <http://www.revistasumma.com/economia/2551-contratos-por-lineas-moviles-superan-a-las-residenciales-en-guatemala.html>
- Las comunicaciones móviles serán la base de crecimiento del sector en América Latina.* (07 de Septiembre de 2010). Recuperado el 27 de Septiembre de 2010, de América economía : <http://www.americaeconomia.com/negocios-industrias/las-comunicaciones-moviles-seran-la-base-de-crecimiento-del-sector-en-america-la>
- Bates, S., & Smith, T. (2010). Introduction to Sharepoint Technologies. En S. Bates, & T. Smith, *SharePoint 2010 User's Guide: Learning Microsoft's Business Collaboration Platform* (pág. 505). New York: Apress.
- Batz, H. (15 de Marzo de 2010). *Creciente afición por celulares.* Recuperado el 30 de Septiembre de 2010, de Titulares Guatemala.
- Bogue, R. (12 de Septiembre de 2006). *Fiddler Can Make Debugging Easy.* Recuperado el 10 de Enero de 2011, de Developer.com: <http://www.developer.com/lang/jscript/article.php/3631066/Fiddler-Can-Make-Debugging-Easy.htm>
- Cassavoy, L. (2009). *What Makes a Smartphone Smart?* Recuperado el 20 de Septiembre de 2010, de About.com: http://cellphones.about.com/od/smartphonebasics/a/what_is_smart.htm
- Cavanaugh, E. (2006). *Web services: Benefits, challenges, and a unique, visual development solution.* Recuperado el 20 de julio de 2010, de Altova: www.altova.com/whitepapers/webservices.pdf
- Claro. (2010). *Internet para tu claro.* Recuperado el 31 de enero de 2010, de Claro: <http://www.claro.com.gt/Movil/Postpago/Internetentuclaro.aspx>
- Cook, C. (6 de noviembre de 2000). *Introduction to Encryption.* Recuperado el 30 de septiembre de 2010, de Symantec Sites: <http://www.symantec.com/connect/es/articles/introduction-encryption>
- De Herrera, C. (11 de Noviembre de 2009). *Windows CE/ Windows Mobile Versions.* Recuperado el 20 de Noviembre de 2010, de Pocket FAQ: <http://www.pocketpcfaq.com/wce/versions.htm>

- Glasscoe, M. (13 de Agosto de 1998). *The Global Positioning System*. Recuperado el 30 de Septiembre de 2010, de The Source California Integrated GPS Network Education Module: <http://scign.jpl.nasa.gov/learn/gps1.htm>
- Gray, P., & Byun, J. (23 de Marzo de 2001). *Customer Relation Management*. Recuperado el 10 de Julio de 2010, de Crito: www.crito.uci.edu/papers/2001/crm.pdf
- Grundgeiger, D. (2002). .NET Framework. En *Visual Basic .NET Programming* (pág. 464). O'Reilly.
- Gunzer, H. (Marzo de 2002). *Introduction to Web Services*. Recuperado el 20 de Julio de 2010, de Daimi: www.daimi.au.dk/~thomasr/Wearable/intro_to_web_services_wp.pdf
- Harris, S. (2008). *CISSP EXAM GUIDE* (Cuarta ed.). New York, USA: McGraw-Hill.
- Hjerde, M. (15 de abril de 2008). *Mobile Screen Size Trends*. Recuperado el 20 de septiembre de 2010, de Sender11: <http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html>
- Larry, W. (2005). *What Is Information Assurance (IA) ?* Recuperado el 20 de Septiembre de 2010, de SelfGrowth: <http://www.selfgrowth.com/articles/Westfall1.html>
- Litchfield, S. (Septiembre de 2010). *A detailed comparison of Series 60 (S60) Symbian Smartphones*. Recuperado el 20 de Noviembre de 2010, de 3-Lib: <http://3lib.ukonline.co.uk/s60history.htm>
- Makofsky, S. (6 de junio de 2003). *The .NET Compact Framework*. Recuperado el 30 de Septiembre de 2010, de Ptgmedia: <http://ptgmedia.pearsoncmg.com/images/0321133528/samplechapter/makofskych12.pdf>
- Microsoft. (2009). *.NET Framework Overview*. Recuperado el 30 de Septiembre de 2010, de Microsoft Site: <http://www.microsoft.com/net/overview.aspx>
- Microsoft. (4 de Agosto de 2010). *Windows Mobile*. Recuperado el 20 de 11 de 2010, de MSDN: <http://msdn.microsoft.com/library/bb847935.aspx>
- Movistar. (2011). *Internet en tu movistar*. Recuperado el 31 de Enero de 2011, de Movistar: <http://www.movistar.com.gt/?secc=Contenido&id=57>
- Myerson, J. (2006). *Web Service Architectures*. Recuperado el 20 de julio de 2010, de Web Services Architectures: www.webservicesarchitect.com/content/webservicesarchitectures.pdf
- Pressman, R. S. (2001). *Software Engineering*. Nueva York: McGraw Hill.

- Rodgers, K., & Howlett, D. (2000). *What is CRM?* Recuperado el 10 de Octubre de 2010, de Solica: www.solica.com/brochures/WhatIsCRM.pdf
- Rodriguez Ventosa, F., & Secilla, J. P. (s.f.). *Situación actual y tendencias*. Recuperado el 30 de Septiembre de 2010
- Sklavos, D. (11 de Diciembre de 2005). *Notebook Memory (RAM) Guide*. Recuperado el 20 de Noviembre de 2010, de Notebook Review: <http://www.notebookreview.com/default.asp?newsID=2612>
- Software, P. (20 de Diciembre de 2010). *Common CPU's*. Recuperado el 27 de Diciembre de 2010, de CPU Benchmark: http://www.cpubenchmark.net/common_cpus.html
- Spillmann, J., & Shmedding, R. (28 de Abril de 2009). *Microsoft Visual Studio Introduction*. Recuperado el 30 de Septiembre de 2010, de University of Freiburg: <http://cg.informatik.uni-freiburg.de/teaching/VisualStudio8%20Introduction.pdf>
- Vaidya, V. (26 de agosto de 2009). *Best practices for designing Mobile Application*. Recuperado el 30 de 09 de 2010, de Usercentric: <http://www.usercentric.com/news/2009/08/26/best-practices-designing-mobile-applications>
- W3Schools. (2010). *Web Statistics and Trends*. Recuperado el 20 de septiembre de 2010, de W3Schools.com: http://www.w3schools.com/browsers/browsers_display.asp
- Woyke, E. (10 de mayo de 2009). *The Most Powerful Smart Phones*. Recuperado el 20 de septiembre de 2010, de Forbes: <http://www.forbes.com/2009/10/03/iphone-blackberry-nokia-technology-wireless-smartphone.html>

X. APÉNDICES

A. Algoritmos de cifrado para envío de credenciales

1. Introducción. Para manejar la respuesta del servidor se utilizó una modificación de algoritmo de TripleDES, del cual ya se disponía una implementación de la librería que el Framework proveía lo cual aseguró que cualquier persona que esté espiando la red, no tenga acceso a la información que se está intercambiando. Para enviar las credenciales al servidor en la petición se optó por utilizar otro algoritmo que se desarrolló conjuntamente con el proyecto del servidor CRM. Este algoritmo utiliza varios componentes que se describirán en la siguiente sección.

