

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Aplicación de Tecnología en la Industria Cafetalera de Guatemala

Trabajo de graduación en modalidad de megaproyecto presentado por

Carlos André Aguilar Flores

Josué Daniel Cabrera Loaiza

Juan José Paiz Leiva

Luisa Mercedes Portillo Díaz

Sandra Paola García Sandoval

para optar al grado académico de Licenciados en Ingeniería

Mecatrónica

2014

Aplicación de Tecnología en la Industria Cafetalera de Guatemala

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Aplicación de Tecnología en la Industria Cafetalera de Guatemala

Trabajo de graduación en modalidad de megaproyecto presentado por
Carlos André Aguilar Flores
Josué Daniel Cabrera Loaiza
Juan José Paiz Leiva
Luisa Mercedes Portillo Díaz
Sandra Paola García Sandoval

para optar al grado académico de Licenciados en Ingeniería
Mecatrónica

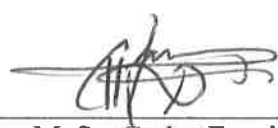
2014

Vo. Bo. :

(f) 
Ing. Kurt Kellner

Tribunal Examinador:

(f) 
Ing. Kurt Kellner

(f) 
M. Sc. Carlos Esquit

(f) 
M. Sc. Roberto Saravia

Fecha de aprobación: Guatemala 25 de Noviembre de 2014

CONTENIDO

	Página
CONTENIDO.....	v
LISTA DE CUADROS	x
LISTA DE FIGURAS	xii
RESUMEN.....	xvii
I. INTRODUCCIÓN.....	1
II. OBJETIVOS.....	3
A. GENERALES.....	3
B. ESPECÍFICOS	3
III. JUSTIFICACIÓN.....	4
IV. MARCO TEÓRICO	5
A. PROCESO DE SECADO DEL CAFÉ.....	5
B. TIPOS DE MEZCLADORAS.....	6
C. HORNOS DE LEÑA Y APLICACIÓN DE QUEMADORES DE GAS.....	7
D. QUEMADORES MODULANTES.....	8
E. TRANSFERENCIA DE CALOR.....	9
F. SISTEMAS DE CONTROL.....	9
G. CONTROLADOR PROPORCIONAL, INTEGRAL, DERIVATIVO (PID)	10
H. PÉRDIDA DE CARGA EN TUBERÍAS.....	12
I. CÁLCULO DE CAUDALES EN TUBERÍAS.....	14
J. MEDICIÓN Y SENSORES DE TEMPERATURA.....	14
K. MEDICIÓN Y SENSORES DE HUMEDAD.....	16
L. MEDICIÓN Y SENSORES DE FLUJO DE AIRE.....	16
M. SENSOR DE NIVEL HC-SR04.....	17
N. DIAGRAMAS P&ID	18
O. CONVERTIDOR DIGITAL A ANALÓGICO (DAC).....	19
P. VARIADORES DE FRECUENCIA	21
Q. ARRANQUE DE MOTORES TRIFÁSICOS DE TIPO ESTRELLA-DELTA.....	24
R. ACEROS Y SUS CARACTERÍSTICAS	26

S. RODAMIENTOS Y TIPOS DE RODAMIENTOS	27
T. ENGRANAJES Y TIPOS DE ENGRANAJES.....	30
U. COMUNICACIÓN SERIAL.....	32
V. PROTOCOLO ONE-WIRE.....	32
W. HMI	33
X. SERVIDORES VIRTUALES	34
Y. PROGRAMACIÓN DE PÁGINAS WEB	34
Z. PROTOCOLO FTP	35
AA. MICRO-CONTROLADORES	36
BB. GENERADORES ELÉCTRICOS.....	38
CC. CONTROL DE VELOCIDAD DE PLANTA GENERADORA DIÉSEL	39
DD. FUMIGACIÓN POR ASPERSIÓN	40
EE. ROYA.....	40
FF. UBUNTU	40
GG. SCRIPT	40
HH. CRON.....	41
II. CRONTAB	41
JJ. DJANGO	41
KK. DIGITALOCEAN	43
LL. NGNIX	43
MM.GUNICORN	44
NN. SUPERVISOR.....	44
OO. GOOGLE MAPS API	44
PP. GOOGLE CHART API.....	45
QQ. BASE DE DATOS	45
RR. POSTGRESQL.....	46
SS. DREAMWEAVER CC	47
TT. CSS.....	47
UU. JAVASCRIPT	47
VV. JQUERY.....	47
WW. SECURE FILE TRANSFER PROTOCOL (SFTP).....	48
XX. PUTTY	48
YY. SENSOR DE PRESIÓN	49

ZZ. SENSOR DE CAUDAL	50
AAA. BLUETOOTH	51
BBB. SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)	51
CCC. SECURE DIGITAL (SD)	51
DDD. PYTHON	52
EEE.LENGUAJE C	52
FFF. RASPBIAN	52
GGG. IMPRESORA 3D.....	52
HHH. CAD.....	53
V. IMPLEMENTACIÓN DE SENSORES PARA EL CONTROL DE TEMPERATURA EN SECADORAS ESTÁTICAS Y FLUJO DE AIRE EN SECADORAS TIPO GUARDIOLA Y MONITOREO DE HUMEDAD Y TEMPERATURA EN LAS CÁMARAS DE SECADO DE CAFÉ EN LA FINCA “LA FAMA”	54
A. METODOLOGÍA.....	54
B. DISEÑO	55
C. IMPLEMENTACIÓN	64
D. RESULTADOS	64
E. ANÁLISIS DE RESULTADOS.....	75
F. CONCLUSIONES.....	78
G. RECOMENDACIONES	79
VI. MÓDULO DE CONTROL INTEGRAL DEL PROCESO DE SECADO DEL CAFÉ Y DEL GENERADOR DIESEL DESDE UNA RASPBERRY PI VÍA WEB DE LA FINCA “LA FAMA”	80
A. METODOLOGÍA.....	80
B. DISEÑO	81
C. RESULTADOS	84
D. ANÁLISIS DE RESULTADOS.....	98
E. CONCLUSIONES.....	101
F. RECOMENDACIONES	101
VII. MECANISMO PARA EL SECADO HOMOGÉNEO DEL GRANO DE CAFÉ DE LA FINCA “LA FAMA”	103
A. METODOLOGÍA.....	103
B. DISEÑO	105
C. RESULTADOS	133

D.	ANÁLISIS DE RESULTADOS	138
E.	CONCLUSIONES	140
F.	RECOMENDACIONES	140
VIII.	MÓDULO FUMOGUA IMPLEMENTACIÓN DE SENSORES, ANÁLISIS Y TRANSMISIÓN DE LA INFORMACIÓN DE DATOS.....	141
A.	METODOLOGÍA.....	141
B.	DISEÑO	141
C.	RESULTADOS	149
D.	ANÁLISIS DE RESULTADOS.....	159
E.	CONCLUSIONES.....	162
F.	RECOMENDACIONES	162
IX.	MÓDULO FUMOGUA DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS Y VISUALIZACIÓN EN PÁGINA WEB.....	163
A.	METODOLOGÍA.....	163
B.	DESARROLLO.....	164
C.	RESULTADOS	176
D.	ANÁLISIS DE RESULTADOS.....	184
E.	CONCLUSIONES.....	187
F.	RECOMENDACIONES	188
X.	CONCLUSIONES.....	189
XI.	RECOMENDACIONES.....	190
XII.	BIBLIOGRAFÍA	191
XIII.	ANEXOS.....	198
A.	CÓDIGO FUENTE PARA MONITOREO DE TEMPERATURA Y HUMEDAD	198
B.	CÓDIGO DE PROGRAMA DE MEDICIÓN Y REGULACIÓN DE PLANTA AUXILIAR DE GENERACIÓN.....	204
C.	CÓDIGO DE PROGRAMA DE HMI.....	206
D.	CÓDIGO DE PROGRAMA DE RECOLECCIÓN DE DATOS EN RASPBERRY PI.....	223
E.	CÓDIGO DE SEGUNDO PROGRAMA DE RECOLECCIÓN DE DATOS EN RASPBERRY PI.....	230
F.	PROGRAMA PIC18F45K22 PARA TOMA DE DATOS DE LOS SENSORES Y ENVIO. 231	
G.	PROGRAMA RASPBERRY PI.....	245

H.	PLANO DE PIÑÓN DE ACELERADOR DE MOTOR DIESEL.....	254
I.	PLANO DE CREMALLERA DE ACELERADOR DE MOTOR DIESEL.....	255
J.	DIAGRAMA DE FLUJO DE DATOS DEL PROCESO DE SECADO DE CAFÉ.....	256
XIV.	GLOSARIO.....	258

LISTA DE CUADROS

	Página
1. Sistema de Ziegler-Nichols	11
2. Representación de simbología de diagrama P&ID	56
3. Diámetro y área de las tuberías	59
4. Largos de las tuberías.....	60
5. Mediciones de prueba con LM35 y pirómetro con porcentajes de error	68
6. Mediciones de pruebas de temperatura y humedad.....	70
7. Voltaje de salida de un DAC, con señales de 5V y salida luego del amplificador	72
8. Tasa de transferencia de calor requerida por cada secadora estática.....	73
9. Cálculo de caudales en tuberías	74
10. Matriz de pugh para evaluar los mecanismos del acelerador.....	81
11. Funcionamiento de mecanismo regulador de velocidad.....	96
12. Listado de materiales utilizados.....	98
13. Matriz de Pugh para la evaluación del diseño y desempeño de los mecanismos elegidos	107
14. Fuerzas ejercidas en los soportes del eje.....	108
15. Fuerza total ejercida por cada una de las partes en el plano X-Y.....	108
16. Fuerza total ejercida por cada una de las partes en el plano Y-Z	108
17. Especificaciones del motor seleccionado	113
18. Velocidades resultantes de la reducción del tren de engranes	113
19. Características para el diseño de engranes de la caja reductora	114
20. Características para el piñón y corona para la reducción en el tren de engranes.....	115
21. Factores para el cálculo del esfuerzo real de flexión.....	118
22. Factores para el cálculo del esfuerzo de flexión permisible.....	120
23. Factores para el cálculo del esfuerzo de contacto real	122
24. Factores para el cálculo del esfuerzo de contacto permisible.....	123
25. Propiedades del material de los ejes de la caja reductora.....	123
26. Cuadro que muestra el esfuerzo de fatiga de los ejes.....	125
27. Torques resultantes	125
28. Fuerzas a las que están sometidos los ejes de la caja reductora	126
29. Diámetro mínimo para los ejes de la caja reductora.....	133
30. Dimensiones de cada parte del mecanismo	134
31. Rodamiento seleccionado para el eje del mecanismo	134
32. Variables para el cálculo de la fuerza necesaria para movilizar el café	135
33. Sensor de presión	149
34. Sensor de caudal	150
35. Adafruit ultimate GPS.....	150
36. Bluetooth HC-05.....	150
37. Parámetros de operación	151
38. Prueba de datos enviados desde el microcontrolador.....	152
39. Prueba final de datos enviados desde el microcontrolador	152
40. Prueba final datos que son enviados a la base de datos.....	153
41. Tabla desempeño de jornaleros	181

42. Tabla world_datos_fincas de la Base de Datos.....	181
43. Tabla de desempeño de los jornaleros.....	183
44. Tabla world_datos_fincas de la base de datos.....	183

LISTA DE FIGURAS

	Página
1. Patio de secado de granos.....	5
2. Secadoras tipo Guardiola, foto tomada en finca "La Fama".....	6
3. Mezclador de doble cinta o helicoidal.....	6
4. Mezcladora de paletas.....	7
5. Quemador de gas.....	8
6. Diagrama de un quemador modulante.....	9
7. Diagrama de bloques de un sistema PID.....	11
8. Coeficientes de pérdida.....	13
9. Diagrama de Moody.....	13
10. Conexión del LM35 para mediciones de 2 a 150°C.....	15
11. Circuito interno LM35.....	15
12. Módulo ultrasónico HC-SR04.....	17
13. Parámetros eléctricos de módulo HC-SR04.....	18
14. Señales de funcionamiento del módulo HC-SR04.....	18
15. Letras de identificación en un sistema P&ID.....	19
16. DAC con ponderación binaria.....	20
17. DAC en escalera R/2R.....	20
18. DAC en escalera R/2R de 2 bits.....	21
19. Variador de frecuencia Allen-Bradley.....	21
20. Tarjeta de control ACS 300.....	22
21. Diagrama de bloques de un ACS 300.....	22
22. Diagrama de entradas digitales ACS 300.....	23
23. Diagrama recomendado para conexión 3-wire del ACS 300.....	23
24. Rampas de aceleración y desaceleración para ACS 300.....	24
25. Configuración estrella (a) y delta (b).....	25
26. Arranque por transición abierta (izquierda) y transición cerrada (derecha).....	25
27. Rodamiento de bolas.....	27
28. Rodamiento de rodillo.....	28
29. Rodamiento de bolas a rótula.....	28
30. Rodamiento de rodillos a rótula.....	29
31. Rodamiento de simple efecto.....	29
32. Rodamiento de doble efecto.....	30
33. Engrane recto.....	30
34. Engranajes helicoidales.....	31
35. Engranajes cónicos.....	31
36. Tornillo sin fin.....	32
37. Paquete de medición para Visual Studio.....	33
38. Ejemplo de código HTML.....	34
39. Ejemplo de código PHP.....	35
40. Modelo funcional del protocolo FTP.....	36
41. Arquitectura interna del micro-controlador.....	36

42.	Componentes principales del Olimexino-328	37
43.	Raspberry Pi modelo B+	38
44.	Planta auxiliar de generación eléctrica de finca La Fama.....	39
45.	Estructura de Django	42
46.	Sistemas operativos de DigitalOcean	43
47.	Comunicación Supervisor-Gunicorn-Nginx	44
48.	Características de PostgreSQL	46
49.	Entorno FileZilla	48
50.	Entorno PuTTY	49
51.	Sensor de presión 100psi	50
52.	Sensor de caudal HQ-A68-1	50
53.	Raspbian	52
54.	Impresora 3D a base de plástico	53
55.	Observación de procesos utilizados en finca "La Fama"	54
56.	Diagrama P&ID de los sistemas diseñados	55
57.	Identificación de tuberías para cálculo de caudales	59
58.	Diagrama de flujo de monitoreo de variables	65
59.	Envío serial de mediciones de sensores	66
60.	Prueba de laboratorio con sensores e indicadores luminosos	66
61.	Prueba de laboratorio con indicadores luminosos indicando la temperatura del aire	67
62.	Prueba de sensores LM35 y medición por medio de un pirómetro	67
63.	Medición de temperatura por medio de un pirómetro	69
64.	Prueba de medición de temperatura	69
65.	Prueba de medición de humedad	69
66.	Diagrama de bloques sistema de control de temperatura	71
67.	Diagrama de flujo sistema de control de temperatura	71
68.	Lógica analógica para control de temperatura	72
69.	Circuito DAC con amplificador para determinar la entrada al quemador	72
70.	Diagrama de bloques del sistema de control de flujo de aire	73
71.	Diagrama de flujo de control de flujo de aire	73
72.	Esquema eléctrico para control de variador de frecuencia con arranque estrella-delta	74
73.	Relación distancia-voltaje de mecanismo acelerador de motor diesel.....	82
74.	Relación ángulo-distancia de mecanismo regulador de velocidad	83
75.	Simulación de mecanismo acelerador con el software Autodesk Inventor 2014.....	83
76.	Diagrama de conexión de sensores y actuadores a Olimexino 328	84
77.	Diagrama de conexión del sistema completo	84
78.	Diagrama de flujo de programa de medición y regulación de planta auxiliar	85
79.	Diagrama de flujo de programa de HMI.....	86
80.	Diagrama de flujo de programa de recolección de datos en Raspberry Pi	87
81.	Diagrama de flujo de segundo programa de recolección de datos en Raspberry Pi	88
82.	Sección de aplicación python para control de velocidad del motor diesel	89
83.	Aplicación python para monitoreo de secadoras estáticas y guardiolas	89
84.	Sección de aplicación python para visualización de último ingreso de datos.....	90
85.	Visualización de temperatura en generador diesel. Aplicación de visual c#	90
86.	Visualización de voltaje en generador diesel. Aplicación de visual c#	91

87.	Visualización de nivel de combustible en generador diesel. Aplicación de visual c#.....	91
88.	Visualización de datos de guardiola 1. Aplicación de visual c#.....	92
89.	Visualización de datos de guardiola 2. Aplicación de visual c#.....	92
90.	Visualización de datos de secadora estática 1. Aplicación de visual c#.....	93
91.	Visualización de datos de secadora estática 2. Aplicación de visual c#.....	93
92.	Almacenamiento de datos en archivo excel generado con aplicación de visual c#.....	94
93.	Datos recibidos desde Raspberry Pi almacenados para su uso en aplicación de visual c#.....	94
94.	Circuito esquemático de rectificador de voltaje AC.....	95
95.	Circuito en placa de cobre de rectificador de voltaje AC.....	95
96.	Circuito rectificador de voltaje AC soldado en placa de cobre.....	96
97.	Mecanismo acelerador de motor diésel.....	97
98.	Despliegue de datos via web.....	97
99.	Pilas donde se coloca el café.....	103
100.	Instrumento utilizado para movilizar el café.....	104
101.	Representación de planos de trabajo para el análisis de fuerzas.....	109
102.	Diagrama de corte del eje en el plano X-Y.....	109
103.	Momento flector del eje en el plano X-Y.....	110
104.	Ángulo de flexión del eje en el plano X-Y.....	110
105.	Flexión en el eje en el plano X-Y.....	110
106.	Diámetro ideal para el eje sugerido por Inventor para el plano X-Y.....	111
107.	Diagrama de corte del eje en el plano Y-Z.....	111
108.	Momento flector del eje en el plano Y-Z.....	112
109.	Ángulo de flexión del eje en el plano Y-Z.....	112
110.	Flexión del eje en el plano Y-Z.....	112
111.	Diámetro ideal del eje sugerido por Inventor para el plano Y-Z.....	113
112.	Cálculo de interferencia en relación al número de dientes.....	115
113.	Tabla de factores de sobrecarga.....	116
114.	Factor dinámico.....	116
115.	Factor de forma de Lewis.....	117
116.	Factor de confiabilidad.....	119
117.	Vida de diseño (L) en horas, dependiendo de la aplicación.....	119
118.	Valores del factor por ciclo de esfuerzo utilizando el número de ciclos de carga.....	120
119.	Módulo de elasticidad de los materiales.....	121
120.	factor de ciclos de esfuerzo.....	122
121.	gráfica que muestra la resistencia a la fatiga en función de la resistencia a la tensión.....	124
122.	Factor de confiabilidad.....	124
123.	Diagrama de corte de las fuerzas verticales del eje de entrada.....	127
124.	Diagrama de momento de las fuerzas verticales del eje de entrada.....	127
125.	Diámetro ideal del eje de entrada.....	127
126.	Diagrama de corte de las fuerzas horizontales del eje de entrada.....	128
127.	Diagrama de momento de las fuerzas horizontales del eje de entrada.....	128
128.	Diámetro ideal para las fuerzas horizontales.....	128
129.	Diagrama de corte de las fuerzas horizontales del eje intermedio.....	129
130.	Diagrama de momento de las fuerzas horizontales del eje intermedio.....	129
131.	Diámetro ideal para las fuerzas horizontales del eje intermedio.....	129

132. Diagrama de corte para las fuerzas verticales del eje intermedio	130
133. Diagrama de momento de las fuerzas horizontales del eje intermedio	130
134. Diámetro ideal para las fuerzas verticales del eje intermedio	130
135. Diagrama de corte de las fuerzas verticales del eje de salida	131
136. Diagrama de momento de las fuerzas verticales del eje de salida	131
137. Diámetro ideal de las fuerzas verticales para el eje de salida	131
138. Diagrama de corte de las fuerzas horizontales del eje de salida	132
139. Diagrama de momento de las fuerzas horizontales del eje de salida	132
140. Diámetro ideal para las fuerzas horizontales del eje de salida	132
141. Diseño de paleta para el mecanismo diseñada en Autodesk Inventor	135
142. Soporte de la paleta del mecanismo diseñado en Autodesk Inventor	135
143. Eje del mecanismo diseñado en Autodesk Inventor	136
144. Vista isométrica del mecanismo diseñado en Autodesk Inventor	136
145. Vista de planta del mecanismo diseñado en Autodesk Inventor	136
146. Vista lateral del mecanismo diseñado en Autodesk Inventor	137
147. Soldadura de las paletas al eje del mecanismo	137
148. Estructura final del sistema de potencia mecánico	137
149. Sensor de presión Honeywell	142
150. Adafruit Ultimate GPS	142
151. Módulo de comunicación	143
152. Posición de acople	144
153. Acople prototipo 1	144
154. Acople prototipo 2	145
155. Acople final	145
156. Acople conectado en la bomba	146
157. Diagrama de flujo programa del microcontrolador	146
158. Diagrama de flujo programa de la Central	148
159. Acople de sensores	150
160. Conexiones hacia el microcontrolador	151
161. Primera prueba real realizada	154
162. Prueba final con datos reales	154
163. Función para transformar los datos de la presión en una escala	155
164. Primera parte de la función escala - presión	155
165. Segunda parte de la función escala - presión	156
166. Función para transformar los datos de caudal en una escala	156
167. Primera parte función escala - caudal	157
168. Segunda parte de la función escala - caudal	157
169. Muestra la escala de la presión en el rango de 0 a 10	158
170. Escala del caudal	158
171. Escala después de analizar la escala de la presión y el caudal	159
172. Droplets en DigitalOcean	164
173. Selección de tamaño de Droplet	164
174. Añadir dominio	165
175. DNS en DigitalOcean	165
176. Nameserver Settings en GoDaddy	166

177.	Configuración de PuTTY	166
178.	ventana PuTTY, ingreso usuario y password	167
179.	página obtenida cuando funciona Django	168
180.	Uso de FileZilla	176
181.	Página Inicio	176
182.	Página Nosotros	177
183.	Página Producto	177
184.	Página de Contacto	178
185.	Formulario de Acceso a información	178
186.	Usuarios para los clientes	178
187.	Calendario para filtrar información	179
188.	Mapa con filtros	179
189.	Puntos con Roya	180
190.	Gráfica del desempeño de los Jornaleros	180
191.	Mapa para ANACAFE	181
192.	Recorrido con dispositivo ensamblado a la bomba	182
193.	Desempeño bajo en mapa	182
194.	Gráfico de uso del equipo	182
195.	Diagrama de flujo de datos del proceso de secado de café	257

RESUMEN

El megaproyecto consiste en implementar tecnologías para aumentar la eficiencia en las fincas de café. Se diseñaron sistemas de automatización para el proceso de secado de café. Se diseñó un sistema compuesto por un quemador de gas, controlado por medio de un sistema proporcional para así mantener la temperatura en un rango deseado. Para ingresar el aire caliente dentro de las secadoras tipo guardiola se diseñó un sistema que consiste en un variador de frecuencia para el motor del ventilador. Se diseñó un mecanismo que realice el movimiento de granos, tomando en cuenta los materiales adecuados, fuerzas necesarias y las dimensiones de las cámaras de secado.

Se diseñó un sistema que regula el nivel de voltaje de la señal eléctrica de un generador, que será inyectada a la red interna de la finca “La Fama”. También se implementaron indicadores de temperatura y nivel de diésel en el tanque del motor. Se desarrolló una aplicación para sistemas operativos Windows en Visual C# que tiene como objetivo mostrar los datos obtenidos por los sensores de las secadoras y de la generación de energía eléctrica. Además esta aplicación permite generar un archivo Excel donde se obtienen los datos del proceso de secado que son ingresados desde la Raspberry Pi y actualiza de forma constante una página web donde se muestran las mediciones principales.

Se creó un dispositivo que realiza el monitoreo de fumigación, lo cual provee al caficultor la información sobre la fumigación de cada jornada y la forma en la que el jornalero está utilizando el equipo. Esto se realizó por medio de un sistema que obtiene datos del recorrido del jornalero y una escala de uso del equipo, obtenido por el caudal y presión manejada en la bomba. También se elaboró un dispositivo para la toma y análisis de fotos de hojas con roya. Esta información fue desplegada en una página web con acceso a los caficultores.

I. INTRODUCCIÓN

El campo de diseño de la ingeniería busca la implementación de nuevos sistemas o métodos que no sólo satisfagan una necesidad determinada, sino que optimicen algunos procesos, los cuales permiten la producción de nuevos productos, reducción de costos de operación y producción, y mejorar la calidad. El presente trabajo de graduación consta de dos partes, automatización del proceso de secado de café y monitoreo de fumigación.

Para la parte de automatización del proceso de secado de café se trabajó en la finca “La Fama”. Se buscó dar una nueva alternativa para la homogenización de la temperatura del grano del café. Para comenzar con el diseño de la máquina se tomó como referencia mezcladoras de sólidos ya existentes en el mercado, se tuvo que analizar cada uno de estos para verificar si el grano del café sería dañado al estar en movimiento. Asimismo se tuvo que tomar en cuenta los materiales adecuados para el diseño, simulación y construcción. El material tenía que ser capaz de estar en ambiente húmedo y capaz de soportar fuerzas moderadas, por lo que tenía que analizarse con detalle cuál sería el ideal para este tipo de mezcladora. Se realizaron simulaciones en el ensamblaje de piezas para conocer los esfuerzos a los que estaría sometido el eje y para conocer el rodamiento adecuado para adaptar en los extremos del mecanismo.

Se creó un sistema de monitoreo de temperatura y humedad dentro de las cámaras de secado del café, por medio de un sensor DHT22. También se midió la temperatura del aire que entra a las cámaras de secado, por medio de un sensor LM35. Se diseñó un sistema de control de temperatura del aire que entra a las cámaras de secado mediante un quemador modulante. Para el control del quemador modulante, se colocó un comparador analógico donde se tiene una señal de referencia y la medición de temperatura del LM35, para el cálculo del error. Con el valor del error se puede llegar a determinar si se requiere de mayor o menor temperatura.

Se diseñó un sistema de control de flujo de aire para las secadoras tipo guardiola, donde se utiliza un variador de frecuencia para modificar la velocidad de giro del motor que mueve el ventilador. Se tiene un comparador analógico para el cálculo del error, teniendo una señal de referencia que puede ser modificada por el operador y una señal del sensor de flujo de aire. Se realizó un esquema eléctrico para el control del variador de frecuencia, con el valor del error se selecciona la acción a tomar, ya sea acelerar o desacelerar el variador. También se hicieron cálculos de pérdida de carga para encontrar los valores de los caudales en cada una de las entradas de las secadoras tipo guardiola.

Se desarrollaron diferentes programas de recolección de datos, medición de variables, control de generación de energía eléctrica y despliegue de datos. Para esto se utilizó un Olimexino-328 el cual es una versión mejorada del ya conocido Arduino. Para la planta auxiliar de generación de energía eléctrica se desarrollaron sistemas de medición de temperatura del motor diésel, y medición de nivel de combustible. Este Olimexino-328 envía los datos medidos a una Raspberry Pi que recolecta estos datos y los datos medidos en las cuatro secadoras del beneficio de café. En la Raspberry Pi se ejecutan dos programas paralelos uno de ellos posee una interfaz gráfica para que el usuario ingrese datos de los procesos de secado y pueda controlar el nivel de voltaje generado en la planta auxiliar. Este proceso de regulación de voltaje se diseñó para tener un error máximo del 5%.

El segundo programa que corre la Raspberry Pi no posee interfaz gráfica y solamente recibe los datos enviados desde el Olimexino-328 encargado de medir variables de las secadoras. Los datos recibidos son almacenados en un archivo .txt para que la computadora central por medio de un HMI realice una petición FTP para obtener una copia de los archivos. En la computadora central se muestran los valores medidos en las cuatro secadoras del beneficio y los variables medidas en la planta auxiliar. Estos datos son mostrados al usuario en forma de gráficas y se le permite realizar una descarga en formato Excel de los datos medidos.

La fumigación en el cultivo del café es necesaria en la mayoría de casos, ya que existen muchos agresores, entre estos está la roya. La roya es un hongo que ataca directamente a las hojas de la mata dañándolas evitando que la planta se alimente, si no se llevan a cabo acciones en contra de la roya la planta se morirá y en el transcurso enfermará a otras matas causando una posible plaga ya que como medio de transporte utiliza el aire.

Debido a estos problemas es necesaria la creación de una tecnología capaz de monitorear las partes ya fumigadas de la finca y las que se necesitan fumigar. Teniendo en cuenta la calidad adecuada de la fumigación y no reparar área ya fumigada para no echar a perder el producto aplicado.

Por lo que se creó un dispositivo con un sensor de caudal, de presión y un GPS para lograr el monitoreo los datos recolectados durante la jornada de fumigación se transferirán inalámbricamente a una central para su análisis y posterior envío a la base de datos.

Con en análisis anterior se puede determinar si el caudal y la presión de la bomba a la hora de fumigar son óptimas, de lo contrario se puede verificar las áreas mal fumigadas para volver a fumigar evitando que se propague la roya. En nuestro caso una fumigación óptima está con una presión de 30psi a 50psi y un caudal entre 0.7 l/min a 1.3 l/min.

II. OBJETIVOS

A. GENERALES

Implementar tecnologías que permitan la automatización del secado de café y el monitoreo de la fumigación en las fincas de café.

B. ESPECÍFICOS

- a. Diseñar un sistema de control y monitoreo de temperatura y humedad, de bajo costo, para las secadoras estáticas y tipo guardiola utilizadas en el proceso de secado del café en la finca “La Fama”.
- b. Diseñar y construir un mecanismo de bajo costo capaz de homogenizar el secado del grano de café en las secadoras estáticas de la finca “La Fama”.
- c. Diseñar y construir un sistema de control de velocidad de bajo costo para un motor diésel y desarrollar una aplicación que despliegue estadísticas de los procesos de secado realizados en la producción de café en finca “La Fama”.
- d. Diseño e implementación de una base de datos y página web, que provea a los caficultores información sobre la fumigación que se ha realizado en su finca; e información a ANACAFE para conocer los sectores de roya en todas las fincas.
- e. Diseñar y construir el sistema de sensores para recolectar información de la operación de la bomba y el sistema de transmisión de esta información a los servidores informáticos.

III. JUSTIFICACIÓN

En Guatemala el grano de café es el segundo producto más exportado, por lo que es importante la implementación de tecnología en sus procesos, para así tener un mejor control de los factores que pueden afectar la calidad del mismo. El café, como cualquier otro cultivo, requiere de un procedimiento de fumigación específico para poder combatir distintos tipos de hongos; por ejemplo, la roya que es el hongo que más ha perjudicado a los caficultores en los últimos años. Por otro lado, el control de los procesos en los que pasa el grano de café, antes de la exportación, es sumamente importante para llevar la calidad que lo caracteriza.

Durante el proceso de secado de café, dos de los puntos más importantes que se deben tomar en cuenta son los siguientes: la temperatura del aire que ingresa a las secadoras y humedad del grano. La humedad del café durante el proceso de secado está relacionada con el tiempo que se encuentra en contacto con calor. La temperatura a la que debe estar la cámara de secado del café debe permanecer constante para no afectar la calidad del mismo.

En la finca La Fama no se posee un sistema de control para el generador secundario lo cual puede tener consecuencias tan graves como el daño de maquinaria y/o equipos eléctricos/electrónicos que se tengan conectados a la red eléctrica interna de la finca. Actualmente este problema es solucionado de forma superficial mediante la aceleración manual del motor diésel de forma periódica durante todo el tiempo que este generador secundario esté en funcionamiento.

La roya toma al año un aproximado de 760,000 quintales de café, lo que representa una pérdida de \$372,000,000.00 y se estima que para el 2014 la roya genere directa e indirectamente unos 115,000 desempleos en Guatemala (según datos de PROMECAFE Guatemala). Por ese motivo es necesario que idealmente cada caficultor posea un sistema de monitoreo en tiempo real el cual informa, como se está fumigando.

Se planteó la opción de construir un dispositivo de monitoreo, que pueda ayudar a los caficultores a tener un control sobre su finca y trabajadores. En Guatemala, no existe un sistema, que guíe al fumigador para que pueda realizar de manera efectiva su labor. Contratar supervisores para que estén corrigiendo a los trabajadores para que fumiguen de manera eficiente, sería costoso. Por lo que este dispositivo sería capaz de mostrarle a los cafetaleros, que trabajador no está realizando de manera eficiente la fumigación.

IV. MARCO TEÓRICO

A. PROCESO DE SECADO DEL CAFÉ

Al finalizar el proceso de lavado de café, se pasa al secado del mismo. Éste consiste en bajar la humedad del grano hasta un 10-12% dependiendo del valor establecido por la finca, por medio de la aplicación de calor. Se pueden aplicar diferentes métodos de secado, entre ellos se encuentra el secado natural y mecánico. (AnaCafé, s.f.)

Para la medición de humedad en los granos, se puede realizar mediante la medición de la humedad relativa del aire, en donde se realizan curvas de comparación entre la humedad del grano y la humedad del aire. Una vez realizadas las curvas de relación se puede determinar la humedad en la que se encuentra el grano al medir la humedad relativa del aire. (de Dios, 1996)

1. **Secado natural.** El secado natural consiste en la utilización del calor del sol para secar el grano de café. Se requiere de un espacio grande y plano para colocar capas de 5 – 6 cm de grosor de grano de café. El grano requiere que se mueva de manera constante, para garantizar que los granos de abajo se sequen de la misma manera que los de arriba. Una ventaja de utilizar este método, es que los gastos en equipo son casi nulos y el costo de mantenimiento es bajo. La mayor desventaja radica en la cantidad de sol que haya por día y no se puede utilizar el método en días de lluvia. (AnaCafé, El beneficiado húmedo, s.f.)

Figura 1 Patio de secado de granos



(AnaCafé, El beneficiado húmedo, s.f.)

2. **Secado mecánico.** El secado mecánico consiste en la utilización de secadoras tipo guardiola, la cual consiste en un contenedor giratorio, en la cual se aplica aire caliente a través de un horno o caldera con ayuda de un ventilador (para impulsar el aire caliente en el grano). Es recomendable que el aire se encuentre entre 50-60°C para secar de una manera eficiente y no quemar el grano. (AnaCafé, El beneficiado húmedo, s.f.)

Figura 2 Secadoras tipo Guardiola, foto tomada en finca "La Fama"



B. TIPOS DE MEZCLADORAS

El mezclado es un proceso en donde generalmente varios ingredientes entran en contacto para formar una mezcla homogénea. Sin embargo, existen mezcladoras que solamente buscan homogenizar un material para tener una sustancia más fina o para cambiar la temperatura del mismo. Existen varios tipos de mezcladoras, se describirán dos diferentes: (QuimiNet, 2006)

1. **Mezclador helicoidal.** Este tipo de mezcladoras son consideradas poco costosas, sencillas y compactas. Consisten básicamente de dos helicoides, las cuales pueden realizar dos movimientos, dependiendo de la construcción: arrastrar el material al centro o hacia los laterales. Una de las ventajas de este tipo de mezcladoras es la rapidez de mezclado, además de poder tratar con materiales pesados. Una de las desventajas es la contaminación en el material y el dimensionamiento limitado del mecanismo. (Trevi, 2007)

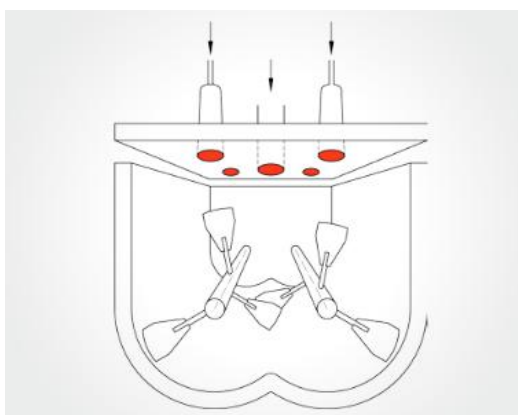
Figura 3: Mezclador de doble cinta o helicoidal



(TecnMixer, S.A., 2010)

2. Mezclador de paletas. Los mezcladores de paletas están diseñados para trabajos pesados. Probablemente es el tipo de mezclador más antiguo, pero eficiente. Las paletas están montadas en un eje giratorio, por lo que el material es empujado alrededor del recipiente siguiendo una trayectoria circular. La distribución de las paletas permite que exista alta turbulencia y una buena mezcla del material. La higiene en este tipo de mecanismo es alta, debido a su facilidad de limpieza. Existen paletas corredizas, paletas de doble movimiento, paletas con movimiento planetario o agitadores. (TECAM, 2012)

Figura 4: Mezcladora de paletas



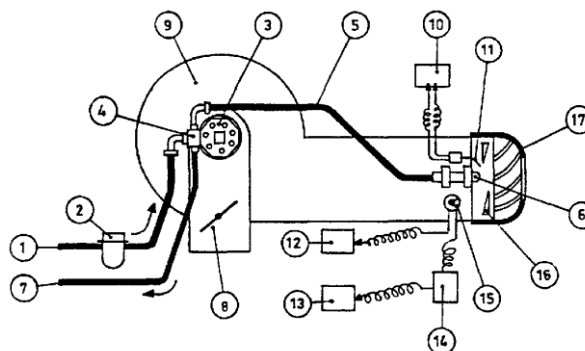
(LASKA, s.f.)

C. HORNOS DE LEÑA Y APLICACIÓN DE QUEMADORES DE GAS

Un horno de leña consiste en un cuarto pequeño cerrado, donde se coloca leña para producción de calor, la leña es un combustible económico y de fácil combustión. Una desventaja de la utilización de leña es el espacio que ocupa y el acarreo de la misma puede resultar incómodo y complicado. También se requiere de una persona que se encargue de abastecer el horno y regular la temperatura, lo cual hace que la transmisión de calor sea de manera irregular. (de Dios, 1996)

Para reducir el error del operario, se puede reemplazar el uso de leña por gases o residuos naturales, tales como bagazo de caña de azúcar, biogás, carbón vegetal, entre otros. De la misma manera, la utilización de éstos reduce la contaminación de los granos. Cuando se utiliza gas, se debe tener mucha precaución para prevenir fugas del mismo. Una desventaja es que se debe controlar la presión de atomización y estos sufren desgastes y obstrucciones frecuentes, por lo que requieren mantenimientos frecuentes. (de Dios, 1996)

Figura 5 Quemador de gas



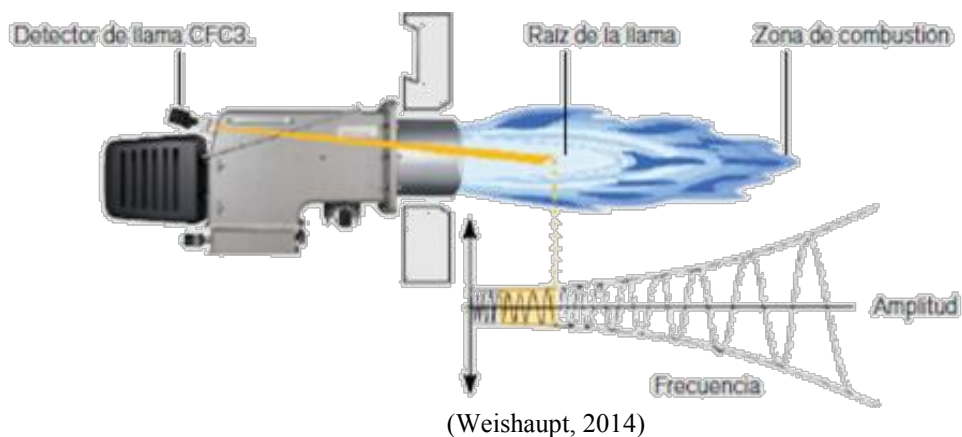
- | | |
|---|----------------------------------|
| 1. Conducto de llegada del combustible | 8. Válvula de regulación de aire |
| 2. Filtro | 9. Ventilador |
| 3. Bomba | 10. Transformador |
| 4. Regulador de presión | 11. Electrodo |
| 5. Conducto de la alimentación al gliceur | 12. Contactor |
| 6. Gliceur (pico) | 13. Motor |
| 7. Conducto de retorno de combustible | 14. Relais |
| | 15. Célula fotorresistente |
| | 16. Deflector |
| | 17. Boca |

(de Dios, 1996)

D. QUEMADORES MODULANTES

Un quemador modulante tiene la capacidad de controlar el tamaño de la llama de salida, con el fin de mantener la variable requerida, por medio del control de potencia. El principio de modulación se logra por medio de la separación entre el quemador y la estación de precalentamiento y bombeo. Los quemadores modulantes pueden controlar el nivel del gas y el volumen de aire por medio de válvulas y un control PID para asegurar que la salida cumpla con los requerimientos que se indiquen por el operador. Cuando se requiera utilizar un quemador modulante, se debe determinar correctamente el necesario, ya que si la función se encuentra fuera de las capacidades del quemador se puede llegar a tener un control on/off (encendido/apagado) causando así una pérdida de energía y un sistema poco eficiente. Para evitar este problema se debe tomar en cuenta el tamaño del quemador, la capacidad, combustión y se deben fijar correctamente los parámetros del control PID. La mayoría de quemadores modulantes poseen un control digital de la combustión, el cual aumenta la eficiencia del mismo. Los gases más utilizados por estos quemadores son el gas natural, gasóleo, gasoil. (SabienTechnology, s.f.) (Weishaupt, 2014)

Figura 6 Diagrama de un quemador modulante



E. TRANSFERENCIA DE CALOR

Los procesos de transferencia de calor se pueden dividir por conducción, convección y radiación. La conducción se basa en el cambio de momentum (cantidad de movimiento de moléculas) a causa de un cambio de temperatura por medio de contacto, ésta ocurre de un objeto que se encuentra a una temperatura alta a uno de menor temperatura. La convección se da al mezclar un fluido caliente con uno más frío, también se puede dar por medio de un fluido que hace contacto con un cuerpo o tuberías a diferentes temperaturas. La radiación es la energía emitida por un sólido o líquido, que al hacer contacto con otro ocurre una transferencia de calor. La tasa de transferencia de calor es la cantidad de calor que se transmite por unidad de tiempo, ésta se da únicamente cuando existe diferencia de temperatura. La tasa de transferencia de calor se encuentra dada por la ley de conducción de calor de Fourier: (Fonseca) (Inzunza)

$$H = \frac{\Delta Q}{\Delta t} \quad (1)$$

Donde:

H representa la tasa de transferencia de calor

ΔQ representa el cambio de calor

Δt representa el cambio del tiempo

F. SISTEMAS DE CONTROL

Existen dos diferentes tipos de sistemas de control, el sistema de control de lazo abierto y el sistema de control de lazo cerrado.