2. Componentes. Para funcionamiento de este algoritmo se necesitan los siguientes elementos.

- La lista de caracteres permitidos.
- La tabla de codificación.

Toda esta información puede estar guardada en un archivo XML o utilizar una clase que contenga todos estos valores definidos.

3. Pasos para el cifrado. Para mostrar el uso de este algoritmo se procederá a utilizar un ejemplo: Se debe encriptar el mensaje hola, con los siguientes componentes:

- Caracteres válidos: h, o, l, a, e
- Tabla de codificación

Tabla 22 La tabla para el algoritmo de codificación

	H	L	a	e	o
h	L	A	e	o	h
l	A	E	o	h	l
a	E	O	h	l	a
e	O	H	l	a	E
o	H	L	a	e	o

La tabla de codificación debe cumplir con que las columnas y filas deben tener como elementos los caracteres válidos y evitar que una misma fila tenga elementos repetidos.

El procedimiento es el siguiente:

- Se inicializa la variable para el texto cifrado.
- Se toma el primer carácter.
- Se genera un carácter aleatorio entre los caracteres válidos, el cual será la llave, por ejemplo a.
- Se busca en la tabla la celda con columna del carácter y fila de la llave, que en este caso se obtiene la letra e, este será el carácter codificado, entonces se forma la cadena “ea”, es decir se concatena el carácter codificado y la llave.
- Se agrega al texto cifrado el resultado anterior.
- Se realiza los pasos 3 a 5 con los demás caracteres.

Supóngase que en los otros caracteres se obtuvieron las llaves h, l y e, al final se obtiene el texto cifrado “eahhelle”. Se debe mencionar que el texto cifrado siempre es doble del texto original, lo cual le da una desventaja en cuanto al tamaño.

4. Pasos para el descifrado. Se utilizará el ejemplo anterior para mostrar los pasos del descifrado:

- Primero se divide el texto cifrado en parejas, en caso de ser número impar se regresa un mensaje de error.
- Se inicializa el texto plano.
- A cada pareja se le realiza el siguiente procedimiento:
 - Se sabe que el segundo elemento es la llave. Por lo que se busca en la fila, el primer elemento de la pareja y se determina cuál es la columna. Dicho valor es el carácter del texto plano.
 - Se agrega el resultado al texto plano.

Por ejemplo, para la pareja “ea” se reconoce que el carácter h es la columna que puede generar el primer elemento en la fila de a. De esta manera se obtiene que el texto plano que es “hola”.

En el caso de la aplicación se utilizó una tabla que abarcara los caracteres permitidos para una contraseña o nombre usuario.

B. Manual del usuario

1. Primeros Pasos.

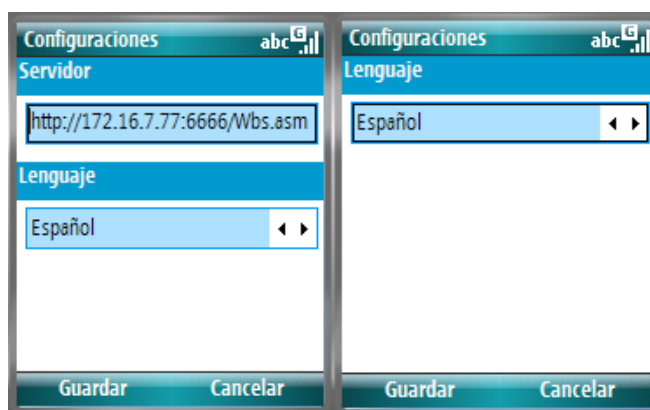
a. Pantalla de inicio de sesión. En esta pantalla el usuario deberá colocar el dominio, su nombre y contraseña para ingresar a la aplicación, además podrá indicar al sistema que guarde estos datos para futuras sesiones. En caso que presione la opción **Menú** podrá escoger si navegar a **la pantalla de configuración** o salir de la aplicación. Para iniciar sesión sólo debe ingresar la información requerida y presionar la opción **Acceder**, después se le presentará la pantalla de la lista de tareas.

Ilustración 19 Pantalla de inicio de sesión



b. Pantalla de configuración. El usuario podrá cambiar la dirección del servidor donde se almacena la información de las tareas, además encontrará la opción para cambiar el lenguaje de la aplicación (en este caso sólo se dispone de los idiomas español e inglés). Si presiona la opción **Guardar** se actualizarán los cambios que realizó y regresará a la pantalla de inicio de sesión, la opción **Cancelar** regresa a esta pantalla sólo que no guarda los cambios. Como se observa en las dos imágenes anteriores hay dos vistas de esta pantalla, la primera se presenta cuando el usuario todavía no ha iniciado sesión, la otra vista cuando no necesita configurar la dirección del servidor.

Ilustración 20 Pantalla de configuración



c. Pantalla de lista de tareas. En esta pantalla el usuario podrá ver la lista de tareas que tiene asignadas y en caso que no tenga, se desplegará el mensaje “La lista está vacía” (como se puede observar en los primeras imágenes de este apartado). Si presiona la opción **menú** podrá escoger la siguiente tarea que desea realizar: **Refrescar**, actualiza la información de la lista de tarea; **Configuración**, se dirige a la pantalla de configuración; **Mi posición**, cambia a la pantalla de localización; **Log out**, cierra la sesión y regresa a la pantalla de inicio de sesión; **Salir**, cierra la aplicación. Si presiona la opción **Ver tarea** cambiará al visor de tareas donde se mostrará la información de la tarea seleccionada.

Ilustración 21 Lista de tareas



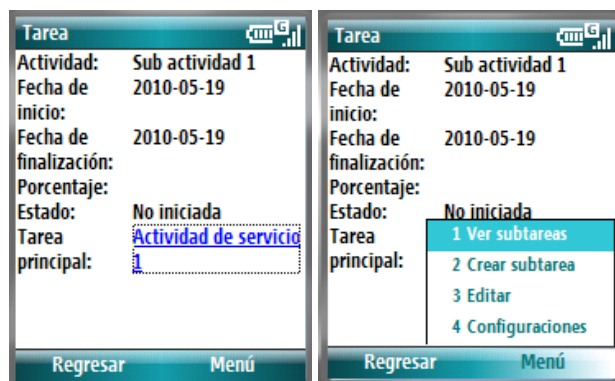
d. Pantalla de localización. El usuario podrá conocer su posición a través de esta pantalla. La primera vez que la aplicación se ejecute sólo se mostrará al usuario que esta información no se ha determinado, por lo que tendrá que actualizarlo. Cuando presione la opción **Actualizar**, se activará el dispositivo de GPS para determinar su posición actual y realizará una petición para determinar la dirección y obtener el mapa correspondiente a las nuevas coordenadas. La opción **Regresar** es para cambiar a la pantalla de lista de tareas.

Ilustración 22 Pantalla de localización



d. Visor de tareas. Esta pantalla se encarga de mostrar la información de la tarea seleccionada. Si presiona la opción **Regresar** cambiará a la pantalla de lista de tareas. En cambio si selecciona la opción **Menú** podrá escoger entre las siguientes opciones: Ver sub-tareas, se mueve a la pantalla de lista de sub-tareas; Crear sub-tareas, cambia a la pantalla para crear nuevas sub-tareas; Editar, se le presentará el editor de tareas; Configuraciones, donde podrá ver la pantalla de configuración del visor. Si algunos de los campos están subrayados y de color azul entonces son enlaces que el usuario podrá seleccionar para ver la información asociada a estos campos.

Ilustración 23 Visor de tareas



e. Pantalla de lista de sub-tareas. Esta pantalla tiene las mismas funcionalidades que la pantalla de la lista de tareas, la diferencia consiste en que presenta las sub-tareas que están relacionadas con una tarea. El menú también es muy parecido, sólo que se cambia la opción **Regresar** por **Logout** para que regrese al visor de tareas para ver la tarea seleccionada.

Ilustración 24 Lista de sub-tareas



f. Pantalla para crear nuevas sub-tareas. A través de esta pantalla el usuario podrá crear nuevas sub-tareas y asignárselas a otro usuario. En cada campo deberá ingresar la información pertinente y después presionar la opción **Guardar** para almacenar la nueva sub-tarea en servidor o presionar **Cancelar** para regresar al visor de tareas.

Ilustración 25 Pantalla de nuevas sub-tareas

g. Pantalla para configurar el visor. Utilizando esta pantalla el usuario podrá escoger cuáles son los campos que se mostrarán en el visor de tareas o que se modificará en el editor. Si se desea guardar los cambios sólo debe presionar la opción **Guardar**, en caso contrario se presiona **Cancelar** y luego regresará al visor de tareas.

Ilustración 26 Pantalla de configuración del visor

Field	Editar	Mostra
ID	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Actividad	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Fecha de inicio	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Fecha de finalización	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Porcentaje	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Teléfono	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons: Guardar, Cancelar

h. Editor de tareas. El usuario podrá actualizar la información de la tarea seleccionada utilizando esta pantalla. Los campos que podrá modificar son los que activó en la pantalla para configurar el visor. Para actualizar la información modificada en el servidor se deberá presionar la opción **Guardar**, en caso contrario deberá escoger la opción **Cancelar**, después regresará al visor de tareas.

Ilustración 27 Editor de tareas

Fields: Porcentaje, Estado (No iniciada), Incidente (Incidente número)