1. **Sistema de control de lazo cerrado.** Para un sistema de control de lazo cerrado se tiene una característica importante: posee una referencia, lo cual es el objetivo del sistema. Este sistema también es conocido como un sistema realimentado, ya que la salida del sistema es comparada con la referencia, donde la diferencia entre ambas será la entrada al controlador. El controlador se encarga de actuar sobre la variable de entrada, para que dicho sistema llegue de nuevo al valor de referencia. La ventaja principal de utilizar un sistema de lazo cerrado, es que es posible llegar a un valor deseado de manera automática, sin recurso humano, donde se reducen errores y se realiza de una manera rápida. (Rodríguez, 2013) (Ramírez, 2014)

2. **Sistema de control de lazo abierto.** Un sistema de control de lazo abierto, carece de retroalimentación, por lo que la salida únicamente depende de la entrada. Al no tener retroalimentación, el sistema no tiene la capacidad de realizar cambios en la salida, por lo que se ve afectado por las perturbaciones que son causadas en la entrada. (Rodríguez, 2013) (Ramírez, 2014)

G. CONTROLADOR PROPORCIONAL, INTEGRAL, DERIVATIVO (PID)

Un PID utiliza un sistema de control de lazo cerrado (con retroalimentación) el cual tiene la capacidad de corregir el error en la salida comparándolo con una señal de referencia; esto con el fin de manejar un elemento de control. Un PID utiliza un algoritmo con tres parámetros: el proporcional, integral y derivativo. La ecuación que describe al sistema es la siguiente: (Mazzone, 2002)

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \quad (2)$$

Donde:

$e(t)$ representa la señal del error

K_p representa la ganancia proporcional

T_i representa el tiempo integral

T_d representa el tiempo derivativo

Para determinar los valores de las constantes K_p, T_i , y T_d se utiliza el método de Ziegler-Nichols, para el cual se requiere encontrar el valor crítico de la ganancia proporcional K_p y el período de oscilación (representado como P_c). Para encontrar dichos valores se deben fijar en cero los valores de integral y derivativo; a continuación se aumenta gradualmente el valor de P hasta que el sistema empiece a oscilar, en

ese momento el valor de K_p es crítico y el valor de P_c puede ser determinado. Una vez encontrados dichos valores, se utiliza la siguiente tabla para determinar los valores del proporcional, T_i y T_d : (Instruments N. , 2011)

Cuadro 1 Sistema de Ziegler-Nichols

Sistema de control	P	T_i	T_d
P	$0.5K_p$	-	-
PI	$0.45K_p$	$P_c/1.2$	-
PID	$0.60K_p$	$0.5P_c$	$P_c/8$

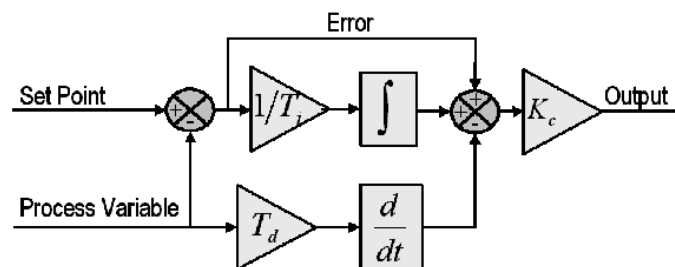
(Instruments N. , 2011)

1. **Componente proporcional.** El componente proporcional depende únicamente del error producido, es decir, la diferencia entre la referencia y la señal de salida. La ganancia proporcional K_p determina el tiempo de reacción del sistema al error, por lo que mientras más grande sea dicha ganancia, mayor será el tiempo de reacción, pero si éste llega a ser muy grande el sistema empezará a oscilar y, asimismo, causar inestabilidad. (Instruments N. , 2011)

2. **Componente integral.** El componente integral realiza la suma del error durante el tiempo, con el fin de corregir el error del sistema. La respuesta de la integral crece durante el tiempo, a menos que éste sea cero, por lo que causa que la diferencia entre la referencia y la salida sea llevada a cero. (Instruments N. , 2011)

3. **Componente derivativo.** El componente derivativo determina el tiempo de reacción cuando el error es producido. Cuando el tiempo derivativo T_d aumenta, el sistema reacciona mejor y más rápido a los cambios de error producidos, generalmente se utiliza en valores bajos ya que un valor alto puede causar inestabilidad en el sistema. (Instruments N. , 2011)

Figura 7 Diagrama de bloques de un sistema PID



(Instruments N. , 2011)

H. PÉRDIDA DE CARGA EN TUBERÍAS

Las pérdidas de carga en tuberías son las presiones que requiere la sustancia para vencer la fricción del medio por donde se mueven, en este caso, las tuberías. Las pérdidas de carga se ven afectadas por las dimensiones de las tuberías (diámetros y largos), velocidad de la sustancia, densidad, rugosidad de las paredes de las tuberías y el coeficiente de fricción de los mismos. Se debe determinar las pérdidas de carga de cada elemento que constituye las tuberías, como tramos rectos, codos, obstáculos, entre otros. (Palau)

Para el cálculo de pérdidas de carga, en cualquier tubería se utiliza la ecuación (4), donde se expresan en función del coeficiente de fricción de la tubería; el coeficiente de fricción se determina por medio del diagrama de Moody. También se debe tomar en cuenta que existen pérdidas menores por uniones, codos, válvulas, ensanchamientos, entre otros; las cuales son determinadas por la ecuación (5). Éstas se encuentran en función de un coeficiente de pérdida, el cual se determina por medio de tablas. (Cengel & Cimbala, 2010)

$$h_L = f \frac{LV_{prom}^2}{2Dg} \quad (3)$$

Donde:

h_L representa la pérdida de carga

L representa el largo de la tubería

f representa el coeficiente de fricción

V_{prom} representa la velocidad promedio del flujo

D representa el diámetro de la tubería

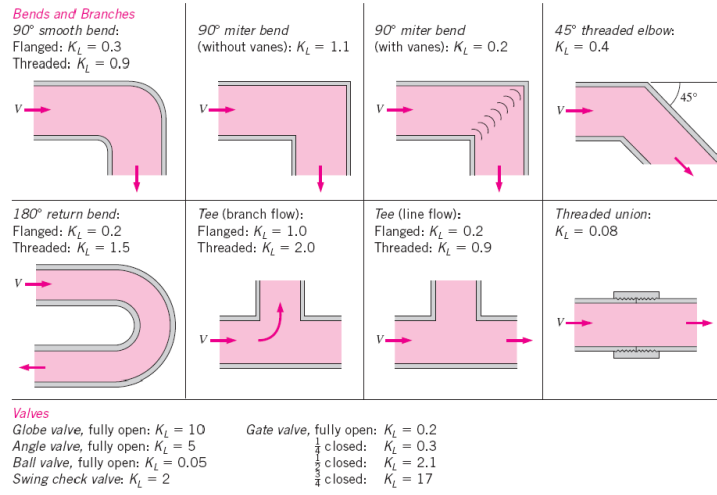
g representa el valor de la gravedad (9.81 m/s^2)

$$h_L = K \frac{V_{prom}^2}{2g} \quad (4)$$

Donde:

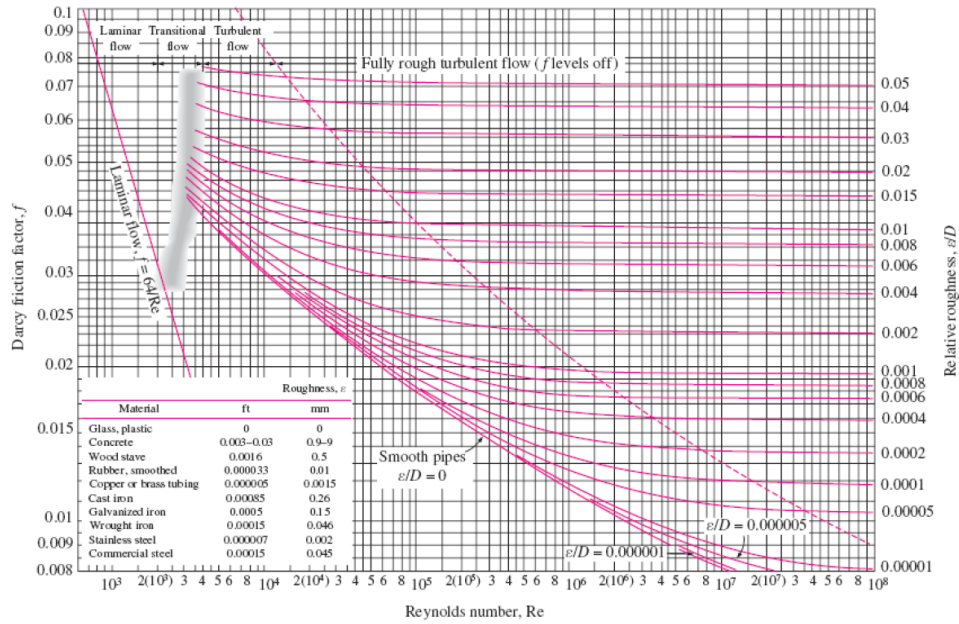
K representa el coeficiente de pérdida

Figura 8 Coeficientes de pérdida



(Cengel & Cimbala, 2010)

Figura 9 Diagrama de Moody



(Cengel & Cimbala, 2010)

I. CÁLCULO DE CAUDALES EN TUBERÍAS

Cuando se tienen tuberías en serie, el caudal es igual en cada una de ellas y la pérdida de carga total es la suma de las pérdidas en cada una. Por otro lado, cuando se tienen tuberías en paralelo, la suma del caudal de cada tubería es igual al caudal total y las pérdidas de carga son iguales para cada tubería. Para el cálculo de los caudales, se debe tomar en cuenta el principio de conservación de la masa, es decir que la misma masa que entra es la que sale. Por medio de las pérdidas de carga, se puede llegar a determinar la velocidad promedio del flujo, por lo que para determinar el caudal del mismo se utiliza la siguiente ecuación: (Cengel & Cimbala, 2010)

$$Q = VA \quad (5)$$

Donde:

Q representa el caudal

V representa la velocidad del flujo

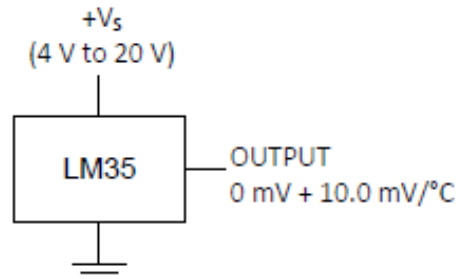
A representa el área transversal de la tubería

J. MEDICIÓN Y SENSORES DE TEMPERATURA

La temperatura es un valor que se encuentra relacionado con la energía interna de un sistema termodinámico, es decir, el movimiento de las partículas; mientras más movimiento exista, mayor temperatura. Para medir la temperatura existen varios instrumentos, entre los cuales se encuentran los termómetros de vidrio, bimetálicos, termopares, pirómetros ópticos, entre otros.

1. **Sensor LM35.** El sensor LM35 es un circuito integrado de precisión para medición de temperatura, éste proporciona una salida analógica de voltaje que representa la temperatura en grados centígrados, de manera lineal; por lo que la lectura se debe pasar por un convertidor analógico a digital para interpretar la lectura. Una ventaja del LM35 es que no requiere de una calibración externa para sus mediciones y tiene la capacidad de cubrir un rango de temperatura desde -55 hasta 150°C, siendo éste un sensor de bajo costo. Como voltaje de alimentación se puede colocar una señal de -0.2 a 35V. Para su instalación se recomienda pegar el sensor en una superficie que no sobrepase los 0.01°C del medio a medir, ya que la temperatura del sensor resultará errónea al estar midiendo la temperatura de la superficie. También se recomienda cubrir los cables con epoxy para asegurar que la temperatura de los cables se encuentre a la misma temperatura del medio a medir. (TexasInstruments, 2013)

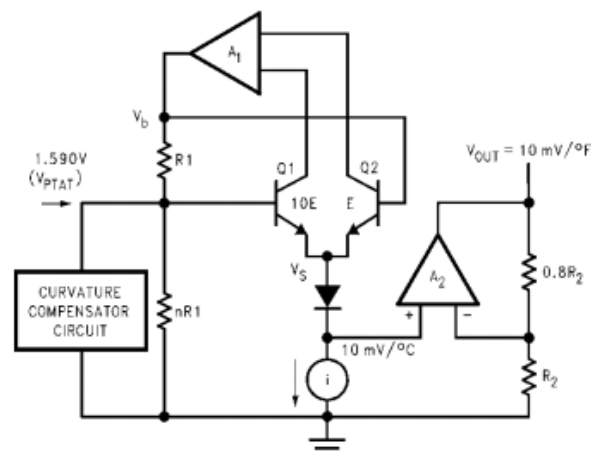
Figura 10 Conexión del LM35 para mediciones de 2 a 150°C



(TexasInstruments, 2013)

El LM35 se encuentra compuesto por dos transistores y dos amplificadores, como se puede observar en la siguiente figura. Se utiliza un transistor, donde el voltaje base-emisor depende de la temperatura del medio, pero se tiene el problema que la relación no es lineal; para ello se implementa un circuito integrado para arreglarlo. En el transistor Q1 pasa un décimo de la densidad de corriente que pasa por Q2, por lo que dicha diferencia genera un voltaje proporcional a la temperatura del medio. El amplificador A1 se encarga de forzar un voltaje en la base de Q1 y el amplificador A2 se encarga de sacar el voltaje que se relaciona de manera lineal con la temperatura. (NationalSemiconductor, 1995)

Figura 11 Circuito interno LM35



(NationalSemiconductor, 1995)

2. **Pirómetro.** Un pirómetro es un instrumento de medición de temperatura de alta precisión, sin tener un contacto directo con la superficie a medir. Estos poseen un mecanismo óptico, el cual permite la medición de temperatura en objetos en movimiento o en aplicaciones donde no se pueda tener contacto directo. Los pirómetros de láser tienen la capacidad de medir a largas distancias,

dependiendo de sus especificaciones. Los pirómetros se basan en el principio de la irradiación de energía de un objeto, donde la cantidad de energía es proporcional a la cuarta potencia de la temperatura. (PCE, s.f.)

K. MEDICIÓN Y SENSORES DE HUMEDAD

La humedad es la cantidad de agua por unidad de volumen, a una temperatura y presión específicas. La humedad se ve afectada directamente por la temperatura, por lo que, mientras mayor temperatura mayor humedad que puede tener el aire. La humedad se puede describir como la relación de la presión parcial del vapor de agua entre la presión de saturación de la misma. Para una correcta medición se recomienda seleccionar correctamente un punto de medición, teniendo precaución de no colocar los sensores en lugares con entradas de aire, ya que la medición puede resultar inestable e incorrecta. Se debe considerar si los rangos de humedad a medir son altos o bajos, determinar si las condiciones en las que se van a realizar las mediciones no son de altas temperaturas o alta presión, ya que se debe seleccionar el sensor adecuado para cada uno de los casos. (Vaisala, 2012)

1. **Sensor DHT22.** El sensor DHT22, también conocido como AM2302, posee una tecnología que permite medir la humedad y temperatura del medio. La medición de humedad la realiza por medio de un capacitor y la temperatura la mide por medio de un termistor; posee también un convertidor analógico a digital para el envío de la información. Éste proporciona una salida digital, por medio del protocolo one-wire. La salida digital posee en los 16 bits más significativos la información de humedad, en los siguientes 16 bits la información de temperatura y en los últimos 8 bits envía una verificación de los datos. El DHT22 trabaja en los rangos de temperatura de -4 a 80°C y 0 a 100% de humedad. Este dispositivo posee una memoria programable, la cual guarda el coeficiente de calibración, por lo que permite la correcta calibración del mismo. (Liu)

L. MEDICIÓN Y SENSORES DE FLUJO DE AIRE

Para la medición de flujo de aire, se puede realizar por medio de la medición de la velocidad del aire, conociendo el área transversal de la tubería. Para la medición de la velocidad del aire se pueden utilizar anemómetros, los cuales se basan en un álabe rotativo, donde se mide la velocidad angular de la misma, siendo ésta proporcional a la velocidad del fluido. Por otro lado, también se pueden utilizar directamente sensores de caudal. (Cavallé Oller, 2004)

1. **Transductor de velocidad y caudal del aire EE75.** El transductor de velocidad y caudal de aire EE75 es un sensor que posee una tecnología que permite medir bajas y altas velocidades de flujo aire. Éste puede ser utilizado en temperaturas desde -40 a 120°C, por lo que posee una compensación

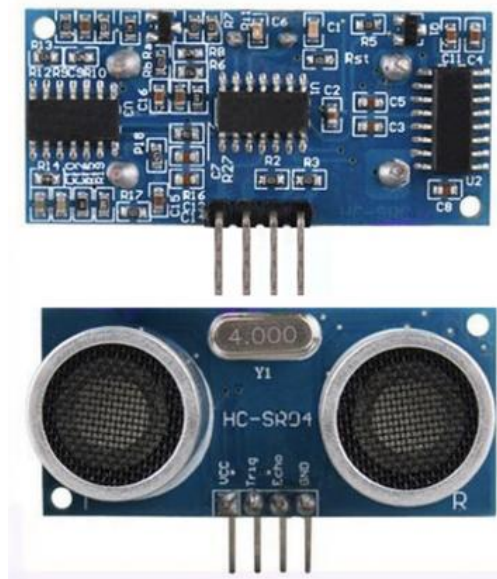
de temperatura, lo cual lo ayuda a eliminar la dependencia de las mediciones a la temperatura. Éste puede medir desde 0 a 40m/s, con tiempos de respuesta de 1.5 a 40 segundos configurables. El transductor tiene una señal de salida de voltaje de 0 a 10V ó 0 a 5V, según el usuario lo configure, también se tiene una señal de salida de corriente de 4 a 20mA. (Instruments P. , s.f.)

M. SENSOR DE NIVEL HC-SR04

El eco se produce cuando una onda sonora, viajando a través de un medio, llega a una superficie que interrumpe su trayectoria dando como resultado un rebote de parte de dicha onda. En la medición de nivel el proceso consiste en el envío de un pulso ultrasónico y medir el tiempo que a este le toma regresar al sensor debido al rebote (eco). De esta forma podemos conocer la distancia entre el sensor y la superficie que deseamos medir. (Espinoza)

El sensor ultrasónico HC-SR04 utiliza este método de medición ofreciendo un rango de 2cm a 400cm. Su operación no se ve afectada por la luz del sol, o materiales oscuros como en otros sensores de la industrial. El módulo está diseñado para ser fácil de usar e instalar, incluyendo el emisor y receptor en el módulo. (Cytron Technologies, 2013)

Figura 12. Módulo ultrasónico HC-SR04.



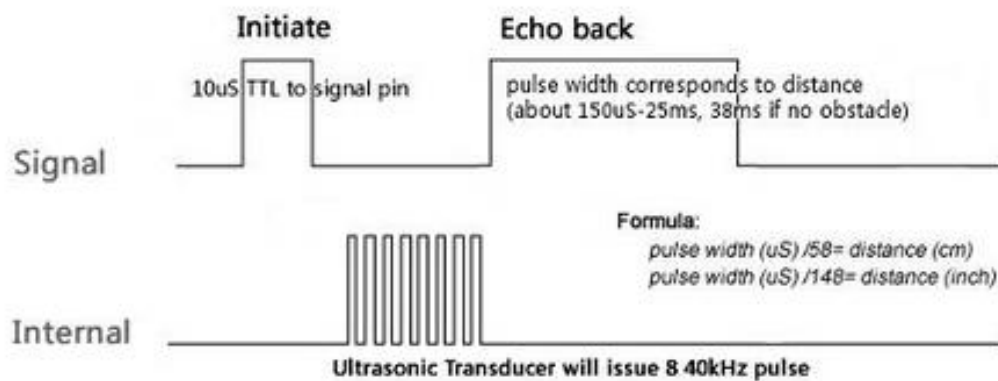
(Cytron Technologies, 2013)

Figura 13. Parámetros eléctricos de módulo HC-SR04.

Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

(ElecFreaks, 2012)

Figura 14. Señales de funcionamiento del módulo HC-SR04.



(Cytron Technologies, 2013)

N. DIAGRAMAS P&ID

Un diagrama P&ID tiene como fin la identificación de equipo industrial, identificación de procesos, condiciones de diseño, de una manera sencilla en proyectos de tuberías e instrumentación. El nombre viene por las siglas en inglés “piping and instrumentation diagram”. Los diagramas P&ID se realizan por medio de símbolos que representan cada uno de los componentes, controles, equipos, líneas de tubería, entre otros; dichos símbolos basados en la Norma ISA S5.1 que habla sobre instrumentación y automatización. Para la identificación de un instrumento, se coloca un círculo con letras y números dentro de él. Los números que se encuentran en la parte inferior representan el número de lazo al que pertenece. Las letras en la parte

superior representan la función del instrumento, la primera indica la variable que se va a controlar, las siguientes dos o tres, representan la función de salida del instrumento. (JTP)

Figura 15 Letras de identificación en un sistema P&ID

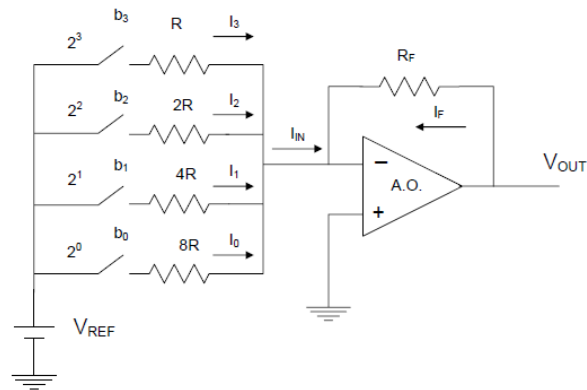
	FIRST-LETTER (4)		SUCCEEDING-LETTERS (3)		
	MEASURED OR INITIATING VARIABLE	MODIFIER	READOUT OR PASSIVE FUNCTION	OUTPUT FUNCTION	MODIFIER
A	Analysis (5, 19)		Alarm		
B	Burner, Combustion		User's Choice (1)	User's Choice (1)	User's Choice (1)
C	User's Choice (1)			Control (13)	
D	User's Choice (1)	Differential (4)			
E	Voltage		Sensor (Primary Element)		
F	Flow Rate	Ratio (Fraction) (4)			
G	User's Choice (1)		Glass, Viewing Device (9)		
H	Hand				High (7, 16, 16)
I	Current (Electrical)		Indicate (10)		
J	Power	Scan (7)			
K	Time, Time Schedule	Time Rate of Change (4, 21)		Control Station (22)	
L	Level		Light (11)		Low (7, 15, 16)
M	User's Choice (1)	Momentary (4)			Middle, Intermediate (7, 15)
N	User's Choice (1)		User's Choice (1)	User's Choice (1)	User's Choice (1)
O	User's Choice (1)		Orifice, Restriction		
P	Pressure, Vacuum		Point (Test) Connection		
Q	Quantity	Integrate, Totalize (4)			
R	Radiation		Record (17)		
S	Speed, Frequency	Safety (8)		Switch (13)	
T	Temperature			Transmit (18)	
U	Multivariable (6)		Multifunction (12)	Multifunction (12)	Multifunction (12)
V	Vibration, Mechanical Analysis (19)			Valve, Damper, Louver (13)	
W	Weight, Force		Well		
X	Unclassified (2)	X Axis	Unclassified (2)	Unclassified (2)	Unclassified (2)
Y	Event, State or Presence (20)	Y Axis		Relay, Compute, Convert (13, 14, 18)	
Z	Position, Dimension	Z Axis		Driver, Actuator, Unclassified Final Control Element	

(JTP)

O. CONVERTIDOR DIGITAL A ANALÓGICO (DAC)

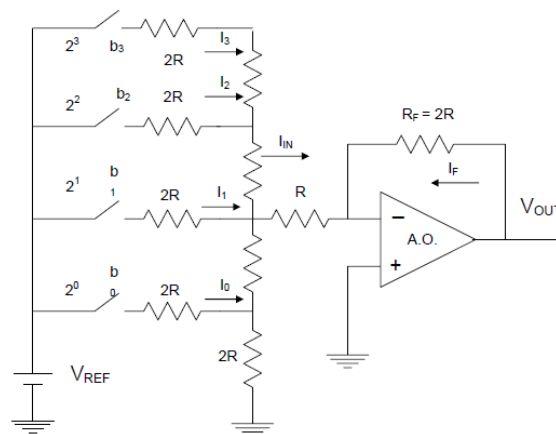
Para la realización de un convertidor digital a analógico se puede realizar por medio de varias configuraciones, entre ellas se encuentra el convertidor con ponderación binaria, en el cual se conectan varias resistencias en paralelo, donde cada una de ellas representa un bit. La principal desventaja es que se requieren varias resistencias que cumplan con el valor de $2^n R$, con tolerancias bajas para que la precisión no se vea afectada. Otra configuración es el convertidor R/2R, el cual no requiere gran número de resistencias de diferentes valores. (Floyd, 2000)

Figura 16 DAC con ponderación binaria



(Floyd, 2000)

Figura 17 DAC en escalera R/2R

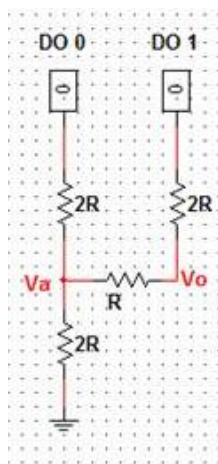


(Floyd, 2000)

Cuando se tienen salidas digitales (0,1) se puede realizar un DAC en escalera R/2R de manera que en la salida del mismo se tengan variaciones de voltaje, dependiendo de las entradas. Para el siguiente circuito, el voltaje de salida va representado por la ecuación: (NationalInstruments, 2011)

$$V_o = \frac{1}{4}D_0 + \frac{1}{2}D_1 \quad (7)$$

Figura 18 DAC en escalera R/2R de 2 bits



(National Instruments, 2011)

P. VARIADORES DE FRECUENCIA

Un variador de frecuencia, también conocido como variador de velocidad, es un dispositivo electrónico que controla la velocidad síncrona del motor trifásico, par mecánico o torque. Estos pueden ser utilizados para aplicaciones que requieran de cambios de velocidad en motores o incluso se pueden utilizar para realizar un arranque suave del motor. Al utilizar un variador de frecuencia en un motor trifásico se obtienen varias ventajas tales como la reducción de corriente y par de arranque, se puede tener un frenado dinámico, es posible monitorearlo mientras el motor se encuentra funcionando, se tiene una protección por sobrecarga, es posible la variación de la velocidad síncrona, entre otros. (Ventura Nava, 2008)

Figura 19 Variador de frecuencia Allen-Bradley



(Ventura Nava, 2008)

1. Variador de frecuencia ABB ACS 300. El variador de frecuencia de la familia ACS 300 es un driver de PWM, utilizado en motores trifásicos de 0.5 a 15HP de potencia. Éste provee protección contra la sobrecarga del motor, esto se realiza por medio de la medición de la corriente del motor y cuando ésta aumenta, se desconecta el motor para evitar daños. Para controlar el variador de frecuencia, se puede realizar por medio del panel de control, señales externas o comunicación serial. Para indicar la velocidad de referencia, se debe enviar una señal analógica a la entrada AI. Es posible modificar las señales de entrada, ya sea una señal de voltaje de 2 a 10V o una señal de corriente de 4 a 20mA, por medio del jumper X5. (Systems, 1998)

Figura 20 Tarjeta de control ACS 300

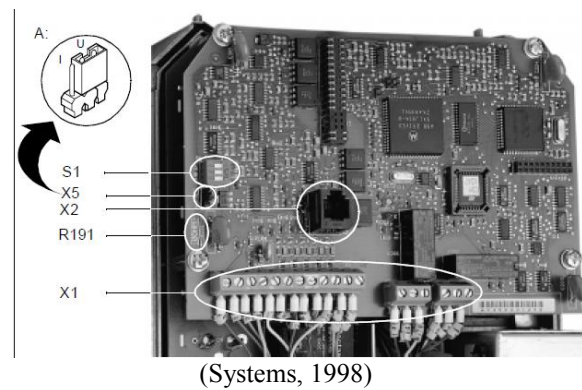
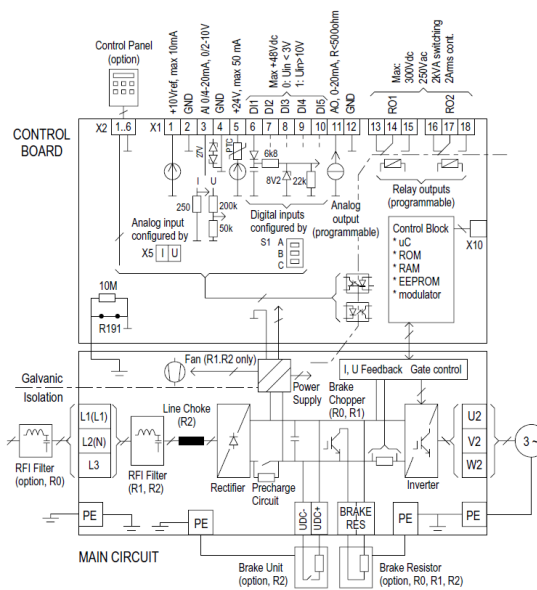
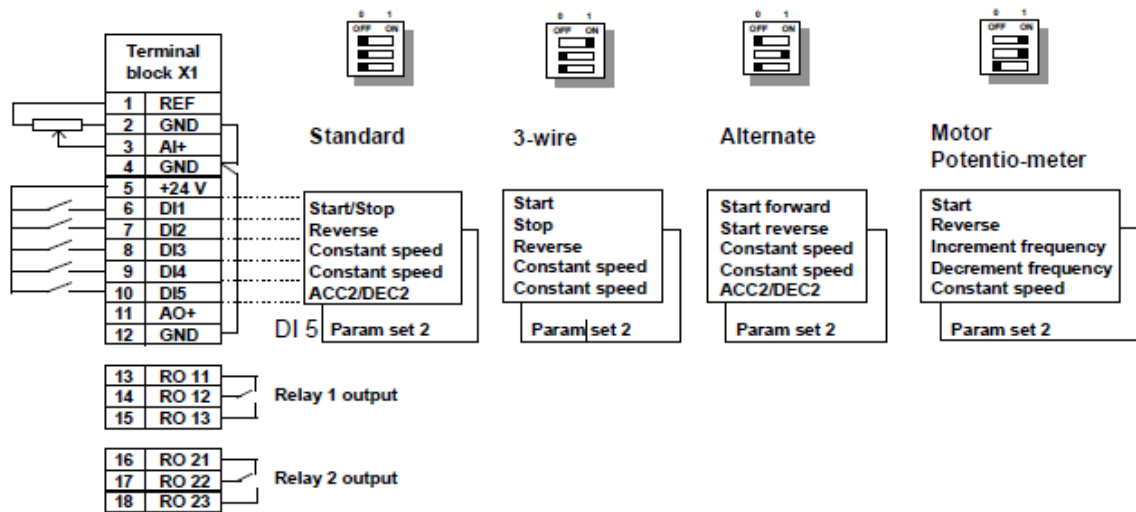


Figura 21 Diagrama de bloques de un ACS 300



Las entradas digitales del variador de frecuencia ACS 300, pueden ser por medio de cierre de relés para la aplicación de 24V en la entrada o aplicar entre 24 a 48V desde una fuente externa, donde es interpretado como un cero lógico cuando no hay voltaje y un uno lógico cuando se ha cerrado un contacto o se aplica entre 24 a 48 V. Las posibles configuraciones de entradas digitales se muestran en la siguiente figura. (Systems, 1998)

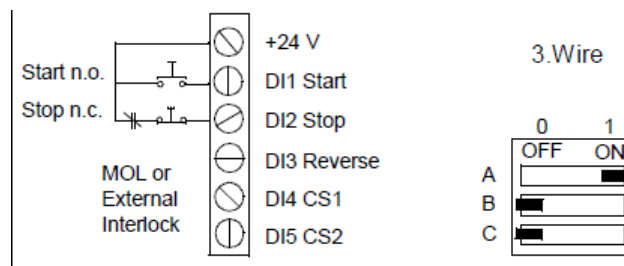
Figura 22 Diagrama de entradas digitales ACS 300



(Systems, 1998)

Para una aplicación industrial, el fabricante recomienda una conexión 3-wire, es decir, tres cables para una señal de inicio y paro, para lidiar con cualquier emergencia. Se utiliza un push-button normalmente abierto para el botón de inicio y uno normalmente cerrado para el botón de paro. A continuación se muestra el diagrama recomendado por el fabricante. (Systems, 1998)

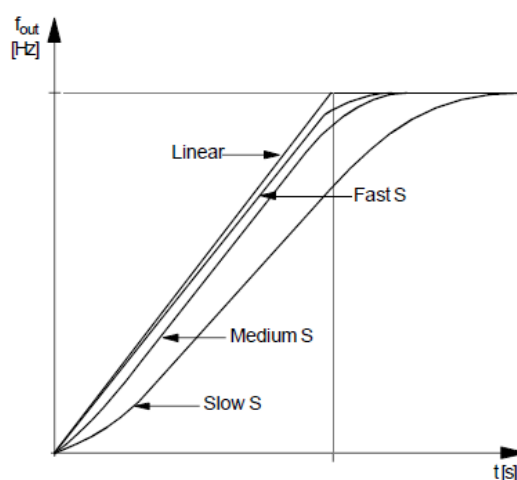
Figura 23 Diagrama recomendado para conexión 3-wire del ACS 300



(Systems, 1998)

Antes de iniciar el funcionamiento, se deben configurar los parámetros del mismo, se debe fijar una frecuencia mínima y máxima de operación, tiempos de respuesta, señales de referencia, comportamiento de aceleración y desaceleración, entre otras. El comportamiento de aceleración y desaceleración del ACS 300 puede ser lineal, lento, medio y rápido, este se debe seleccionar dependiendo del tiempo de respuesta deseado. Se selecciona lineal cuando se requiere una aceleración constante, lento para aplicaciones con una rampa sobre 15 segundos, medio 1.5 segundos y rápido menos de un segundo. (Systems, 1998)

Figura 24 Rampas de aceleración y desaceleración para ACS 300



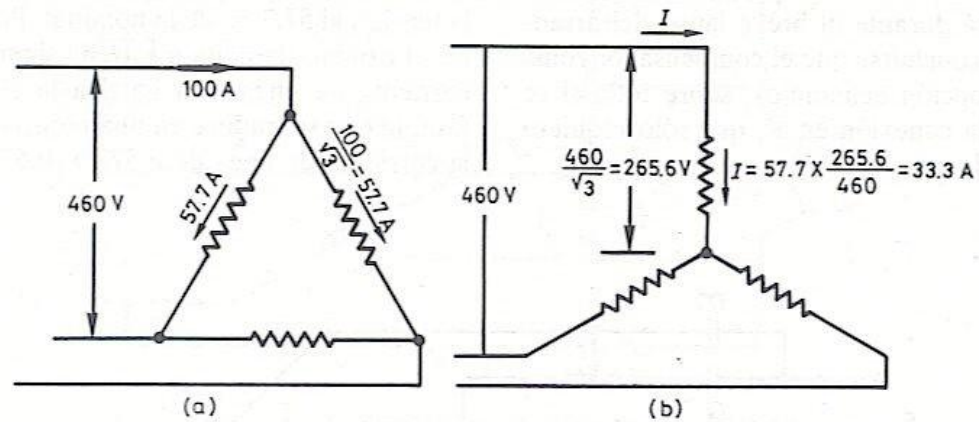
(Systems, 1998)

Q. ARRANQUE DE MOTORES TRIFÁSICOS DE TIPO ESTRELLA-DELTA

Para poder utilizar el arranque tipo estrella-delta en un motor trifásico, éste debe ser embobinado. Éste consiste en modificar las conexiones del motor para un arranque en estrella y luego funcione en conexión delta. Principalmente se utiliza este arranque para la impulsión de cargas centrífugas, tales como ventiladores, bombas centrífugas, entre otros. Otra aplicación es cuando el par mecánico o torque que se requiere es reducido. (Ventura Nava, 2008)

Cuando se realiza un arranque en conexión estrella la corriente de la línea es $1/3$ de la corriente que fluye en conexión delta y la corriente del devanado es 1.73 veces mayor que en la conexión delta. El voltaje en cada fase del devanado es el voltaje nominal dividido entre $\sqrt{3}$, es decir que se reduce 42.3%. Se debe tener en cuenta que el cambio de conexiones puede causar una corriente alta, por lo que se recomienda colocar una reactancia en serie para limitar la corriente de transición. Los cambios mencionados se pueden observar en la siguiente figura. (Ventura Nava, 2008)

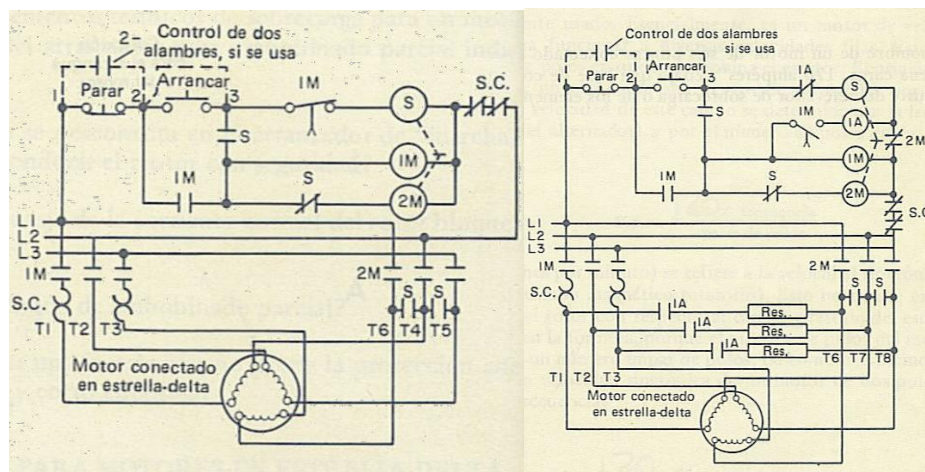
Figura 25 Configuración estrella (a) y delta (b)



(Ventura Nava, 2008)

Para la realización del cambio estrella-delta, se utiliza un temporizador, quien se encarga de controlar el tiempo. Cuando el operador presiona el botón de arranque se energizan los contactos para la conexión estrella y se inicia el control del tiempo. Cuando ha pasado el tiempo suficiente, el temporizador desconecta el primer contacto y se conecta a otros formando la conexión delta. Notar que durante el cambio mencionado existe un momento del tiempo en el que el motor no se encuentra conectado a nada, lo cual se le conoce como “transición abierta”. Para arreglar la transición abierta, se colocan resistencias entre los contactos, de esta manera se logra tener continuidad hacia el motor todo el tiempo. (Ventura Nava, 2008)

Figura 26 Arranque por transición abierta (izquierda) y transición cerrada (derecha)



(Ventura Nava, 2008)

R. ACEROS Y SUS CARACTERÍSTICAS

El acero es una aleación de hierro, carbono, manganeso y otros elementos. El carbono es el que tiene mayor importancia en este material debido a su resistencia, ductilidad y dureza. Los demás elementos brindan otras características en el acero, tales como: tenacidad, resistencia a la corrosión, resistencia a altas temperaturas, templabilidad, entre otras. Los elementos que pueden alearse son: azufre, fósforo, silicio, níquel, cromo, molibdeno y vanadio. (Mott, 2006)

Existen diferentes maneras de clasificar a los aceros, pero en este caso se tomará en cuenta la aplicación que tienen:

1. **Aceros al carbono.** La mayoría de aceros son aceros al carbono, aproximadamente un 90% de los metales. Con este tipo de aceros se puede fabricar diversas maquinarias, chasis de carros, estructuras para la construcción. Estos pueden someterse a tratamientos térmicos como el recocido, temple y revenido. (CAP, S.A., 2000)

2. **Aceros aleados.** Los elementos que se utilizan para este tipo de aceros son el vanadio, molibdeno, manganeso (1.65%), silicio (0.60%) y cobre (0.60%). Estos se pueden clasificar en tres diferentes categorías: (CAP, S.A., 2000)

a. **Aceros estructurales.** Este tipo de aceros tienen bajo porcentaje de carbón y están laminados en caliente. Con frecuencia se encuentran en láminas, placas, barras, vigas y otros perfiles estructurales. Generalmente son utilizados en ejes, engranajes y palancas; también en estructura de edificios, puentes, automóviles, barcos y otras aplicaciones semejantes. El contenido de carbono varía entre 0.25 a 6%. (Mott, 2006)

b. **Aceros para herramientas.** Este tipo de aceros tiene que contar con varias características, especialmente: alta resistencia al desgaste, alta tenacidad, alta dureza y baja deformación al estar sometidos a tratamientos térmicos. Asimismo, estos aceros tienen que tener resistencia a la alta velocidad, trabajo en frío, trabajo en caliente y resistencia al choque. Son empleados básicamente para cortar y construir herramientas como taladros, fresas, escariadores, machos de roscar. (CAP, S.A., 2000)

c. **Aceros especiales.** En esta clasificación del acero se caracteriza por su resistencia a las altas temperaturas y a la corrosión; además, el contenido de cromo supera el 12%. Existen dos diferentes tipos, tales como: (CAP, S.A., 2000)

1) Aceros de baja aleación ultraterresistente. Este tipo de acero se caracteriza por ser económico, debido a que contiene una menor cantidad de los elementos costosos, pero a pesar de ello reciben tratamientos especiales para poder tener una mayor resistencia. Una de las aplicaciones de este tipo de aceros es en estructura de edificios. (CAP, S.A., 2000)

2) Aceros inoxidable. La principal característica para este tipo de aceros es la alta resistencia a la corrosión cuando se utilizan en ambientes con mucha humedad; además, debe de contener 10% de cromo como mínimo, la mayoría está entre el rango de 12-18%. Algunos aceros inoxidable tienen una alta dureza, otros son más resistentes, especialmente al estar sometidos a tratamientos térmicos elevados. Algunas de las aplicaciones para este tipo son: la arquitectura, tuberías, tanques, equipos quirúrgicos, unión de huesos humanos, entre otras. (Mott, 2006)

S. RODAMIENTOS Y TIPOS DE RODAMIENTOS

Los rodamientos son elementos mecánicos que permiten el giro rotativo entre dos piezas y para soportar cargas radiales, axiales o combinadas. El objetivo principal de estos es disminuir la fricción y garantizar estabilidad al momento estar sometidos los elementos a cargas. (Gutierrez, 2008)

3. Rodamientos según la carga aplicada:

a. Rodamientos de cargas radiales. Este tipo de rodamientos son capaces de soportar cargas que están dirigidas en dirección perpendicular al eje de rotación. En este tipo de rodamientos se sub-clasifican en:

1) Rodamiento rígido de bolas. Son los rodamientos más utilizados, especialmente por ser económicos y por sus múltiples aplicaciones. Son capaces de operar a altas velocidades y son fáciles de diseñar; además, no requieren de mucho mantenimiento. (Gutierrez, 2008)

Figura 27: Rodamiento de bolas



(DirectIndustry, s.f.)

2) Rodamiento de rodillos. Este tipo de rodamiento se caracteriza por soportar grandes magnitudes de cargas radiales, además de soportar altas velocidades. Estos tienen guiados los rodillos en unas pequeñas pestañas en uno de los aros para permitir que gire libremente en el soporte. (Gutierrez, 2008)

Figura 28: Rodamiento de rodillo



(NKE Bearings, s.f.)

3) Rodamiento de bolas a rótula. Este tipo de rodamiento es parecido al de bolas solamente que esta cuenta con doble hilera. Las ventajas de este tipo de rodamiento son la auto-alineación y que tienen mayor adaptación a las cargas que puede estar sometido el eje. (Gutierrez, 2008)

Figura 29: Rodamiento de bolas a rótula



(NKE Bearings, s.f.)

4) Rodamiento de rodillos a rótula. Este tipo de rodamiento cuenta con doble hilera de rodillos. Tienen la capacidad de soportar grandes cantidades de carga, especialmente radiales y axiales; también tienen la capacidad de auto-alineación. (Gutierrez, 2008)

Figura 30: Rodamiento de rodillos a rótula

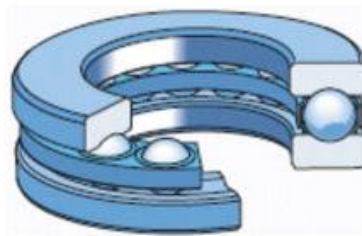


(NKE Bearings, s.f.)

b. Rodamientos de cargas axiales. Este tipo de rodamiento puede soportar cargas que actúan en dirección al eje de rotación. Dentro de este tipo de rodamientos están:

1) Rodamientos de simple efecto. Estos rodamientos cuentan con dos aros, entre los cuales se colocan las bolas. Permite que uno de los aros gire con el eje y el otro se quede estático como soporte. (Gutierrez, 2008)

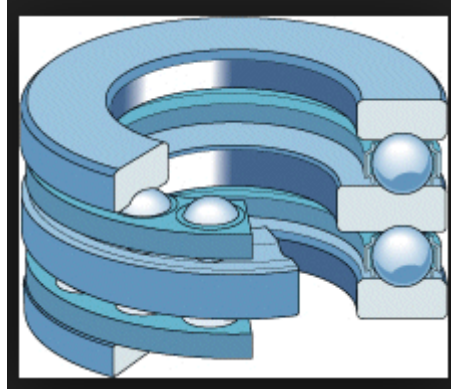
Figura 31: Rodamiento de simple efecto



(DirectIndustry, s.f.)

2) Rodamientos de doble efecto. Estos rodamientos tienen el mismo funcionamiento que los de simple efecto, con la diferencia que el aro central es el que gira con respecto al eje y los otros dos absorben las cargas y actúan como soportes. (Gutierrez, 2008)

Figura 32: Rodamiento de doble efecto



(SKF, s.f.)

c. Rodamientos de cargas mixtas. Los rodamientos de cargas mixtas son capaces de soportar tanto cargas radiales como axiales. Muchos de estos rodamientos soportan cargas en ambas direcciones del eje de rotación y también al estar sometidos a cargas angulares. (Gutierrez, 2008)

T. ENGRANAJES Y TIPOS DE ENGRANAJES

El engrane es una rueda con dientes capaz de transmitir movimiento giratorio de un elemento a otro. Estos se pueden clasificar en cuatro diferentes grupos:

4. Engranés rectos. Estos tienen sus dientes paralelos al eje de rotación, es decir, son capaces de producir movimiento rotacional de un eje a otro de manera paralela. Es el más sencillo y común de utilizar, especialmente por la forma de sus dientes. (Budynas, 2008)

Figura 33: Engrane recto



(DirectIndustry, s.f.)

5. **Engranés helicoidales.** Estos engranes tienen los dientes inclinados con relación al eje de rotación. Las ventajas de estos engranes es que no producen mucho ruido, debido a su acoplamiento en los dientes de manera angular y permiten transmitir potencia entre ejes que no se encuentran en paralelo. La desventaja es que pueden producir cargas de empuje y pares de flexión. (Budynas, 2008)

Figura 34: Engranés helicoidales



(DirectIndustry, s.f.)

6. **Engranés cónicos.** Estos engranajes se utilizan principalmente para transmitir movimiento entre dos ejes que se intersecan. Las caras de soporte de diente de los engranes son cónicas. Existen engranes cónicos de dientes rectos, engranes cónicos en espiral y engranes hipoides. (Budynas, 2008)

Figura 35: Engranés cónicos

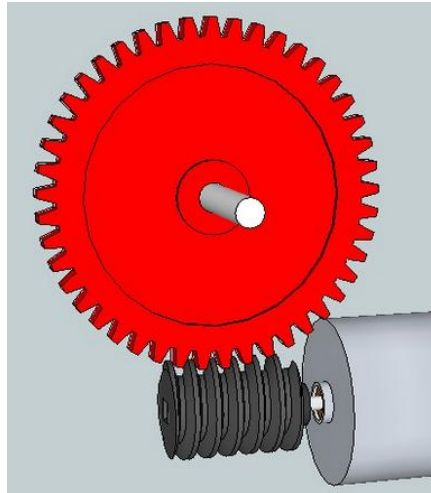


(DirectIndustry, s.f.)

7. **Tornillo sinfín.** El tornillo sinfín permite la transmisión de movimiento entre ejes que se encuentran perpendicularmente. Cada vez que el tornillo sinfín realiza una revolución el

engranaje acoplado avanza un número de dientes igual al número de acoples del tornillo. Estos se utilizan cuando la relación de velocidades entre dos ejes es alta. (Budynas, 2008)

Figura 36: Tornillo sin fin



(Platea, s.f.)

U. COMUNICACIÓN SERIAL

La comunicación serial se da cuando se envían bits uno detrás del otro, a dicha comunicación se le llama asíncrona. Para que la comunicación serial entre dos dispositivos funcione, se deben configurar ciertos parámetros; tales como el bit de inicio, para indicar al receptor que se está iniciando la comunicación. El bit de paro para indicar que se ha terminado de transmitir una palabra y el bit de paridad para identificación de errores. Para una correcta comunicación se debe fijar una velocidad específica, tanto en el receptor como en el emisor; esta velocidad es la que indica cuántos bits por segundo son enviados y recibidos, es por ello que deben estar configurados a la misma velocidad. (Morales, 2004)

V. PROTOCOLO ONE-WIRE

El protocolo one-wire permite una comunicación serial por medio de un solo cable, ésta va definida por un maestro y un esclavo. Se la llama maestro al dispositivo que indica cuándo realizar una acción o lectura de datos y el esclavo es el que responde a dichas indicaciones. Para la salida one-wire del maestro se debe colocar una resistencia pull-up a 5V para su correcto funcionamiento. El protocolo indica los valores lógicos de voltaje, entre 0 y 0.8V es un cero lógico y entre 2.2V en adelante un uno lógico. La comunicación entre esclavo y maestro es en ambas direcciones, pero una a la vez. (Wong, 2002)

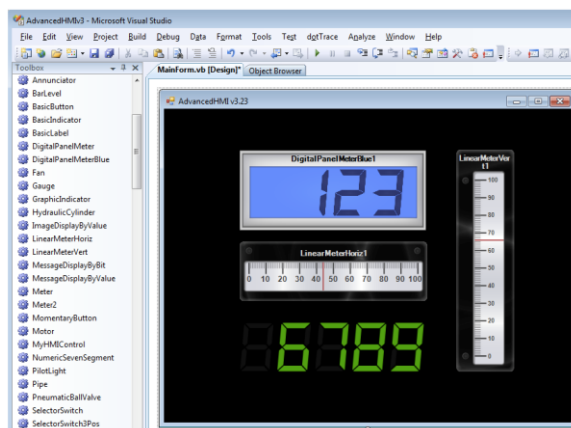
W. HMI

La Interfaz Hombre Maquina (HMI por sus siglas en inglés) permite presentar al operario una vista completa del estatus de los procesos realizados en la planta de producción. Estas pueden implementarse con dispositivos especialmente diseñados para esta función, como lo son los paneles de operación, o pueden realizar su función desde una computadora. Es necesario que todos los dispositivos del sistema de control y monitoreo posean la configuración necesaria para entablar comunicación con el HMI. (Universidad Nacional de Quilmes, 2011)

En cualquier proceso industrial es inevitable encontrar al menos una interfaz hombre-máquina. La capacidad de mostrar información de forma gráfica ha convertido las HMI como la mejor opción para la supervisión y control de un proceso industrial. Para procesos sofisticados se prefiere el uso de dispositivos como los PLCs o DCS pero para tareas relativamente sencillas el uso de una computadora se ha vuelto cada vez más común. Para poder elegir un sistema de HMI indicado para la aplicación que se vaya a realizar se deben tomar en cuenta características como: capacidad de almacenamiento de datos, calidad y cantidad de despliegue de información, comunicación con red interna o internet, software de facil configuración y facilidad de soporte técnico. (Electroindustria, 2013)

Unos de los lenguajes de programación que se utilizan comúnmente para el desarrollo de HMIs en computadoras es Visual C o Visual Basic. Esto se debe a la facilidad que tiene permitir comunicación con dispositivos de campo, actualización de datos de forma dinámica, permitir que el operador envíe señales al proceso utilizando botones, controles ON/OFF y ajustes continuos. (Universidad Nacional de Quilmes, 2011)

Figura 37. Paquete de medición para Visual Studio.