Buttons: Guardar, Cancelar

C. Diagramas UML

1. Diagrama de casos de uso

Ilustración 28 Diagrama de casos de uso

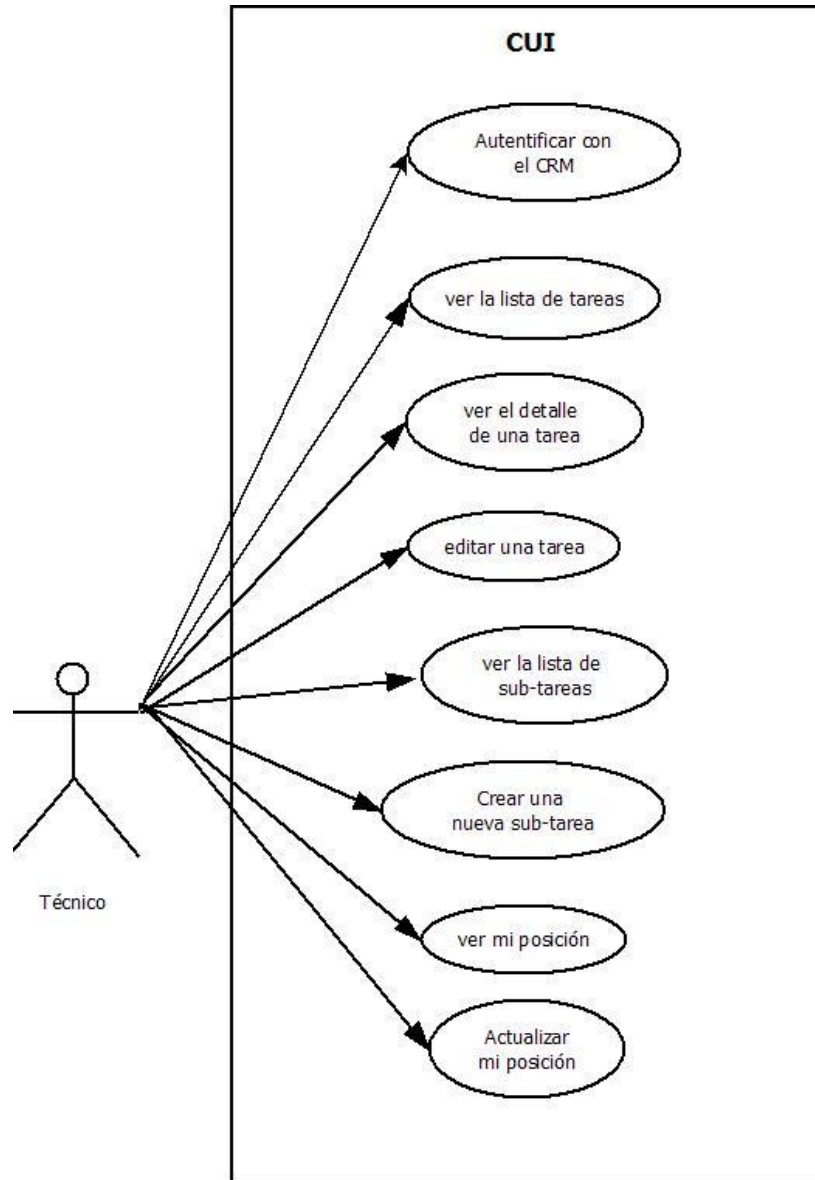
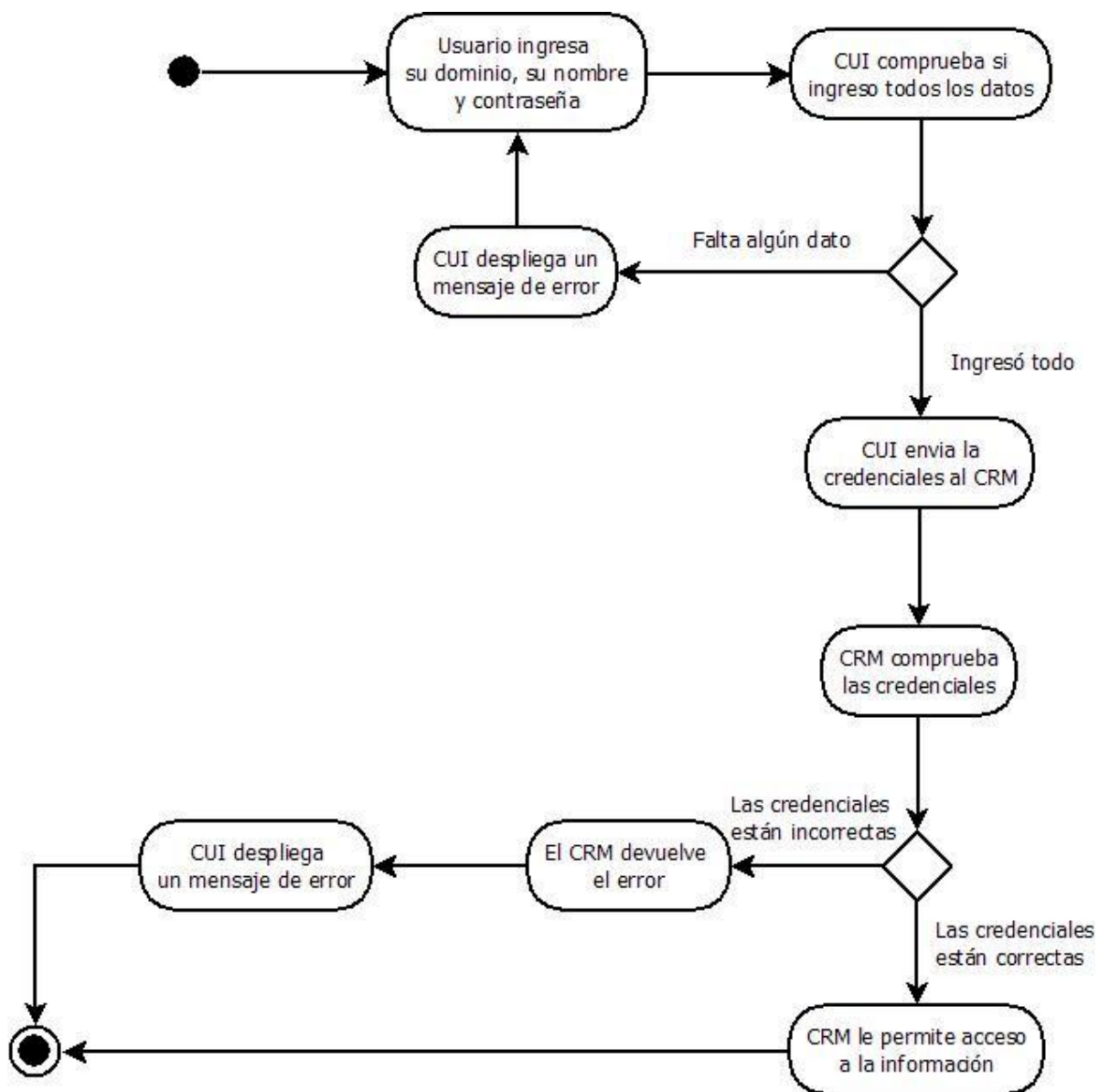


Diagrama de casos de usos de CUI

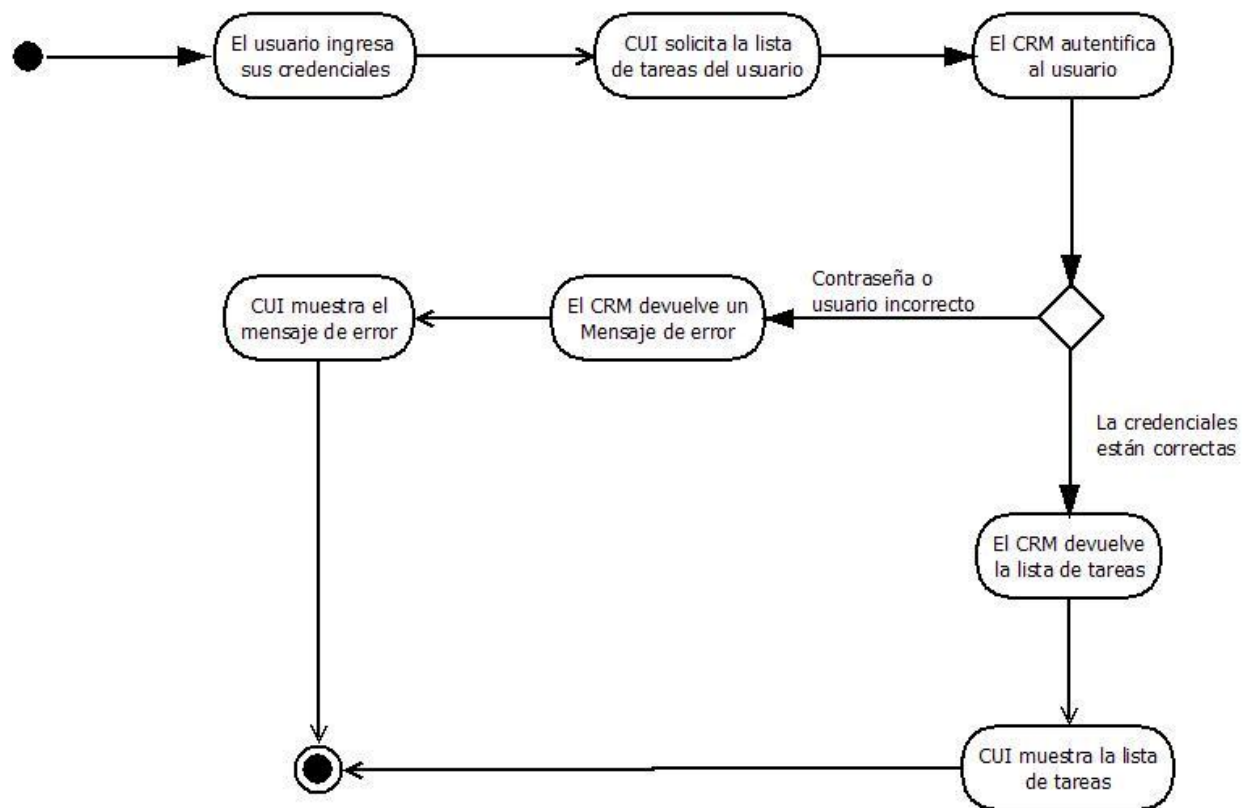
2. Diagramas de actividades

Ilustración 29 Diagrama de actividades –Autenticación con el CRM



**Diagrama de actividades
Autenticación con el CRM**

Ilustración 30 Diagrama de actividades - Consulta de lista de tareas



**Diagrama de actividades
Consulta de la lista de tareas**