(Advanced HMI, 2014)

X. SERVIDORES VIRTUALES

Un servidor web se ejecuta en un ordenador para que este ser mantenga a la espera de que un cliente realice alguna solicitud. Las respuestas a estas solicitudes son interpretadas por el navegador web que recibe código, normalmente html, para luego mostrar el resultado al usuario. Instalar un servidor real para cada dominio o sitio web que se posee no es una opción rentable por lo que los servidores virtuales han obtenido popularidad para usos personales y empresariales. Utilizando un servidor virtual podemos re direccionar cada domino a sub carpetas ubicadas en el servidor. Los servidores que ofrecen el servicio de alquiler de servidores virtuales han obtenido el nombre del “host” o “servidor web” y a la acción de alquilar se le conoce como “hosting”. (Andreu, 2011)

Existen servidores virtuales que ofrecen sus servicios de forma gratuita tanto para uso personal como para uso empresarial. Entre los servicios que ofrecen estos servidores se encuentra: espacio en disco duro, normalmente 1.5GB, editor de código en línea, plantillas para facilitar el diseño de páginas web y sistemas sencillos para carga y descarga de archivos. (Oleaga, 2013)

Y. PROGRAMACIÓN DE PÁGINAS WEB

1. HTML. HTML es el lenguaje que entienden los navegadores web. Este lenguaje fue desarrollado por el científico llamado Tim Berners-Lee quien en 1990 buscaba facilitar el acceso a documentos para científicos universitarios. Estas fueron las bases para la web como la conocemos hoy en día. Al escribir código html agregamos etiquetas para identificar cada sección de la estructura que llevará la página web. Estas etiquetas le indican al navegador web dónde mostrar el texto o gráfico. Para crear archivos html no se necesita ningún editor sofisticado ni instalar software extra. Html es escrito en texto puro por lo que cualquier tipo de editor de texto es herramienta suficiente para la tarea. (Gary Shelly, 2008)

Figura 38. Ejemplo de código HTML.

```
<html>
<head>
  The Head
  <title>
  </title>
</head>
<body>
  The Body
</body>
</html>
```

(goodellgroup, 2004)

2. Lenguaje PHP. El lenguaje de programación php se caracteriza por su robustez, versatilidad y modularidad. Los programas desarrollados en php son interpretados por el servidor web para ser transferidos al navegador y mostrados al usuario. El código php es embebidos directamente en el código html. Este lenguaje sigue la corriente "Open Source" por lo que el intérprete puede tener acceso fácil al código fuente de forma gratuita. Php resulta un lenguaje de programación sencillo de comprender por su flexibilidad. Los programadores familiarizados con lenguajes como C, Perl o Java encuentran similitudes de sintaxis por lo que la adaptación se vuelve un proceso fácil y rápido. Así como es gratuito es un lenguaje multiplataforma, trabaja sobre la mayoría de servidores web y tiene la disponibilidad de comunicarse con más de 20 tipos de bases de datos. (Cobo, 2005)

Figura 39. Ejemplo de código PHP.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo ";Hola, soy un script de PHP!";
    ?>