Ilustración 31 Diagrama de actividades - Mostrar una tarea

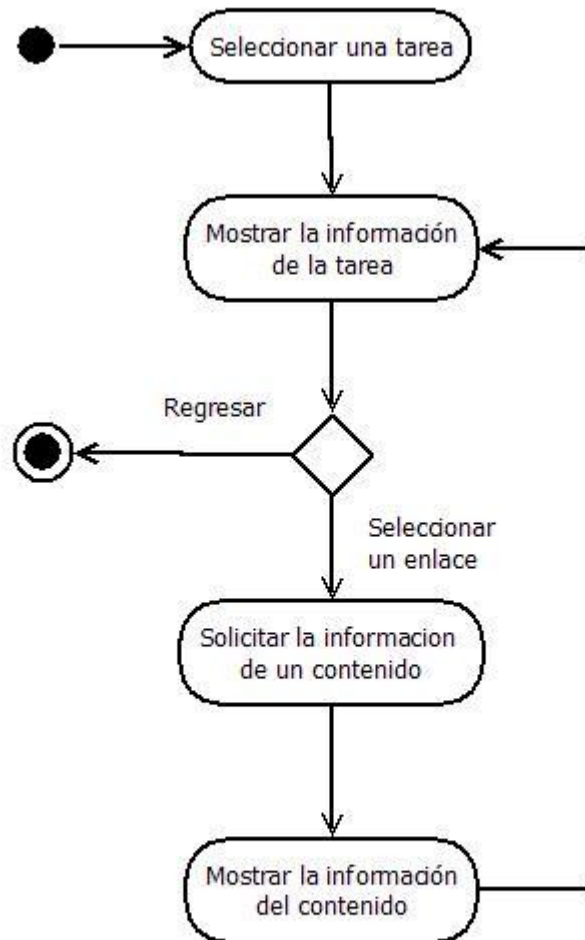


Diagrama de actividades Desplegar la información de una tarea

Ilustración 32 Diagrama de actividades - Editar tareas

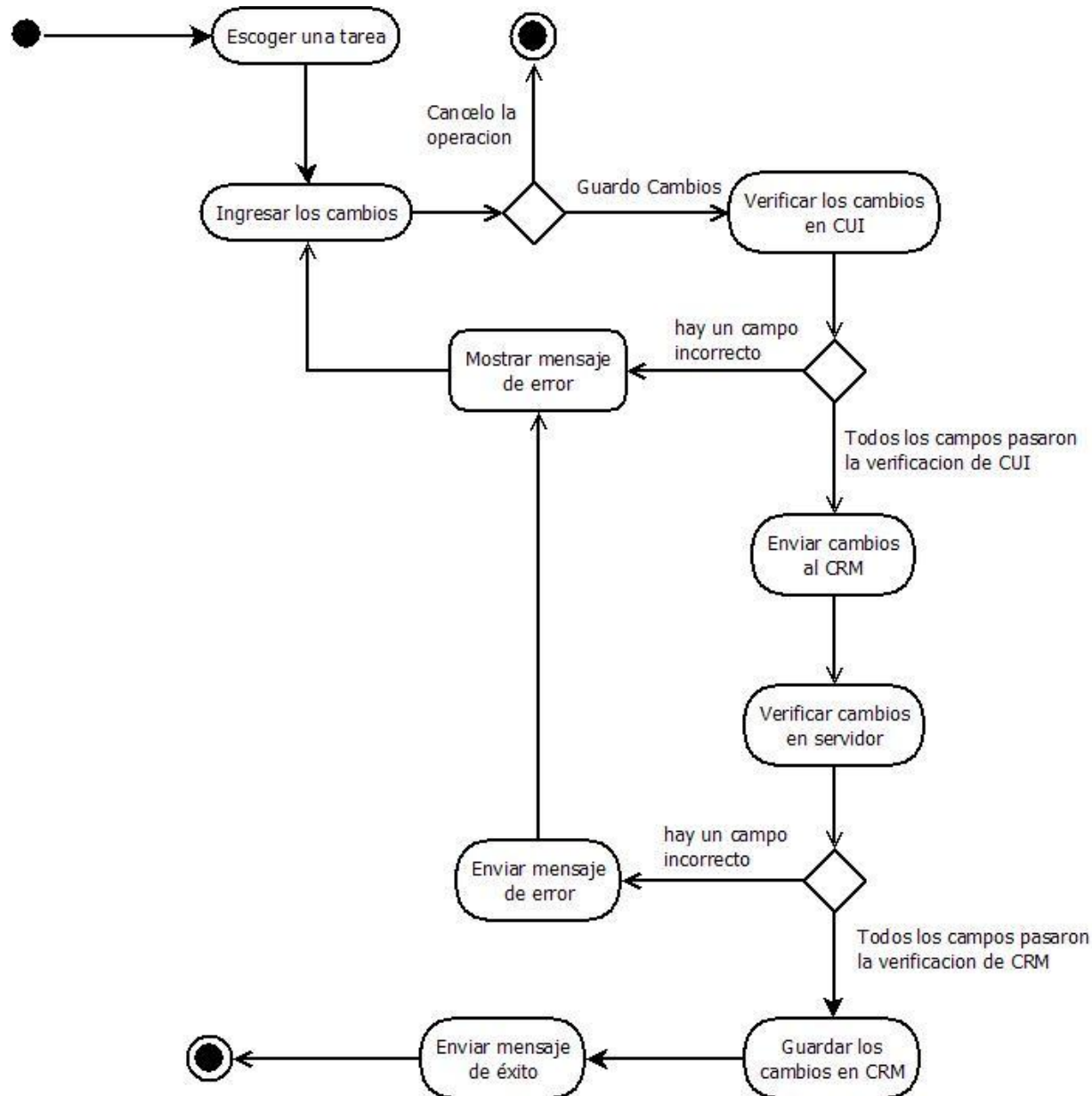
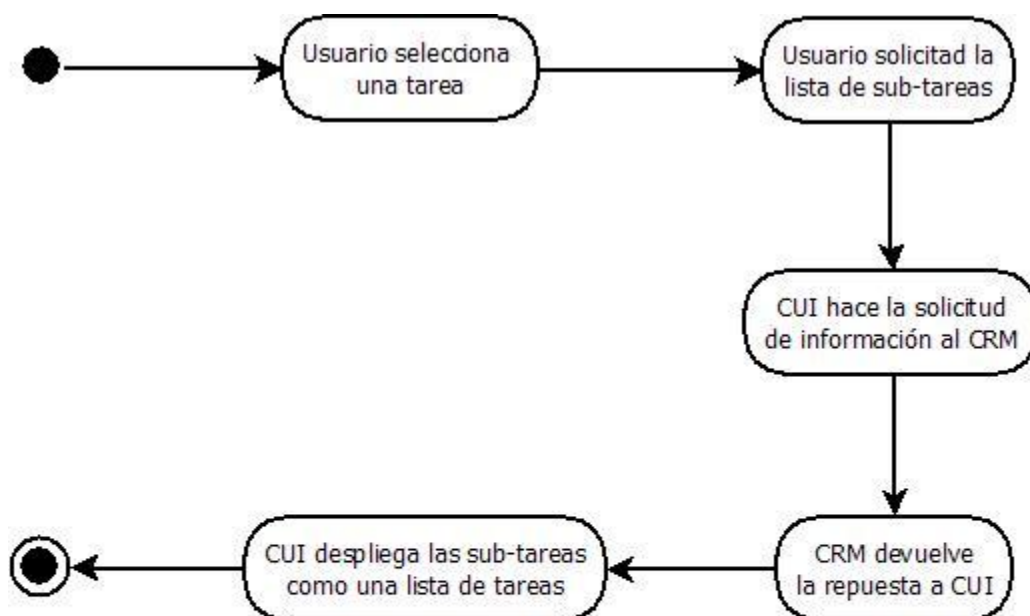