  </body>
</html>

```

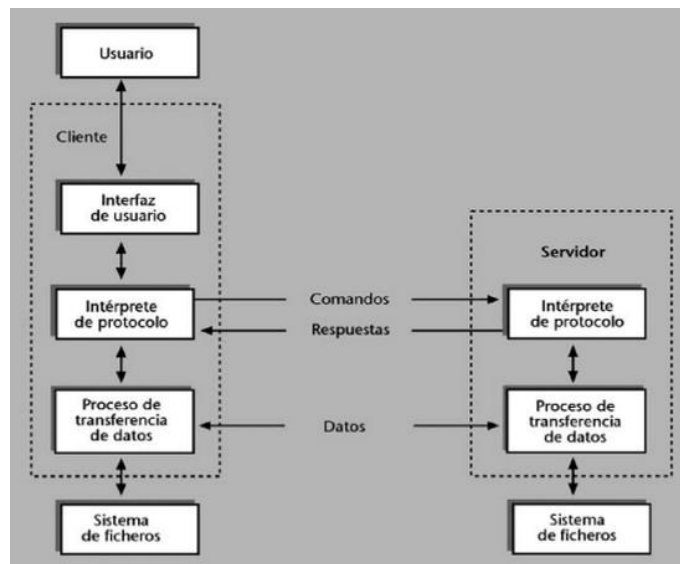
(The PHP Group, 2014)

Z. PROTOCOLO FTP

El protocolo FTP es el más utilizado para la transferencia de archivos entre diferentes sistemas. Se basa en un modelo cliente/servidor con el cual permite que tanto cliente como servidor obtenga archivos el uno del otro. El protocolo proporciona operaciones para que el cliente pueda modificar archivos ubicados en el servidor, dichas operaciones son: borrar archivos, cambiarles nombre, crear y borrar directorios, entre otras. Uno de los beneficios más importantes de este protocolo es que posee interoperabilidad entre sistemas distintos. No se debe especificar detalles de la estructura interna de los sistemas o la organización de los contenidos de los archivos. Para el modelo general del protocolo, tanto en el cliente como en el servidor, intervienen dos entidades. El intérprete es una de estas entidades, este se encarga de la transferencia de comandos. En la parte del usuario el intérprete convierte la solicitud en una serie de comandos FTP para

luego enviarlos al servidor. La segunda entidad es el proceso de transferencia de datos. Está bajo el control del intérprete de protocolo y se encarga de intercambiar los datos que sean requeridos. (Ordinas, 2008)

Figura 40. Modelo funcional del protocolo FTP.

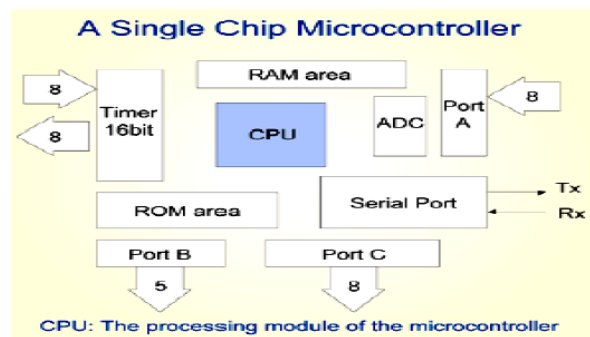


(Ordinas, 2008)

AA. MICRO-CONTROLADORES

Un micro-controlador es un dispositivo de integración de alta escala, el cual tiene la capacidad de controlar varios procesos. Éste es un sistema cerrado, el cual incorpora una micro-computadora en un chip. Un micro-controlador se encuentra conformado por un reloj interno, memoria programable, memoria de datos, un procesador (CPU) y circuitos de entradas y salidas. (Canto Q.)

Figura 41 Arquitectura interna del micro-controlador

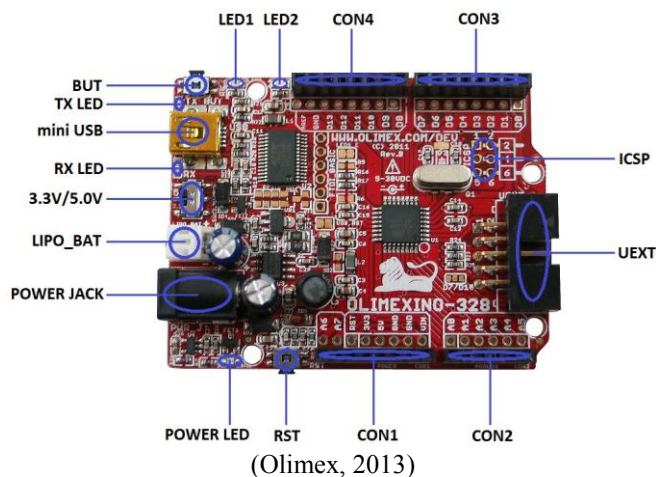


(Canto Q.)

1. **Arduino.** Arduino es una plataforma abierta al público que provee facilidades en el proceso de diseño de proyectos electrónicos. Posee diferentes entradas y salidas analógicas y digitales, en los cuales se pueden conectar diferentes sensores, luces, motores, controladores, entre otros. Posee una plataforma para programar el micro-controlador, lo cual lo hace de una forma fácil. Arduino proporciona diferentes librerías, las cuales se pueden utilizar en cualquier aplicación que el usuario desee. (Arduino, 2014)

2. **Olimexino-328.** El Olimexino-328 es una tarjeta que puede ser utilizada como un Arduino UNO, con una tarjeta ATmega328 para uso industrial. Éste puede ser alimentado con fuentes externas de 9 a 30V, vía USB o baterías de Li-ion de 3.7V. Entre las mejoras del Olimexino 328 y un Arduino es que, por los niveles de alimentación externa, se puede conectar a una fuente industrial de 24V. Los componentes pueden trabajar en temperaturas entre -25 a 85°C y todos vienen protegidos contra ruido, lo cual lo hace perfecto para trabajar en ambientes industriales. (Olimex, 2013)

Figura 42 Componentes principales del Olimexino-328



3. **Raspberry pi.** Ya sea que se desee ver archivos multimedia, ingresar a la web, aprender y experimentar la computadora del tamaño de una tarjeta de crédito llamada Raspberry Pi es la plataforma indicada. Su versatilidad forma parte de sus grandes ventajas. Los usos más comunes que se le puede dar son: computadora de propósito general, plataforma para proyectos electrónicos y plataforma educativa. El objetivo inicial fue desarrollar una plataforma educativa para alentar a niños a explorar con las computadoras, además incluye diferentes intérpretes y compiladores previamente cargados. A comparación de otros microcontroladores la Raspberry Pi le permite al usuario tener mayor flexibilidad y amplios horizontes para la creación de proyectos de programación y/o electrónicos. Por ejemplo, para

realizar un control simple de temperatura es mucho más práctico utilizar un arduino. Si se desea tener acceso al control desde la web y poder descargar un historial de los datos medidos es recomendable utilizar una Raspberry Pi. (Matt Richardson, 2012)

Figura 43. Raspberry Pi modelo B+.



(Matt Richardson, 2012)

Como toda computadora esta posee un sistema operativo que inicia al arrancar la computadora y permite a los programas tener acceso al hardware. El sistema operativo que incluye de fábrica es una versión de Linux llamado Raspbian. La idea de unir la Raspberry Pi con Linux es una gran combinación debido a que Linux es un sistema operativo gratuito. (Matt Richardson, 2012)

BB. GENERADORES ELÉCTRICOS

La principal manera de obtener energía eléctrica en la actualidad es con centrales termo eléctricas o hidroeléctricas las cuales utilizando turbinas y alternadores transforman la energía mecánica en energía eléctrica. Por medio de inducción electromagnética se produce un voltaje que puede ser generado debido a la rotación de bobinas de alambre a través de un campo magnético o por la rotación del campo en el centro de las bobinas de alambre. El voltaje inducido por el generador en cualquier instante de tiempo depende de la densidad del flujo de campo por el cual se mueve un conductor, velocidad del conductor en movimiento y el ángulo con el cual el conductor corta las líneas de flujo. Un generador de corriente alterna es también llamado alternador y debe su movimiento giratorio al uso de turbinas de vapor, hidroturbinas o motores diesel. Los alternadores pequeños son comúnmente utilizados junto a un motor diésel y se utilizan en plantas de suministro de emergencia. (Harper, 2000)

Los sistemas de generación diésel de velocidad constante son, en su mayoría, sistemas poco eficientes y de elevado costo de mantención, en especial cuando este es utilizado para pargas que están por debajo del

valor nominal del sistema. Esto produce un fenómeno llamado “wetstacking” el cual es una acumulación de residuos en las cámaras del motor, producto de una combustión incompleta del combustible. Para aumentar la eficiencia de dichos sistemas los fabricantes sugieren una carga mínima de 30% al 40% de la nominal. Al momento de dimensionar un sistema de generación diésel (de la misma forma que cualquier otro sistema de generación) el valor de demanda máxima juega un gran papel, pero a lo largo del tiempo en el que trabaja dicho sistema de generación muy pocas veces se alcanza este valor. Es por esto que para lograr una operación óptima en estos sistemas es utilizado un sistema de generación diésel operados a velocidad variable. (J. B. Andriulli, 1999)

Figura 44. Planta auxiliar de generación eléctrica de finca La Fama.



CC. CONTROL DE VELOCIDAD DE PLANTA GENERADORA DIÉSEL

Un sistema automático de generación eléctrica en una planta auxiliar diésel debe controlar tanto el nivel de voltaje como la frecuencia de salida. En algunos alternadores el voltaje depende del voltaje de alimentación del alternador y la frecuencia es proporcional a las revoluciones por minuto del motor diésel. En otros alternadores la frecuencia es controlada de forma automática y el voltaje varía en función de la velocidad de giro del eje del motor. Para controlar la velocidad del eje del motor es necesario, en la mayoría de los casos, implementar un sistema electromecánico con el cual, de forma electrónica se controlan las variables medidas y de forma mecánica se acciona el acelerador del motor para inducir una variación en las mediciones. El uso de sistemas digitales para el control tiene las ventajas de ser pequeño y de bajo peso, alimentados, por lo regular, con una fuente de 5 voltios y presentan gran flexibilidad al momento de realizar modificaciones (no se requiere cambiar el diseño del hardware, fácilmente se modifica el funcionamiento por medio de software). (Mendoza, 2003)

DD. FUMIGACIÓN POR ASPERSIÓN

Método mediante el cual se esparce pequeña gotas de plaguicida bien sea concentrado emulsionable, polvo mojable, concentrado soluble o micro encapsulado. Permite el control del producto aplicado, salvaguardando así las superficies de contacto. Este método es el más efectivo y usado por las mayoría de empresas de fumigación, nosotros usamos combinaciones entre polvo mojable y emulsionable para una mayor efectividad. El polvo mojable hace que queden partículas adheridas a las superficies generando residualidad y el emulsionable hace un mayor control de eliminación de la plaga. (Fumigaciones Bay-Were C.A., 2014)

EE. ROYA

Esta es una enfermedad del café, que se presentó en Guatemala desde 1980, afecta a las hojas maduras principalmente. Dependiendo la severidad, esta puede provocar una caída intensa de hojas y pérdida también en la producción. En los años 2010 y 2011 se observaron repuntes de roya en diferentes regiones del país, no existe un patrón definido en el comportamiento de este hongo, aun para la misma finca de un año para otro.

Además de la utilización preventiva de fungicidas, se debe incluir el monitoreo de la enfermedad y buenas prácticas de cultivo, enfocándose en una fertilización balanceada. Las fincas con una alta producción, son más vulnerables a la enfermedad. (Anacafe, s.f.)

El hongo de la roya y la violencia amenazan con empobrecer aún más a las pequeñas comunidades rurales de Guatemala, en donde la mayoría de sus habitantes depende casi en su totalidad del café, uno de los más importantes productos agrícolas de exportación de este país.

FF. UBUNTU

Ubuntu es un sistema operativo que trabaja a código abierto, fue desarrollado entorno al kernel Linux. Ubuntu se basa en los siguientes principios: el software debe ser gratuito, la gente puede usarlo en su lengua materna y puede hacerlo independientemente sus limitaciones. Además los usuarios deben ser libre de personalizar o modificar el software del modo que el desee. (Mediawiki, 2012)

GG. SCRIPT

Un script es un archivo de órdenes o pequeño programa con el que poder automatizar ciertas tareas o acciones en un ordenador. Suele ser un archivo de texto plano, en el que se insertan las órdenes que se quieren realizar. En Ubuntu (o cualquier distribución Linux) se puede realizar un script con comandos de

consola para lograr, por ejemplo, actualizar el sistema, instalar alguna aplicación, hacer un backup de archivos, o todo a la vez. (Henares, 2013)

HH. CRON

El nombre cron viene del griego chronos que significa “tiempo”. En el sistema operativo Unix, este es un administrador de procesos en segundo plano, el cual ejecuta procesos o guiones en periodo de tiempo regular. Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab.

El cron se inicia de `/etc/rc.d/` o `/etc/init.d` dependiendo de la distribución del sistema operativo que se tenga. Cron se ejecuta en segundo plano, se revisa cada minuto la tabla de tareas del crontab, para buscar tareas que se deban cumplir. El usuario puede agregar comandos o scripts con tareas a cron para automatizar algunos procesos. (Lucain, 2010)

II. CRONTAB

Este es un archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado. Crontab verifica la fecha y hora en la cual se debe ejecutar el script o comando y lo realiza en segundo plano. Crontab es la manera más sencilla de administrar tareas de cron en sistemas multiusuario, ya sea como simple usuario de sistema o usuario root. (Lucain, 2010)

JJ. DJANGO

Este es un framework web, el cual es código abierto, escrito en Python y permite construir aplicaciones web de manera más rápida y con menos código. Fue inicialmente desarrollado para la gestión de aplicaciones web de páginas que eran orientadas a noticias de World Online. Después de un tiempo se liberó bajo licencia BSD y se centró en automatizar todo lo posible y adquirió el principio DRY (Don't Repeat Yourself).

El objetivo de Django es facilitar la creación de sitios web que son complejos. Este framework pone énfasis en el re-uso, conectividad y extensibilidad de componentes, el desarrollo rápido y el principio no lo repitas. Python es la base de este framework, ya que es utilizado en todas partes, incluso en configuraciones, archivos y modelos de datos. (Garcia, 2013)

Framework Web. En general, con el término framework, se refiere a una estructura de software que se compone de piezas personalizadas e intercambiables, que sirven para el desarrollo de una aplicación. En

otras palabras, un framework se puede considerar como una aplicación genérica que se encuentra incompleta, pero se puede configurar para añadirle las piezas necesarias para construir una aplicación correcta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, se puede definir como un conjunto de componentes, que conforman un diseño reutilizable, el cual facilita y agiliza el desarrollo de sistemas Web.

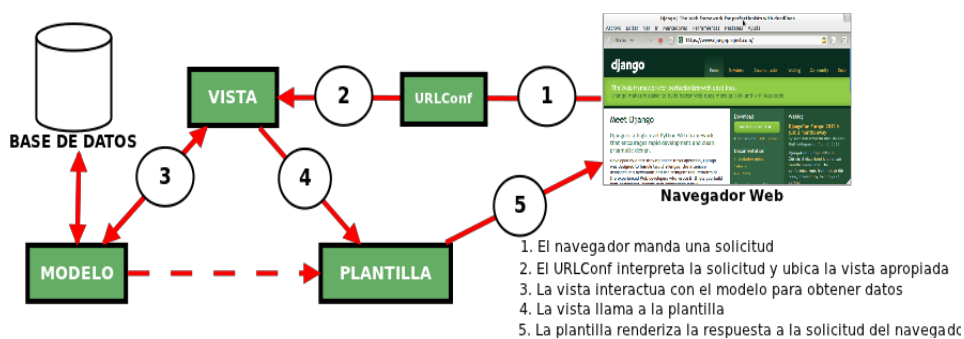
1. Modelo. El modelo es el que define los datos almacenados en la base de datos, este se encuentra en forma de clases de Python. Cada tipo de dato almacenado, se encuentra en una variable con ciertos parámetros, estas variables también poseen métodos. Todo esto permite indicar y controlar el comportamiento de los datos. (Infante, 2012)

2. Vista. La vista se presenta en forma de funciones en Python, el propósito de estas funciones determinar qué datos serán visualizados, además permite retornar páginas HTML para que se desplieguen, redireccionar a otras páginas, entre otras funciones. El ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista.

La vista es la encargada de realizar tareas como enviar correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. La vista no tiene relación con el estilo de presentación de los datos, solo se encarga de los datos. (Infante, 2012)

3. Plantilla. La plantilla es una página HTML, donde se pueden incluir etiquetas de Django para obtener la información obtenida por las vistas, en sí no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, etc). (Infante, 2012)

Figura 45: Estructura de Django



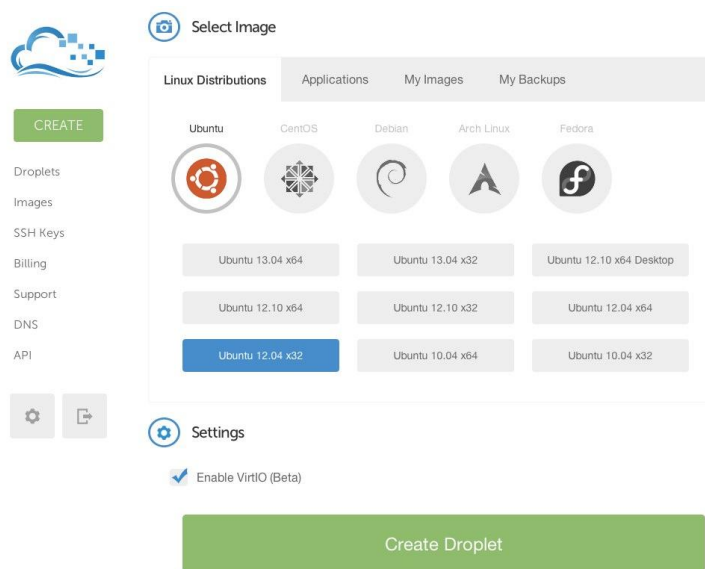
4. GeoDjango. Es un módulo contrib incluido para Django, este permite crear aplicaciones Web geográficas. Esto se realiza por medio de extensiones ORM para realizar queries que manipulen datos espaciales, también permite realizar operaciones geométricas. (Django, 2014)

KK. DIGITALOCEAN

Es un hosting VPS, este es barato y fácil de utilizar. El servicio provee un precio bajo, a partir de 5 US\$/mes, y las facturas por hora. DigitalOcean utiliza discos duros SSD y una red rápida que ofrecen velocidad de trabajo a los servidores.

Por otra parte, el servicio promete estar en actividad un 99.99% en la red, energía y disponibilidad del servidor virtual. Se pueden crear Droplets, en donde se escoge el sistema operativo para trabajar, este cuenta con Ubuntu, CentOS, Debian, ArchLinux y Fedora. (Fridman, 2013)

Figura 46: Sistemas operativos de DigitalOcean



LL. NGNIX

Es un servidor HTTP libre, de código abierto, de alto rendimiento y que puede ser utilizado para actuar tanto como proxy HTTP reverso, como proxy IMAP/POP3. Nginx es conocido por ser ligero, debido a que consume pocos recursos del sistema, es estable y además su configuración es sencilla, comparándolo con Apache2. Es servidor web porque es el encargado de devolver el contenido estático de la aplicación. (Viera, Nginx, 2013)

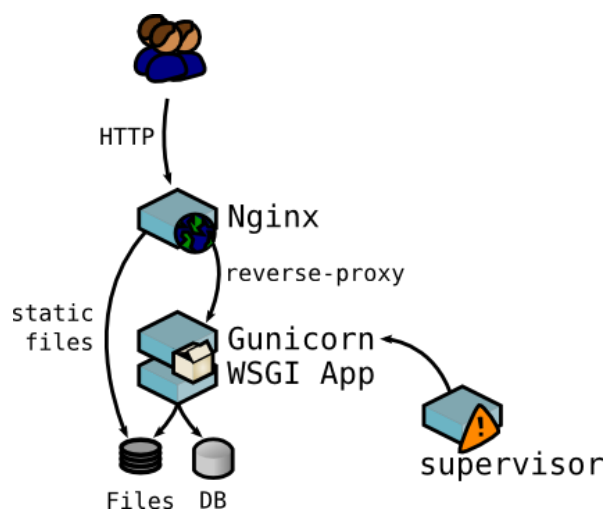
MM. GUNICORN

Es el componente que nos ofrece el servidor de aplicaciones como tal. Lo que hace es servir la aplicación, utilizando el estándar WSGI. En realidad se puede decir que el servidor de aplicaciones se construye con la suma de Supervisor y Gunicorn. Gunicorn es, en realidad, el servidor de aplicaciones, pero se utiliza, además, Supervisor para controlar el número de instancias del servidor de aplicaciones que tendremos y el arranque/rearranque y parada de dichos servidores. (Viera, Gunicorn, 2013)

NN. SUPERVISOR

Es una aplicación hecha en Python que permite gestionar procesos. En el fichero de configuración indicamos cuántos servidores y qué capacidad (workers) tendrá cada uno, así como dónde se escribirán los logs. (Estrella, 2013)

Figura 47: Comunicación Supervisor-Gunicorn-Nginx



OO. GOOGLE MAPS API

Es un servicio de Google que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

Google Maps son los responsables de su facilidad de uso por cualquier usuario: el sistema de deslizamiento de imagen, acoplado a la carga dinámica de nuevas imágenes; la adaptación del mapa al tamaño de ventana del navegador; la interfaz minimalista; la posibilidad de cambiar de tipo de mapa en un clic.

Como todas las demás aplicaciones Google, Maps descansa poderosamente sobre la utilización de JavaScript. La carga y el deslizamiento de imagen no podrían efectuarse sin este código.

PP. GOOGLE CHART API

Su uso es muy sencillo, mediante peticiones http a una determinada url (<http://chart.apis.google.com>) y con unos determinados parámetros se nos devolverá una imagen. Es necesario que el formato de los datos que se van mostrar en la gráfica cumplan un formato determinado.

Una tabla concreta (que puede cambiar en función del tipo de gráfico que se muestra): para los más sencillos, como las gráficas de barras y tarta, bastará con una tabla con dos columnas, clave/valor, pero para otro tipo de gráficos es necesario estructuras más complejas, que podemos consultar en la documentación que Google brinda en su web oficial.

QQ. BASE DE DATOS

Una base de datos se define como una serie de datos que están organizados y se encuentran relacionados entre sí. Estos datos son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Características: entre las características principales de los sistemas de bases de datos se pueden mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.

- Acceso a través de lenguajes de programación estándar.

RR. POSTGRESQL

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Figura 48: Características de PostgreSQL

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

1. PostGIS. Es un módulo para el motor de bases de datos objeto-relacional PostgreSQL, que añade soporte para tipos geográficos y lo habilita para ser usado como contenedor de información geoespacial, permitiéndole realizar operaciones de análisis geográfico. PostGIS ha sido desarrollado por Refrations Research Inc., empresa de consultoría y desarrollo especializada en aplicaciones de bases de datos y sistemas SIG responsable de proyectos como uDig. Está disponible bajo licencia GNU General Public License, sigue las especificaciones OpenGIS y cumple la norma Simple Feature Specification for SQL del Open Geospatial Consortium.

PostGIS dispone de una interfaz de usuario con herramientas para la gestión de datos, soporta funciones de básicas de topología, transformación de coordenadas, validación de datos, programación de APIs, etc. En sus próximos desarrollos tiene previsto proporcionar funcionalidades topológicas más avanzadas, soportar el almacenamiento de información raster e incorporar herramientas para la realización de cálculos de rutas, gestión de redes, superficies 3D y features complejas como curvas y splines.

SS. DREAMWEAVER CC

Con este programa se puede desarrollar diferentes contenidos Web, de una manera más efectiva y rápida. Se pueden realizar un enlace con los archivos CSS y además realizar un diseño web adaptable, esto con el fin de que la página se adapte a los diferentes tamaños de resolución. Permite ver la página web en un explorador determinado, para observar si el diseño está como se requiere.

TT. CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple el cual describe cómo se va a desplegar un documento en la pantalla. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

UU. JAVASCRIPT

Es un lenguaje utilizado para crear pequeños programas, que pueden ser insertados en una página web y en programas orientados a objetos más complejos. Con Javascript se pueden crear diferentes efectos e interactuar con los usuarios. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

El código javascript podemos encontrarlo dentro de las etiquetas `<body></body>` de nuestras páginas web. Por lo general se insertan entre: `<script></script>`. Su sintaxis es similar a la usada en Java y C, al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún Framework.

VV. JQUERY

Es una biblioteca gratuita de Javascript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos. Otra de las grandes ventajas de jQuery es que se enfoca en simplificar los scripts y en acceder/modificar el contenido de una página web. Finalmente, jQuery agrega una cantidad impresionante de efectos nuevos a Javascript, los cuales podrán ser utilizados en los sitios Web.

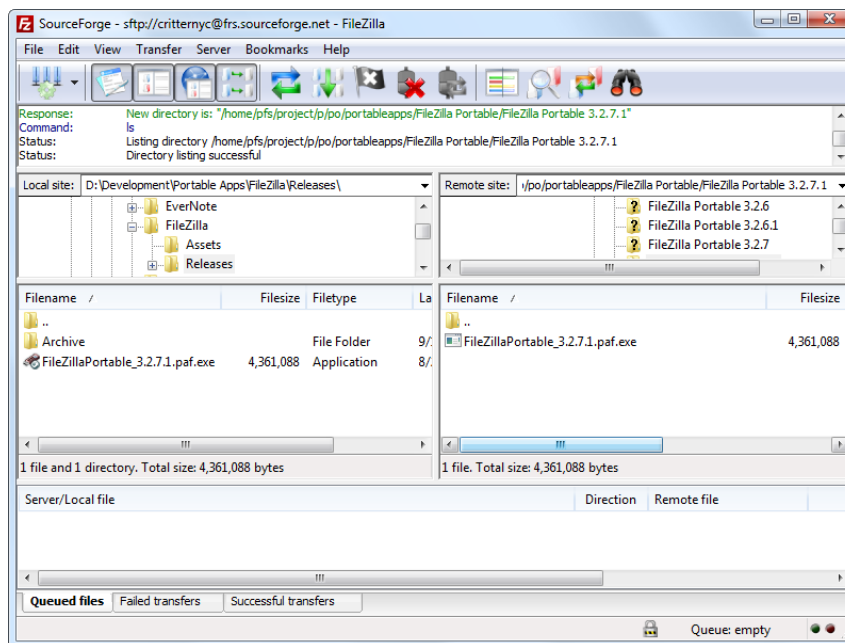
WW. SECURE FILE TRANSFER PROTOCOL (SFTP)

SSH (Secure SHell) File Transfer Protocol, también conocido como SFTP o Secure File Transfer Protocol) es un protocolo de nivel de aplicación, el cual proporciona la funcionalidad para poder transferir y manipular archivos, sobre un flujo de datos fiable. Se utiliza SSH para proporcionar seguridad a los datos con los cuales se están trabajando, aunque permite ser usado con otros protocolos de seguridad. La seguridad no proviene directamente del protocolo SFTP, sino SSH en este caso.

Para enviar archivos a un servidor que soporte este protocolo, todos los servidores o instancias VPS Linux lo incluyen por defecto, deberemos conocer: IP remota, puerto (por defecto es el 22), usuario y contraseña.

A. Filezilla. Es un programa gratuito para el sistema operativo Windows, este da la capacidad de distribuir archivos por medio de FTP (File Transfer Protocol) o SFTP. Se utiliza el puerto 22, ya que es SFTP el tipo de protocolo.

Figura 49: Entorno FileZilla



XX. PUTTY

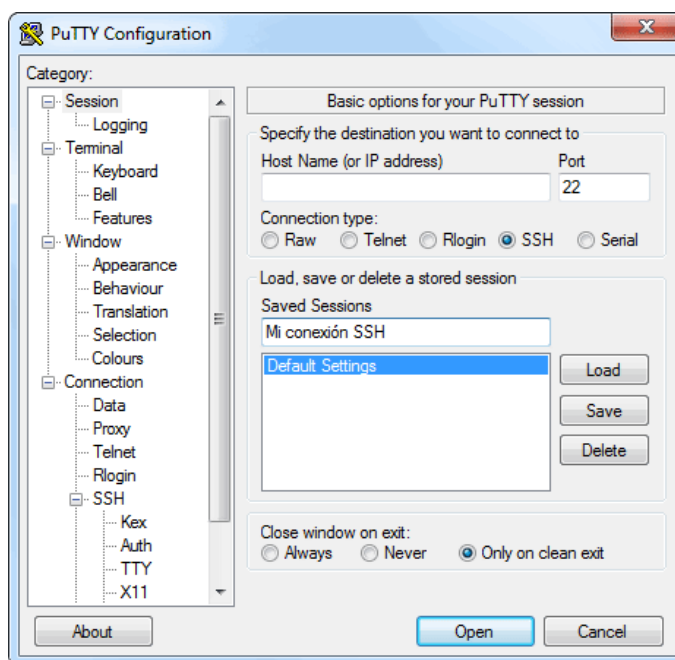
PuTTY es un cliente SSH y Telnet con el que podemos conectarnos a servidores remotos iniciando una sesión en ellos que nos permite ejecutar comandos. El ejemplo más claro es cuando empleamos PuTTY

para ejecutar comandos en un servidor VPS y así poder instalar algún programa o configurar alguna parte del servidor.

Tras ejecutar PuTTY se nos muestra el apartado Session con las opciones básicas y tan sólo hay que:

- Introducir la IP o Hostname del servidor remoto.
- Seleccionar el puerto (normalmente para conectar a través de SSH es el 22 por defecto).
- Seleccionar en connection type la opción SSH (ya suele venir marcada por defecto).
- Hacemos click en el botón Open.
- Puede que se nos muestre alguna advertencia, la cual aceptamos y tras unos segundos nos solicitará nuestro nombre de usuario y contraseña para iniciar sesión en el servidor remoto.

Figura 50: Entorno PuTTY



YY. SENSOR DE PRESIÓN

Los sensores o transductores de presión son dispositivos que transforman la magnitud física de la presión a una magnitud proporcional en voltaje. Esto se logra de diferentes maneras por ejemplo con la separación de un par de membranas, la separación se puede medir con la técnica de capacitancia.

Sensor 100psi cuenta con una medición de 0 a 100psi de acero alimentado con 5v, su rango de salida es de 0.5v – 4.5v lineal siendo 0.5v 0psi 2.5v 50psi y 4.5v 100psi. Se puede sobrecargar hasta 200psi sin que el sensor se sufra daños irreparables. El rango de temperatura va desde -40 hasta 125 °C. se puede observar en la “Figura 51: Sensor de presión 100psi”. (TopCeller, 2014)

Figura 51: Sensor de presión 100psi



ZZ. SENSOR DE CAUDAL

Los sensores o transductores de caudal son dispositivos que transforman la magnitud física de la cantidad de fluido por unidad de tiempo que pasa en cierto diámetro. Unos de los sensores de caudal más comunes son los de frecuencia, estos sensores convierten el caudal a una frecuencia. El voltaje de la frecuencia varía dependiendo de la fuente de alimentación.

Sensor de Caudal HQ-A68-1 de plástico resistente a golpes, se puede alimentar con un voltaje máximo de 100v DC y el sensor va a responder con el mismo voltaje en frecuencia, dependiendo del caudal. El caudal máximo que soporta es de 30 l/min y un mínimo de 0.2 l/min. Puede operar en un rango de temperatura entre -40 a 150°C y a una presión máxima de 1.6Mpa. (Uxcell, 2014)

Figura 52: Sensor de caudal HQ-A68-1



AAA. BLUETOOTH

La tecnología Bluetooth empezó en 1994, en la universidad de Suecia Lund, aunque la versión 1.0 fue liberada en 1999. La compañía Ericsson empezó a estudiar distintas formas de comunicación entre teléfonos celulares con otros dispositivos, por lo que después de muchas pruebas llegaron a la conclusión de que la forma más efectiva de hacerlo era por medio de radiofrecuencia. En 1998 se creó Bluetooth SIG (por sus siglas Special Interest Group) es un grupo de compañías trabajando juntas para definir y promover las especificaciones Bluetooth las compañías que la conformaron fueron: Ericsson, Intel, IBM, Toshiba y Nokia.

Bluetooth es una especificación para un dispositivo que describe como se puede interconectar con teléfonos celulares, Asistentes Personales Digitales, ordenadores, microcontroladores, etc. Utilizando una conexión inalámbrica de corto alcance que no necesita de visión directa entre los dispositivos a conectar. Esta conexión opera en la banda de 2.4 GHz. Tiene la habilidad de atravesar obstáculos como paredes y madera, por lo que resulta de gran utilidad en lugares de trabajo. (Gonzalez, 2008)

BBB. SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)

El sistema de GPS (Global Positioning System) inicio en 1973 y tiene como objetivo la determinación de las coordenadas espaciales de puntos respecto de un sistema de referencia mundial. Se puede observar cualquier punto del planeta a cualquier momento siempre y cuando estén satélites de transmisión cerca.

Para la obtención de coordenadas el sistema se basa en determinar las distancias de cuatro satélites simultáneamente como mínimo. Las distancias se obtienen a partir de las señales emitidas por los satélites, y resabidas por instrumentos, especialmente diseñados para transformar las señales a datos útiles, entre estos están: latitud, longitud, hora, etc. (Huerta, Mangiaterra, & Noguera, 2005)

CCC. SECURE DIGITAL (SD)

Secure Digital es una asociación global de empresas que se encarga de simplificar el uso y extender la vida de las memorias. Muchas empresas utilizan estos estándares para crear sus dispositivos como Panasonic, SanDisk Corporation y Toshiba Corporation. La corporación se fundó en el año 2000. (SD Association, 2014)

DDD. PYTHON

Es un lenguaje de programación, que tiene como objetivo ser fácil de utilizar y de leer. Fue creado por Guido van Rossum a principios de los años 90. Es un lenguaje de script, con variables dinámicas (sin declarar el tipo de variable al inicio), multi-plataforma y orientado a objetos. Entre sus cualidades posee una gran gama de librerías que ya trae por defecto o se le pueden añadir como por ejemplo: bluetooth, sys, etc. (González Duque, 2008)

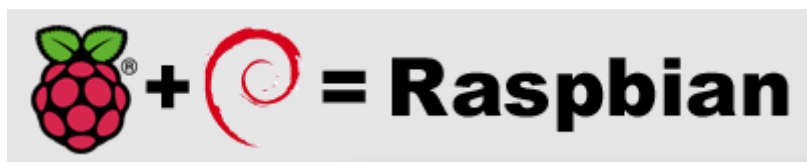
EEE. LENGUAJE C

Es un lenguaje de programación que no está ligado a ninguna computadora en específico, esto le da mucha diversidad. Puesto que es muy diverso puesto se pueden programar computadoras y microcontroladores. Es un lenguaje que se tienen que inicializar las variables a utilizar antes de utilizarlas. (Kernighan & Ritchie, 1991)

FFF. RASPBIAN

Es un sistema operativo libre basado en Debian y optimizado para utilizarlo en la Raspberry Pi. Raspbian viene con más de 35,000 paquetes, pre-compilado para una fácil instalación. Raspbian sé completó en junio del 2012. En la “Figura 53: Raspbian” se puede observar el logo. (Raspbian, 2012)

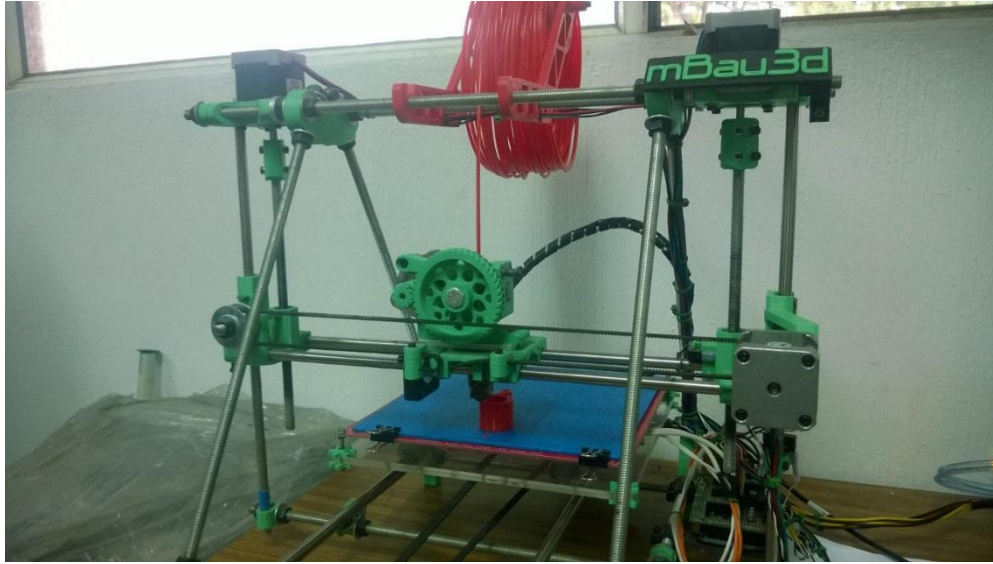
Figura 53: Raspbian



GGG. IMPRESORA 3D

La tecnología 3D se basa en crear capas de plástico, metales, etc. estas pequeñas capas de material van dando forma al objeto. Esto no es realmente una tecnología nueva ya que existe de manera comercial desde 1986 pero hasta ahora está creando un gran impacto en la sociedad, sobre todo las impresoras de plástico a bajo costo, como se puede observar en la “Figura 54: Impresora 3D a base de plástico.” (Andrei, 2014)

Figura 54: Impresora 3D a base de plástico.



HHH. CAD

Se trata de una herramienta imprescindible en cualquier disciplina técnica y de diseño. CAD o Dibujo Asistido por Computadora permite la creación de modelos en 2D y 3D, también presenta herramientas para realizar modificaciones fácilmente. Los modelos 3D pueden ser exportados directamente en planos 2D o hacia una impresora 3D.

V. IMPLEMENTACIÓN DE SENSORES PARA EL CONTROL DE TEMPERATURA EN SECADORAS ESTÁTICAS Y FLUJO DE AIRE EN SECADORAS TIPO GUARDIOLA Y MONITOREO DE HUMEDAD Y TEMPERATURA EN LAS CÁMARAS DE SECADO DE CAFÉ EN LA FINCA “LA FAMA”

A. METODOLOGÍA

Se visitó la finca “La Fama” para observar los procesos utilizados para el tratamiento del café, para así poder determinar posibles soluciones para tener un mejor control de los mismos. Se observó que la mayoría de procesos son realizados a mano, por lo que se puede implementar tecnología para aumentar la eficiencia de las máquinas utilizadas. Se identificó la necesidad de implementar sistemas automáticos en las secadoras de la finca. En las secadoras estáticas un sistema que regule la temperatura de secado del grano y el monitoreo continuo de temperatura y humedad. En las secadoras tipo guardiola un control de flujo de aire en las tuberías y al igual que en las estáticas, control continuo de temperatura.

Figura 55 Observación de procesos utilizados en finca "La Fama"



Se realizó una revisión bibliográfica para determinar la tecnología que se puede utilizar para dichos procesos y hacerlos más eficientes. Se investigó el funcionamiento de varios equipos para control de temperatura y humedad, se investigaron formas de medición de flujo de aire y diversos sensores.

Se analizó un sistema de inyección de gas para el horno de las secadoras estáticas, para la reducción de uso de leña. También se analizó el sistema de tuberías de la finca para determinar los cálculos necesarios para conocer el flujo de aire de entrada en cada secadora tipo guardiola. Se analizaron los tipos de sensores para medición de flujo de aire y así determinar el sensor adecuado.

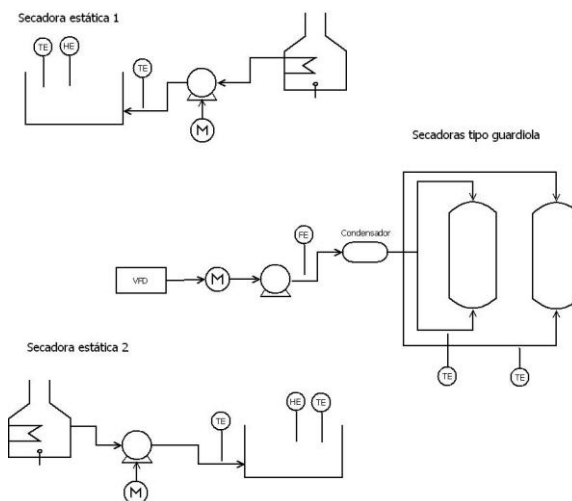
Se hicieron los cálculos necesarios para la selección de inyectores de gas para los hornos de las secadoras estáticas, luego se diseñó el sistema de control proporcional para el control de la temperatura. Luego se realizaron los cálculos de flujo de aire en las tuberías de las secadoras tipo guardiola, se seleccionó el sensor a utilizar y se diseñó el sistema de control de flujo de aire. Se diseñó un programa que mantiene un control continuo de temperatura y humedad dentro de las cámaras de secado.

Se llevaron a cabo diferentes pruebas del monitoreo de temperatura y humedad, donde se varió la temperatura medida en rangos cercanos a los manejados en la finca “La Fama” y así poder determinar la confiabilidad de los sensores. Una vez comprobado el correcto funcionamiento del mismo, se unió al módulo que recibe los datos y se realizaron pruebas de envíos de datos durante varias horas.







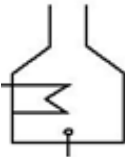
Por último se hicieron pruebas con la unión de los módulos en el ambiente industrial en los que estarían trabajando, para determinar el correcto funcionamiento del sistema de monitoreo.

B. DISEÑO

Figura 56 Diagrama P&ID de los sistemas diseñados



Cuadro 2 Representación de simbología de diagrama P&ID

Símbolo	Representación
	Elemento de temperatura (sensor)
	Elemento de humedad (sensor)
	Motor trifásico
	Ventilador
	Condensador
	Variador de frecuencia
	Horno con calentamiento externo, quemador de gas

1. Diseño y análisis de sistema de control proporcional de temperatura para secadoras estáticas. Luego de investigar las formas de medición de temperatura, humedad y flujo de aire y de haber realizado todos los cálculos necesarios, se seleccionaron los sensores a utilizar. Para la medición de temperatura se seleccionó un sensor LM35 y para la medición de humedad un DHT22, ya que ambos adaptan una tecnología que les permite medir con precisión y tienen un precio bajo en el mercado.

Se diseñó un sistema de control proporcional de temperatura, por medio de un quemador de gas para mantener los rangos de temperatura deseados, midiendo la temperatura por medio de un sensor LM35. Para el control del quemador se propuso el uso de un comparador analógico, donde se coloca un valor de referencia y se compara con la salida del sensor para realizar la acción necesaria con el quemador.

Cálculos realizados para la correcta selección de un quemador de gas

Secadora estática 1

- Entran entre 40 a 50 quintales (qq) de café húmedo
- Salen entre 20 a 25 qq de café en seco
- Tiempo aproximado: entre 32 a 40 horas
- Energía necesaria para evaporar un kilogramo (kg) de agua: $h_{fq} = 2441.7 \text{ kJ/kg}$

Tasa de transferencia de calor requerida:

$$\dot{Q}_{sec1} = \frac{m_{agua}}{\Delta t} * h_{fq}$$

Suponiendo los extremos

$$m_{agua} = 50qq - 20qq = 30qq = 1360 \text{ kg}$$

$$\Delta t = 32 \text{ horas} = 115000 \text{ seg}$$

$$\dot{Q}_{sec1} = \frac{1360 \text{ kg}}{115000 \text{ seg}} * 2441.7 \frac{\text{kJ}}{\text{kg}} = 28.9 \text{ kW}$$

Como al calentar el aire, éste debe pasar por un serpentín y ser impulsado por un ventilador, y las cámaras de secado se encuentran abiertas, se asume un 50% de eficiencia

$$\dot{Q}_{sec1} = \frac{28.9 \text{ kW}}{0.5} = 57.8 \text{ kW} = 197000 \text{ BTU/h}$$

Secadora estática 2

- Entran entre 90 a 100 quintales (qq) de café húmedo
- Salen entre 50 a 60qq de café en seco
- Tiempo aproximado: entre 32 a 40 horas
- Energía necesaria para evaporar un kilogramo (kg) de agua: $h_{fq} = 2441.7 \text{ kJ/kg}$

Tasa de transferencia de calor requerida:

$$\dot{Q}_{sec2} = \frac{m_{agua}}{\Delta t} * h_{fq}$$

Suponiendo los extremos

$$m_{agua} = 100qq - 60qq = 40qq = 1813 \text{ kg}$$

$$\Delta t = 32 \text{ horas} = 115000 \text{ seg}$$

$$\dot{Q}_{sec2} = \frac{1813 \text{ kg}}{115000 \text{ seg}} * 2441.7 \frac{\text{kJ}}{\text{kg}} = 38.5 \text{ kW}$$

Asumiendo un 50% de eficiencia

$$\dot{Q}_{sec2} = \frac{38.5 \text{ kW}}{0.5} = 77 \text{ kW} = 262700 \text{ BTU/h}$$

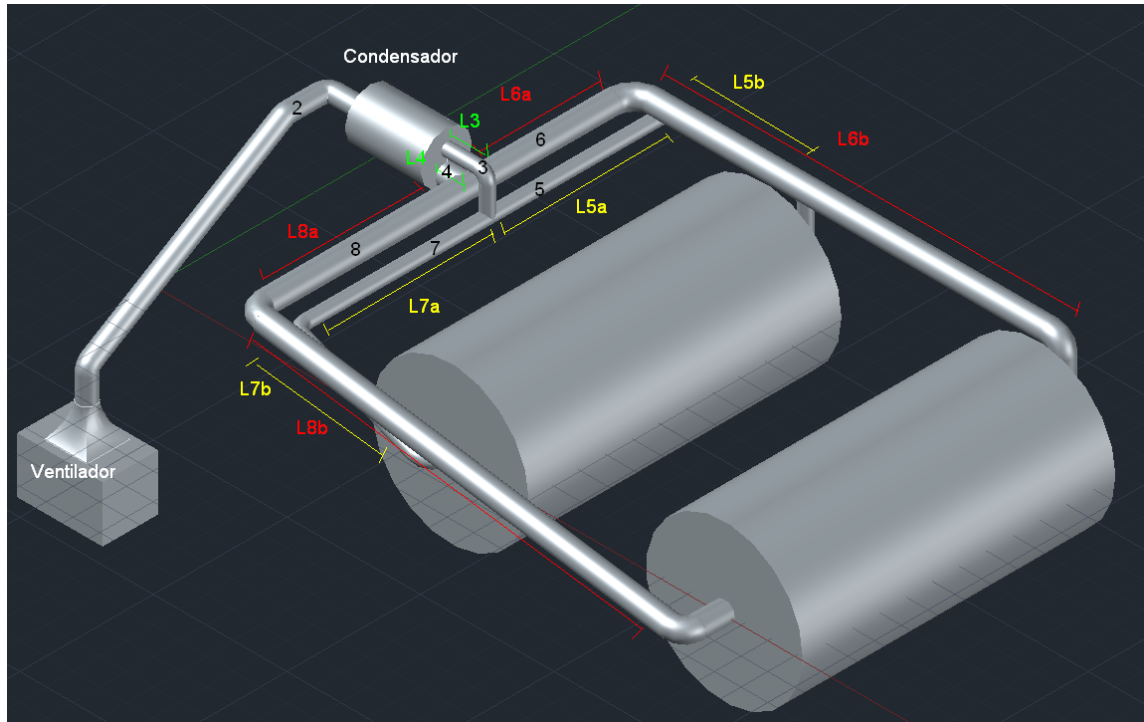
Para dicho sistema de control de temperatura se colocó un quemador modulante, el cual tiene la capacidad de modular el tamaño de su llama, dependiendo de la temperatura deseada. Los dos quemadores necesarios para las secadoras, deben tener una capacidad mayor de 77kW (262700 BTU/h), los cuales son de los más pequeños en la industria de quemadores.

2. Diseño y análisis de sistema de control de flujo de aire para secadoras tipo Guardiola. Se diseñó un sistema de control de flujo de aire por medio de un variador de frecuencia para controlar la velocidad del motor que mueve al ventilador que se encarga de soplar el aire en las tuberías. Por medio de un sensor de flujo de aire se determina el flujo en el que se encuentra el aire y con un comparador y una señal de referencia se realizan las acciones necesarias para mantener un flujo de aire deseado.

Para dicho sistema de control de flujo de aire, se seleccionó un variador de frecuencia ABB ACS 300. Éste trabaja en un rango de 0 a 500Hz suficiente para controlar diferentes valores de flujo de aire. Tiene una entrada de 0 a 10V y la señal de salida es una señal analógica de 4 a 20mA. Por último, el sensor de caudal seleccionado fue un transductor de velocidad y caudal de aire EE75, por sus altos rangos de velocidad que mide (entre 0 a 40m/s) y es apto para trabajar a temperaturas de hasta 120°C. La señal de salida se encuentra en la estándar de 4 a 20mA, por lo que hace más sencilla la lectura de las mediciones.

Para determinar el flujo de aire en cada tubería, primero se debe conocer el caudal provocado por el ventilador del sistema. Luego se realizan cálculos de pérdida de carga en cada tubería para encontrar la velocidad del aire y con el conocimiento del área de cada una, se calcula el flujo del mismo.

Figura 57 Identificación de tuberías para cálculo de caudales



Se puede observar que la tubería 4 y 3 son tuberías en paralelo, por lo que la suma de los caudales de las dos será el valor del caudal total que entra desde el condensador. También la tubería 8 es paralela con la tubería 6 y la tubería 7 paralela con la 5. A continuación se muestran cuadros con las medidas de cada tubería.

Cuadro 3 Diámetro y área de las tuberías

Diámetros	
D2 (m)	0.3
D3 (m)	0.42
D4 (m)	0.42
D5 (m)	0.28
D6 (m)	0.42
D7 (m)	0.28
D8 (m)	0.42

Áreas	
A2 (m ²)	0.0707
A3 (m ²)	0.1385
A4 (m ²)	0.1385
A5 (m ²)	0.0616
A6 (m ²)	0.1385
A7 (m ²)	0.0616
A8 (m ²)	0.1385

Cuadro 4 Largos de las tuberías

L3 (m)	0.9
L4 (m)	0.7
L5a (m)	2.44
L5b (m)	0.91
L6a (m)	2.45
L6b (m)	4.53
L7a (m)	2.15
L7b (m)	0.91
L8a (m)	2.44
L8b (m)	4.2

Cálculo del caudal en la salida del condensador, teniendo el valor del caudal en la entrada del mismo, por medio de un sensor de caudal.

Q_1 representa el valor del caudal inicial, medido por un sensor de caudal

Q_2 representa el valor del caudal en la salida del condensador

T_1 representa la temperatura en la entrada del condensador, temperatura ambiente

T_2 representa la temperatura en la salida del condensador

Asumiendo que no existen pérdidas de carga, sabemos que el flujo másico será constante

$$\dot{m}_1 = \rho_1 Q_1$$

$$\dot{m}_2 = \rho_2 Q_2$$

$$\dot{m}_1 = \dot{m}_2$$

$$\rho_1 Q_1 = \rho_2 Q_2$$

$$\rho_1 = \frac{\rho_2 Q_2}{Q_1}$$

Como el aire es un gas ideal, sabemos que

$$P_1 = \rho_1 R T_1$$

$$P_2 = \rho_2 RT_2$$

$$\rho_1 RT_1 = \rho_2 RT_2$$

$$\rho_1 = \frac{\rho_2 RT_2}{RT_1} = \frac{\rho_2 T_2}{T_1}$$

$$\frac{\rho_2 Q_2}{Q_1} = \frac{\rho_2 T_2}{T_1}$$

$$Q_2 = \frac{Q_1 T_2}{T_1}$$

Cálculo de caudales para tuberías 3 y 4

$$Q = AV$$

Donde:

Q representa el valor del caudal

A representa el área transversal de la tubería

V representa la velocidad del aire

$$Q_2 = Q_3 + Q_4 = A_3 V_3 + A_4 V_4$$

Las pérdidas de carga de la tubería 3 serán iguales a las pérdidas de la tubería 4, por ser tuberías en paralelo.

$$h_3 = \left(\frac{f_3 L_3}{D_3} + k \right) \frac{V_3^2}{2g}$$

$$h_4 = \frac{f_4 V_4^2 L_4}{2g D_4}$$

$$\left(\frac{f_3 L_3}{D_3} + k \right) \frac{V_3^2}{2g} = \frac{f_4 V_4^2 L_4}{2g D_4}$$

$$V_3 = \sqrt{\frac{f_4 V_4^2 L_4 D_3}{(f_3 L_3 + k D_3) D_4}} = V_4 \sqrt{\frac{f_4 L_4 D_3}{(f_3 L_3 + k D_3) D_4}}$$

Notamos que tenemos dos ecuaciones con dos incógnitas, por lo que resolvemos el sistema de ecuaciones.

$$Q_2 = A_3 V_4 \sqrt{\frac{f_4 L_4 D_3}{(f_3 L_3 + k D_3) D_4}} + A_4 V_4 = V_4 \left(A_3 \sqrt{\frac{f_4 L_4 D_3}{(f_3 L_3 + k D_3) D_4}} + A_4 \right)$$

$$V_4 = \frac{Q_2}{\left(A_3 \sqrt{\frac{f_4 L_4 D_3}{(f_3 L_3 + k D_3) D_4}} + A_4 \right)}$$

Una vez encontrado el valor de V_4 podemos proceder a calcular el valor de V_3 . Con el valor de las velocidades y las áreas conocidas, podemos calcular los caudales en ambas tuberías.

Cálculo de caudales para tuberías 5 y 7

$$Q_4 = Q_5 + Q_7 = A_5 V_5 + A_7 V_7$$

Las pérdidas de carga de la tubería 5 serán iguales a las pérdidas de la tubería 7, por ser tuberías en paralelo.

$$h_5 = \left(\frac{f_5 (L_{5a} + L_{5b})}{D_5} + 3k \right) \frac{V_5^2}{2g}$$

$$h_7 = \left(\frac{f_7 (L_{7a} + L_{7b})}{D_7} + 2k \right) \frac{V_7^2}{2g}$$

$$\left(\frac{f_5 (L_{5a} + L_{5b})}{D_5} + 3k \right) \frac{V_5^2}{2g} = \left(\frac{f_7 (L_{7a} + L_{7b})}{D_7} + 2k \right) \frac{V_7^2}{2g}$$

$$V_5 = V_7 \sqrt{\frac{(f_7 (L_{7a} + L_{7b}) + 2k D_7) D_5}{(f_5 (L_{5a} + L_{5b}) + 3k D_5) D_7}}$$

Notamos que tenemos dos ecuaciones con dos incógnitas, por lo que resolvemos el sistema de ecuaciones.

$$Q_4 = A_5 V_7 \sqrt{\frac{(f_7(L_{7a} + L_{7b}) + 2kD_7)D_5}{(f_5(L_{5a} + L_{5b}) + 3kD_5)D_7}} + A_7 V_7 = V_7 \left(A_5 \sqrt{\frac{(f_7(L_{7a} + L_{7b}) + 2kD_7)D_5}{(f_5(L_{5a} + L_{5b}) + 3kD_5)D_7}} + A_7 \right)$$

$$V_7 = \frac{Q_4}{\left(A_5 \sqrt{\frac{(f_7(L_{7a} + L_{7b}) + 2kD_7)D_5}{(f_5(L_{5a} + L_{5b}) + 3kD_5)D_7}} + A_7 \right)}$$

Una vez encontrado el valor de V_7 podemos proceder a calcular el valor de V_5 . Con el valor de las velocidades y las áreas conocidas, podemos calcular los caudales en ambas tuberías.

Cálculo de caudales para tuberías 6 y 8

$$Q_3 = Q_6 + Q_8 = A_6 V_6 + A_8 V_8$$

Las pérdidas de carga de la tubería 6 serán iguales a las pérdidas de la tubería 8, por ser tuberías en paralelo.

$$h_6 = \left(\frac{f_6(L_{6a} + L_{6b})}{D_6} + 3k \right) \frac{V_6^2}{2g}$$

$$h_8 = \left(\frac{f_8(L_{8a} + L_{8b})}{D_8} + 2k \right) \frac{V_8^2}{2g}$$

$$\left(\frac{f_6(L_{6a} + L_{6b})}{D_6} + 3k \right) \frac{V_6^2}{2g} = \left(\frac{f_8(L_{8a} + L_{8b})}{D_8} + 2k \right) \frac{V_8^2}{2g}$$

$$V_6 = V_8 \sqrt{\frac{(f_8(L_{8a} + L_{8b}) + 2kD_8)D_6}{(f_6(L_{6a} + L_{6b}) + 3kD_6)D_8}}$$

Notamos que tenemos dos ecuaciones con dos incógnitas, por lo que resolvemos el sistema de ecuaciones.

$$Q_3 = A_6 V_8 \sqrt{\frac{(f_8(L_{8a} + L_{8b}) + 2kD_8)D_6}{(f_6(L_{6a} + L_{6b}) + 3kD_6)D_8}} + A_8 V_8 = V_8 \left(A_6 \sqrt{\frac{(f_8(L_{8a} + L_{8b}) + 2kD_8)D_6}{(f_6(L_{6a} + L_{6b}) + 3kD_6)D_8}} + A_8 \right)$$

$$V_8 = \frac{Q_3}{\left(A_6 \sqrt{\frac{(f_8(L_{8a} + L_{8b}) + 2kD_8)D_6}{(f_6(L_{6a} + L_{6b}) + 3kD_6)D_8}} + A_8 \right)}$$

Una vez encontrado el valor de V_8 podemos proceder a calcular el valor de V_6 . Con el valor de las velocidades y las áreas conocidas, podemos calcular los caudales en ambas tuberías.

Notar que, como los valores de los coeficientes de fricción deben ser aproximados por medio del diagrama de Moody, se debe iterar para encontrar el valor correcto de cada uno de ellos.

3. **Diseño de arranque de motores.** Se diseñó un arranque de motor de tipo estrella-delta para el motor que mueve el ventilador de las secadoras tipo guardiola, para evitar picos de corriente en el mismo. El diagrama del diseño utilizado se puede observar en la sección de resultados, Figura 62.

C. IMPLEMENTACIÓN

1. **Programación en Arduino.** Se realizaron programas en código Arduino para la medición de temperatura y humedad, con el fin de monitorear dichas variables. Todos los datos medidos son enviados a otro módulo por medio de comunicación serial, para ser mostrados en forma de gráficas.

2. **Pruebas de laboratorio.** Se realizaron pruebas de laboratorio para determinar el correcto funcionamiento de los sensores. Con ayuda de un pirómetro infrarrojo se comprobaron las mediciones del sensor LM35. Luego se realizaron otras pruebas con todos los sensores, para determinar el funcionamiento de todos los LM35 y los DTH22 (temperatura y humedad). También se realizaron diversas pruebas de comunicación con el módulo que recibe los datos.

D. RESULTADOS

1. **Sistema de monitoreo de temperatura y humedad en cámaras de secado de café**

Figura 58 Diagrama de flujo de monitoreo de variables

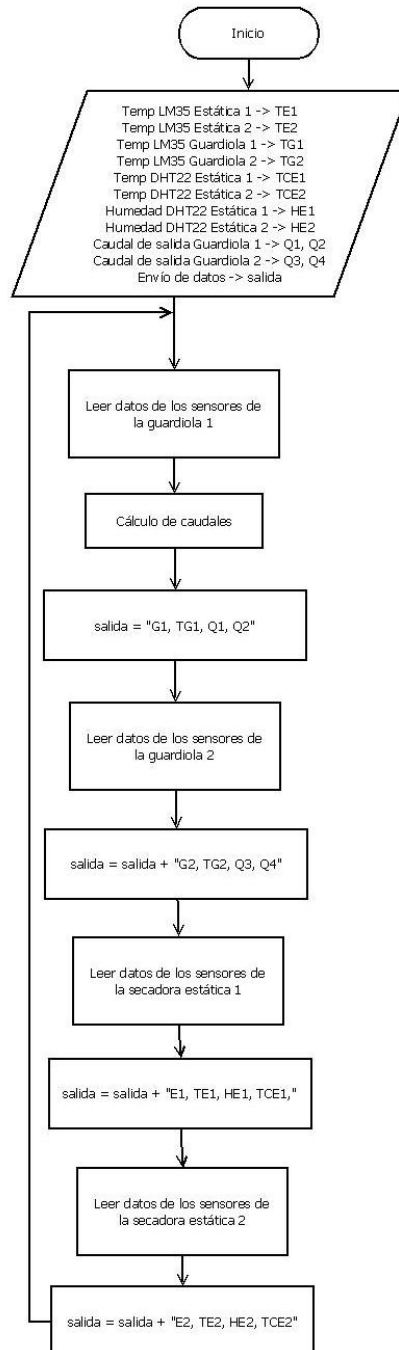


Figura 59 Envío serial de mediciones de sensores

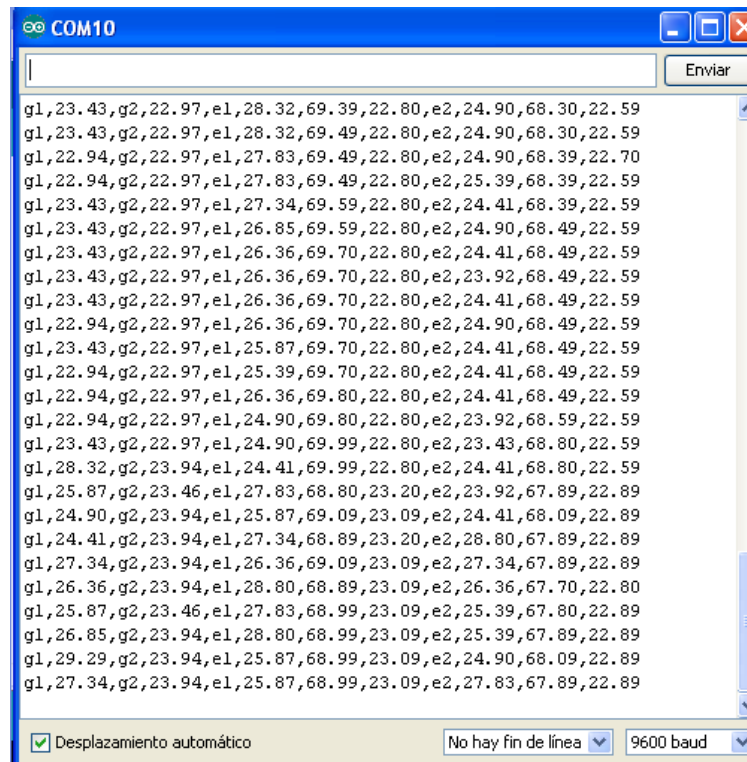


Figura 60 Prueba de laboratorio con sensores e indicadores luminosos

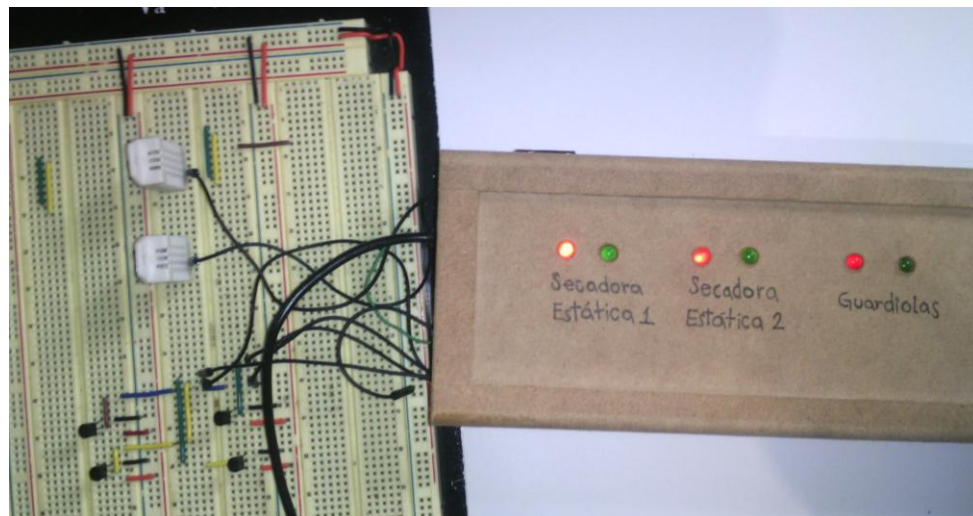
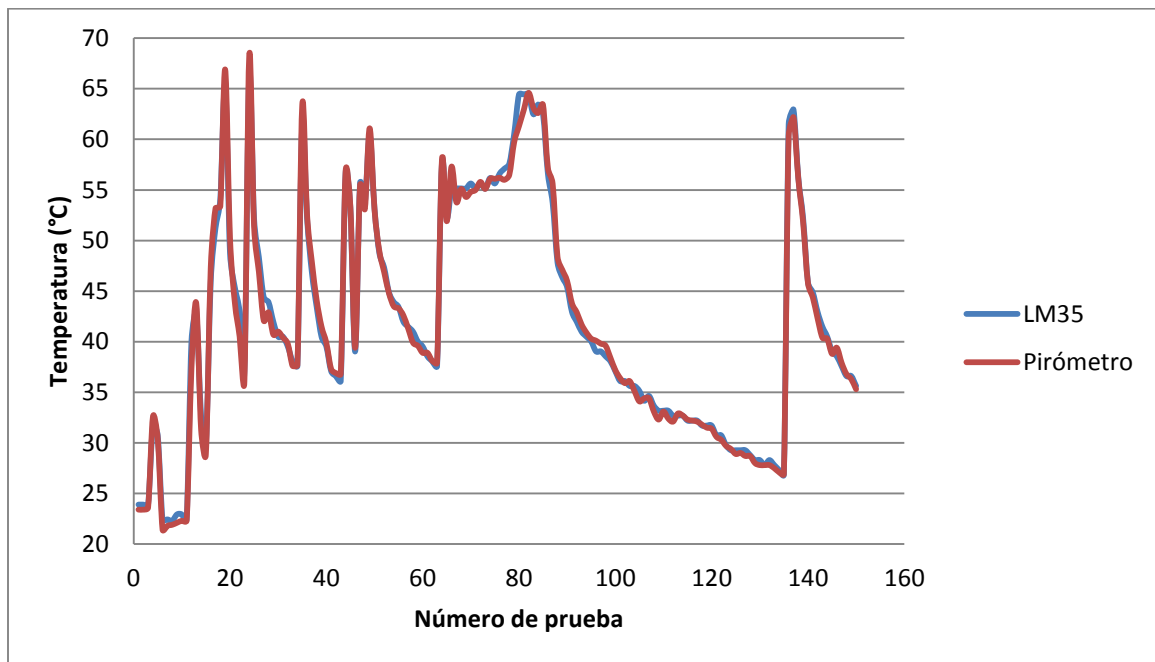


Figura 61 Prueba de laboratorio con indicadores luminosos indicando la temperatura del aire



Figura 62 Prueba de sensores LM35 y medición por medio de un pirómetro



Cuadro 5 Mediciones de prueba con LM35 y pirómetro con porcentajes de error

No. Prueba	LM35 (°C)	Pirómetro (°C)	% error
1	23.9	23.4	2.14
2	23.9	23.4	2.14
3	23.9	23.6	1.27
4	31.71	32.6	2.73
5	30.74	30.4	1.12
6	22.4	21.4	4.67
7	22.44	21.8	2.94
8	22.3	21.9	1.83
9	22.93	22.1	3.76
10	22.93	22.3	2.83
11	22.44	22.3	0.63
12	40.02	37.5	6.72
13	42.95	43.8	1.94
14	32.69	31.3	4.44
15	29.76	28.9	2.98
16	45.88	47.3	3.00
17	51.25	53.2	3.67
18	54.18	53.4	1.46
19	65.41	66.9	2.23
20	48.8	50.3	2.98
21	45.39	43.9	3.39
22	42.95	40.5	6.05
23	37.09	36.1	2.74
24	66.38	68.3	2.81
		Promedio % error	1.5%

Figura 63 Medición de temperatura por medio de un pirómetro



Figura 64 Prueba de medición de temperatura

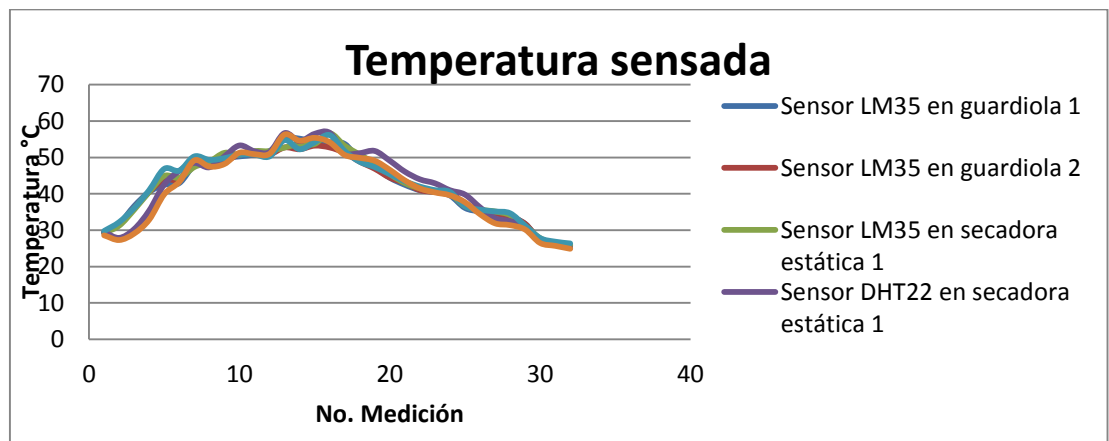
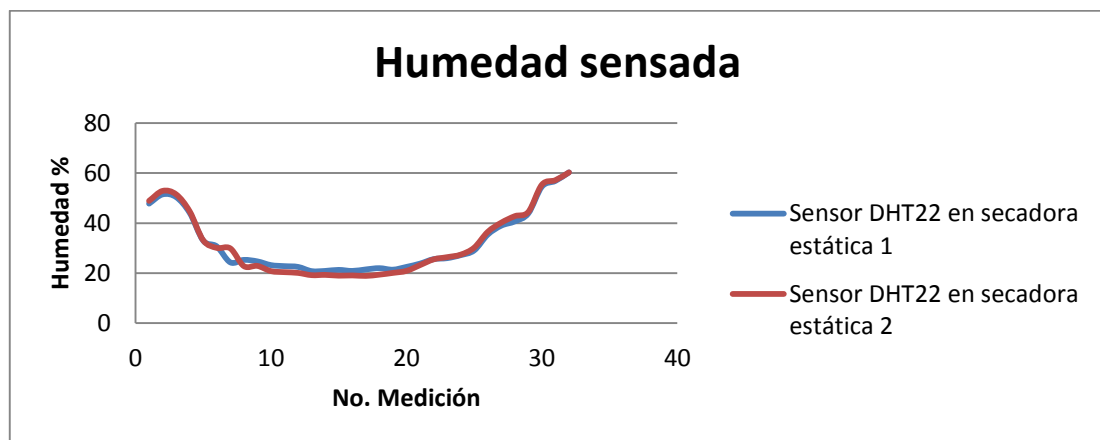


Figura 65 Prueba de medición de humedad



Cuadro 6 Mediciones de pruebas de temperatura y humedad

No. Medición	Lm35 guardiola 1 (°C)	Lm35 guardiola 2 (°C)	Lm35 estática 1 (°C)	DHT22 estática 1 Humedad (%)	DHT22 estática 1Temp (°C)	LM35 estática 2 (°C)	DHT22 estática 2 Humedad (%)	DHT22 estática 2Temp (°C)
1	29.29	29.32	29.29	47.79	29.49	29.78	48.9	28.59
2	31.73	31.76	31.24	51.49	27.89	32.22	52.89	27.3
3	36.62	36.15	35.64	50.4	30.3	36.13	51.49	29.09
4	40.52	40.55	40.52	44.1	35.4	41.01	44.69	32.9
5	42.48	43.96	44.92	32.99	42.6	46.87	33.1	39.89
6	42.96	43.96	44.43	30.69	46.29	46.38	30.09	43.29
7	47.85	48.85	47.36	24.3	48.49	50.29	29.89	49.19
8	48.82	48.36	48.82	25.3	47.29	49.31	22.8	47.49
9	49.8	48.85	51.26	24.7	50.29	49.8	22.89	48.19
10	50.29	50.8	50.78	23.2	53.29	50.78	20.8	51.29
11	50.78	50.8	51.75	22.8	51.69	50.78	20.39	50.9
12	51.75	50.8	51.75	22.49	51.49	50.29	20.09	50.99
13	55.17	52.75	52.73	20.8	56.69	54.78	19.19	56.29
14	55.17	52.26	53.71	20.89	54.49	52.26	19.3	54.6
15	54.19	53.24	53.71	21.3	56.49	54.19	18.99	55.39
16	54.19	52.75	56.64	20.89	56.79	56.15	19.09	54.19
17	53.71	51.29	53.22	21.49	51.29	51.75	18.89	50.69
18	49.31	48.85	50.78	21.99	51.19	48.82	19.39	49.9
19	47.36	46.89	48.33	21.39	51.79	47.36	20.09	49.1
20	44.43	44.45	45.41	22.49	49.1	45.41	20.89	46.6
21	42.48	42.99	42.96	23.8	46.1	43.45	23.2	43.69
22	41.01	41.03	41.5	25.59	43.99	41.99	25.49	41.49
23	40.52	40.55	40.52	25.99	42.9	41.01	26.39	40.39
24	39.55	40.55	40.03	27.19	40.99	40.52	27.39	39.6
25	36.13	37.13	36.62	29.09	39.79	36.62	30.19	37.69
26	35.15	35.66	35.64	35.4	36.29	35.64	36.49	34.4
27	34.66	34.69	35.15	38.99	33.29	35.15	40.19	31.89
28	33.69	33.71	34.17	40.69	32.49	34.66	42.79	31.39
29	31.73	31.76	31.24	43.9	31.09	31.24	44.49	30.19
30	27.34	27.36	27.83	54.6	26.89	27.83	55.39	26.49
31	26.36	26.39	26.36	56.89	26.09	26.85	57.19	25.7
32	25.87	25.9	25.87	60.29	25.09	26.36	60.29	24.89

2. Sistema de control de temperatura en secadoras estáticas

Figura 66 Diagrama de bloques sistema de control de temperatura

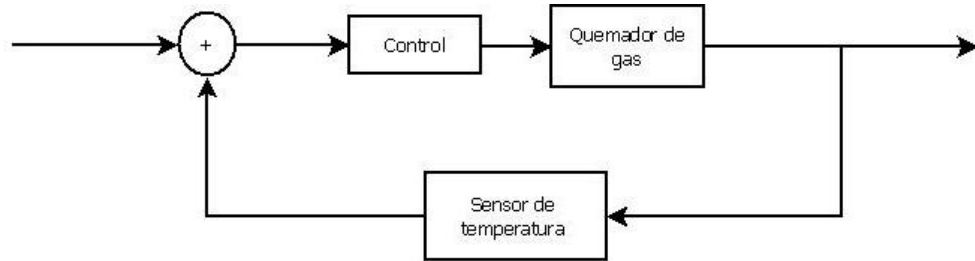


Figura 67 Diagrama de flujo sistema de control de temperatura

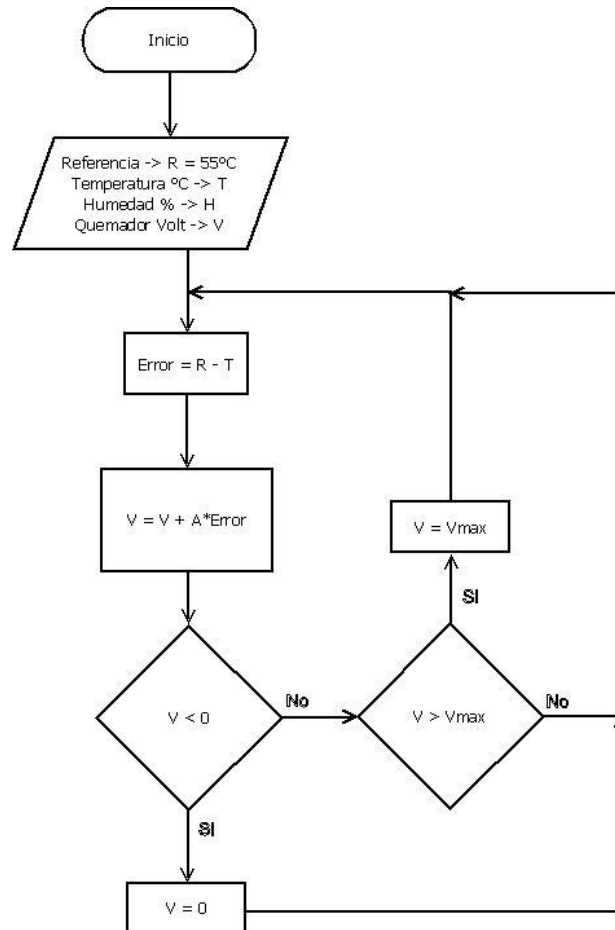


Figura 68 Lógica analógica para control de temperatura

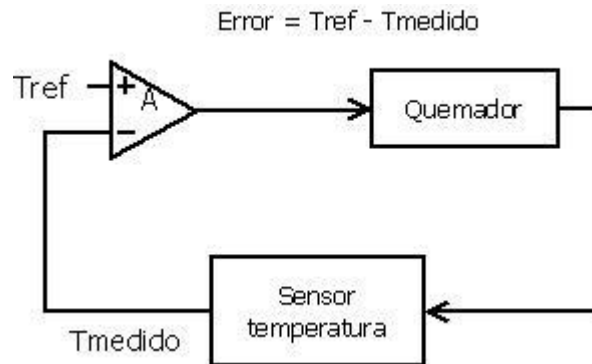
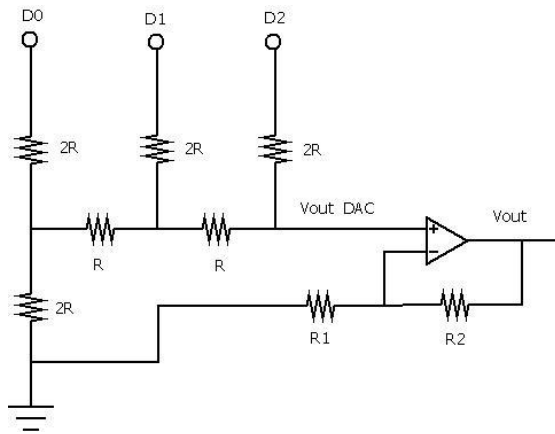


Figura 69 Circuito DAC con amplificador para determinar la entrada al quemador



Cuadro 7 Voltaje de salida de un DAC, con señales de 5V y salida luego del amplificador

Decimal	Binario (D2 D1 D0)	Vout DAC (V)	Vout (V)
0	000	0.00	0.00
1	001	0.625	1.25
2	010	1.250	2.50
3	011	1.875	3.75
4	100	2.500	5.00
5	101	3.125	6.25
6	110	3.750	7.50
7	111	4.375	8.75

Cuadro 8 Tasa de transferencia de calor requerida por cada secadora estática

	Tasa de transferencia de calor	
	kW	BTU/h
Secadora estática 1	57.8	197000
Secadora estática 2	77	262700

3. Sistema de control de flujo de aire en secadoras tipo guardiola

Figura 70 Diagrama de bloques del sistema de control de flujo de aire

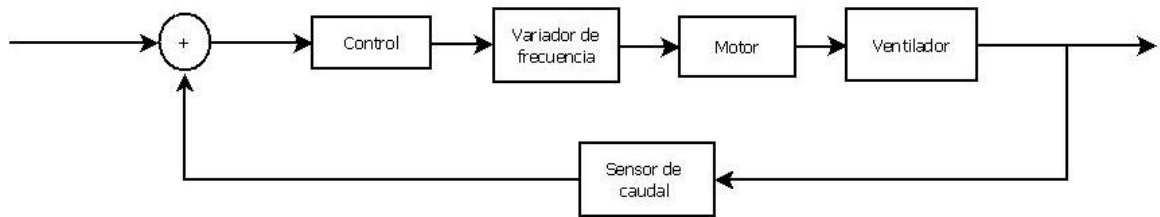


Figura 71 Diagrama de flujo de control de flujo de aire

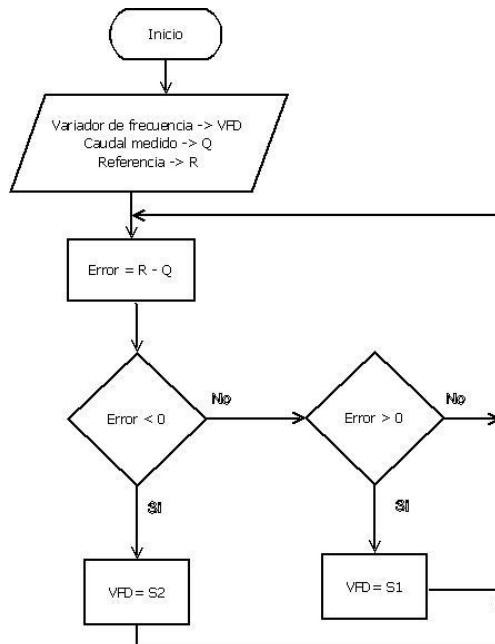
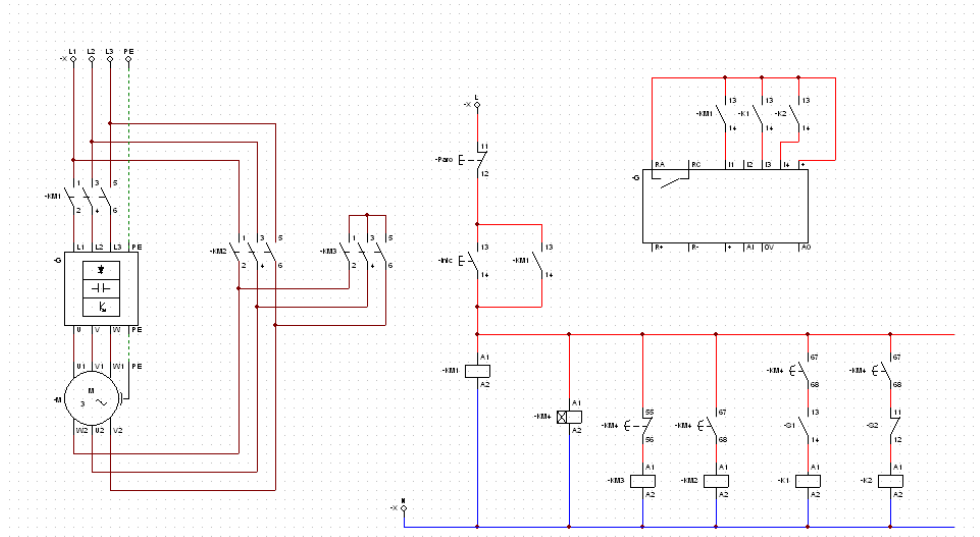


Figura 72 Esquema eléctrico para control de variador de frecuencia con arranque estrella-delta



Cuadro 9 Cálculo de caudales en tuberías

Tubería	Caudal de aire
2	Q_2
3	$A_3 V_4 \sqrt{\frac{f_4 L_4 D_3}{(f_2 L_2 + k D_2) D_4}}$
4	$\frac{A_4 Q_2}{\left(A_3 \sqrt{\frac{f_4 L_4 D_3}{(f_2 L_2 + k D_2) D_4}} + A_4 \right)}$
5	$A_5 V_7 \sqrt{\frac{(f_7 (L_{7a} + L_{7b}) + 2k D_7) D_5}{(f_5 (L_{5a} + L_{5b}) + 3k D_5) D_7}}$
6	$A_6 V_8 \sqrt{\frac{(f_8 (L_{8a} + L_{8b}) + 2k D_8) D_6}{(f_6 (L_{6a} + L_{6b}) + 3k D_6) D_8}}$
7	$\frac{A_7 Q_4}{\left(A_5 \sqrt{\frac{(f_7 (L_{7a} + L_{7b}) + 2k D_7) D_5}{(f_5 (L_{5a} + L_{5b}) + 3k D_5) D_7}} + A_7 \right)}$
8	$\frac{A_8 Q_3}{\left(A_6 \sqrt{\frac{(f_8 (L_{8a} + L_{8b}) + 2k D_8) D_6}{(f_6 (L_{6a} + L_{6b}) + 3k D_6) D_8}} + A_8 \right)}$

E. ANÁLISIS DE RESULTADOS

Para realizar el monitoreo de variables en las secadoras, se optó por la implementación de sensores de temperatura LM35 y sensores de temperatura y humedad DHT22 por su precisión y asequibilidad. Estos se conectaron a una plataforma Arduino para realizar las mediciones y posteriormente enviar los datos por comunicación serial vía UART a otro módulo del proyecto. En el diagrama de flujo de la Figura 48 se puede observar que primero se realiza la medición de temperatura en las secadoras tipo guardiola, luego en las secadoras estáticas. Toda la información es enviada por medio de la variable “salida”. Se colocaron indicadores antes de cada serie de datos, para que el módulo que los recibe pueda identificar de qué secadora proviene cada conjunto de mediciones.

En la Figura 49 se pueden observar los datos enviados. Estos son: temperatura del aire en las guardiolas 1 y 2, temperatura y humedad en la cámara de secado y la temperatura del aire de entrada, en las secadoras estáticas 1 y 2. Se optó por implementar un Olimexino 328 en lugar de un Arduino UNO, ya que su lugar de instalación es un ambiente industrial donde el ruido de los motores y otros equipos pueden afectar su funcionamiento. Se realizaron pruebas de su funcionamiento en la finca “La Fama” y todas las mediciones y envío de datos fueron exitosas.

Para que los operadores puedan tener un mejor control de la temperatura del aire, se implementaron indicadores luminosos. En estos el color verde indica que el aire se encuentra dentro del rango recomendado, el color rojo indica que el aire se encuentra sobre el rango recomendado y mientras los indicadores se mantienen apagados quiere decir que se encuentra debajo de la temperatura deseada. En la Figura 50 se pueden observar los sensores y los indicadores luminosos como resultado de colocar aire muy caliente en todos los sensores y en la Figura 51 se puede observar que en las guardiolas el aire se encuentra debajo del rango ya que no se encuentra encendido ningún indicador. En la secadora estática 2 el aire se encuentra sobre el rango ya que se encuentra encendido el indicador rojo y en la secadora estática 1 se tiene una temperatura deseada ya que se observa el indicador verde.

Para determinar la confiabilidad de las mediciones realizadas por los sensores se realizó la siguiente prueba: mientras se soplaba aire caliente, se realizaron mediciones de temperatura con un sensor LM35 en paralelo con un pirómetro infrarrojo con láser marca Mastercool, con rangos de temperatura entre -50 a 500°C, con 0.5°C de resolución y $\pm 2^\circ\text{C}$ de precisión. Se fue aumentando la temperatura hasta un valor de 70°C, para obtener resultados de valores que sobrepasen el rango que se maneja el aire caliente en la finca “La Fama” y así asegurar el funcionamiento de los sensores.

Se puede observar en la Figura 52 la gráfica resultante de las 150 mediciones realizadas. El porcentaje de error obtenido entre ambos sensores se puede observar en el Cuadro 5, se puede decir que las mediciones del sensor LM35 son confiables, ya que el porcentaje de error promedio es de 1.5% donde el mayor de estos es inferior del 7%. En la temperatura máxima medida (70°C), un porcentaje de error de 7% equivale a una desviación de 4.9°C . Durante el proceso de secado de café se manejan rangos entre $40\text{-}50^{\circ}\text{C}$ en la finca “La Fama”, se recomienda que dichos valores no sobrepasen de 60°C , por lo que una desviación de 4.9°C no afecta la calidad del mismo. Como se puede observar en la Figura 53 para poder realizar la medición con el pirómetro se debía apuntar correctamente al sensor y esperar un pequeño instante para leer la temperatura. Esta demora pudo generar error en las mediciones, ya que no se realizaba la lectura de temperatura en tiempo real.

Una vez ya comprobada la confiabilidad del sensor LM35, se realizó otra prueba de medición de temperatura y humedad con todos los sensores simultáneamente. De la misma manera, se sopló aire caliente a todos los sensores y por medio de la consola de comunicación serial de Arduino se tomaron todas las mediciones. Se puede observar en la Figura 54 gráficas de los 4 sensores LM35 y 2 sensores DHT22, todas las gráficas se encuentran dentro de un mismo rango, algunas variaciones se pudieron haber generado a causa de que los sensores se encontraban separados unos de otros y el aire caliente no llegaba a todos a la misma temperatura; sin embargo se obtuvieron buenos resultados.

En la Figura 55 se observa la humedad medida por ambos sensores DHT22 y se puede observar el mismo comportamiento y valores cercanos de humedad en ambos sensores. Comparando los valores de las mediciones de todos los sensores y siendo los errores de medición bajos, se puede tener confianza en que las mediciones tendrán en promedio un error de 1.5%, ya que anteriormente se comprobó la confiabilidad de un sensor LM35. En el Cuadro 6 se pueden observar todos los valores de temperatura y humedad medidos durante la prueba de laboratorio, los cuales fueron utilizados para realizar las gráficas de la Figura 54 y Figura 55.

Para poder tener control de la temperatura del aire que entra a las secadoras estáticas se diseñó un sistema de control con un quemador de gas. En la Figura 56 se puede observar el diagrama de bloques del sistema, donde, para determinar el voltaje de entrada del quemador modulante y mantener la temperatura dentro de un rango específico, se tiene como realimentación un sensor de temperatura LM35. En la Figura 57 se puede observar el diagrama de flujo del sistema para la selección del voltaje de entrada del quemador. Dicho control se realiza por medio de un sistema de control proporcional, donde se calcula el error entre el valor actual y una referencia y dependiendo de ese resultado se aumenta o disminuye el voltaje de entrada. El cálculo del error se realiza por medio de un comparador analógico denotado en la Figura 58.

Como se está utilizando un micro-controlador para el control de los sistemas y estos no disponen de una salida de voltaje analógica, no es posible controlar directamente el voltaje del quemador. Por esta razón se utilizó un convertidor digital - analógico (DAC por sus siglas en inglés). Se puede observar en la Figura 59 el circuito DAC con un amplificador de voltaje. El DAC posee tres entradas digitales de 5V cada una, por medio de la configuración de las resistencias se tiene una suma de voltajes en la salida dependiendo de las entradas. En el Cuadro 7 se puede observar el voltaje de salida en función de la entrada, se puede observar que utilizando 3 bits se tienen 8 diferentes niveles de voltaje, por lo que si se quiere tener mayores variaciones para tener más control sobre el quemador, se debe aumentar el número de bits en la entrada. El quemador descrito en el sistema diseñado maneja voltajes entre 0 a 10V, por lo que se colocó un amplificador de voltaje con ganancia 2 para tener en la entrada los valores correctos. Se puede observar que por la resolución del DAC utilizada no se obtiene una salida de 5V, por lo que el valor máximo de voltaje en la salida del amplificador es de 8.75V. En el Cuadro 8 se puede observar la tasa de transferencia de calor de cada una de las secadoras estáticas, lo que quiere decir que para la secadora estática 1 se requiere un quemador con capacidad de 57.8kW o mayor. Para la secadora estática 2 se requiere un quemador con capacidad de 77kW o uno mayor para poder realizar el trabajo deseado de manera correcta.

En las secadoras tipo guardiola se requiere tener control del flujo de aire para tener un proceso eficiente de secado de café. En la Figura 60 se observa el diagrama de bloques del sistema de control de flujo de aire, donde se utiliza un variador de frecuencia en el motor que mueve el ventilador que sopla el aire a las secadoras. El fin de utilizar un variador de frecuencia es acelerar o desacelerar el motor para que el ventilador gire con mayor o menor velocidad y de esta manera se logra reducir o aumentar el flujo de aire. Para determinar el flujo de aire se colocó un sensor de caudal justo después del ventilador y por medio de un comparador analógico y una señal de referencia se obtiene el control del mismo. Se buscó diseñar un sensor de flujo de aire propio pero por medio de investigación bibliográfica se determinó que el diseño del mismo escapa los alcances del proyecto, por lo que se optó por implementar un sensor industrial.