Diagrama de actividades
Editar la información de una tarea

Ilustración 33 Diagrama de actividades - Consulta de sub-tareas



**Diagrama de actividades
Consulta de la lista de sub-tareas
de una tarea**

Ilustración 34 Diagrama de actividades - Crear una sub-tarea

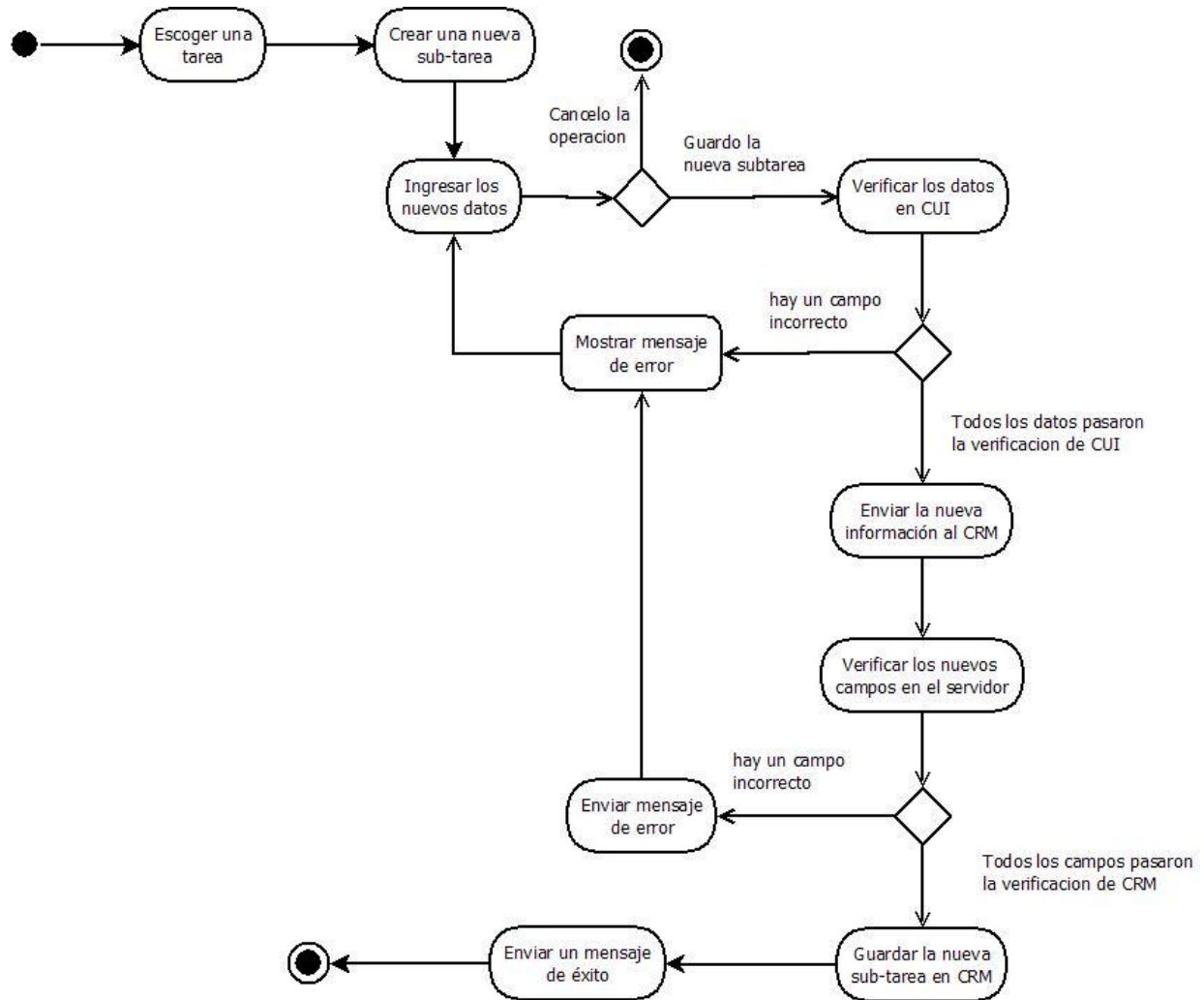


Diagrama de actividades
Crear una nueva sub-tarea

Ilustración 35 Diagrama de actividades – Localización

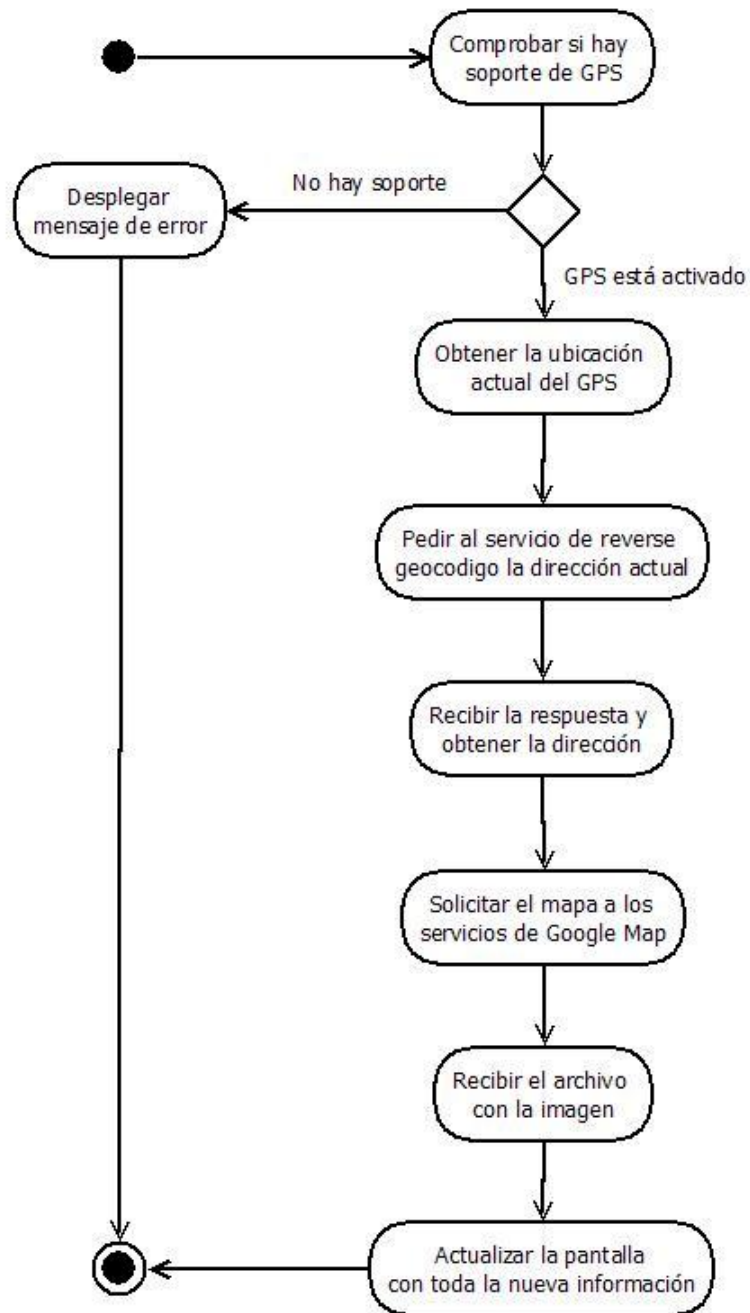


Diagrama de actividades
Actualización de la información de la localización

3. Diagrama de componentes

Ilustración 36 Diagrama de componentes

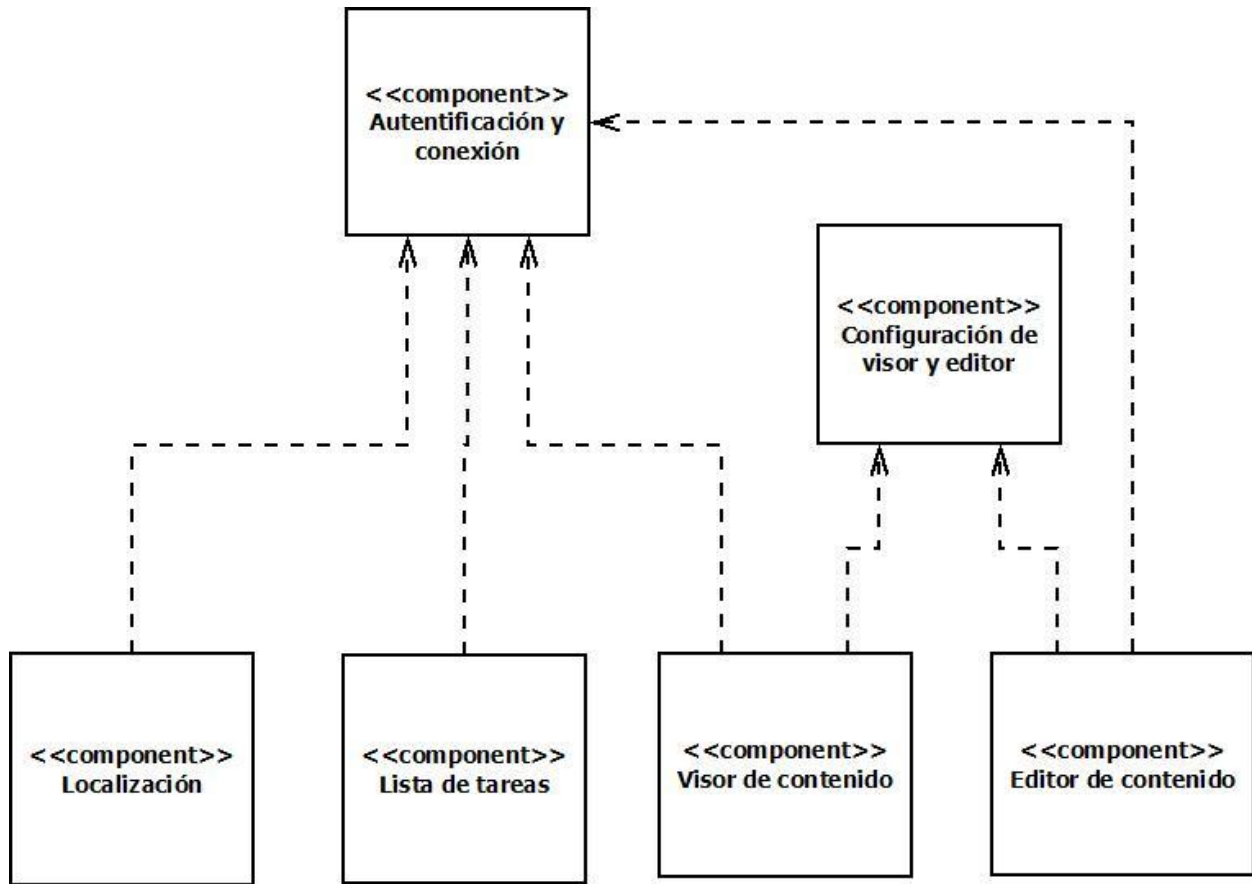


Diagrama de componentes

D. Plantilla utilizada para realizar el análisis

Software Requirements Specification

For

<Project>

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

1. Introduction

a. Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

b. Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

c. Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

d. Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

e. References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

a. *Product Perspective*

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

b. *Product Features*

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

c. *User Classes and Characteristics*

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

d. *Operating Environment*

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

e. *Design and Implementation Constraints*

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

f. *User Documentation*

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

g. *Assumptions and Dependencies*

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

a. *System Feature 1*

<Don't really say "System Feature 1." State the feature name in just a few words.>

1. Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

2. Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3. Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid

inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

b. *System Feature 2 (and so on)*

4. External Interface Requirements

a. *User Interfaces*

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

b. *Hardware Interfaces*

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

c. *Software Interfaces*

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software

components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

d. *Communications Interfaces*

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements

a. *Performance Requirements*

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

b. *Safety Requirements*

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

c. *Security Requirements*

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

d. *Software Quality Attributes*

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

7. Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

8. Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

9. Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>

XI. GLOSARIO

- **API (Application Program Interface)** es un conjunto de rutinas, protocolos y herramientas creadas para el desarrollo de aplicaciones de software.
- **Aplicación nativa:** es una aplicación diseñada para correr en un ambiente de computación (sistema operativo y lenguaje de maquina) sin necesidad de utilizar un software intermediario.
- **Aplicación servidor:** es una aplicación diseñada para ejecutarse en un servidor para proveer información a computadora clientes.
- **Aplicación web:** es cualquier aplicación servidor que utiliza un navegador web como cliente para acceder a la información.
- **C#:** es un lenguaje de programación orientada a objetos desarrollado por Microsoft, basado sobre lenguaje programación C++. Es utilizado principal para el desarrollo de aplicaciones .NET.
- **CheckBox:** es un componente gráfico que le permite a usuario seleccionar varias opciones.
- **Cloud Computing:** es una nueva paradigma de computación donde le permite a las personas o empresas obtener las aplicación de software como un servicio sobre demanda abstrayéndoles la parte técnica de la computación,