Se puede observar en la Figura 61 el diagrama de flujo para el sistema que controla el flujo de aire. Dependiendo del valor del error se selecciona entre S1 y S2: si el error es negativo el caudal actual se encuentra sobre la referencia, por lo que se selecciona S2 para desacelerar el variador de frecuencia; si el error es positivo se requiere acelerar y se selecciona S1. Es posible cambiar la referencia por medio de una computadora que contenga el programa del controlador, por si el operador necesita un flujo más lento o rápido, de acuerdo a la cantidad de café que se requiera secar.

En la Figura 62 se observa el esquema eléctrico del control del variador de frecuencia. El variador de frecuencia selecciona si debe aumentar o disminuir su velocidad por medio de la señal S1 y S2. Si se envía una señal S1 y ya ha terminado el arranque suave del motor, se activa la bobina K1 que conecta una señal

al variador y éste aumenta su velocidad. Si se envía una señal S2 y el arranque ha terminado, se desconecta la bobina K2 y la velocidad del variador disminuye. Se implementó un arranque de motor estrella-delta para evitar los picos de corriente en el motor y evitar daños a los equipos. Se diseñó un arranque estrella-delta, ya que el torque necesario para trabajar no es alto, por ser sólo el movimiento de un ventilador.

Por medio de un solo sensor de flujo de aire a la salida del ventilador, se puede conocer el flujo que entra a todas las guardiolas ya que es un mismo sistema de tuberías. Para la determinación del caudal en todas las salidas, fue necesario realizar cálculos de pérdida de carga. El resultado final para el cálculo de las tuberías se puede observar en el Cuadro 9, los cuales fueron únicamente planteados con variables ya que actualmente no se tiene un sensor de flujo de aire instalado, por lo que no se conoce el flujo inicial. La numeración de las tuberías se puede observar en la Figura 47 y los valores de diámetros, áreas y largos de tuberías pueden observarse en el Cuadro 3 y Cuadro 4 respectivamente. El sistema de flujo de aire no fue implementado, ya que el alcance del proyecto únicamente era el diseño del mismo.

Para ambos sistemas de control diseñados, no se pudo obtener una función de transferencia y realizar un análisis más profundo, ya que el equipo nuevo necesario sobrepasa el presupuesto para la realización del proyecto.

F. CONCLUSIONES

1. Se diseñó e implementó un sistema de monitoreo de temperatura y humedad en las cámaras de secado de café.
2. El error máximo en la medición de temperatura (7%) es aceptable por los rangos manejados en la finca “La Fama”.
3. El Olimexino-328 trabaja correctamente en ambiente industrial sin verse afectado por el ruido de los motores y otros equipos industriales.
4. Se diseñó un sistema de control de temperatura con un inyector de gas, para las secadoras estáticas.
5. Para el sistema de control de temperatura, se requiere un quemador de gas de una capacidad de transferencia de calor de 57.8kW para la secadora estática 1 y uno de 77kW para la secadora estática 2.
6. Se diseñó un sistema de control de flujo de aire para las secadoras tipo guardiola, tomando en cuenta los cálculos de flujo de aire para el sistema de tubería de las mismas.

G. RECOMENDACIONES

- Para una segunda fase del proyecto, se recomienda comprar los equipos seleccionados para los diseños, para así poder obtener la función de transferencia de los mismos y realizar un sistema más eficiente, incluso se podría llegar a implementar un sistema de control PID y no solo proporcional.
- La implementación de un Olimexino-328 funciona bastante bien, pero para mayores funciones y más facilidad, se puede instalar un PLC para el manejo y control de todos los equipos industriales.
- Implementar arranques de motores en todos los motores que se utilizan en la finca, para así evitar picos de corriente que puedan dañar los equipos.
- Obtener curvas de relación entre la humedad de las cámaras de secado de café y la humedad del grano, para automatizar más el proceso.
- Diseñar un sensor de flujo de aire propio a partir del diseño y cálculo de un ventilador de entrada. Esto para poder determinar de mejor forma el caudal y así conocer el flujo de aire en toda la tubería.

VI. MÓDULO DE CONTROL INTEGRAL DEL PROCESO DE SECADO DEL CAFÉ Y DEL GENERADOR DIESEL DESDE UNA RASPBERRY PI VÍA WEB DE LA FINCA “LA FAMA”

A. METODOLOGÍA

A continuación se muestra la metodología que se utilizó para el despliegue de mediciones de temperatura y humedad de secadoras de café y el diseño del control de velocidad del motor diésel.

Se realizó una visita a la finca La Fama en la cual se nos explicaron los procesos de secado que se realizan, la importancia e uso de los datos a medir y el uso que se le da a la planta auxiliar de generación de energía eléctrica. Luego de esta visita se realizaron investigaciones bibliográficas sobre diseño de HMIs, tipos de sensores industriales de temperatura, sensores industriales de nivel, protocolos de transferencia de archivos y métodos de regulación de velocidad de motores de combustión interna. En todas las investigaciones realizadas se analizaron diferentes métodos para desarrollar la sección del módulo a la que correspondía la investigación.

Luego se diseñó y simuló, con la ayuda del software Autodesk Inventor 2014, el sistema a utilizar para la regulación de velocidad del motor diésel para luego iniciar con la fabricación de mecanismo el cual, debido a sus dimensiones, se decidió fabricar en plástico con la ayuda de las impresoras 3D que se poseen en las instalaciones de la universidad. Se diseñó y simuló el circuito rectificador de señal AC utilizado para convertir la señal AC del generador a un nivel de voltaje DC para ser medido por el Olimexino 328. Por último, en esta sección, se desarrolló un programa para la medición de temperatura, nivel y regulación de velocidad del motor diésel. Dicho programa envía los datos medidos vía serial a la Raspberry pi. En la Raspberry pi se desarrolló una aplicación en lenguaje Python para la recolección de datos medidos en las secadoras de café. También se desarrolló una aplicación con la cual los colaboradores de la finca La Fama pueden controlar el nivel de voltaje generado en la planta auxiliar y a la vez pueden ingresar los datos importantes de los procesos realizados en las cuatro secadoras que se encuentran en el beneficio.

Utilizando Visual Studio en sistema operativo Windows se desarrolló una aplicación que presenta las herramientas necesarias para la visualización de datos recolectados por la Raspberry pi. También se desarrolló una página web simple, montada en un servidor virtual gratuito, que muestra los datos medidos en las cuatro secadoras del beneficio y en la planta auxiliar de generación. Luego se instaló un servidor FTP

en la Raspberry pi para que la PC central (cliente del protocolo FTP) tuviera acceso a los archivos con los datos almacenados sin interrumpir los procesos realizados en la Raspberry pi ni en los Olimexinos. Debido a que el servidor web utilizado permite realizar conexiones FTP la PC central también se desempeña como cliente para ese servidor y modifica archivos en el cada minuto.

Junto con los otros módulos trabajados en el megaproyecto se realizaron pruebas en el laboratorio, recibiendo los datos de los sensores y graficándolos en el HMI con actualización de datos cada minuto. Los datos más importantes fueron enviados al servidor web para ser mostrados en la página. Por ultimo, se realizó otra visita a la finca La Fama para realizar pruebas de la recolección de datos y despliegue de los mismo en el HMI. Se recibió retroalimentación por parte del dueño de la finca para poder realizar mejoras al sistema más adelante.

B. DISEÑO

Para el diseño del mecanismo acelerador se tomaron en cuenta cuatro tipos de mecanismos capaces de transformar el movimiento circular de un motor a un movimiento rectilíneo para desplazar el cable acelerador del motor diesel en la dirección deseada. Utilizando el método de selección de pugh se analizaron los siguientes mecanismos: Leva (opción A), piñón-cremallera (opción B), biela-manivela (opción C), tornillo y tuerca (opción D). También se agregó la opción de no instalar ningún mecanismo, lo cual implica no tener control sobre la aceleración, como referencia (opción R).

Para la evaluación de los criterios se toma en cuenta un rango de puntuación dependiendo de la dificultad o el beneficio de la construcción del prototipo de -2 a 2, en donde:

-2 = aspecto no favorable para la realización del prototipo

-1 = aspecto no favorable, pero se puede buscar una solución

0 = sin relevancia (neutro)

1 = importancia

2 = mucha importancia

Cuadro 10. Matriz de pugh para evaluar los mecanismos del acelerador.

Criterios	peso	Conceptos de diseño					Peso por calificación				
		R	A	B	C	D	R	A	B	C	D
Mantenimiento	4	0	1	2	0	2	0	4	8	0	8
Facil instalación	3	0	-1	2	-1	-1	0	-3	6	-3	-3
Facilidad de control	2	0	1	2	1	2	0	2	4	2	4
Facilidad de construcción	1	0	-1	2	1	2	0	-1	2	1	2
							0	2	20	0	11

Tomando el resultado obtenido con la matriz de pugh se decidió diseñar un mecanismo similar a un sistema piñón-cremallera. La pieza mecánica que hace el trabajo de piñón tiene un radio de 70mm dando una separación entre dientes de 10 cm debido a que solamente se implementaron dos dientes perpendiculares. El mecanismo se diseñó utilizando el software Autodesk Inventor 2014 y se realizaron simulaciones de movimientos rotacionales para poder calcular el movimiento lineal que se obtiene en la salida. En la figura 74 se observa que con una rotación de 50 grados del motor se obtiene un desplazamiento lineal de 54.7mm.

Se realizaron mediciones, en la finca, para poder determinar el desplazamiento necesario en el acelerador para producir los niveles de voltaje deseados. Estos resultados se muestran en la figura 73 donde podemos apreciar que para obtener 220v en la salida debemos desplazar el acelerador 6.22mm lo cual equivale a 5.3 grados de giro del motor. Debido a que se necesitan movimientos precisos en la pieza rotacional se decidió utilizar un servomotor controlado con un Olimexino-328. Este produce movimientos en el eje del servomotor de hasta 0.18 grados.

Figura 73. Relación distancia-voltaje de mecanismo acelerador de motor diesel.

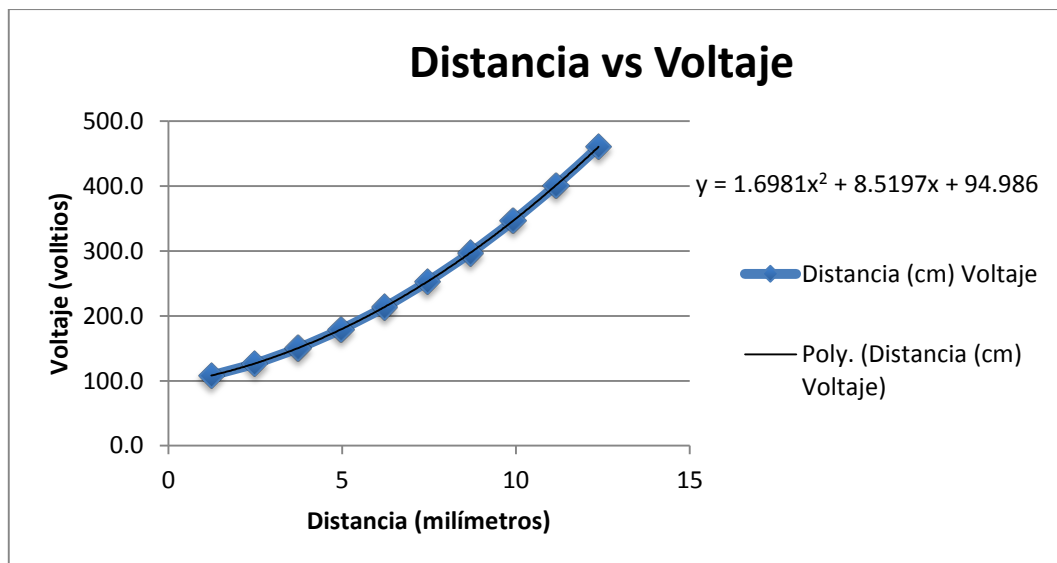


Figura 74. Relación ángulo-distancia de mecanismo regulador de velocidad.

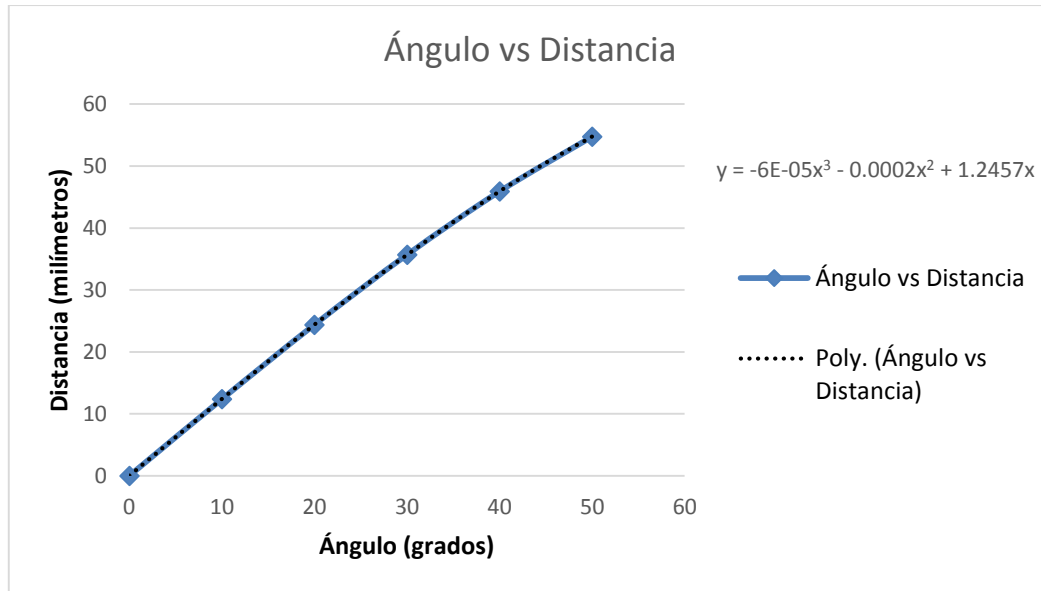
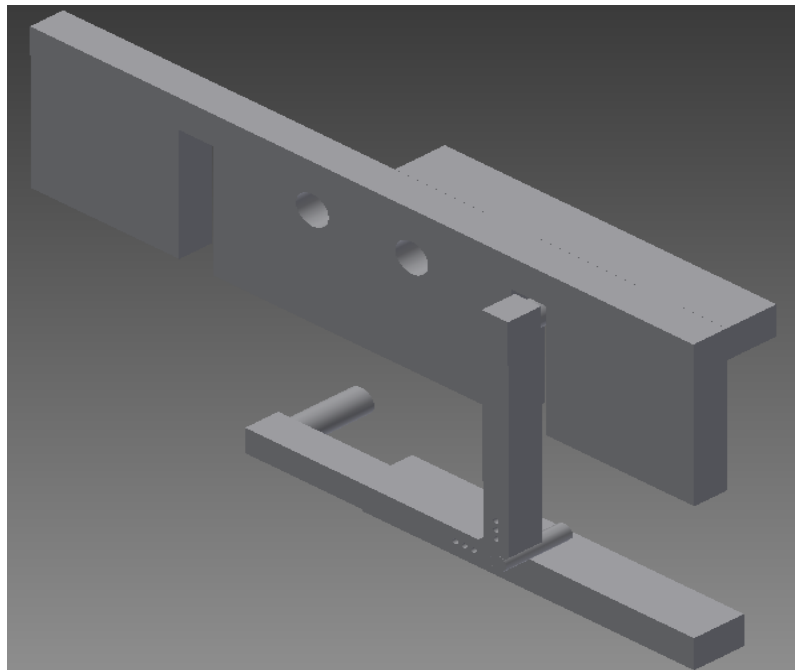


Figura 75. Simulación de mecanismo acelerador con el software Autodesk Inventor 2014.



Los planos del piñón y la cremallera se encuentran en la sección de anexos.

C. RESULTADOS

1. Diagramas de conexión y diagramas de flujo

Figura 76. Diagrama de conexión de sensores y actuadores a Olimexino 328.

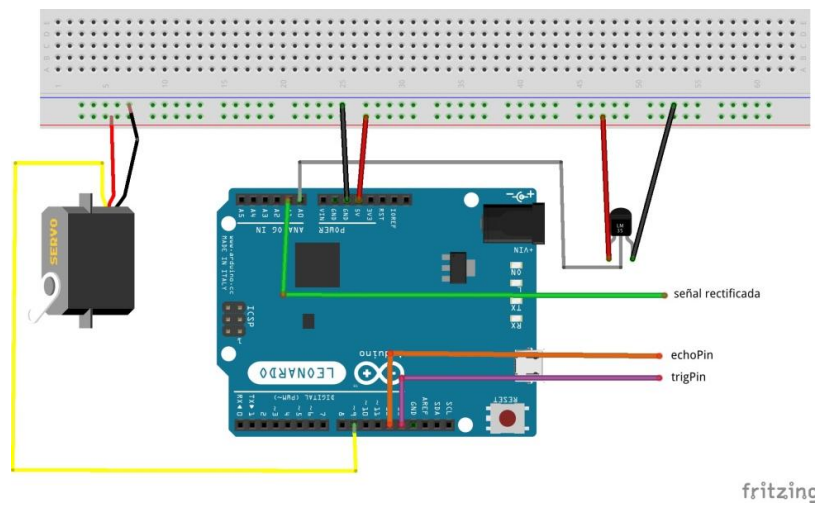


Figura 77. Diagrama de conexión del sistema completo.

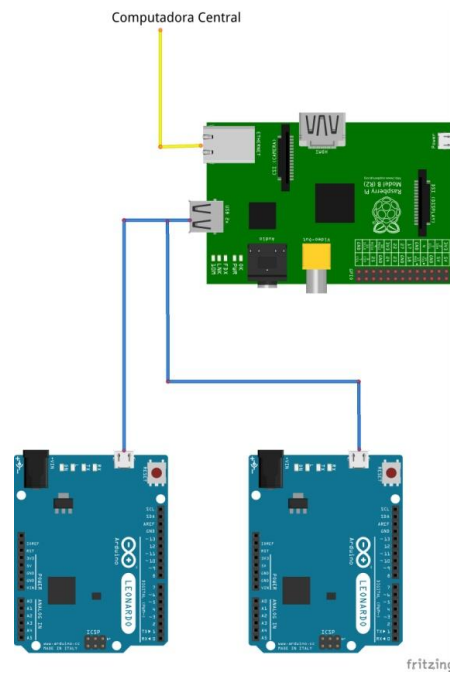


Figura 78. Diagrama de flujo de programa de medición y regulación de planta auxiliar

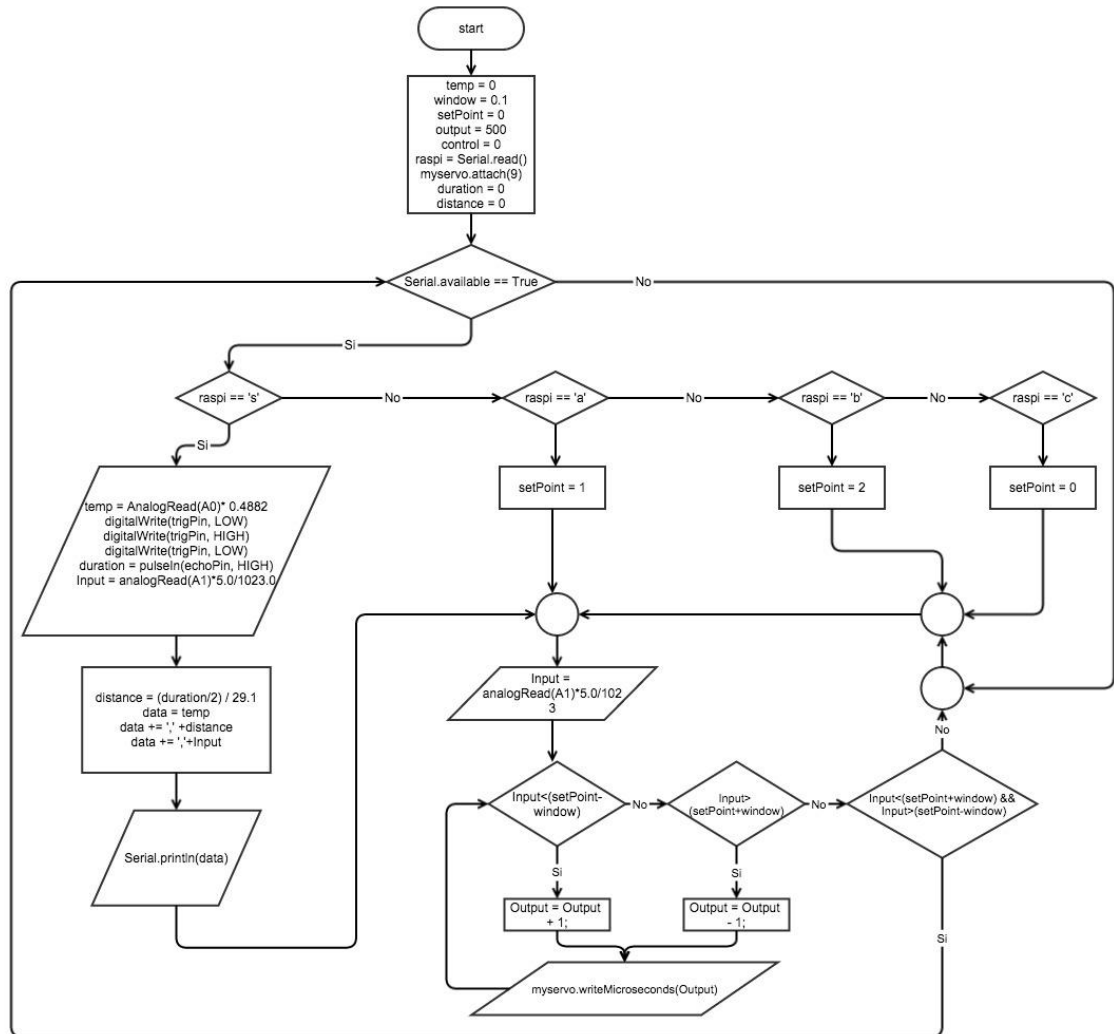


Figura 79. Diagrama de flujo de programa de HMI.

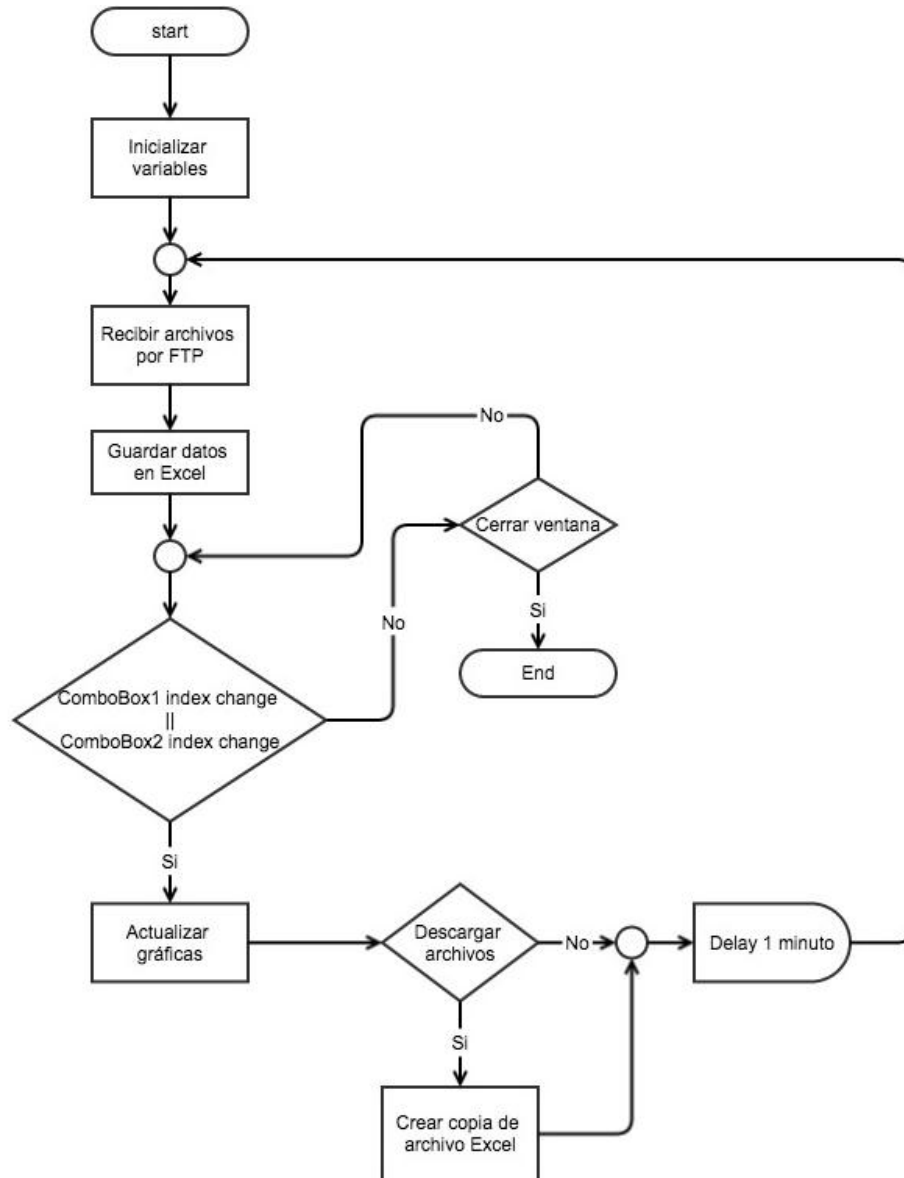


Figura 80. Diagrama de flujo de programa de recolección de datos en Raspberry Pi.

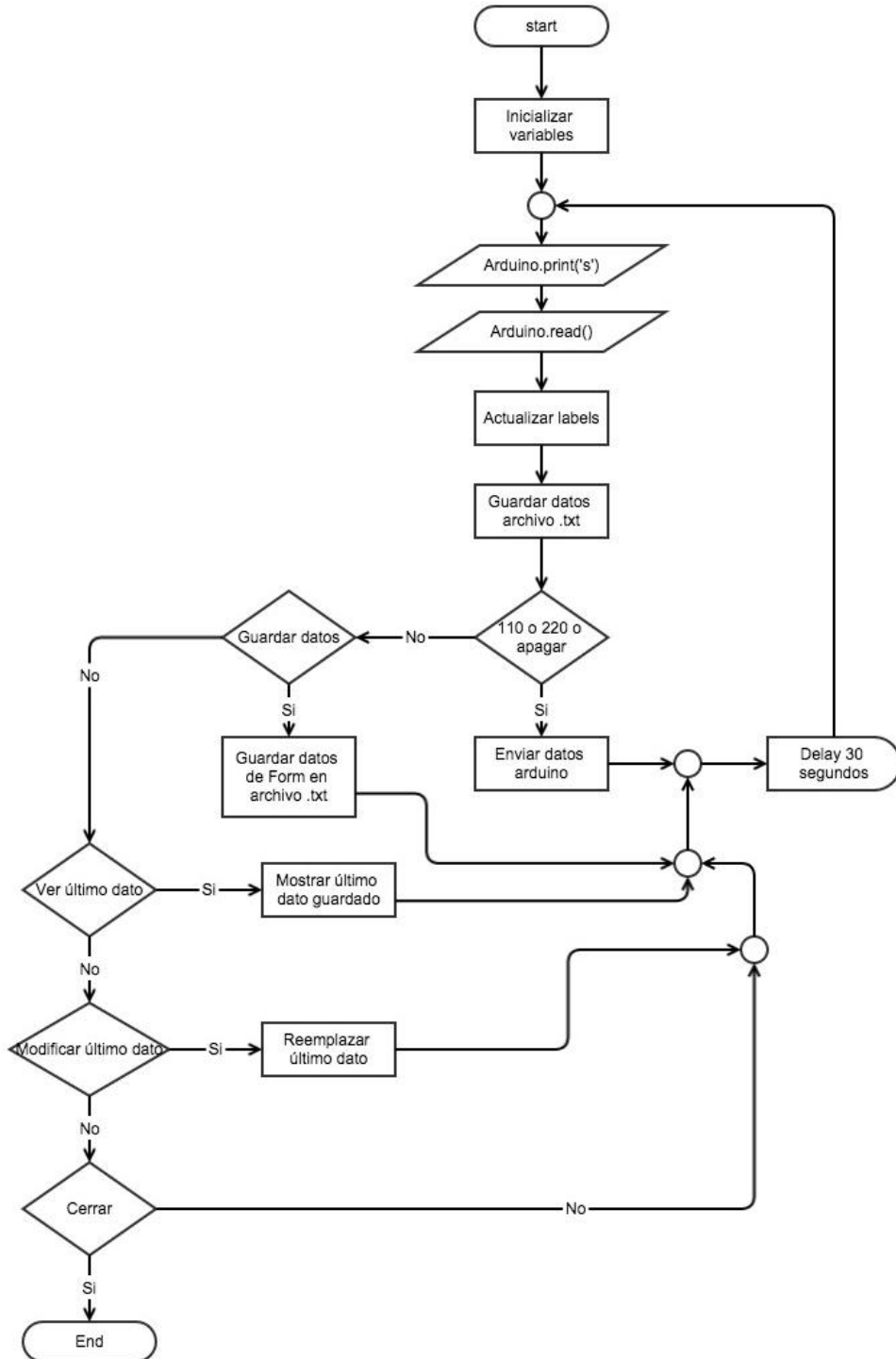
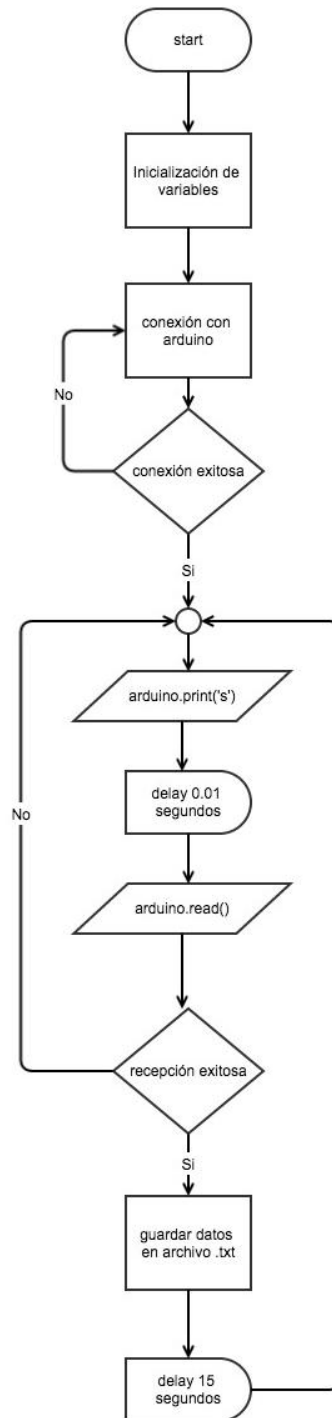


Figura 81. Diagrama de flujo de segundo programa de recolección de datos en Raspberry Pi.



2. HMI en raspberry pi

Figura 82. Sección de aplicación python para control de velocidad del motor diesel.

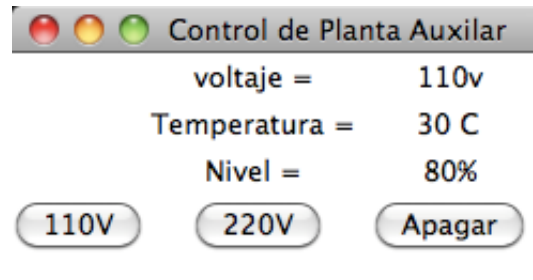


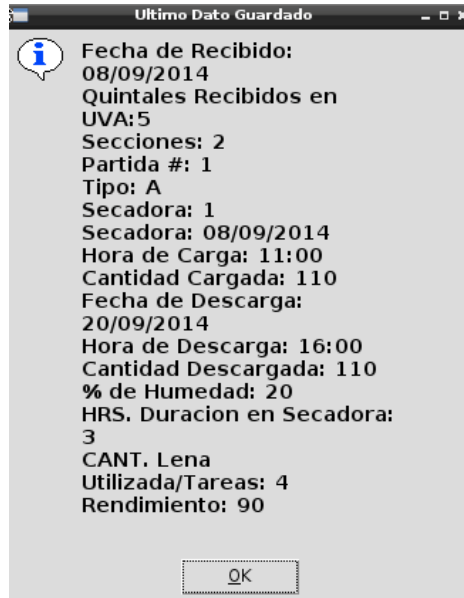
Figura 83. Aplicación python para monitoreo de secadoras estáticas y guardiolas.

The image shows a window titled "Finca La Fama" with a standard macOS-style title bar. The main content area displays a form with the following fields and values:

Fecha de Recibido:	08/09/2014
Quintales Recibidos en UVA:	5
Secciones:	2
Partida #:	1
Tipo:	A
Secadora:	1
Fecha de Carga:	08/09/2014
Hora de Carga:	11:00
Cantidad Cargada:	110
Fecha de Descarga:	10/09/2014
Hora de Descarga:	16:00
Cantidad Descargada:	110
% de Humedad:	20
HRS. Duracion en Secadora:	48
CANT. Lena Utilizada/Tareas:	4
Rendimiento:	90

At the bottom of the window are five buttons: "Guardar Datos", "Ver Ultimo Ingreso", "Modificar Ultimo Ingreso", "Control de planta auxiliar", and "Cerrar".

Figura 84. Sección de aplicación python para visualización de último ingreso de datos.



3. HMI en PC

Figura 85. Visualización de temperatura en generador diesel. Aplicación de visual c#.

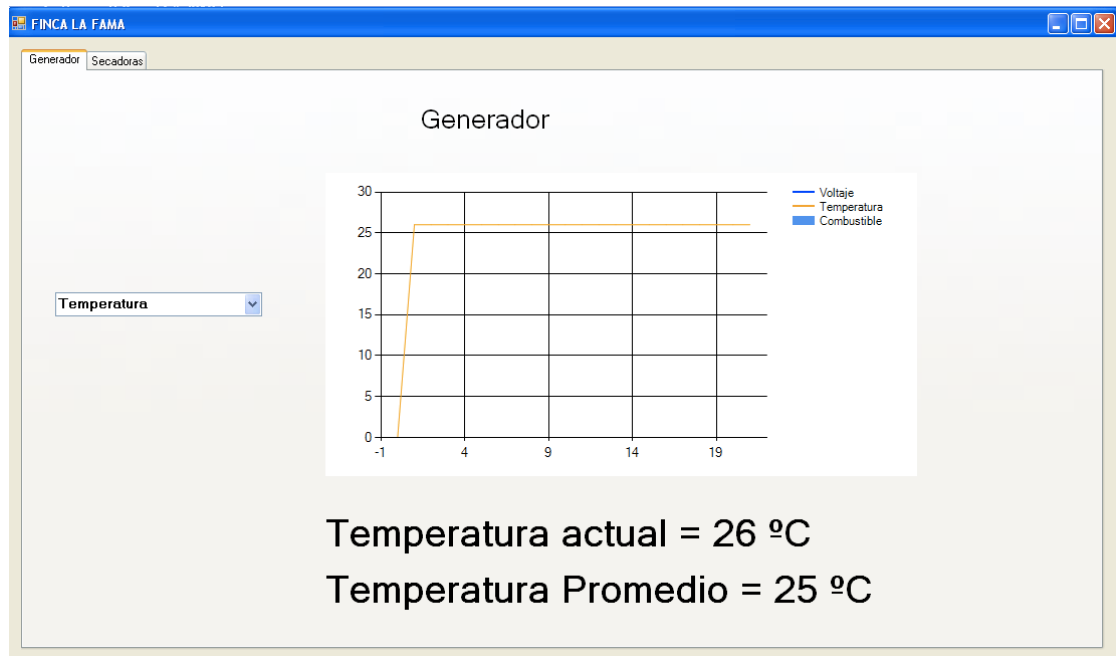


Figura 86. Visualización de voltaje en generador diesel. Aplicación de visual c#.

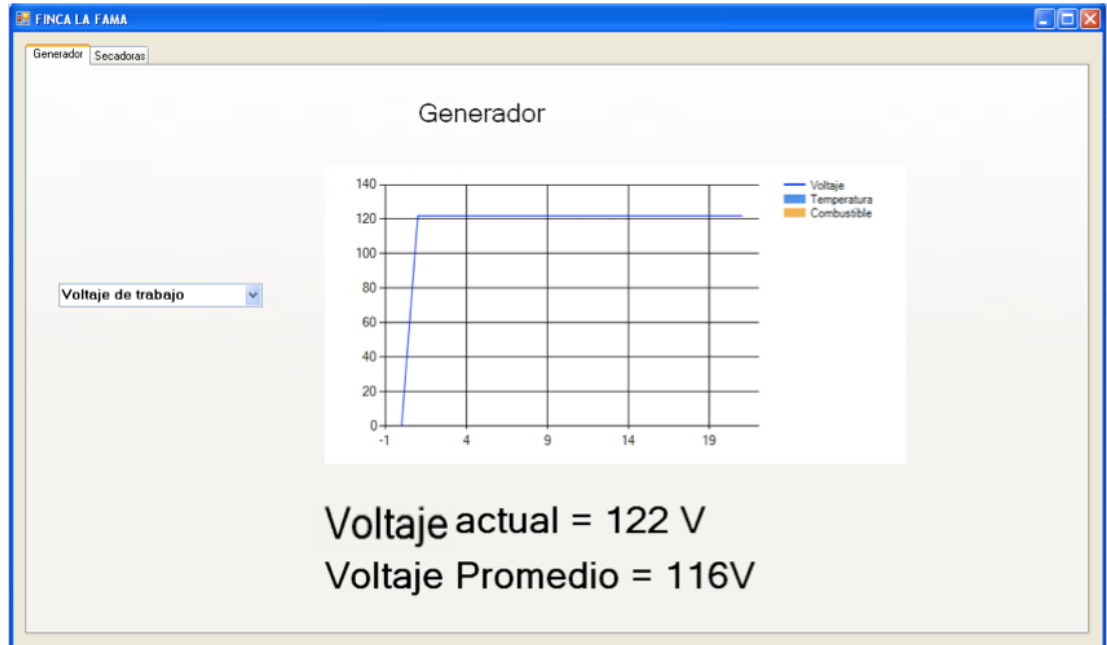


Figura 87. Visualización de nivel de combustible en generador diesel. Aplicación de visual c#.

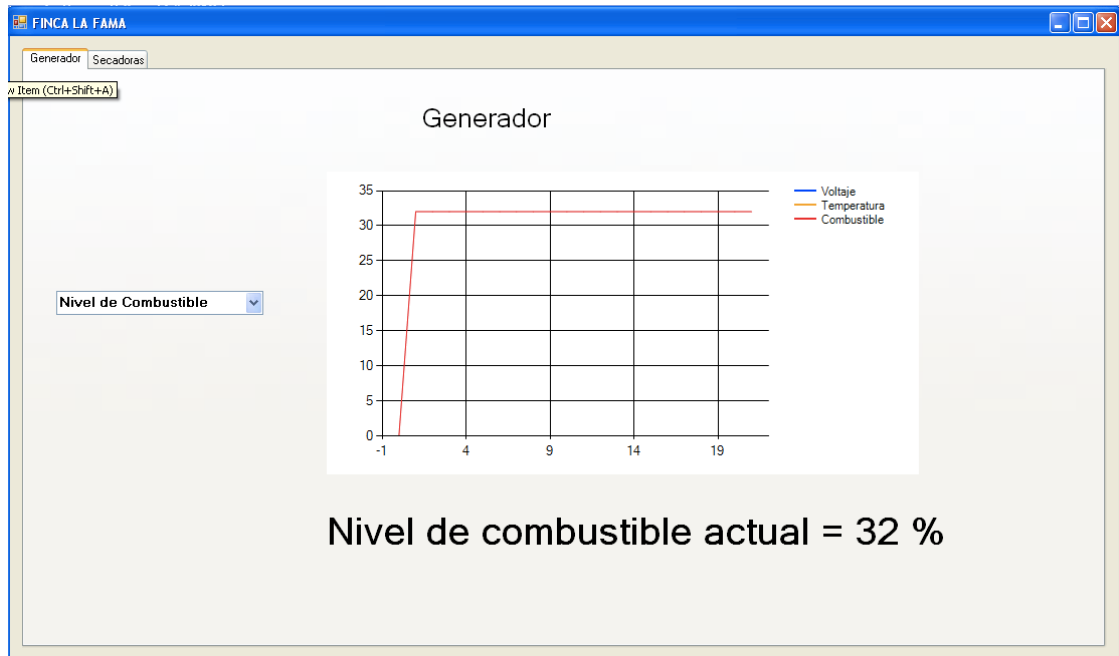


Figura 88. Visualización de datos de guardiola 1. Aplicación de visual c#.

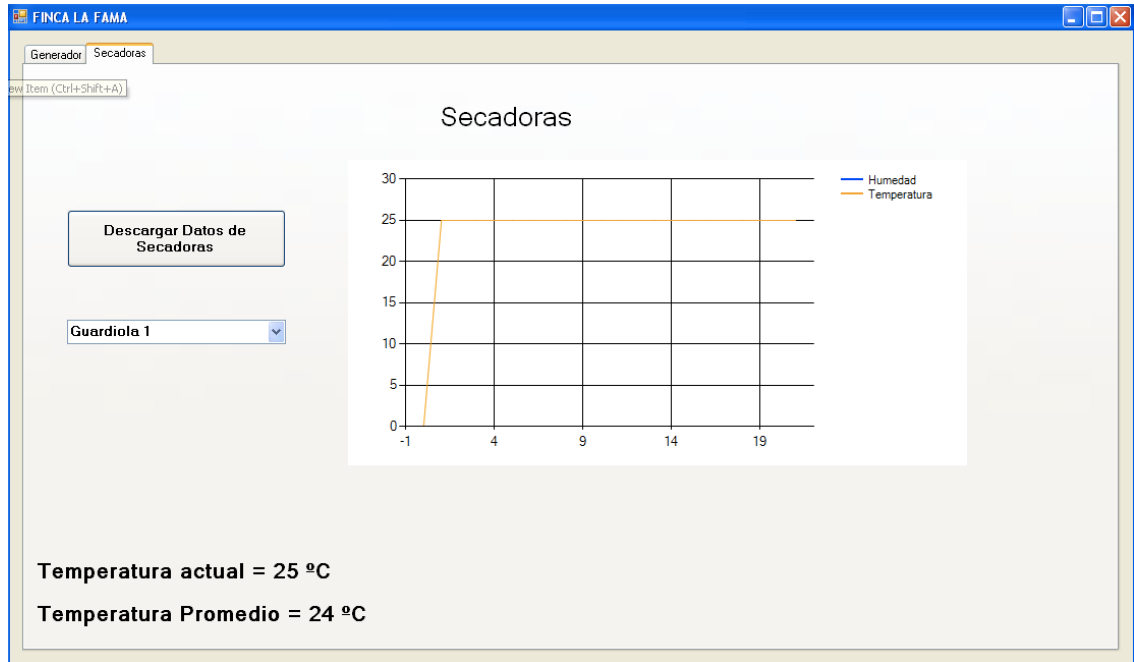


Figura 89. Visualización de datos de guardiola 2. Aplicación de visual c#.

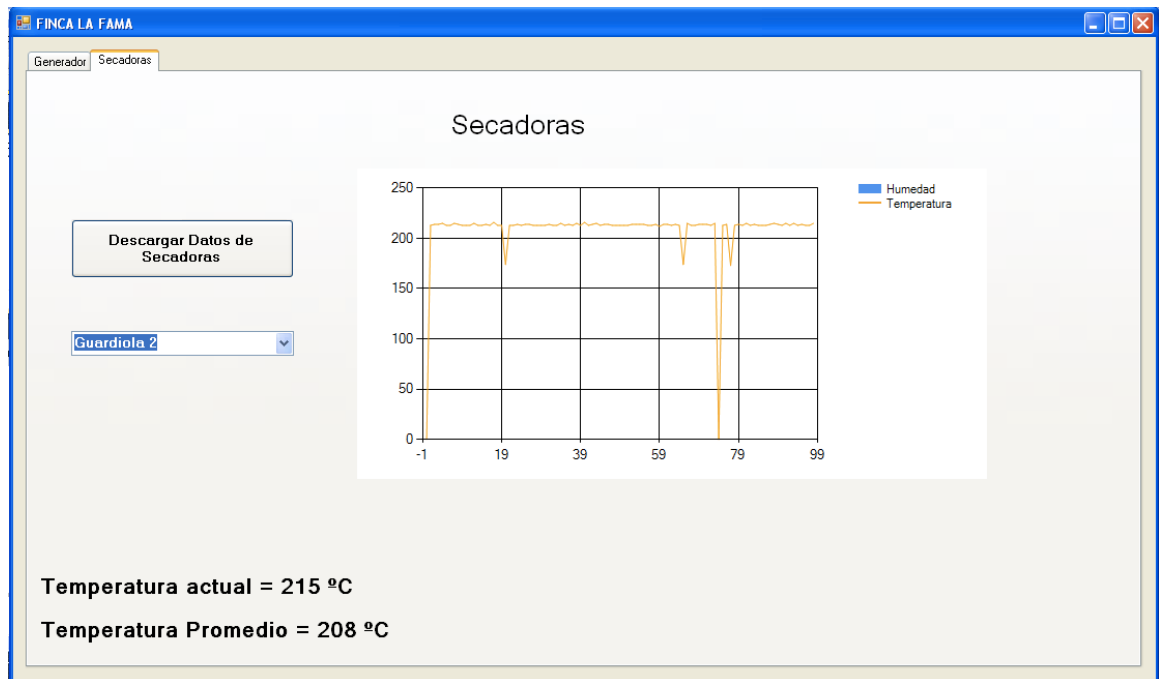


Figura 90. Visualización de datos de secadora estática 1. Aplicación de visual c#.

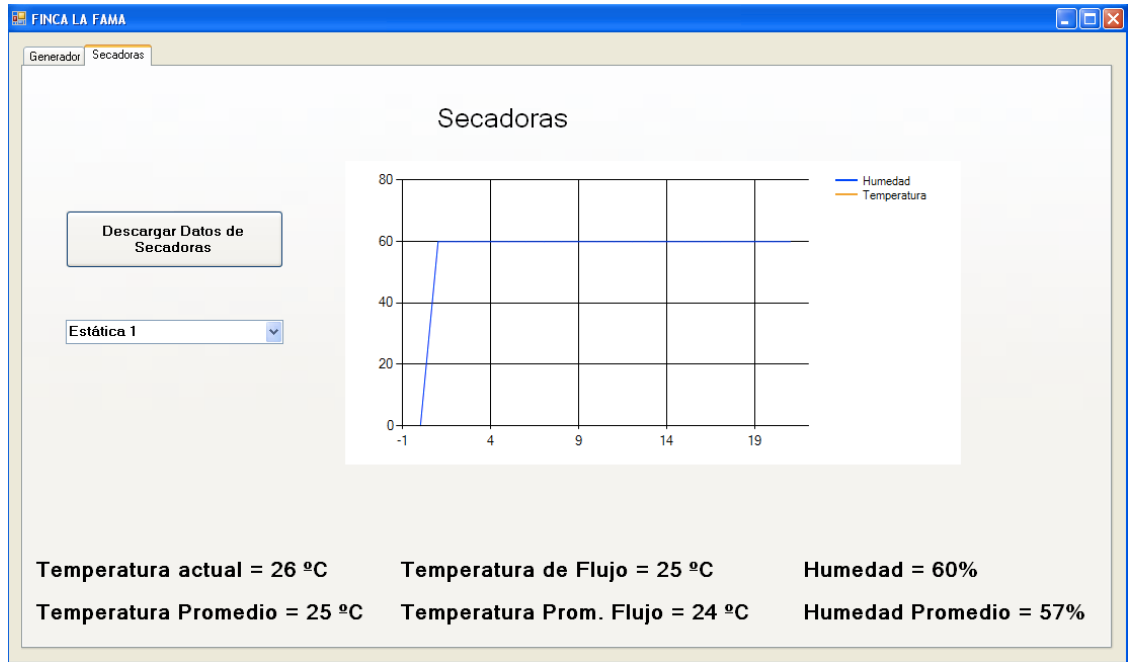
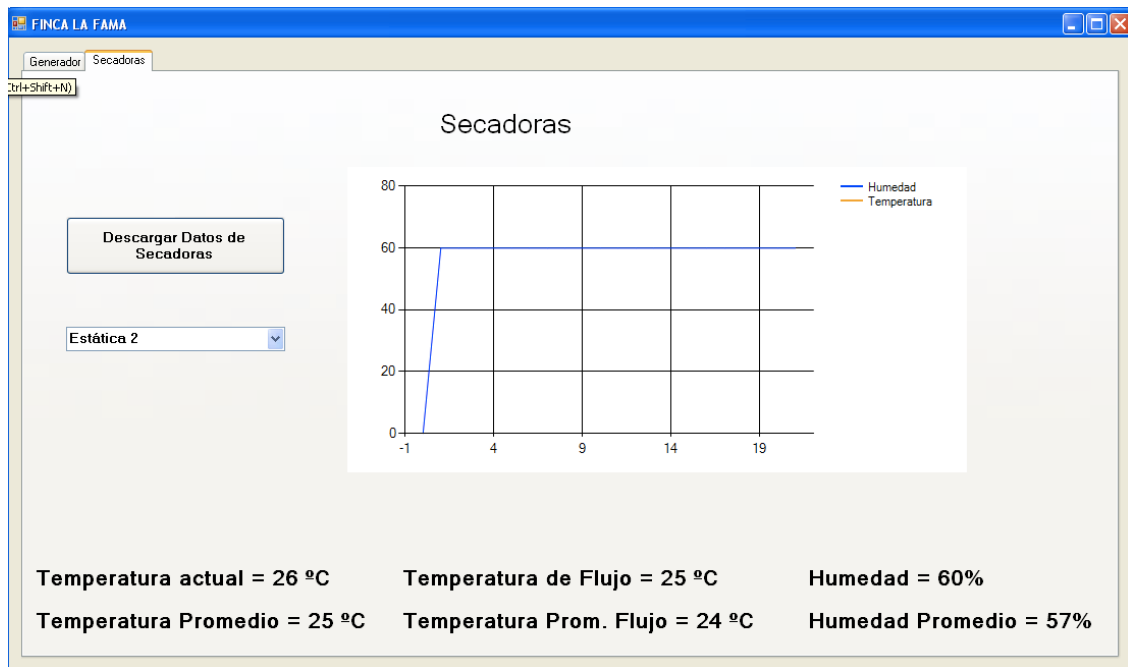


Figura 91. Visualización de datos de secadora estática 2. Aplicación de visual c#.



4. Mecanismo de control de generación de energía eléctrica

Figura 94. Circuito esquemático de rectificador de voltaje AC.

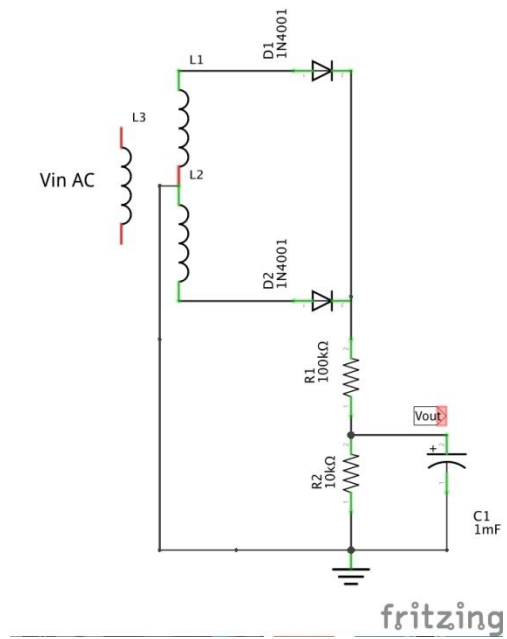


Figura 95. Circuito en placa de cobre de rectificador de voltaje AC.

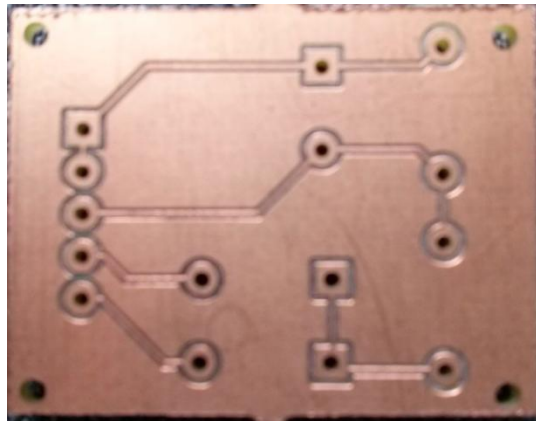


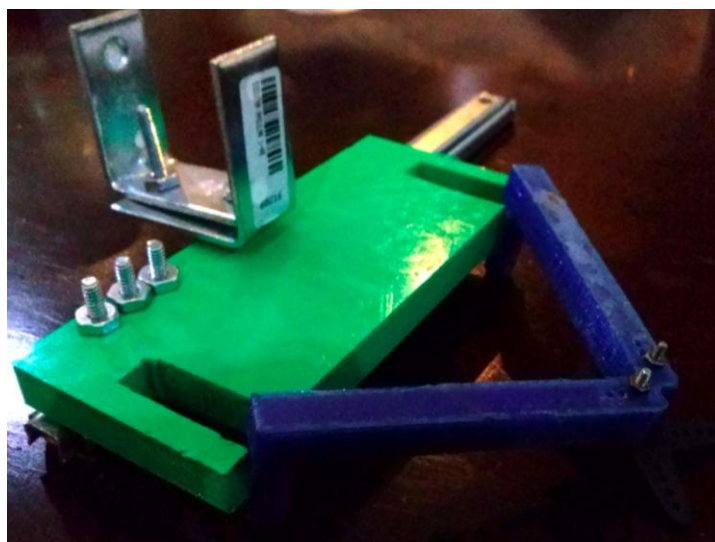
Figura 96. Circuito rectificador de voltaje AC soldado en placa de cobre.



Cuadro 11. Funcionamiento de mecanismo regulador de velocidad.

<i>Ángulo (grados)</i>	<i>Desplazamiento lineal(mm)</i>	<i>Voltaje generado (V)</i>
0	0	95
1	1.24	108.2
2	2.49	126.7
3	3.73	150.4
4	4.97	179.3
5	6.22	213.5
6	7.45	252.8
7	8.69	297.2
8	9.92	346.7
9	11.15	401.2
10	12.38	460.6

Figura 97. Mecanismo acelerador de motor diésel.



5. Página web para despliegue de datos

Figura 98. Despliegue de datos via web.

Finca La Fama x
fincalafama.hostoi.com

FINCA LA FAMA

En esta página se muestra información sobre el control de la finca La Fama

----- DATOS DE SECADORAS Y GENERADOR DIESEL -----

Secadora Estática 1	Secadora Estática 2	Guardiola 1
Temperatura actual: 10	Temperatura actual: 488	Temperatura actual: 493
Humedad: 11	Humedad: 19	Guardiola 2
		Temperatura actual: 213
Generador Diesel		
Voltaje: 4		
Temperatura: 7		
Nivel: 8		

6. Listado de materiales del sistema completo

Cuadro 12. Listado de materiales utilizados.

<i>Material</i>	<i>Sección</i>	<i>Cantidad</i>
Raspberry Pi	Recepción y visualización de datos en beneficio de café.	1
Olimexino 328	Recepción de datos y control de planta auxiliar.	1
Riel de 20 cm de largo	Mecanismo regulador de velocidad.	1
Servomotor	Mecanismo regulador de velocidad.	1
Acople de servomotor	Mecanismo regulador de velocidad.	1
Acople a riel	Mecanismo regulador de velocidad.	1
Sensor ultrasónico HC-SR04	Medición de nivel de combustible en planta auxiliar.	1
Sensor de temperatura Lm35	Medición de temperatura en motor diésel de planta auxiliar.	1
Computadora con sistema operativo Windows	Despliegue de datos central.	1

D. ANÁLISIS DE RESULTADOS

En la Figura 194, en la sección de anexos se muestra el diagrama de flujo de los datos medidos para el proceso de secado de café. El diseño de un control completo de una planta generadora implica el desarrollo de control del arranque, presión interna, control de temperatura y otros factores que no fueron considerados en este trabajo de graduación. Las mediciones realizadas incluyen temperatura del motor, nivel de combustible y nivel de voltaje generado. La medición de temperatura se realizó con el sensor LM35 el cual genera un voltaje lineal y proporcional a la temperatura medida en escala Celsius. Al recibir este dato el Olimexino-328 simplemente lo multiplica por un factor definido para poder obtener el valor real de temperatura de la superficie del motor. Los datos medidos por este sensor son recolectados por un Olimexino-328 que lee el voltaje analógico de salida de este sensor cuando es necesario. Como se observa en la Figura 78 este Olimexino-328 es responsable de realizar mediciones de temperatura del motor, nivel

de combustible del motor y regular el voltaje generado por la planta auxiliar. La conexión de este módulo se puede observar en la Figura 76.

La medición de nivel es realizada con el sensor ultrasónico HC-SR04 el cual se alimenta con 5v DC y recibe un pulso TTL de 10 μ segundos en el pin llamado “trigger” el cual le indica al módulo que debe iniciar la medición. El módulo envía al Olimexino-328 el tiempo que se tardaron los pulsos en regresar al sensor, con este dato se obtiene la distancia de la superficie medida respecto a la posición del módulo ultrasónico solamente multiplicándolo por un factor que incluye la velocidad a la que viaja el sonido a través del aire y el factor de conversión para obtener un resultado en escala de centímetros. Con este dato y conociendo la altura del tanque de combustible se obtiene el porcentaje de combustible existente. El módulo ultrasónico trabaja con una precisión de 3mm por lo que el error obtenido en la medición puede ser despreciado y debido a la aplicación que se está implementando un error de 3mm no es crítico.

El Olimexino-328 responde a los comandos enviados desde la Raspberry pi, como se muestra en la Figura 77, desde esta se puede elegir el nivel de voltaje que se desea generar en la planta auxiliar y se reciben los datos medidos para ser desplegados en un monitor. La regulación de voltaje generado se realiza midiendo el voltaje en una de las tres fases del generador eléctrico utilizando el circuito rectificador y atenuador que se muestra en las Figuras 94-96. Este voltaje es leído por el Olimexino-328 y es comparado con la referencia definida por los comandos recibidos desde la Raspberry pi. Si el voltaje medido es superior al valor de referencia el servomotor gira en la dirección de desaceleración del motor y si el voltaje medido es inferior al voltaje de referencia el servomotor gira en la dirección de aceleración. En la Figura 78 se muestra el diagrama de flujo del programa que realiza este control y las mediciones en la planta auxiliar.

Como se observa en el Cuadro 11 y Figuras 73 y 74 el servomotor debe girar aproximadamente cinco grados para llegar a generar 220v por lo que en el código del Olimexino-328 se utilizó la función `writeMicroseconds()` que incluye la librería de servomotores de Arduino. Esta función permite generar movimientos en el servomotor tan pequeños como 0.18 grados en ambas direcciones. Este es el incremento o decremento que se realiza en el eje del servomotor hasta llegar a que la señal medida se encuentre en un rango de $\pm 5\%$. Los datos mostrados en el Cuadro 11 fueron obtenidos realizando pruebas de campo con el motor diésel de la finca La Fama. Debido a que la aplicación de regulación no es parte de un proceso crítico en la finca, ya que el motor debe ser arrancado de forma manual y no se necesita alta velocidad de respuesta, no se implementó un sistema de control del tipo PID. Este regulador mantiene la señal de salida dentro del rango deseado. Si se desea disminuir el tiempo de respuesta de la señal controlada o eliminar por completo las oscilaciones se recomienda utilizar un control más robusto.

El mecanismo diseñado para convertir el movimiento angular del servomotor en un movimiento lineal se muestra en la Figura 75 y 97. Los datos mostrados en la Figura 74 son mediciones obtenidas con la simulación del mecanismo realizada con el software Autodesk Inventor 2014 para poder tener una referencia del comportamiento final que se obtendrá si el sistema de regulación se instala en la finca La Fama. Dicho mecanismo debe ser colocado sobre un riel para asegurar un movimiento lineal en el eje deseado y sobre una plataforma a tres centímetros abajo del nivel del acelerador del motor. Los planos de este mecanismo se encuentran en la sección de anexos de este trabajo de graduación.

El diagrama de flujo mostrado en la Figura 79 muestra de forma general los procesos realizados por el HMI que muestra gráficas al usuario de los datos medidos por los sensores de las secadoras y la planta auxiliar de generación. Esta interfaz inicialmente se diseñó para ser corrida en la Raspberry pi utilizando Python pero al realizar las pruebas se encontró que el ambiente de la Raspberry pi no soporta correr aplicaciones con gráficos robustos con actualización constante. El programa del HMI realiza una conexión utilizando el protocolo FTP con el servidor de la Raspberry pi para obtener una copia de los archivos donde se almacenan los datos medidos por los dos Olimexinos-328 y los datos importantes del proceso de secado. Al tener una copia de estos archivos el programa del HMI los tabula en un archivo Excel desde el cual se leen los datos cada vez que se refresquen las gráficas que se muestran en pantalla.

Cada minuto se realiza la petición FTP tanto al servidor ubicado en la Raspberry pi como al servidor virtual donde está montada la página web. El HMI genera una copia de sus archivos para guardarlos en el servidor virtual para que el código html de la página web realice la lectura de este cada minuto. Esta página web fue desarrollada de forma simple para mostrar solamente los datos más importantes de las mediciones. El código html realiza una conexión en un archivo css que le permite desplegar la información con un diseño minimalista. Este archivo css forma parte de una plantilla para el desarrollo libre de páginas web para principiantes. El diseño final de la página web se observa en la Figura 98.

En las Figuras 85-91 se muestran las diferentes secciones de la HMI desarrollada en lenguaje C# para sistemas operativos Windows. Estas figuras muestran las mediciones que se realizaron en la visita a la finca La Fama. En las Figuras 92 y 93 se observan los datos, obtenidos por medio de la petición FTP, ya tabulados. El archivo Excel mostrado en la Figura 93 es para uso exclusivo del código del programa. El archivo Excel mostrado en la Figura 92 es una copia del archivo recibido desde la Raspberry pi que se genera cuando el usuario presiona el botón de “Descargar datos de secadoras” que se muestra en las Figuras 88-91.

En la Raspberry pi se ejecutan simultáneamente dos programas en lenguaje Python. El diagrama de flujo de uno de ellos se muestra en la Figura 80. Este programa permite a los colaboradores de la finca

ingresar los datos de los procesos de secado realizados. El programa permite guardar los datos, revisar el último set de datos guardados y si se cometió un error al ingresar los últimos datos se pueden reemplazar con simplemente presionar un botón. Estos son los datos que luego son agregados al Excel que se muestra en la Figura 92. El programa también permite tener acceso a los datos medidos en la planta auxiliar. Estos datos se actualizan cada 30 segundos. La interfaz gráfica de este programa se puede apreciar en las Figuras 82-84.

El segundo programa que se ejecuta en la Raspberry pi no posee interfaz gráfica. Este se corre de forma continua desde el arranque de la computadora hasta que está se apaga. La función de este programa es recolectar los datos obtenidos por el Olimexino-328 utilizado en otro módulo de este megaproyecto y almacenarlos en un archivo de texto para que puedan ser enviados vía FTP a la computadora central. El diagrama de flujo de este programa desarrollado en lenguaje Python se muestra en la Figura 81.

E. CONCLUSIONES

1. Se logró diseñar un sistema regulador de voltaje generado de fácil instalación
2. Se logró diseñar un sistema de monitoreo de temperatura y nivel de combustible para el motor diésel, para su despliegue en Raspberry pi.
3. Se logró la recolección de datos y cálculo de estadísticas de los sistemas desarrollados en el megaproyecto, en tres diferentes plataformas: Raspberry pi, Visual Studio en sistema operativo Windows y via web.

F. RECOMENDACIONES

- El uso de dispositivos, originalmente diseñados para laboratorio, en ambientes industriales no es recomendado. En aplicaciones simples como la que se realizó en este trabajo de graduación presentan buen funcionamiento pero se recomienda el uso de PLCs para implementar sistemas más robustos.
- Como segunda fase de este proyecto se recomienda implementar un control completo de la planta auxiliar de generación. Controlando el arranque del motor de forma remota e implementando un sistema de control con mayor estabilidad.
- Si se desea implementar PLCs en este sistema se recomienda utilizar sensores y actuadores especialmente diseñados para estos ambientes.

- Se recomienda instalar el sistema presentado en este trabajo de graduación en la finca La Fama debido que representa una solución efectiva y de bajo costo para una automatización menor.
- Para aplicaciones con interfaz gráfica y recolección de datos el uso de una Raspberry pi es recomendado, siempre que la aplicación no necesite una interfaz gráfica robusta.
- Visual Studio es una herramienta adecuada para el desarrollo de de HMIs tanto industriales como domésticos.
- El uso de Olimexino-328 en ambientes industriales es indicado para aplicaciones simples.

VII. MECANISMO PARA EL SECADO HOMOGÉNEO DEL GRANO DE CAFÉ DE LA FINCA “LA FAMA”

A. METODOLOGÍA

Primero que se realizó fue conocer el proceso actual que se utiliza para homogenizar la temperatura en el grano de café. Para ellos, se visitó la finca, en donde se nos explicó cómo es el proceso del café, en qué fechas se tiene cosecha y producción. Se conoció toda la planta y las etapas por las que pasa el grano del café hasta que son metidos en costales para ser transportados a otras industrias de procesamiento del grano y de exportación. Se conoció el instrumento con el que movilizan el grano para poder tomar una idea de cómo podría ser el mecanismo a diseñar y se nos dio una pequeña explicación de qué manera se realizaba y con qué frecuencia se realizaba dicho movimiento. Se realizaron mediciones de las pilas y se consultó sobre la cantidad de grano que se moviliza

Figura 99: Pilas donde se coloca el café



Figura 100: Instrumento utilizado para movilizar el café



Luego se hizo una investigación sobre los tipos de mezcladoras. En esta parte se tuvo que tomar como referencia otros tipos de mezcladoras de granos o sólidos para poder conocer cuál sería un diseño apropiado para el mecanismo. Por esta razón, se investigó con detalle cada uno de estos tipos y el movimiento que realizan para verificar si estos dañarían el grano.

Asimismo se investigó de los materiales apropiados para la elaboración del mecanismo. Para poder decidir qué material era el apropiado se investigó el material con el que están elaboradas algunas mezcladoras, y se tomaron en cuenta varios factores a los que estaría sometido el mecanismo. En este caso, se tomó en cuenta el ambiente en el que se está trabajando y la higiene que debe tener por estar trabajando con alimentos.

Se procedió al diseño de la estructura mecánica para sustituir el proceso actual de la finca. Para empezar éste se tomaron como referencia varios diseños como los que se observan en la investigación previa. Para esta parte se tuvo que realizar una secuencia de pasos, para obtener el diseño deseado.