- **ComboBox:** es componente gráfico que combina un textbox con un listbox, lo cual le permite al usuario seleccionar un elemento de la lista o ingresar un nuevo valor.
- **Comercio Electrónico:** es cualquier transacción de negocio cuyos precios o términos esenciales fueron negociados utilizando un sistema informático en line, por ejemplo portal en Internet.
- **Componentes:** es un objeto pre-programado con utilidad determinada que se utiliza para el desarrollo de aplicaciones.
- **Credenciales:** es la información que un usuario debe proveer para identificarse ante un sistema.
- **Editor:** es un programa o pantalla que se utiliza para modificar la información de un elemento o archivo.
- **Framework:** es un conjunto de software, documentación, políticas y procedimientos que soportar aplicaciones de software que se ejecutan sobre una tecnología específica.
- **FTP (File Transfer Protocol)** es un protocolo estándar de Internet que se utiliza para facilitar el intercambio de archivo entre computadora en la red.
- **Geoposicionamiento:** un método que facilita la localización de un punto relativo a la superficie de la tierra, es decir la longitud y latitud.

- **HTTP (Hypertext Transfer Protocol)** es un protocolo de redes que especifica la manera como un cliente y un servidor se comunicara a través de la web.
- **Interfaz de usuario:** es la parte de la aplicación que interactúa con el usuario.
- **Interfaz Web:** es un tipo de interfaz de usuario donde el usuario utiliza páginas web para interactuar con la aplicación.
- **Interoperabilidad:** es la capacidad de comunicar aplicaciones heterogéneas.
- **Java:** es un lenguaje de programación orientada a objetos desarrollada por Sun Microsystems. Todas las aplicaciones que se desarrolla en lenguaje son ejecutadas en una máquina virtual.
- **Label** es un componente grafico que se utiliza para desplegar información.
- **Lenguaje de Programación:** es un conjunto de reglas, símbolos y palabras especiales que proveen una manera de comunicar con la computadora para realizar ciertas tareas.
- **Librerías:** es conjunto de rutinas o clases utilizada para el desarrollo de software.
- **Máquina virtual** es la implementación en software de una máquina que permite la ejecución de aplicaciones en un ambiente aislado.
- **Microsoft .Net Framework** es un Framework desarrollado por Microsoft para el desarrollo aplicación para sistemas operativos Windows.
- **Microsoft .Net Framework Compact** es una versión compacta de Microsoft .Net Framework para dispositivos móviles.
- **Navegador Web:** un programa utilizado para el acceso y despliegue de documentos en la Web.
- **Parámetros:** es un tipo de variables que se utiliza para representar las entradas a un programa o función.
- **Portal Web** es la presentación de información de distintas fuentes de una manera unificada.
- **Programación de bajo nivel** es desarrollo de rutinas o aplicación utilizando lenguaje maquina o lenguaje assembler.
- **Protocolos** es un conjunto de reglas que se utiliza para comunicación entre computadora a través de un red.
- **RAM (Random-access memory)** es un tipo de memoria de computadora utilizada por los programa para realizar sus tareas mientras la computadora esté encendida.
- **Scrolling** es continuo movimiento de texto o imágenes en una pantalla tanto en dirección horizontal como vertical.
- **SDK (Software Development Kit)** es un conjunto de herramientas que permite la creación de aplicaciones para cierto tipo de tecnología de sistema o plataforma.

- **Sistema Operativo** es un conjunto de rutinas y programa que se encarga de controlar los recursos de una computadora y proveer acceso a sus servicios.
- **Timer** es un componente de software que se utiliza para ejecución de tareas en ciertos intervalos de tiempo.
- **Visor:** es una aplicación que se utiliza para desplegar información de un elemento o archivo.
- **W3C (World Wide Consortium)** es la principal organización que se encarga de establecer los estándares para la Web.