Además, se realizaron los respectivos cálculos de los esfuerzos para los rodamientos del eje del mecanismo. Se tuvo que calcular las fuerzas a las que estará sometido el eje para poder seleccionar el rodamiento adecuado, capaz de soportar tantos esfuerzos radiales, como axiales. Luego de realizar los cálculos de las fuerzas ejercidas por las fuerzas de las partes del mecanismo, se tuvo que tomar la decisión de modificar el eje o de cotizar rodamientos que se acoplaran al eje que se había diseñado. Se tomó la segunda opción, solamente se investigó sobre rodamientos comerciales en el mercado guatemalteco para verificar si el diámetro que se había elegido coincidía con las especificaciones y características del rodamiento.

Luego se prosiguió al diseño de engranajes para el sistema de potencia mecánica. Para el diseño de los engranes que forman parte de la caja reductora se tuvo que realizar varios cálculos con relación al motor

disponible. Además, se tuvo que realizar el diseño de los ejes, tomando en cuenta las dimensiones adecuadas y el espacio disponible.

Por último, se realizó el cálculo de la fuerza necesaria para movilizar el grano de café. Para esta parte se tomó en cuenta la reducción de potencia que se diseñó y la velocidad con la que estaría trabajando la estructura mecánica. Para poder tener una mejor visualización del diseño se elaboró un prototipo, el cual era un modelo a escala del mecanismo. En él se podría verificar el funcionamiento que llevaría a cabo en las secadoras estáticas.

B. DISEÑO

1. DISEÑO DE LA ESTRUCTURA MECÁNICA. Para el diseño de la estructura mecánica se tuvo que tomar decisiones con respecto al diseño más apropiado para la función que se necesita en la finta. A continuación se dará una mayor explicación de éstas decisiones y alguno de los cálculos realizados:

a. Aplicar los pasos del proceso de diseño: para esta parte se tuvo que tomar una decisión de qué era lo que se iba a realizar y cuál era realmente la necesidad que tiene el dueño de la finca. También se tomó en cuenta las funciones, requerimientos operaciones, restricciones y especificaciones del mecanismo. Como se muestra a continuación:

1) Identificar necesidad. Evitar que una persona tenga que estar constantemente movilizándolo el café.

2) Definir el problema:

a) Establecer objetivos:

- Movilizar el grano de café, sin dañarlo.
- Homogenizar la temperatura del grano dentro de las cámaras de secado.
- Diseñar con un bajo costo, tomando en cuenta los materiales, la construcción e instalación del mecanismo.

b) Funciones:

- Mezclar/movilizar granos de café.
- Transmitir fuerza necesaria para movilizar una cantidad considerable de café sin dañarlo.
- Al momento de haberse encendido por el operador, poder trabajar sola sin necesidad de un recurso humano.

- Mostrar en pantalla o en la aplicación cuando se tiene alguna anomalía.

c) Requerimientos operacionales:

- Encenderse manualmente, por medio de un botón/palanca/aplicación.
- El mecanismo se desactivará sólo al momento de detectar que la humedad se llegó a la deseada.

d) Restricciones:

- Tiene que tener una dimensión de: 2450x1580mm como máximo.
- Fácil de utilizar, especialmente el control y monitoreo.
- Facilidad de mantenimiento y de desmontaje.
- Tiene que ser un mecanismo liviano, pero a vez lo suficientemente estable para poder lograr movilizar el grano de café.
- No debe de ser un mecanismo ruidoso.

e) Especificaciones:

- Temperatura de operación de 50-60°C.
- Costo de fabricación máxima de Q2,000.00
- Movilizar aproximadamente 10 quintales de café.

b. Evaluación de criterios de diseño y desempeño: para evitar preferencias personales y decisiones subjetivas se evaluaron varios criterios con los que debían de contar los mecanismos para que el funcionamiento fuera el deseado y para que realmente se hiciera una imitación del movimiento realizado con el instrumento actual. Para ello, se utilizó una Matriz de Pugh en donde se evalúan de manera numérica los criterios en cada uno de los diseños. Para la evaluación de los criterios se toma en cuenta un rango de puntuación dependiendo de la dificultad o el beneficio de la construcción del prototipo de -2 a 2, en donde:

- 2 = aspecto no favorable para la realización del prototipo
- 1 = aspecto no favorable, pero se puede buscar una solución
- 0 = sin relevancia (neutro)
- 1 = importancia
- 2 = mucha importancia.

En el siguiente cuadro se muestran los criterios a evaluar y el resultado final para cada uno de los diseños.

Cuadro 13: Matriz de Pugh para la evaluación del diseño y desempeño de los mecanismos elegidos

CRITERIOS DE DISEÑO Y DESEMPEÑO	PESO	CONCEPTO DE DISEÑO		PESO X CALIFICACIÓN	
		Diseño 1	Diseño 2	Diseño 1	Diseño 2
		(Paletas)	(Espiral)		
Mantenimiento	12	0	-1	0	-12
Facilidad de construcción	4	1	-2	4	-8
Costo	19	0	-1	0	-19
Falla/Resistencia	15	-1	0	-15	0
Funcionalidad/eficiencia	1	0	0	0	0
Durabilidad (vida útil)	2	-1	0	-2	0
Material	10	0	0	0	0
Higiénico	7	0	-1	0	-7
Desensamble	13	-1	-2	-13	-26
Grado de innovación	18	1	2	18	36
Movimiento correcto	3	-1	0	-3	0
Consumo de energía	8	0	0	0	0
Dimensionamiento apropiado	6	1	1	6	6
Facilidad de encontrar el porqué de alguna falla	16	1	1	16	16
Tiempo de implementación	11	-1	-2	-11	-22
Tiempo de construcción	5	0	-2	0	-10
Disponibilidad de recursos	17	0	0	0	0
Seguro	9	-1	0	-9	0
Facilidad de uso o arranque	14	-1	1	-14	14
TOTAL				<u>-23</u>	-32

2. CÁLCULO DE RODAMIENTOS PARA LOS EJES DE LA ESTRUCTURA MECÁNICA. Con el diseño de la estructura se realizaron cálculos para conocer cuál rodamiento sería el adecuado para colocar en los extremos del eje de la estructura mecánica. Con ayuda de Autodesk Inventor Profesional 2014 se pudo conocer la masa de cada una de las piezas del mecanismo para luego poder evaluar las cargas a las que estaría sometido el eje. Para este caso se evaluó el mecanismo de manera estática, es decir sin conocer las fuerzas que está aplicando el café. Una alternativa para el cálculo de la masa de las partes del mecanismo es por medio de la densidad del material utilizado (acero inoxidable) y el volumen de éstas.

Cuadro 14: Fuerzas ejercidas en los soportes del eje

DATOS OBTENIDOS POR INVENTOR				
	Masa		Fuerza	
<i>Paleta</i>	7.175	kg	70.38675	N
<i>Soporte paleta</i>	1.586	kg	15.55866	N
<i>Eje</i>	37.132	kg	364.26492	N

Cuadro 15: Fuerza total ejercida por cada una de las partes en el plano X-Y

FUERZA TOTAL EN EL PLANO X-Y	
<i>Paletas (8)</i>	563.09 N
<i>Soportes (8)</i>	124.47 N
<i>Eje</i>	2914.11936 N
<i>TOTAL</i>	3,601.68 N

Cuadro 16: Fuerza total ejercida por cada una de las partes en el plano Y-Z

FUERZA TOTAL EN EL PLANO Y-Z	
<i>Paletas (10)</i>	703.8675 N
<i>Soportes (10)</i>	155.59 N
<i>Eje</i>	3642.6492 N
<i>TOTAL</i>	4,502.10 N

A continuación se muestran algunos diagramas que se utilizaron para poder verificar si el diámetro del eje seleccionado era el correcto. Se analizó dos diferentes planos en donde eran ejercidos los pesos de las

paletas del mecanismo. Primero se muestran los diagramas para el plano X-Y, luego los diagramas del plano Y-Z.

Cada uno de los planos se muestra en el siguiente diagrama.

Figura 101: Representación de planos de trabajo para el análisis de fuerzas

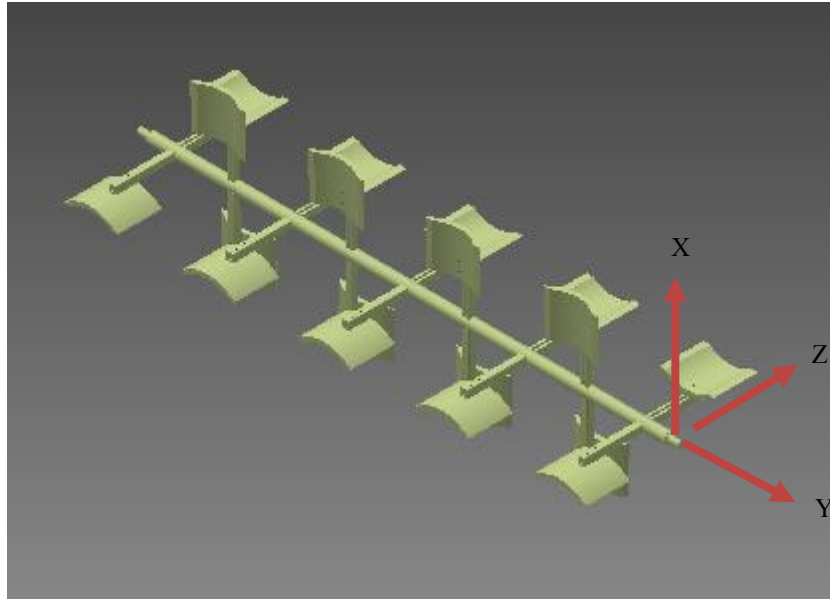


Figura 102: Diagrama de corte del eje en el plano X-Y

Cada una de estas representa a dos paletas

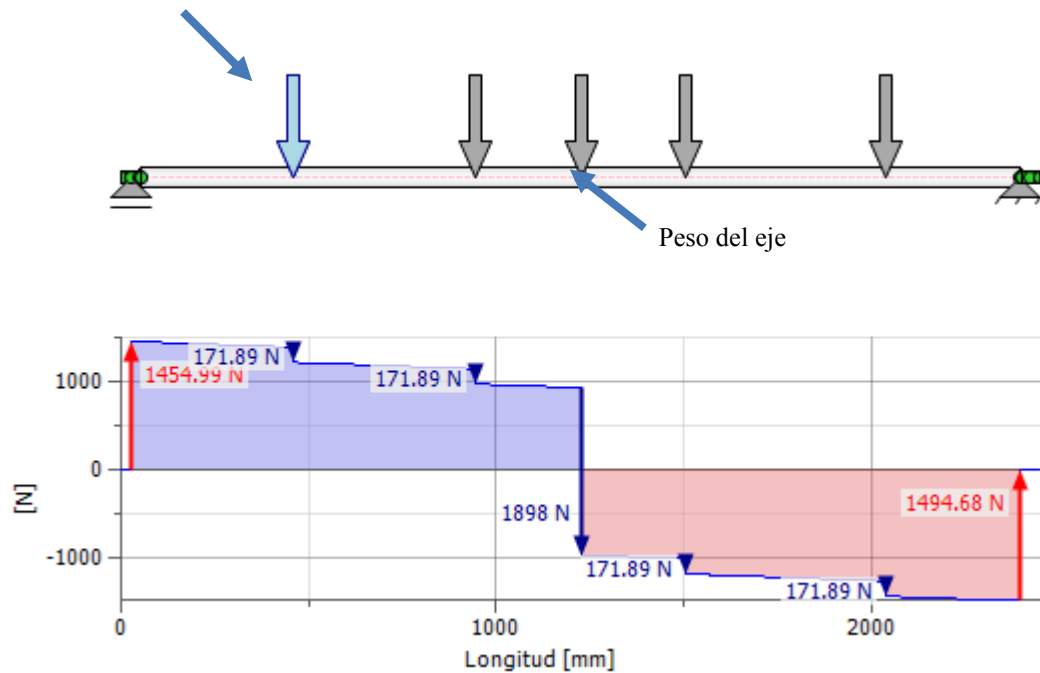


Figura 103: Momento flector del eje en el plano X-Y

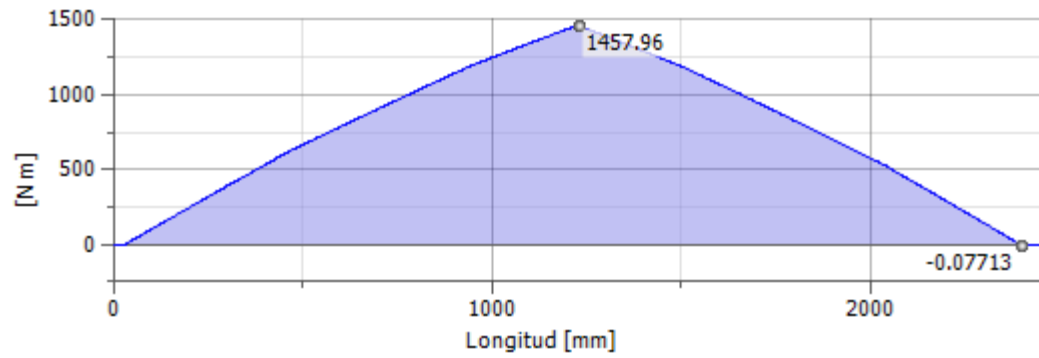


Figura 104: Ángulo de flexión del eje en el plano X-Y

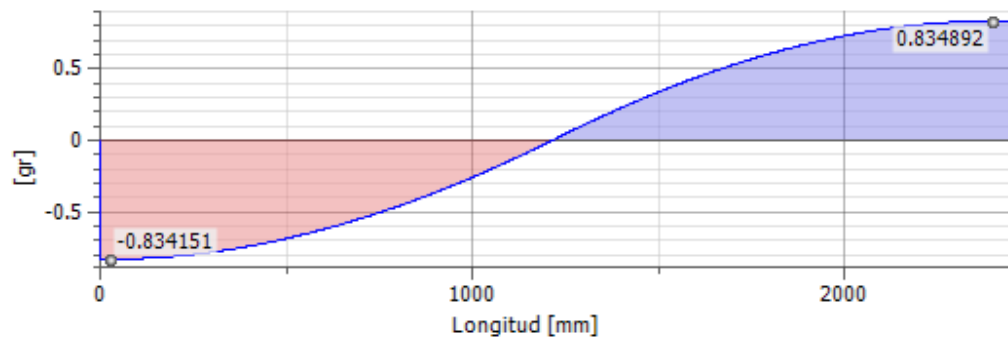


Figura 105: Flexión en el eje en el plano X-Y

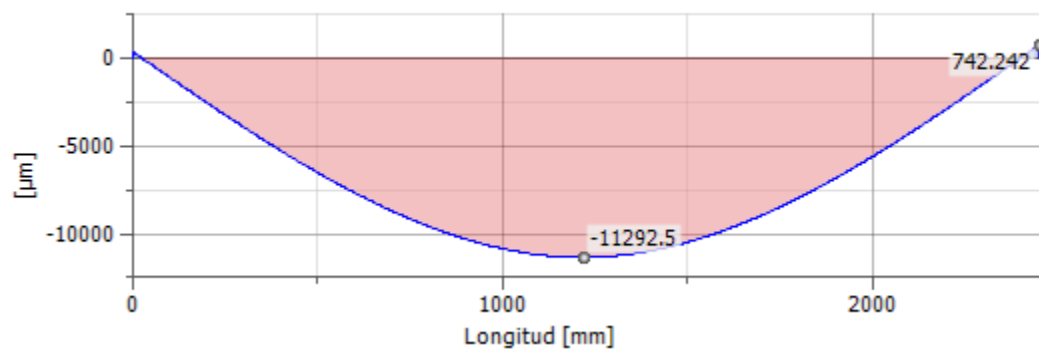


Figura 106: Diámetro ideal para el eje sugerido por Inventor para el plano X-Y

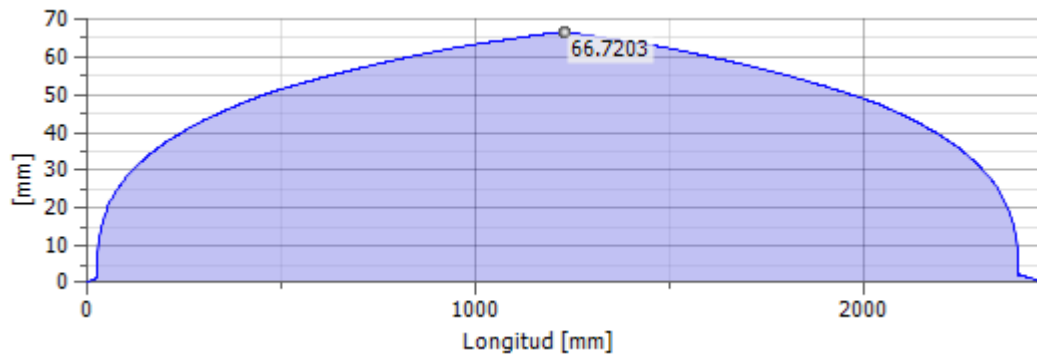
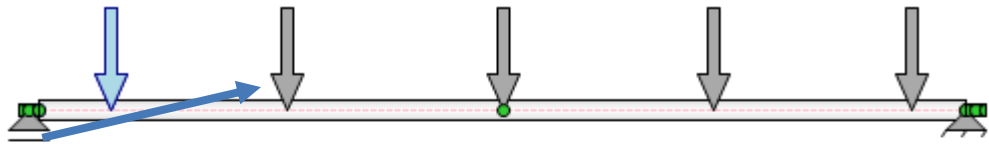


Figura 107: Diagrama de corte del eje en el plano Y-Z



Cada una de éstas representa a dos paletas

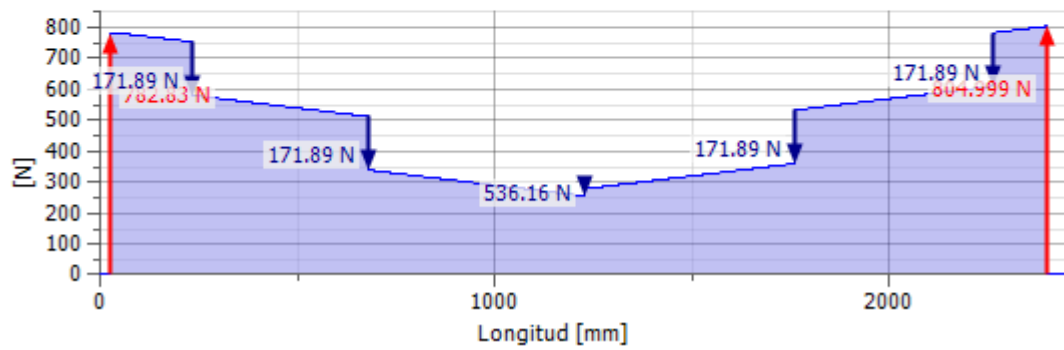


Figura 108: Momento flector del eje en el plano Y-Z

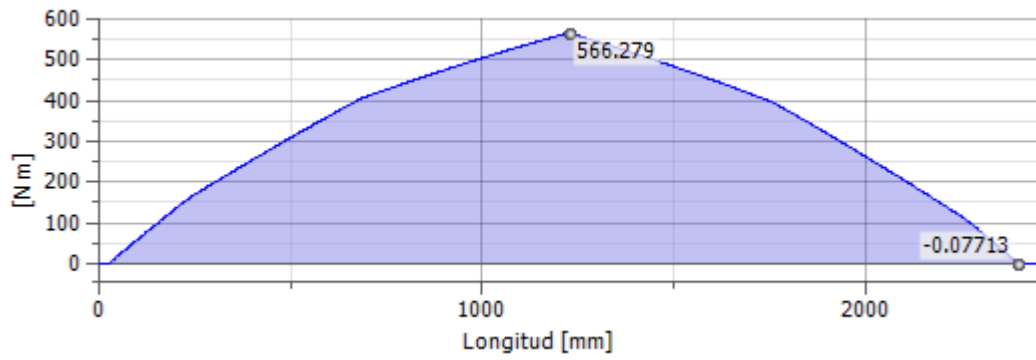


Figura 109: Ángulo de flexión del eje en el plano Y-Z

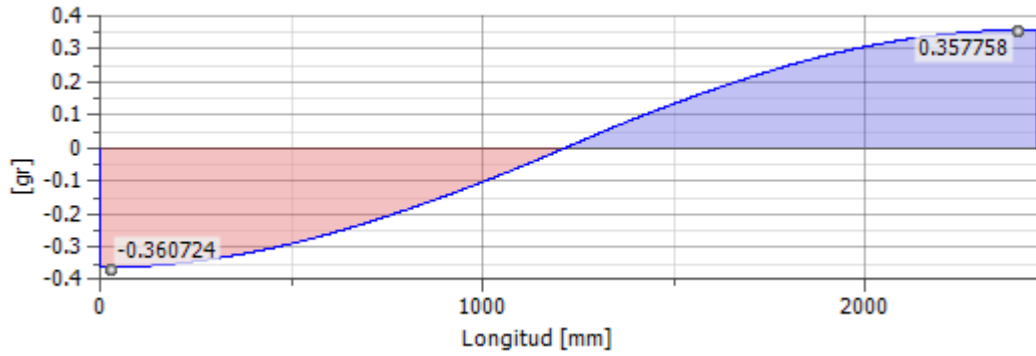


Figura 110: Flexión del eje en el plano Y-Z

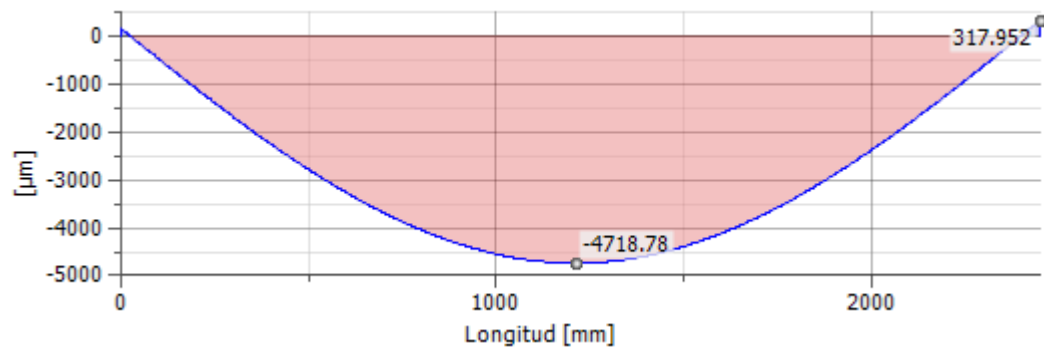
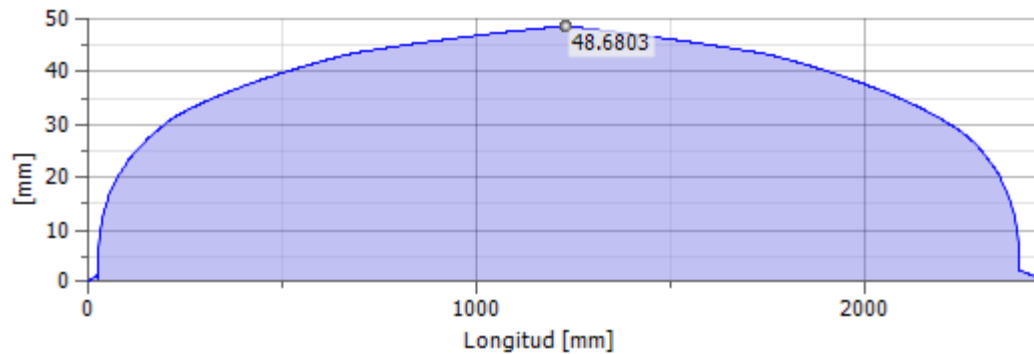


Figura 111: Diámetro ideal del eje sugerido por Inventor para el plano Y-Z



3. DISEÑO DE ENGRANES DEL SISTEMA DE POTENCIA MECÁNICO

a. Se investigó cuál es el motor con el que cuenta la finca y el más comercial para poder diseñar un sistema de reducción de potencia apropiado para movilizar el mecanismo.

Cuadro 17: Especificaciones del motor seleccionado

CARACTERÍSTICAS DEL MOTOR

<i>Potencia (H)</i>	7.5	HP
<i>Voltaje</i>	220	V
<i>Corriente</i>	2.2	A
<i>Velocidad</i>	1750	rpm

b. Se decidió la cantidad de reducciones que se diseñarían. Para este caso se tuvo que realizar dos reducciones, en donde las velocidades se observan en el siguiente cuadro:

Cuadro 18: Velocidades resultantes de la reducción del tren de engranes

VELOCIDADES DE LOS ENGRANES EN EL TREN

<i>Velocidad piñón 1 (nP)</i>	1750	rpm
<i>Velocidad corona 1 (nG)</i>	250	rpm
<i>Velocidad piñón 2 (nP2)</i>	250	rpm
<i>Velocidad corona 2 (nG2)</i>	35.71	rpm

c. Luego se tuvo que tomar en cuenta otras características para diseñar el tren de engranes. Se eligió el número de dientes para los piñones y las coronas, ancho de caras, distancia entre centros, entre otras características. El valor del tren se tomó en cuenta con las rpms que se tenían de entrada y el valor que se quería obtener en la salida, por lo que sólo se hizo una división.

Cuadro 19: Características para el diseño de engranes de la caja reductora

CARACTERÍSTICAS DEL TREN DE ENGRANES	
Valor del tren (e)	25
Número de dientes del piñón 1 (Na)	17
Número de dientes de la corona 1 (Nb)	119
Número de dientes del piñón 2 (Nc)	18
Número de dientes de la corona 2 (Nd)	126
Ángulo de presión (ϕ)	20 °
Módulo métrico	0.125 in
Distancia entre centros (reducción 1)	8.5 in
Distancia entre centros (reducción 2)	9 in
Ancho de cara (F)	1.5 in

d. Luego se realizaron cálculos de algunas características de reducción, es decir, las velocidades y cargas transmitidas de los elementos mecánicos de las dos reducciones. El paso diametral fue un criterio que se seleccionó, solamente por la cantidad de dientes con las que se eligieron. El factor para dientes se coloca con un valor de 1 cuando la profundidad es completa. Para el diámetro de paso, tanto de la corona como del piñón se calculó con la siguiente ecuación:

$$D_p = \frac{\text{Número de dientes}}{\text{Paso diametral}} \quad (\text{i})$$

e. Para el cálculo de las velocidades se utilizó la siguiente ecuación:

$$V = \frac{n\pi D_p}{12} \quad (\text{ii})$$

f. Se calculó la carga transmitida en cada una de las reducciones del tren de engranes, para ello se utilizó la siguiente ecuación:

$$W_t = \frac{33000 H}{V} \quad (\text{iii})$$

Cuadro 20: Características para el piñón y corona para la reducción en el tren de engranes

CARACTERÍSTICAS DE LAS REDUCCIONES EN EL TREN DE ENGRANES		
Paso diametral (P)	8	
Factor para dientes de profundidad completa (k)	1	
Diámetro de paso del piñón 1 (dP)	2.125	in
Diámetro de paso de la corona 1 (dG)	14.875	in
Diámetro de paso del piñón 2 (dP)	2.25	in
Diámetro de paso de la corona 2 (dG)	15.75	in
Velocidad del piñón 1 (VP)	973.566	ft/min
Velocidad de la corona 1 (VG)	973.566	ft/min
Carga transmitida por el piñón 1 (WtP)	254.220	Lb
Carga transmitida por la corona 1 (WtG)	254.220	Lb
Velocidad del piñón 2 (VP)	147.262	ft/min
Velocidad de la corona 2 (VG)	147.262	ft/min
Carga transmitida por el piñón 2 (WtP)	1680.676	Lb
Carga transmitida por la corona 2 (WtG)	1680.676	Lb

g. Se verificó si el diseño del engrane que se había realizado no contaba con interferencia en los dientes. En la siguiente tabla se muestra cómo se verifica ese dato.

Figura 112: Cálculo de interferencia en relación al número de dientes

Número máximo de dientes	
Para evitar interferencia entre un piñón de profundidad total con ángulo de presión de 20° y engranes de profundidad total de varios tamaños	
Número de dientes del piñón	Dientes de engrane máximos
17	1309
16	101
15	45
14	26
13	16

(Mott, 2006)

h. Tomando en cuenta las características de los engranes diseñados, se prosiguió calculando los esfuerzos a los que estarían sometidos estos. Se tuvo que tomar en cuenta el valor de varios factores, como de sobrecarga, dinámicos, de tamaño, espesor, entre otros.

1) El factor de sobrecarga (K_o) se tiene que considerar el incremento en la fuerza tangencial W_t , que pueden provocar las variaciones de carga, vibraciones, choques, cambios de velocidad, entre otros. Como lo que se está diseñando es una máquina mezcladora se considera que la fuente de potencia es uniforme debido a que es un motor eléctrico el que se estará utilizando y la máquina impulsada será considerada de tipo moderada, debido a que es una mezcladora de granos. Entonces con estas decisiones, $K_o = 1.25$.

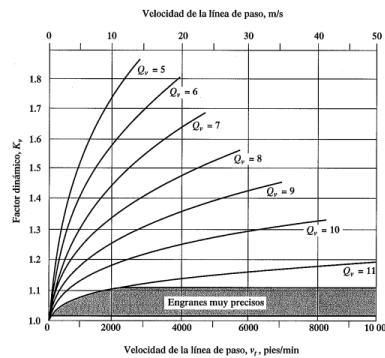
Figura 113: Tabla de factores de sobrecarga.

Máquina impulsada			
Fuente de potencia	Uniforme	Impacto moderado	Impacto pesado
Uniforme	1.00	1.25	1.75
Impacto ligero	1.25	1.50	2.00
Impacto medio	1.50	1.75	2.25

(Budynas, 2008)

i. El factor dinámico (K_v) se toma en cuenta para imprecisiones en la fabricación y acoplamiento de los dientes del engrane en movimiento. Primero se debe de tomar en cuenta el número de calidad de la máquina, ya que este será igual al número de nivel de exactitud en la transmisión (Q_v), esto se elige dependiendo de la velocidad a la que estará trabajando la máquina. En este caso, como la velocidad es de 1947.13 ft/min, entonces el número de calidad está dentro de un rango de 8-10; se eligió un valor de 10. Con estos valores ya se puede conocer el factor dinámico de la máquina por medio de una gráfica que se muestra en la Figura 28, entonces $K_v = 1.19$.

Figura 114: Factor dinámico



(Budynas, 2008)

1) El factor de tamaño (K_s) se calculó utilizando el ancho de cara del diente (F), el paso diametral (P) y el factor de forma de Lewis (Y). Como se muestra en la siguiente ecuación:

$$K_s = 1.192 \left(\frac{F\sqrt{Y}}{P} \right)^{0.0535} \quad (\text{iv})$$

2) El factor de alineación del acoplamiento (C_{ma}) por ser un motor con el que se está trabajando y una máquina mezcladora, entonces se tomará como unidades comerciales cerradas de engranes. Este factor es calculado para conocer el factor de distribución de carga. Para ello se utilizó la siguiente ecuación:

$$C_{ma} = 0.127 + 0.0158F - 1.093 * 10^{-4}F^2 \quad (\text{v})$$

3) El factor de proporción (C_p) se calculó utilizando la siguiente ecuación:

$$C_p = \frac{F}{10D_p} - 0.0375 + 0.0125F \quad (\text{vi})$$

4) El factor de distribución de carga (K_m) permite conocer la distribución no uniforme aplicada al engrane, para ello se calculó con la siguiente ecuación:

$$K_m = 1.0 + C_p + C_{ma} \quad (\text{vii})$$

5) El factor de Lewis se seleccionó utilizando la siguiente tabla, la cual depende del número de dientes del engrane:

Figura 115: Factor de forma de Lewis

Number of Teeth	Y	Number of Teeth	Y
12	0.245	28	0.353
13	0.261	30	0.359
14	0.277	34	0.371
15	0.290	38	0.384
16	0.296	43	0.397
17	0.303	50	0.409
18	0.309	60	0.422
19	0.314	75	0.435
20	0.322	100	0.447
21	0.328	150	0.460
22	0.331	300	0.472
24	0.337	400	0.480
26	0.346	Rack	0.485

(Budynas, 2008)

6) Y el factor geométrico de flexión es el mismo valor del factor de forma de Lewis, por esa razón en la tabla se muestra el mismo valor. Después de seleccionar todos los factores necesarios para el cálculo del esfuerzo real, se utilizó la siguiente ecuación:

$$\sigma = W_t K_o K_v K_s \frac{P_d K_m K_B}{F J} \quad (\text{viii})$$

En esta tabla se muestra el resumen de todos los factores calculados con las ecuaciones, gráficas y tablas que se mostraron previamente.

Cuadro 21: Factores para el cálculo del esfuerzo real de flexión

FACTORES PARA EL CÁLCULO DEL ESFUERZO DE FLEXIÓN		
Factor de sobrecarga (K _o)	1.25	
Factor dinámico (K _v)	1.19	
Factor de tamaño del piñón (K _{sP})	1.06	
Factor de tamaño de la corona (K _{sG})	1.07	
Factor de distribución de carga reducción 1 (K _m)	1.202	
Factor de distribución de carga reducción 2 (K _m)	1.198	
Factor de espesor del aro (K _b)	1	
Factor de forma de Lewis del piñón (Y _P)	0.303	
Factor de forma de Lewis de la corona (Y _G)	0.480	
Factor de proporción del piñón (C _{pf})	0.052	
Factor de alineación del acoplamiento (C _{ma})	0.150	
Factor geométrico de resistencia a la flexión del piñón (J _P)	0.303	
Factor geométrico de resistencia a la flexión de la corona (J _G)	0.480	
Número de calidad	10	
Esfuerzo real de flexión aplicado al piñón 1 (σ _p)	2243.94	psi
Esfuerzo real de flexión aplicado a la corona 1 (σ _G)	10038.21	psi
Esfuerzo real de flexión aplicado al piñón 2 (σ _p)	15656.38	psi
Esfuerzo real de flexión aplicado a la corona 2 (σ _G)	70038.25	psi

j. Después se prosiguió en conocer cuál sería el cálculo del esfuerzo de flexión permisible, por lo que se tuvo que tomar en cuenta otros factores.

1) El factor de confiabilidad (K_r) se eligió utilizando la tabla que se muestra a continuación. Es decir, se seleccionó una confiabilidad del 99%

Figura 116: Factor de confiabilidad

Reliability	$K_R (Y_Z)$
0.9999	1.50
0.999	1.25
0.99	1.00
0.90	0.85
0.50	0.70

(Budynas, 2008)

2) Con el factor de temperatura se eligió como 1, debido a que no se estarían trabando con altas temperaturas, por lo que no era significativo su valor.

3) El factor de seguridad se seleccionó tomando en cuenta que se está trabajando con motores, por esa razón se colocó un valor de 1.5

4) El número de ciclos de carga se utilizó la siguiente tabla y ecuación:

Figura 117: Vida de diseño (L) en horas, dependiendo de la aplicación

TABLA 9-7 Vida de diseño recomendada

Aplicación	Vida de diseño (h)
Electrodomésticos	1000-2000
Motores de avión	1000-4000
Automotriz	1500-5000
Equipo agrícola	3000-6000
Elevadores, ventiladores industriales, transmisiones de usos múltiples	8000-15 000
Motores eléctricos, sopladores industriales, maquinaria industrial en general	20 000-30 000
Bombas y compresores	40 000-60 000
Equipo crítico en funcionamiento continuo durante 24 h	100 000-200 000

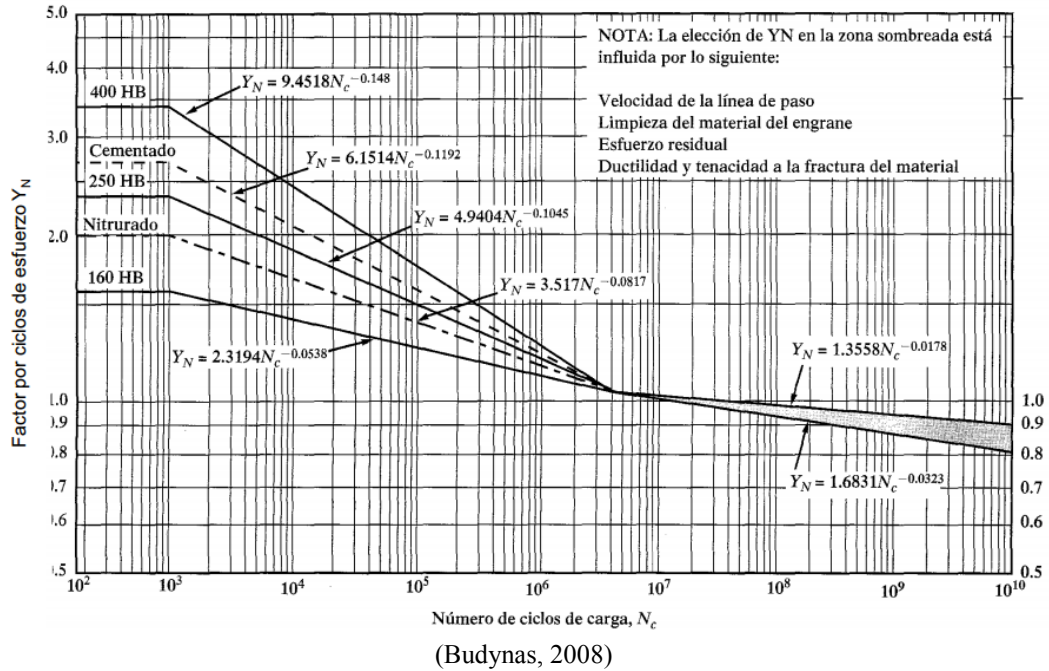
(Mott, 2006)

$$N_c = 60Lnq \quad (\text{ix})$$

En donde L es la vida de diseño, n son las revoluciones por minuto y q es el número de aplicaciones de carga por revolución, la cual generalmente es 1.

5) Por último, se calculó el factor por ciclos del esfuerzo utilizando la siguiente gráfica.

Figura 118: Valores del factor por ciclo de esfuerzo utilizando el número de ciclos de carga



Cuadro 22: Factores para el cálculo del esfuerzo de flexión permisible

FACTORES PARA EL CÁLCULO DEL ESFUERZO DE FLEXIÓN PERMISIBLE

Factor de confiabilidad (K_r)	1
Factor de temperatura (K_T)	1
Factor de seguridad (S_F)	1.5
Número de ciclos de carga del piñón 1 (N_{cP})	420000000
Número de ciclos de carga de la corona 1 (N_{cG})	840000000
Factor de ciclos de esfuerzo en el piñón 1 (Y_{nP})	0.85
Factor de ciclos de esfuerzo en la corona 1 (Y_{nG})	0.91
Número de esfuerzo de flexión en el piñón 1 (STP)	3959.90 psi
Número de esfuerzo de flexión en la corona 1 (STG)	16546.49 psi
Número de ciclos de carga del piñón 2 (N_{cP})	600000000
Número de ciclos de carga de la corona 2 (N_{cG})	85714285.71
Factor de ciclos de esfuerzo en el piñón 2 (Y_{nP})	0.91
Factor de ciclos de esfuerzo en la corona 2 (Y_{nG})	0.91
Número de esfuerzo de flexión en el piñón 2 (STP)	25807.21 psi
Número de esfuerzo de flexión en la corona 1 (STG)	115447.67 psi

k. Luego se analizó el esfuerzo de contacto real al que estarían sometidos los engranes.

1) Se escogió cuál sería el coeficiente elástico, este depende del material seleccionado. Para este caso se utilizó acero y con ayuda de la siguiente tabla se seleccionó el valor.

Figura 119: Módulo de elasticidad de los materiales

Material del piñón	Módulo de elasticidad del piñón E_p psi (MPa)*	Material y módulo de elasticidad de la corona E_G , lb/pulg ² (MPa)*					
		Acero 30×10^6 (2×10^5)	Hierro maleable 25 x 10^6 (1.7×10^5)	Hierro nodular 24×10^6 (1.7×10^5)	Hierro fundido 22×10^6 (1.5×10^5)	Bronce al aluminio 17.5×10^6 (1.2×10^5)	Bronce al estaño 16×10^6 (1.1×10^5)
Acero	30×10^6 (2×10^5)	2 300 (191)	2 180 (181)	2 160 (179)	2 100 (174)	1 950 (162)	1 900 (158)
Hierro maleable	25×10^6 (1.7×10^5)	2 180 (181)	2 090 (174)	2 070 (172)	2 020 (168)	1 900 (158)	1 850 (154)
Hierro nodular	24×10^6 (1.7×10^5)	2 160 (179)	2 070 (172)	2 050 (170)	2 000 (166)	1 880 (156)	1 830 (152)
Hierro fundido	22×10^6 (1.5×10^5)	2 100 (174)	2 020 (168)	2 000 (166)	1 960 (163)	1 850 (154)	1 800 (149)
Bronce al aluminio	17.5×10^6 (1.2×10^5)	1 950 (162)	1 900 (158)	1 880 (156)	1 850 (154)	1 750 (145)	1 700 (141)
Bronce al estaño	16×10^6 (1.1×10^5)	1 900 (158)	1 850 (154)	1 830 (152)	1 800 (149)	1 700 (141)	1 650 (137)

Relación de Poisson = 0.30.

(Budynas, 2008)

2) El factor de condición superficial (C_f) se seleccionó como 1, por tener un acabado superficial.

3) El factor geométrico de resistencia se calculó con la siguiente ecuación:

$$I = \frac{\cos \phi \sin \phi}{2m_N} \frac{m_G}{m_G + 1} \quad (x)$$

En donde: ϕ es el ángulo de presión, m_N es la relación de repartición, que es 1 para engranes rectos y m_G es la relación de engranes.

4) Contando con todos los factores necesarios para el cálculo del esfuerzo de contacto, se utilizó la siguiente ecuación para saber el valor:

$$\sigma_c = C_p \sqrt{W_t K_o K_v K_s \frac{K_m C_f}{D_p F I}} \quad (xi)$$

Cuadro 23: Factores para el cálculo del esfuerzo de contacto real

FACTORES PARA EL CÁLCULO DEL ESFUERZO DE CONTACTO REAL

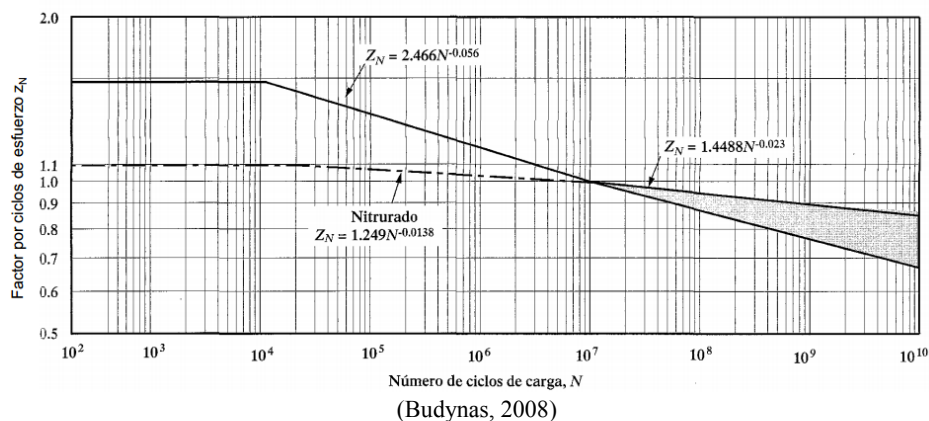
Coefficiente elástico (Cp)	2300
Factor de la condición superficial (Cf)	1
Factor geométrico (I)	0.155
Esfuerzo real de contacto aplicado al piñón 1 (σ_{CP})	71631.78 psi
Esfuerzo real de contacto aplicado a la corona 1 (σ_{CG})	27241.38 psi
Esfuerzo real de contacto aplicado al piñón 2 (σ_{CP})	178698.85 psi
Esfuerzo real de contacto aplicado a la corona 2 (σ_{CG})	67958.70 psi

1. Como último paso para el cálculo de engranes se calculó el esfuerzo de contacto permisible. Para ello se utilizó la siguiente ecuación:

$$S_c = \frac{\sigma_{SH} K_T K_r}{Z_N CH} \quad (\text{xii})$$

- 1) El factor de seguridad se eligió con el mismo valor anterior, por estar trabajando con motores, es decir de 1.5.
- 2) El factor de relación de durezas (CH) generalmente se elige como valor 1, para luego por elegir el material.
- 3) El factor de ciclos de esfuerzo de resistencia (Z_N) se calculó utilizando la siguiente gráfica.

Figura 120: factor de ciclos de esfuerzo



Cuadro 24: Factores para el cálculo del esfuerzo de contacto permisible

FACTORES PARA EL CÁLCULO DEL ESFUERZO DE CONTACTO PERMISIBLE		
Factor de seguridad (SH)		1.5
Factor de relación de durezas (CH)		1
Factor de ciclos de esfuerzo de resistencia a la picadura del piñón (ZNP)		0.75
Factor de ciclos de esfuerzo de resistencia a la picadura de la corona (ZNG)		0.85
Número de esfuerzo de contacto admisible del piñón 1 (ScP)	143263.568	psi
Número de esfuerzo de contacto admisible de la corona 1 (ScG)	48073.01783	psi
Número de esfuerzo de contacto admisible del piñón 2 (ScP)	357397.7074	psi
Número de esfuerzo de contacto admisible de la corona 1 (ScG)	119927.115	psi

4. DISEÑO DE LOS EJES DEL SISTEMA DE POTENCIA MECÁNICO. Para el diseño de los ejes se tuvo que tomar una decisión con respecto al material con los que se estarían fabricando.

a. Se tuvo que decidir un diámetro aproximado del eje para comenzar el diseño, al igual que factor de seguridad y de qué manera estaría acoplado, tanto el motor como el mecanismo, a la caja reductora.

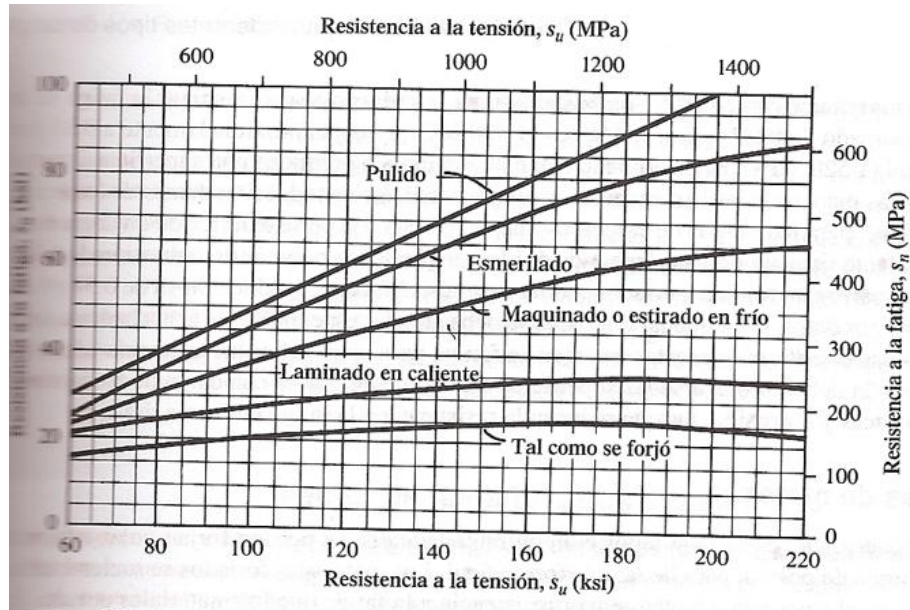
Cuadro 25: Propiedades del material de los ejes de la caja reductora

PROPIEDADES DEL MATERIAL		
Acero AISI 1040 estirado en frío		
Resistencia a la tensión	80000	psi
Resistencia de fluencia	71000	psi
Diámetro aproximado del eje	1.5	in
Diámetro de la Polea de entrada y salida (A)	10	in
Diámetro interno del Rodamiento eje 1 y 2 (B/D)	1.188	in
Diámetro interno del Rodamiento eje 3	1.578	in
Diámetro de paso del engrane recto 1 (C)	2.125	in
Diámetro de paso del engrane recto 2	14.875	in
Diámetro de paso del engrane recto 3	2.25	in
Diámetro de paso del engrane recto 4	15.75	in
Porcentaje de elongación	12	%
Factor de seguridad (N)	2.5	
Factor Kt para radios entrantes pequeños	2.5	
Factor Kt para cuñeros ciegos	2	
Polea colocada a 60°	60	°

b. Se calculó el esfuerzo de fatiga al que estaría sometido el eje.

1) Primero se tuvo que calcular la resistencia a la fatiga teórica del material. Para ello se utilizó la siguiente figura.

Figura 121: gráfica que muestra la resistencia a la fatiga en función de la resistencia a la tensión



(Mott, 2006)

2) Para poder seleccionar el factor de material se tomó en cuenta un acero forjado, por lo que se obtiene un valor de 1 para dicho factor.

3) Para el factor del tipo de esfuerzo se tomaron en cuenta esfuerzos flexionantes y el valor correspondiente a ese tipo de esfuerzos es de 1.

4) Para el factor de confiabilidad se decidió que se tendría una confiabilidad del 99%. Utilizando la siguiente figura se obtuvo el valor.

Figura 122: Factor de confiabilidad

Factores de confiabilidad aproximados C_R	
Confiabilidad deseada C_R	
0.50	1.0
0.90	0.90
0.99	0.81
0.999	0.75

(Mott, 2006)

Cuadro 26: Cuadro que muestra el esfuerzo de fatiga de los ejes

CÁLCULO DEL ESFUERZO DE FATIGA (Sn')		
Resistencia a la fatiga teórica (Sn)	30000	psi
Factor de material (Cm)	1	
Factor del tamaño de la sección (Cs)	0.8	
Factor del tipo de esfuerzo (Cst)	1	
Factor de confiabilidad (Cr)	0.81	
S'n	19440	psi

c. Se calcularon los torques a los que estaría sometido tomando en cuenta la potencia y las rpms.

Cuadro 27: Torques resultantes

CÁLCULO DE TORQUES		
Potencia	7.5	hp
N de entrada	1750	rpm
N del eje intermedio	250	rpm
N de salida	35.71	rpm
Torque Polea de entrada	270	Lb*in
Torque Engrane 1	270	Lb*in
Torque Engrane 2	1890	Lb*in
Torque Engrane 3	1890	Lb*in
Torque Polea de salida	13231.6	Lb*in
Torque Engrane 4	13231.6	Lb*in

Se calcularon las fuerzas a las que estaría sometido el eje, tanto en el eje vertical como en el horizontal.

Cuadro 28: Fuerzas a las que están sometidos los ejes de la caja reductora

CÁLCULO DE FUERZAS
EJE 1**POLEA DE ENTRADA**

Fx 40.50 Lb

Fy 70.15 Lb

ENGRANE 1

Fuerza Tangencial (Ft) 254.12 Lb

Fuerza radial (Fr) 92.49 Lb

EJE 2**ENGRANE 2**

Fuerza Tangencial (Ft) 254.12 Lb

Fuerza radial (Fr) 92.49 Lb

ENGRANE 3

Fuerza Tangencial (Ft) 1680.20 Lb

Fuerza radial (Fr) 611.47 Lb

EJE 3**ENGRANE 4**

Fuerza Tangencial (Ft) 1680.20 Lb

Fuerza radial (Fr) 611.47 Lb

POLEA DE SALIDA

Fx 1984.74 Lb

Fy 3437.67 Lb

d. Para poder obtener el valor de los diámetros mínimos se tuvo que realizar diagramas de corte y momento para poder utilizar la siguiente ecuación:

$$D_{\min} = \left[\frac{32 \cdot N}{\pi} \sqrt{\left(\frac{K_t M_f}{S_n} \right)^2 + \frac{3}{4} \left(\frac{T}{S_y} \right)^2} \right]^{1/3} \quad (\text{xiii})$$

Para obtener los diagramas se utilizó Autodesk Inventor 2014, los cuales se muestran a continuación.

- 1) Eje de entrada:
 - a) Fuerzas verticales

Figura 123: Diagrama de corte de las fuerzas verticales del eje de entrada

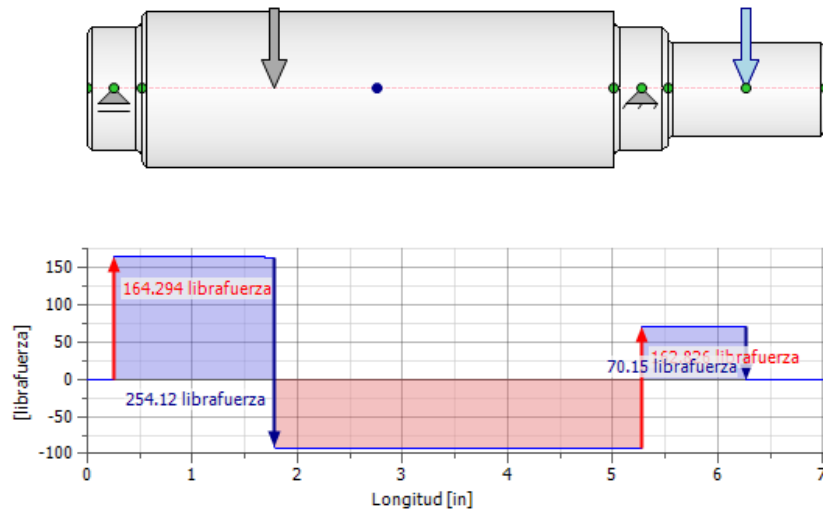


Figura 124: Diagrama de momento de las fuerzas verticales del eje de entrada

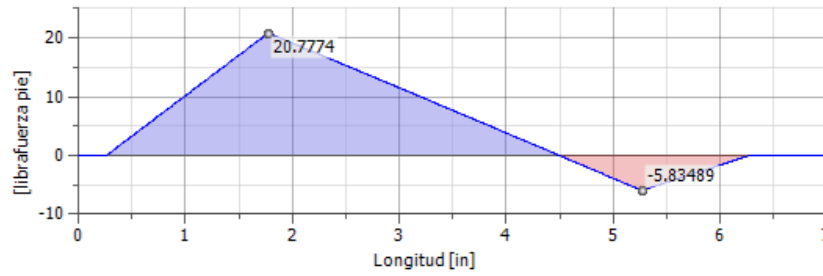
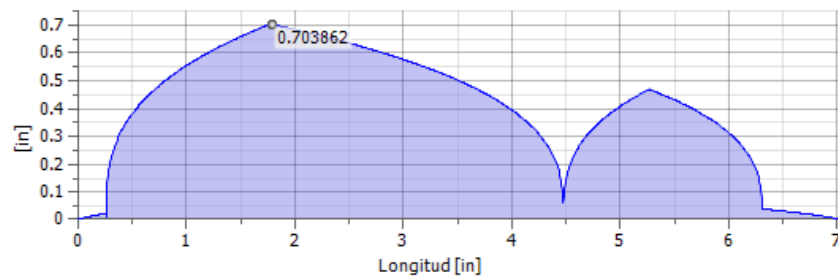


Figura 125: Diámetro ideal del eje de entrada



b) Fuerzas horizontales

Figura 126: Diagrama de corte de las fuerzas horizontales del eje de entrada

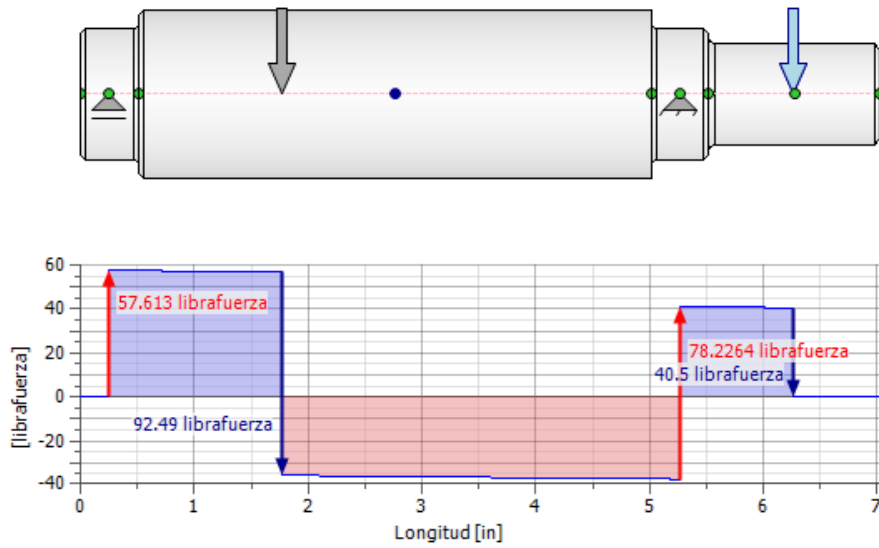


Figura 127: Diagrama de momento de las fuerzas horizontales del eje de entrada

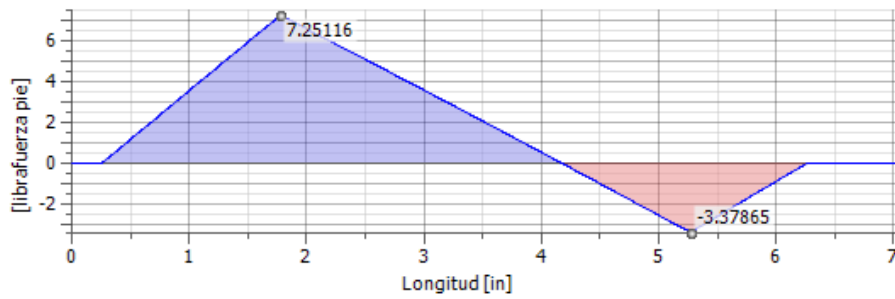
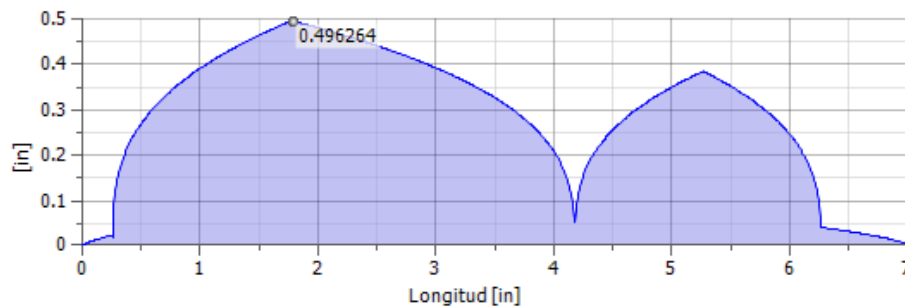


Figura 128: Diámetro ideal para las fuerzas horizontales



- 2) Eje intermedio:
 - a) Fuerzas horizontales:

Figura 129: Diagrama de corte de las fuerzas horizontales del eje intermedio

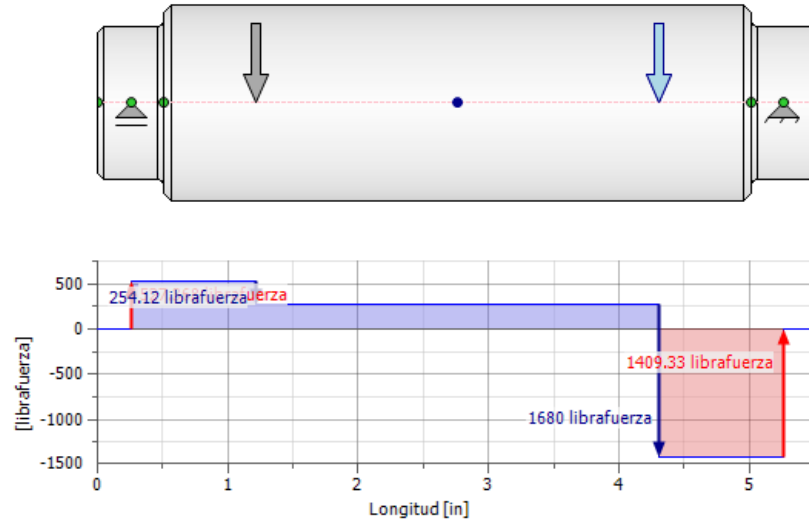


Figura 130: Diagrama de momento de las fuerzas horizontales del eje intermedio

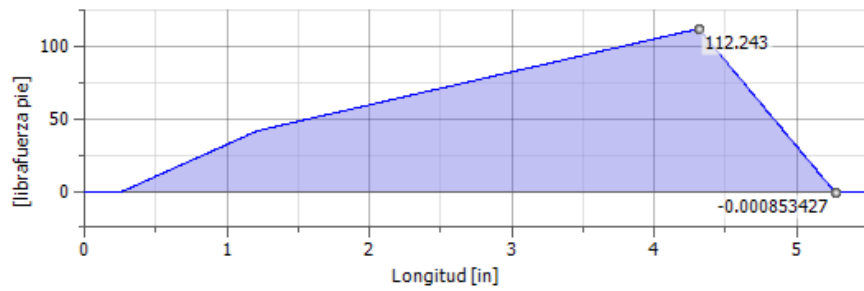
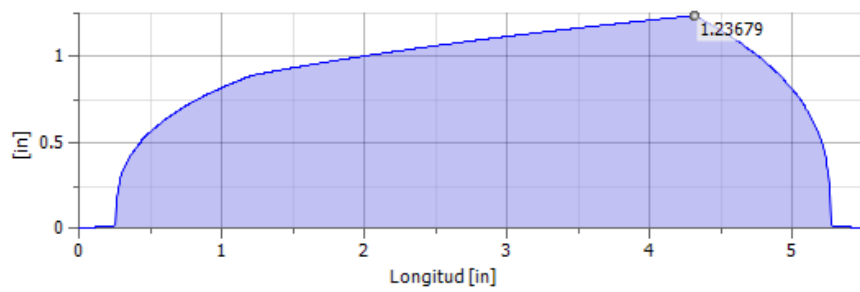


Figura 131: Diámetro ideal para las fuerzas horizontales del eje intermedio



b) Fuerzas verticales:

Figura 132: Diagrama de corte para las fuerzas verticales del eje intermedio

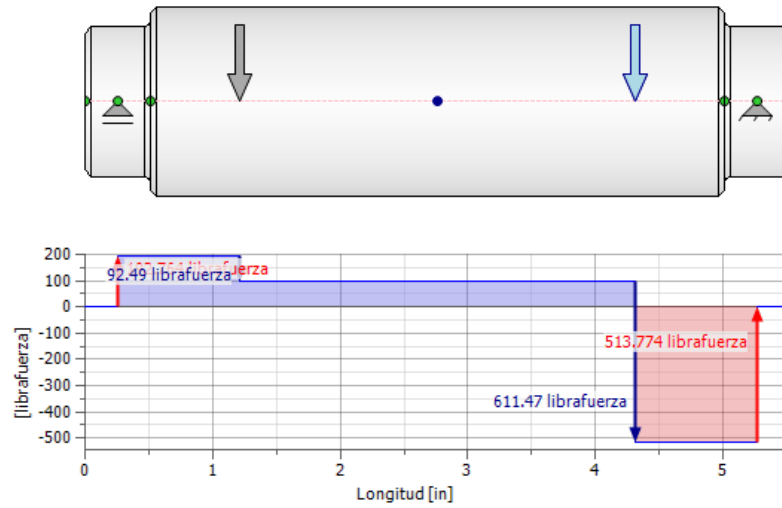


Figura 133: Diagrama de momento de las fuerzas horizontales del eje intermedio

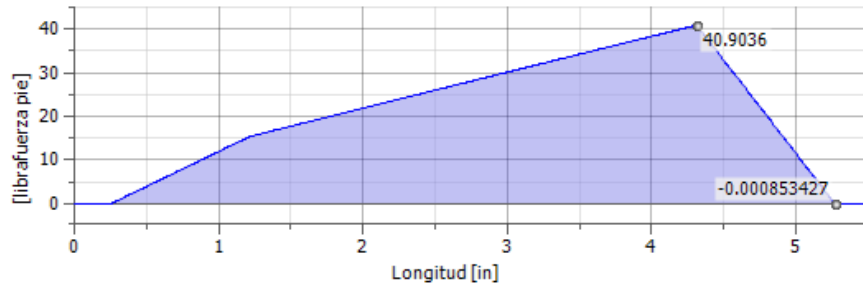
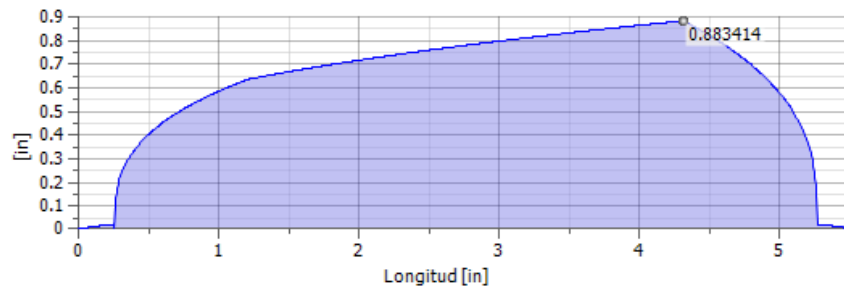


Figura 134: Diámetro ideal para las fuerzas verticales del eje intermedio



- 3) Eje de salida:
 - a) Fuerzas verticales

Figura 135: Diagrama de corte de las fuerzas verticales del eje de salida

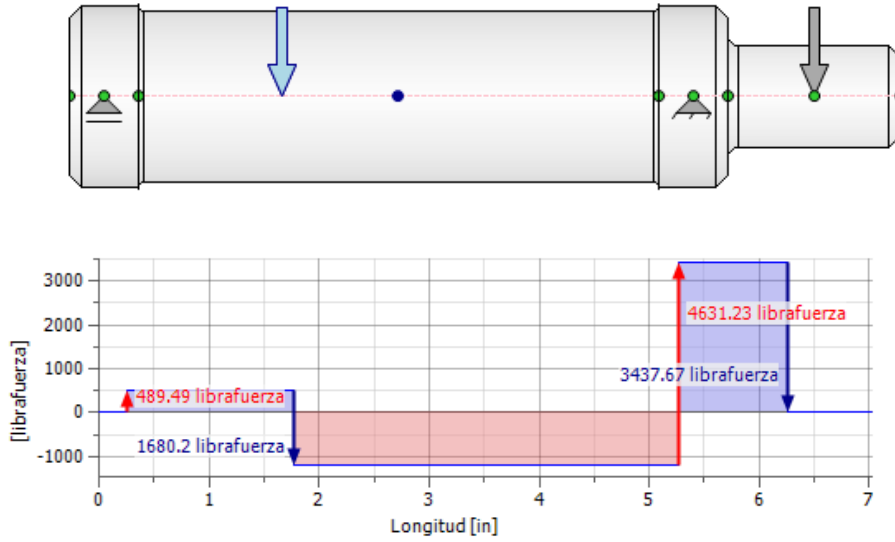


Figura 136: Diagrama de momento de las fuerzas verticales del eje de salida

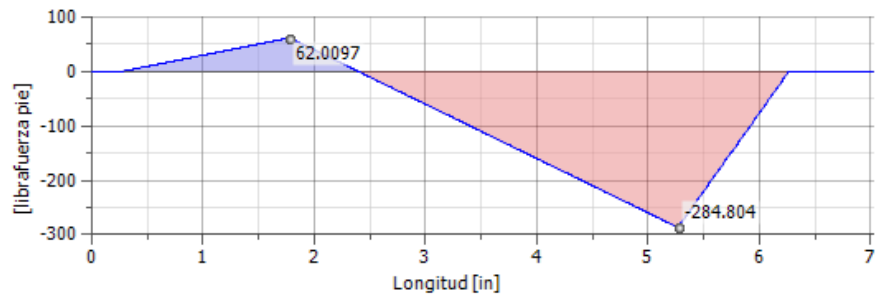
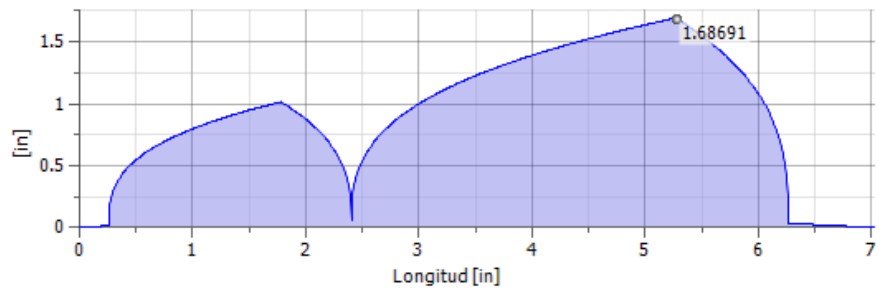


Figura 137: Diámetro ideal de las fuerzas verticales para el eje de salida



b) Fuerzas horizontales:

Figura 138: Diagrama de corte de las fuerzas horizontales del eje de salida

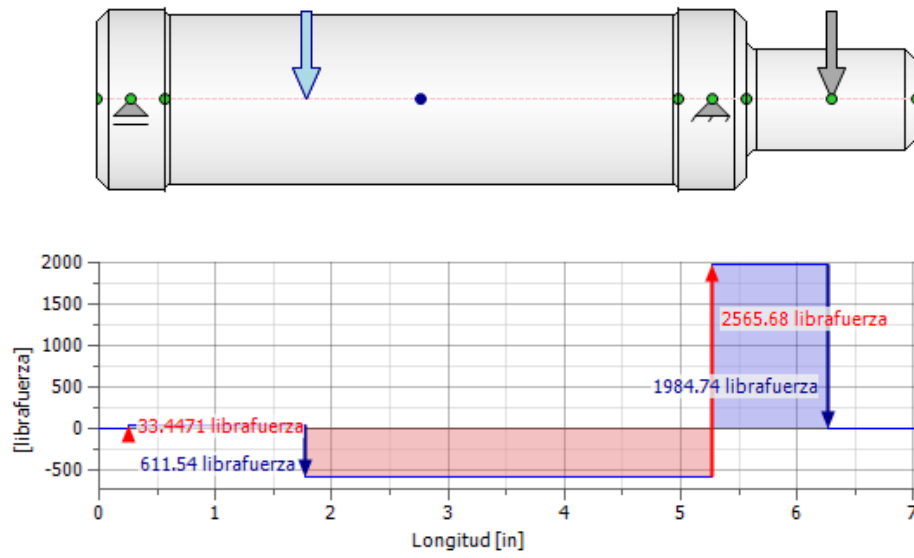


Figura 139: Diagrama de momento de las fuerzas horizontales del eje de salida

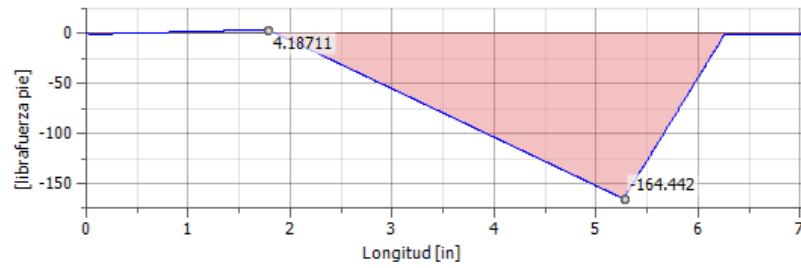
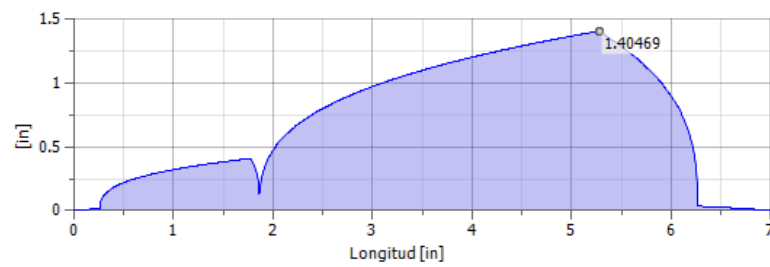


Figura 140: Diámetro ideal para las fuerzas horizontales del eje de salida



e. Se calcularon los diámetros utilizando la ecuación xii, en donde se obtuvieron los siguientes valores para cada eje.

Cuadro 29: Diámetro mínimo para los ejes de la caja reductora

DIÁMETROS MÍNIMOS		
EJE DE ENTRADA		
Diámetro mínimo polea de entrada	0.44	in
Diámetro mínimo rodamiento 1	0.48	in
Diámetro mínimo engrane 1	0.44	in
Diámetro mínimo rodamiento 2	0.41	in
EJE INTERMEDIO		
Diámetro mínimo rodamiento 1	0.84	in
Diámetro mínimo engrane 2	0.84	in
Diámetro mínimo engrane 3	0.84	in
Diámetro mínimo rodamiento 2	0.39	in
EJE DE SALIDA		
Diámetro mínimo polea de salida	1.49	in
Diámetro mínimo rodamiento 1	1.50	in
Diámetro mínimo engrane 4	1.49	in
Diámetro mínimo rodamiento 2	0.61	in

C. RESULTADOS

Luego de llevar a cabo el proceso de diseño y las simulaciones del movimiento del mecanismo se logró desarrollar un sistema mecánico capaz de movilizar el grano de café en las secadoras estáticas. Se pudo seleccionar de manera correcta los rodamientos que estarían soportando el mecanismo. Además, se logró diseñar la transmisión de potencial diseñando un tren de engranes de una caja reductora. Asimismo, con ayuda del diseño del sistema eléctrico mecánico se pudo conocer la fuerza necesaria para movilizar el café dentro de las cámaras, de esta manera se pudo corroborar si el mecanismo estaría actuando de manera correcta.

Cuadro 30: Dimensiones de cada parte del mecanismo

PARTES DEL MECANISMO		
PALETA		
<i>Ancho</i>	25	cm
<i>Alto</i>	20	cm
<i>Profundidad</i>	5	cm
SOPORTE PALETA		
<i>Ancho</i>	3	cm
<i>Alto</i>	26.5	cm
<i>Profundidad</i>	3	cm
EJE		
<i>Diámetro mayor</i>	5	cm
<i>Largo</i>	245	cm

Cuadro 31: Rodamiento seleccionado para el eje del mecanismo

RODAMIENTO INSERTANTE		
UC206		
<i>Diámetro externo</i>	62	mm
<i>Diámetro interno</i>	30	mm
<i>Espesor</i>	38.1	mm
<i>Carga dinámica</i>	20,700	N
<i>Carga estática</i>	11,300	N

Cuadro 32: Variables para el cálculo de la fuerza necesaria para movilizar el café

CÁLCULO DE LA FUERZA PARA MOVILIZAR EL CAFÉ		
Velocidad angular (ω)	183.26	rad/s
Potencia (H)	5592.68	Watts
Torque	30.52	Nm
Torque salida	1495.37	Nm
Radio del mecanismo	0.3715	m
Fuerza (F)	4025.23	N

Figura 141: Diseño de paleta para el mecanismo diseñada en Autodesk Inventor

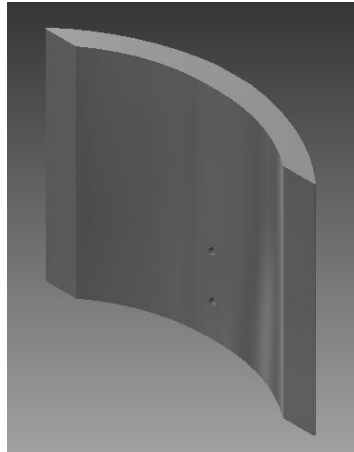


Figura 142: Soporte de la paleta del mecanismo diseñado en Autodesk Inventor

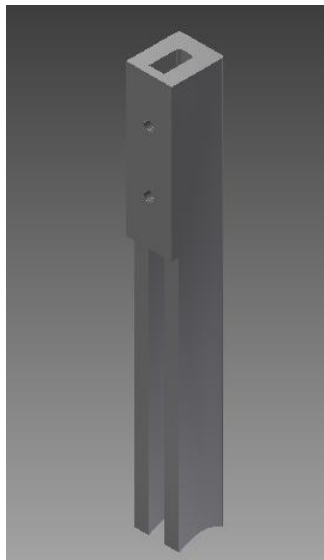


Figura 143: Eje del mecanismo diseñado en Autodesk Inventor



Figura 144: Vista isométrica del mecanismo diseñado en Autodesk Inventor

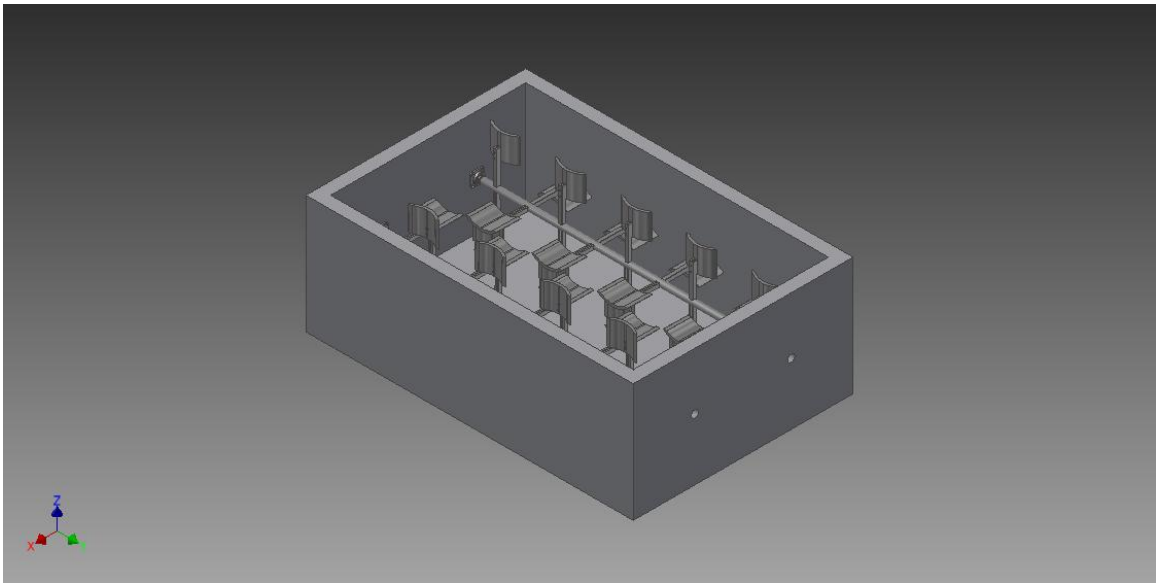


Figura 145: Vista de planta del mecanismo diseñado en Autodesk Inventor

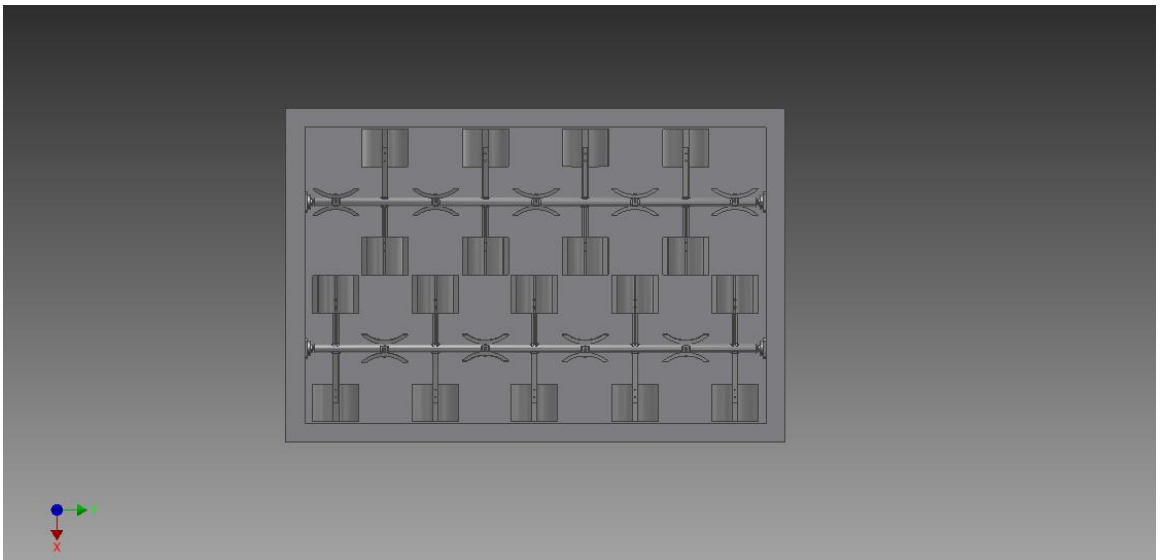


Figura 146: Vista lateral del mecanismo diseñado en Autodesk Inventor

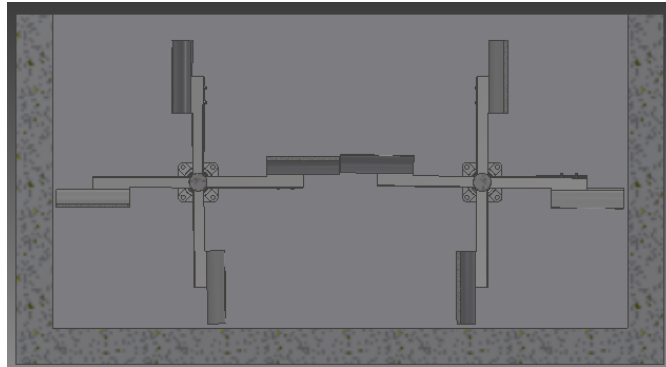


Figura 147: Soldadura de las paletas al eje del mecanismo

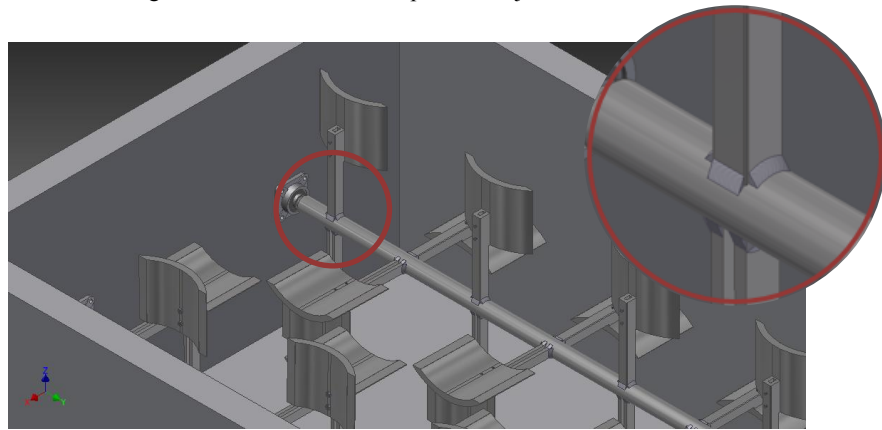
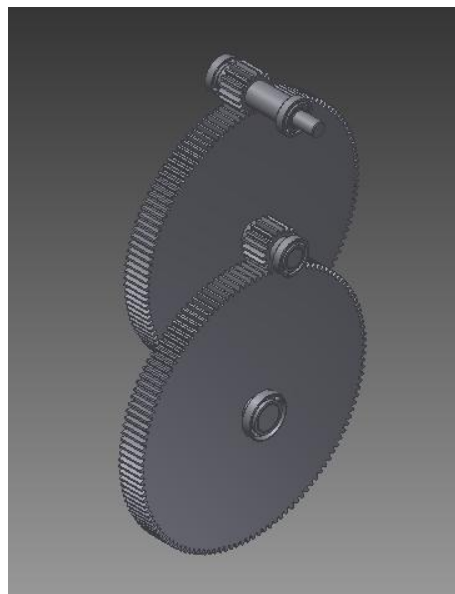


Figura 148: Estructura final del sistema de potencia mecánico



D. ANÁLISIS DE RESULTADOS

Para el diseño de la estructura mecánica para movilizar el café se tuvo que tomar muchas suposiciones y decisiones para poder obtener el mecanismo más óptimo y eficiente. Primero se comenzó identificando el problema, por lo que se tuvo que realizar una visita para conocer con detalle cuáles son las necesidades que tenía el dueño de la finca y de qué manera se pueden automatizar procesos que el realiza de manera manual actualmente. Una de las partes que se optó para poder optimizar fue la manera de movilizar el café dentro de las cámaras de secado, se solicitó a los trabajadores que realizaran una dramatización de cómo se realizaba el movimiento y que se mostrara el instrumento con el que lo realizan. En la Figura 90 se muestra la pala que utilizan, ésta solamente es una cubeta plástica partida a la mitad acoplada a un pedazo de madera. Al momento de decidir de qué forma sería el mecanismo a diseñar se tomó como una de las opciones mecanismos de paletas, especialmente porque era la manera más parecida a lo que ellos realizan actualmente.

En la Figura 130 se puede observar el diseño de la paleta final del mecanismo, ésta cuenta con una forma de un semicírculo principalmente por ser la manera más parecida y delicada de recoger el café al momento de estar en movimiento el mecanismo. No se colocaron paletas planas porque el café se deslizaría en ellas y no estarían tomando pequeñas porciones de café dentro de las paletas. Luego se prosiguió a diseñar los soportes de las paletas, se tuvo dos opciones para realizarla: triangular o rectangular. Para tener un mejor soporte al eje, se optó por soportes rectangulares como se muestra en la Figura 131. Por último se diseñó el eje, en la Figura 132 se puede observar que en los extremos se tiene una reducción de diámetro, esto se debe a que se puedan acoplar de manera correcta los rodamientos, además de tomar en cuenta el costo de éstos, ya que mientras mayor es el diámetro más alto es el costo. Para poder verificar que el diámetro seleccionado para el eje fuera el adecuado se realizaron simulaciones con ayuda de Autodesk Inventor Professional 2014. Se evaluó de manera estática, solamente colocando los pesos de las paletas, soportes y del eje. Se obtuvieron diagramas de corte en cada uno de los planos (Figura 91, Figura 96) para conocer las fuerzas de reacción que se tienen en los soportes para poder seleccionar correctamente los rodamientos.

Asimismo se tomó en cuenta los diagramas del momento flector (Figura 92, Figura 97); los ángulos de flexión, que en este caso no se llegaron a obtener valores mayores a 1° (Figura 93, Figura 98). Se analizó la flexión del eje en los dos planos y no se obtuvieron valores elevados capaces de causar la falla del eje. Por último, se observó diámetro ideal sugerido por Inventor, el promedio de los diámetros en cada plano fue de 58 mm, ver Figura 95, Figura 100. Por lo que el diámetro que al final se seleccionó fue de 50 mm.

En el Cuadro 30 se pueden observar las dimensiones de las paletas y el resto de partes de la estructura mecánica, ésta fue una decisión que se tomó pensando en los esfuerzos y fuerzas a los que estarían sometidas las paletas, el soporte y el eje. Especialmente tomando en cuenta que este último no sufriera una

falla a flexión por el peso y fuerzas del mecanismo. Luego del diseño de cada una de las estructuras y ya conociendo las reacciones del soportes se seleccionó el rodamiento, como se muestra en el Cuadro 31. La capacidad de carga dinámica y estática del rodamiento seleccionado es un valor sobredimensionado al valor obtenido en Inventor, especialmente porque lo que se decidió fue acoplar el rodamiento al diámetro seleccionado y no iterar el valor del diámetro del eje, debido a que si se iba reduciendo el valor la flexión sería mayor y tendería a fallar. Otro dato que se tomó en cuenta en simulación fue el material, tomando en cuenta que estaría en un ambiente húmedo y que se está trabajando con alimentos. Por esta razón, se decidió un acero inoxidable AISI 304, especialmente por las especificaciones y por ser un material poco corrosivo.

La distribución de las paletas dentro del eje se colocó tomando en cuenta que se movilizara la mayor cantidad de café, sin dejar espacios sin hacer contacto para que la humedad y la temperatura se estuvieran homogenizando de la mejor manera. En la Figura 133, se muestra una vista isométrica del ensamblaje de piezas del mecanismo. Asimismo, en la Figura 134 se puede observar con más detalle de qué manera están colocadas y el espaciado que se tiene entre cada paleta. En la Figura 135 se realizó un corte lateral para que se pudiera visualizar que las paletas no estarían entrando en contacto con ninguna de las paredes de las secadoras, especialmente para no dañar el grano ni crear fricción con el mecanismo y que se estuviera un desgaste. En la Figura 136 se observa el tipo de soldadura que se tuvo que realizar con las paletas y el eje, ya que la manufactura no se puede realizar como una sola pieza. El tipo de soldadura que se utilizó fue de aluminio 6061, debido a que es el material más utilizado y comercial especialmente en el mercado guatemalteco.

Además del cálculo de los esfuerzos a los que está sometido el mecanismo se calculó la fuerza necesaria para movilizar el café, de esta manera se podría verificar si el movimiento realizado es el correcto y el necesario para poder movilizar aproximadamente 10 quintales de café. Se tomó en cuenta la velocidad angular del mecanismo, la potencia del motor, el torque que realiza y el torque final de todo el mecanismo y el radio que están recorriendo las paletas al momento de girar el mecanismo. En el Cuadro 32, se puede observar que el valor resultante fue de 2053.69 N, es decir, aproximadamente 461.69 lbs,

Para el diseño del sistema de potencia mecánico, se decidió utilizar una reducción de velocidades con engranes rectos. Estos tienen la capacidad de reducir un máximo de 10:1, por lo que se tuvo que realizar dos reducciones con un tren de engranes. Para la elección de los factores seleccionados para el diseño se tomó en cuenta con qué materiales se está trabajando y el funcionamiento que está realizando el mecanismo. Se decidió que el mecanismo estuviera girando a 35.71 rpms, especialmente porque no se estará realizando de manera constante el movimiento sino que estará trabajando por ciertos períodos de tiempo, como lo realizan actualmente los trabajadores en la finca La Fama. En la Figura 137 se puede

observar el diseño de los engranes rectos generados por Autodesk Inventor Professional 2014 y los ejes acoplados a estos.

Para el diseño sólo se deben colocar ciertos parámetros para que el software pueda generar correctamente lo que se desea obtener. Para el diseño de los ejes se compararon los valores promedios obtenidos con el diseño con los valores promedios obtenidos con Inventor. Para el eje de entrada se obtuvo un porcentaje de error de 19.7%, para el eje intermedio un valor de 20.6% y para el eje de salida 2.96%. Además, se tiene que tomar en cuenta que los valores que se muestran en el Cuadro 29, son los diámetros mínimos permisibles a los ejes pero en este caso se colocó un valor de 1.5 in para que el eje tuviera un mayor factor de seguridad, especialmente porque las revoluciones de entrada tienen un valor alto.

Se tiene en desarrollo la elaboración de un prototipo para que se tenga una mejor visualización del funcionamiento y el diseño de piezas del mecanismo. Con él se podrá observar con detalle el movimiento que estaría realizando dentro de las cámaras de secado.

E. CONCLUSIONES

1. Se logró diseñar un mecanismo capaz de movilizar el café de manera correcta con la velocidad deseada entre 30 y 40 rpms y fuerza necesaria de 4025.23N, sin dañar el grano de café.
2. Se logró el diseño de un sistema de potencia mecánico con engranes rectos, capaz de transmitir el torque y potencia necesaria para movilizar el mecanismo.
3. Uno de los aspectos más importantes a considerar en el diseño de máquinas es determinar los materiales que se utilizarán. Esto se debe a que tienen mucha importancia en los procesos de manufactura, limpieza, costo, durabilidad, resistencia y mantenimiento del equipo.

F. RECOMENDACIONES

- Se recomienda realizar un estudio financiero para establecer la factibilidad de la implementación del proyecto y si realmente se logró la creación de un mecanismo de bajo costo.
- Se recomienda la construcción de un prototipo del sistema de potencia mecánico para verificar alguna falla al momento de estar trabajando de manera automática el mecanismo.
- Se recomienda que el sistema de potencia mecánico se realice utilizando otras alternativas de reducción, como por ejemplo un tornillo sin fin para ahorrar espacio de instalación.

VIII. MÓDULO FUMOGUA IMPLEMENTACIÓN DE SENSORES, ANÁLISIS Y TRANSMISIÓN DE LA INFORMACIÓN DE DATOS.

A. METODOLOGÍA

Se investigó las diferentes formas de monitoreo que se están implementando actualmente en Guatemala, y se propuso una solución para monitorear y mejorar la fumigación del café. Tras haber propuesto una solución, se investigaron diferentes sensores para cubrir nuestras necesidades. Se diseñó un acople para poder colocar los sensores y realizar mediciones en la bomba de fumigación.

Se creó un programa para un microcontrolador y hacer funcionar los sensores propuestos transformando los datos obtenidos a valores digitales. Para reducir la cantidad de datos a almacenar se realizó un promedio.

Luego se buscó una forma de mandar los datos de forma inalámbrica de la bomba de fumigación hacia la central. Ya con la conexión inalámbrica se procedió a enviar los datos guardados a la central. La central analiza los datos de entrada para verificar en donde se realizó una buena fumigación.

Los datos ya analizados se separaron en Finca, Fumogua, 1, fecha, hora, latitud, longitud, presión medida, porcentaje de presión, caudal medido, porcentaje de caudal, escala. Con esto se procedió a crear un archivo .csv para enviarlo a la base de datos.

B. DISEÑO

1. SELECCIÓN DE SENSORES. Se investigaron las variables que se tenían que medir para poder determinar si se está fumigando de forma adecuada. Lo primero que se investigó fue la forma de aplicación, ya que para que sea más efectivo el fungicida tiene que cubrir la parte debajo de la hoja no por encima. También se investigó que tiene que quedar completamente “cubierto” por el fungicida, ya que si no queda completamente cubierto puede ser que en los lugares sin fungicida crezca la roya.

El tipo de roció tiene que ser de forma constante, por lo que se determinó que se necesitaba medir presión y caudal de la bomba.

Para saber en dónde se estaba fumigando se determinó necesario utilizar un GPS.

Para escoger el sensor de presión se buscó un sensor que su medición sea entre 10 a 100 psi ya que esa es la presión de la bomba de fumigación manual, también que pudiera medir la presión en agua y otro parámetro importante es que sea de un tamaño pequeño para facilitar su manejo en la bomba. Se determinó

que el mejor sensor que se encuentra en el mercado fue el sensor Honeywell NBDANN100PGUNV ya que sus dimensiones son de 7mm² y su medición de presión va desde 0psi hasta 100psi. Como máximo soporta 200psi. La salida del sensor es análoga y depende del voltaje de alimentación, este va desde 3v a 12v, sin estar rectificadora ni amplificada. El sensor se puede observar en la “Figura 149: Sensor de presión Honeywell .”.

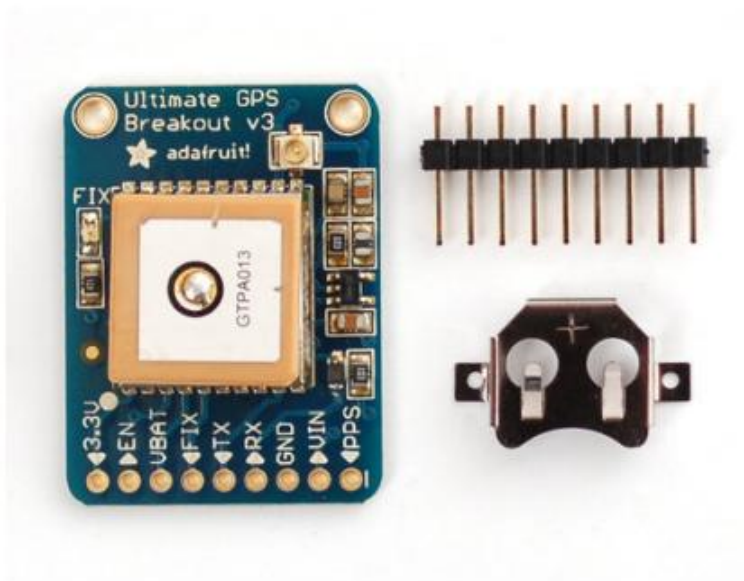
Figura 149: Sensor de presión Honeywell .



Dado que el sensor Honeywell presento problemas al usarse porque su salida tenía que ser rectificadora y amplificada se optó por cambiar de sensor al sensor 100psi. Este presenta las mismas especificaciones que el Honeywell pero su salida esta rectificadora y amplificada se puede observar en “Figura 51: Sensor de presión 100psi”.

El GPS se buscó uno que ofreciera una buena exactitud ya que es de gran importancia en nuestra aplicación, también se seleccionó por sus dimensiones y que fuera de fácil uso. Se optó por utilizar el GPS “Ultimate GPS Breakout v3” utiliza el chip MT3339 ver “Figura 150” este ofrece una exactitud de 1m y sus dimensiones son de 15x15mm.

Figura 150: Adafruit Ultimate GPS

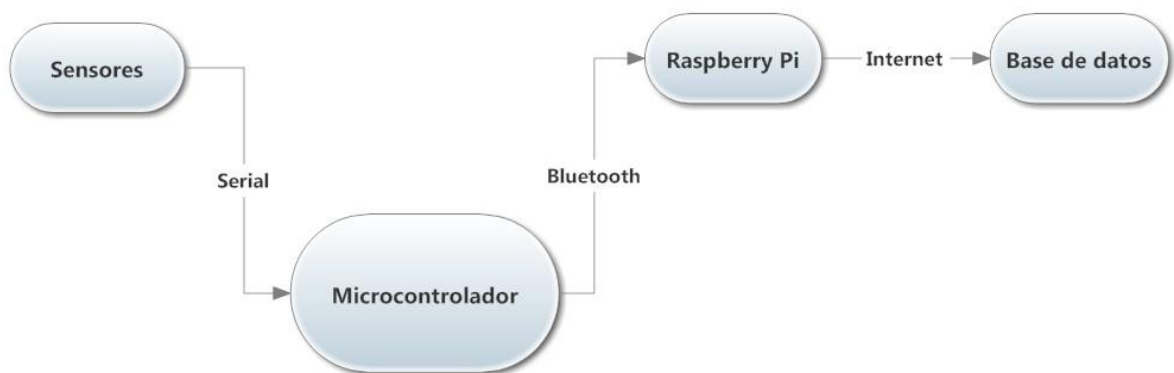


Para determinar que sensor de caudal se investigaron sensores con un rango de caudal entre 1l/m y 0.1 l/m. Dado que el sensor HQ-A68-1 se pudo comprar en Guatemala se optó por utilizarlo. El sensor cuenta con tres terminales dos para la alimentación y una que nos proporciona la salida. La salida del sensor es en frecuencia, dependiendo el caudal es proporcional a la frecuencia de salida. También se puede alimentar con 5v hasta 100v y tiene una corriente de salida de hasta 15mA lo que es exactamente lo que se necesitaba.

Ya con los sensores determinados se dio la necesidad de utilizar un microcontrolador para poder analizar las salidas de los distintos sensores, por lo que se investigó cual microcontrolador era el más apto para nuestra aplicación. Después de investigar se llegó a la conclusión de utilizar el microcontrolador PIC18F45K22 que presenta un bajo consumo energético y hasta 35 salidas, también cuenta con un oscilador interno de 16MHz pudiendo multiplicarlo por 4 llegando hasta 64MHz.

2. **COMUNICACIÓN.** Con el microcontrolador seleccionado se investigó la forma de poder mandar los datos a la base de datos que se encuentra en internet. Después de investigar se llegó a la conclusión de que la mejor opción era por medio de una Raspberry PI. Ya que cuenta con entrada de Ethernet, tiene un procesador que opera a casi 1GHz y su precio es bajo de al rededor Q250. En la “Figura 151: Módulo de comunicación.” se puede observar el diagrama de la conexión, con los diferentes dispositivos.

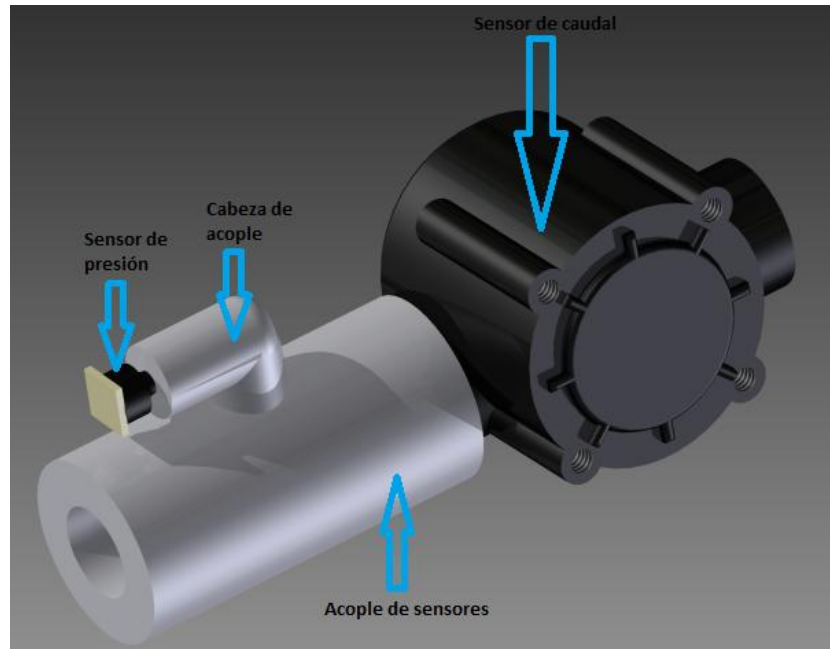
Figura 151: Módulo de comunicación.



3. **ACOPLE DE SENSORES.** Para poder colocar los sensores en la bomba de fumigación se tuvo que crear un acople que fuera fácil de colocar sin necesidad de cambiar grandes piezas ni modificar la bomba. También que soportara golpes. Él acople se diseñó en el programa Adobe Inventor, para luego

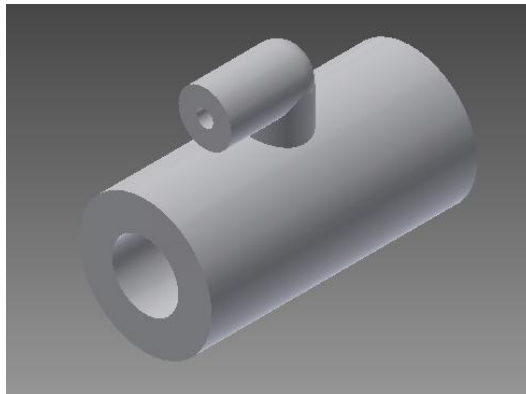
exportarlo en formato .stl (formato para impresoras 3D). En la “Figura 152: Posición de acople” se puede observar la posición de los sensores.

Figura 152: Posición de acople



Se probaron tres prototipos de distintos acoples. El primero mostrado en la “Figura 153: Acople prototipo 1” se desechó porque la impresora 3D no lo podía imprimir debido a que tenía que imprimir la cabeza en el aire sin ningún apoyo por lo que se decidió aumentar el tamaño de la cabeza de acople para poderlo imprimir.

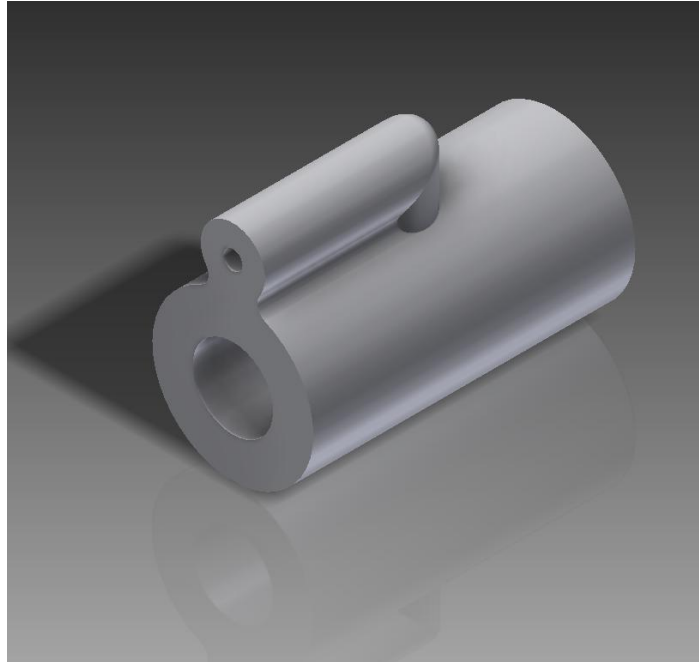
Figura 153: Acople prototipo 1



Ya con el diseño del nuevo Acople ver “Figura 154: Acople prototipo 2” se prosiguió con imprimirlo y probarlo. Tras varias pruebas se pudo notar que se podía escurrir el agua por las paredes del acople por lo

que se investigó algún aditamento para poder impermeabilizarlo. Para impermeabilizarlo primero se le coloco cera de madera y luego un producto en aerosol que se llama NeverWet. Con estos 2 aditamentos se logró impermeabilizar.

Figura 154: Acople prototipo 2



Se diseñó otro acople ver “Figura 155: Acople final” debido a que se cambió el sensor de caudal y presión con distintos diámetros de entrada por lo que se cambiaron estos parámetros en el siguiente prototipo. En la “Figura 156: Acople conectado en la bomba” se puede observar la forma en que se conectaron los sensores finales.

Figura 155: Acople final

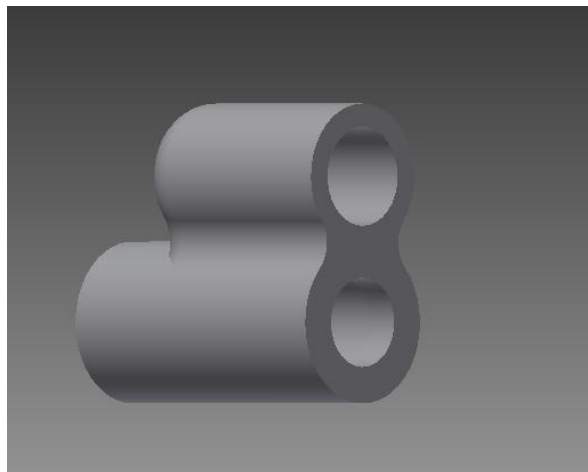
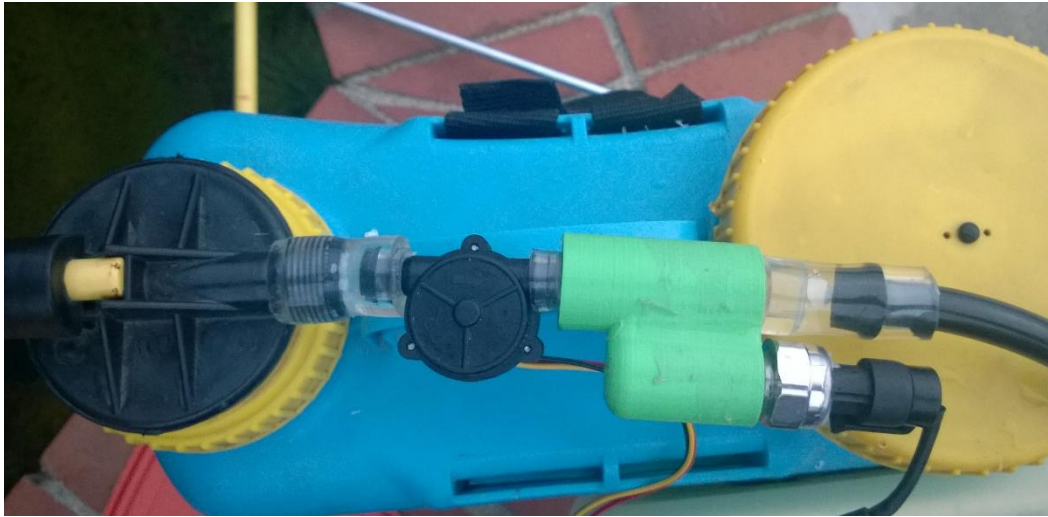
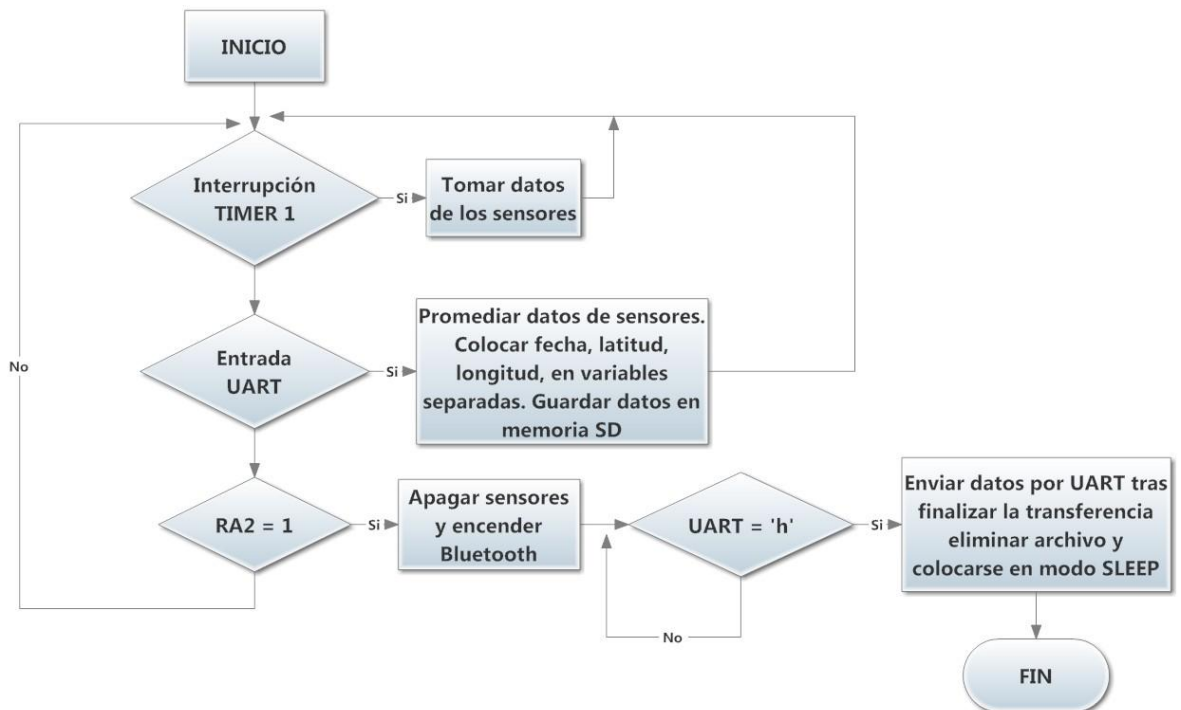


Figura 156: Acople conectado en la bomba



4. PROGRAMAS. Tras haber terminado con él acople y determinado los sensores se prosiguió con realizar el programa en el microcontrolador que tiene como función tomar los datos de los sensores y ordenarlos para poderlos enviar vía Bluetooth. El programa se realizó en el programa MikroC. En la “Figura 157: Diagrama de flujo programa del microcontrolador” se puede observar el diagrama de flujo del programa en una forma resumida.

Figura 157: Diagrama de flujo programa del microcontrolador



Después de crear el proyecto e inicializar el microcontrolador a 64MHz. se comenzó programando la respuesta del sensor de presión, ya que es una señal analógica, por lo que se configuró el conversor analógico-digital en la entrada dos. Con esto se tuvo un número desde 0 hasta 255, este se guardó en una variable para su análisis posterior.

Con el sensor de caudal se creó una forma para poder ver la frecuencia de salida del sensor. La cual se realizó utilizando el TIMER-0 como contador se colocó la señal de salida del sensor en la entrada del contador. Luego se utilizó el TIMER-1 como reloj para crear una interrupción cada 100ms. Dentro de la interrupción se adecuó la cantidad del contador para pasarlo a Hz y se colocó el dato en una variable para su análisis posterior.

Para recibir los datos del GPS se configuró el módulo serial UART a una velocidad de 9600 baudios. Luego se colocó a RX como entrada y TX como salida. En el ciclo principal del programa se espera a que se tenga una entrada del serial. Si existe la entrada entonces es procesada para desechar la información innecesaria. Esto se realiza esperando la secuencia \$GPGGA que tiene la información que se necesita. Si se encuentra \$GPGGA entonces guarda los datos de esa secuencia en una lista. Después se selecciona la latitud, longitud y hora en variables por separado. Notar que la latitud lleva N-S y la longitud E-W.

Ya que los datos a guardar van a ser mucho mayores a la memoria EPROM que tiene el microcontrolador se optó por colocar una memoria flash SD de 2Gb. Para guardar datos en la memoria se utilizó el protocolo SPI. Primero se inicializó el módulo que trae el microcontrolador SPI con todas sus entradas y salidas requeridas. Ya que se tienen los datos en variables separadas, se guarda en memoria después de cada entrada de GPS, está es una vez por segundo. El orden de guardado es hora, latitud, longitud, presión y caudal, separados por comas. Para el guardado se creó un archivo llamado DATOS.txt en modo escritura.

Tras finalizar la jornada el sistema de la bomba de fumigación pasará a modo transferencia de datos, que se activa presionando el botón de la bomba. Estando en modo transferencia se deja de recibir datos del GPS y se desactivan todos los sensores, luego se enciende el Bluetooth. Ya encendido el Bluetooth este esperara a que la central se conecte con él. Esto se verifica esperando el carácter "h", cuando el microcontrolador recibe el carácter pasa a modo transferencia, inicializando la transferencia de los datos guardados en memoria.

En modo transferencia, el microcontrolador coloca el puntero de la memoria SD y lo coloca en 0 para saber cuál es el tamaño del archivo a enviar para enviarle el tamaño a la central. Luego se inicializa la transferencia del archivo. Cuando la transferencia es finalizada se esperan 2 segundos para luego eliminar el archivo y duerme al microcontrolador para evitar gastos innecesarios de energía. El microcontrolador "durmiendo" consume tan solo 20nA de corriente.

Para verificar el programa en el microcontrolador se utilizaron datos ficticios enviados por serial desde una computadora. La cadena transmitida fue la siguiente:

\$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65.

Pero la información que nos interesa solo es la hora, longitud y latitud. El primer bloque de la cadena “GPGGA” nos dice cuál es el protocolo de la información, en este caso es GGA, seguido está la hora “064951.000” en el formato hhmmss.sss, el siguiente bloque “2307.1256” en el formato ddm. mmmm nos dice la latitud seguido por la dirección N (norte) o S (sur). Luego sigue la longitud “12016.4438” en el formato dddmm.mmmm seguido de E (este) o W (oeste). Tras haber enviado la cadena de datos ficticios se verificó en memoria si se habían guardado todos los datos de los sensores y que estuviesen en el orden antes dicho.

Luego se creó el programa del receptor en Python en la Raspberry pi en la “Figura 158: Diagrama de flujo programa de la Central” se puede observar el diagrama de flujo del programa en una forma resumida.

Figura 158: Diagrama de flujo programa de la Central



La programación se comenzó por la conexión bluetooth. Para esto se obtuvo la última versión de los drivers realizando el siguiente proceso:

Primero se abrió una terminal en la Raspberry pi, después se ejecutó el siguiente comando “sudo apt-get update” que actualiza la lista de los drivers no actualizados, luego se descargaron los drivers que no estaban actualizados ejecutando “sudo apt-get upgrade”.

Se instaló el software del bluetooth ejecutando “sudo apt-get install bluetooth bluez-utils blueman”, Luego se ejecutó el comando “hcitool scan” este muestra todos los dispositivos bluetooth en el rango, en nuestro caso se mostró el dispositivo “00:14:03:11:07:36”.

Se creó un programa en Python para poder conectarse al dispositivo “00:14:03:11:07:36”, el programa espera hasta que el dispositivo se encuentre en el rango de transferencia. Cuando lo encuentra se vinculan y espera 5s, luego le envía un carácter de seguridad para asegurarse de que existe la conexión (en nuestro caso es 'h'). El programa en el microcontrolador espera el carácter 'h' para inicializar la transferencia de los datos. La base de los datos que llegan a la central es de la forma hora, latitud, N/S, longitud, E/W, presión, caudal. Luego de que la transferencia haya finalizado se procede a separar todos los datos de entrada en diferentes listas para analizarlas por separado en los futuros pasos. Para cuando el GPS no manda coordenadas se creó un procedimiento en el que se guardan los datos de presión y caudal en listas temporales y realizar un promedio de todos y colocarlos con la coordenada de entrada. Para configurar la hora de entrada en horario Guatemala, se realizó restando 6 a la hora de entrada, para cuando la hora esta entre 0 a 5 se creó un procedimiento para colocarlo entre 18 a 23 respectivamente. Las coordenadas de GPS se transformaron en grados. Luego se analiza la presión y el caudal para colocarlos en una escala de 0 a 10. La escala nos dice que tan bueno ha sido la fumigación en donde 10 es excelente.

Ya con todos los datos analizados y en listas por separado se creó un procedimiento para colocar toda la información en un archivo con extensión .csv y de nombre finca1. Los datos se guardaron en el siguiente orden: Finca.Fumogua, 1, fecha, hora, latitud, longitud, presión medida, porcentaje de presión, caudal medido, porcentaje de caudal, escala. Luego se envía a la base de datos desde el puerto 22 hacia la dirección 107.170.7.174 en la carpeta opt/megaproyecto/geodjango/datos.

C. RESULTADOS

1. SENSORES SELECCIONADOS

Cuadro 33. Sensor de presión

Medición PSI	Dimensiones	Salida digital	Voltaje de Alimentación
0 – 150	4” largo 1/4” diámetro	17 – 89	5

Cuadro 34. Sensor de caudal

Medición flujo	Dimensiones	Diámetro de la boquilla	Flujo mínimo Hz	Flujo máximo Hz	Voltaje de Alimentación
Hz	5.5 x 2 cm	3/8"	0	20	5v

Cuadro 35. Adafruit ultimate GPS

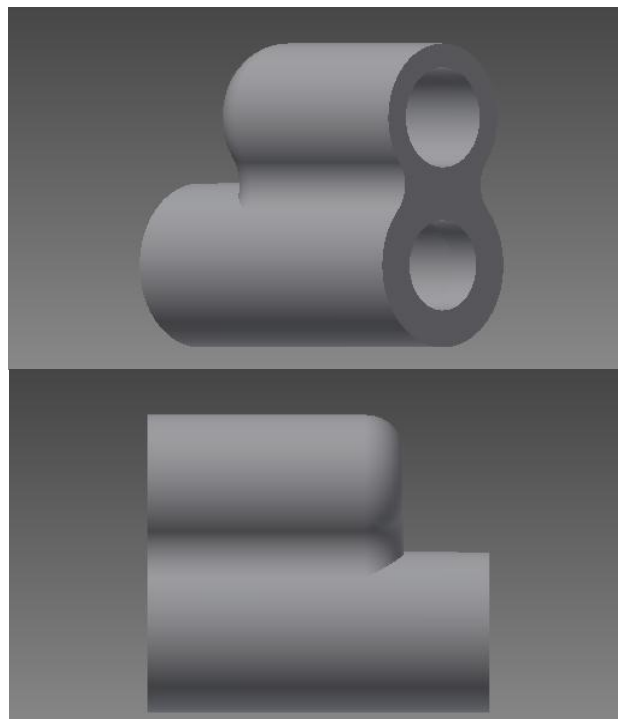
Voltaje de alimentación	Corriente de alimentación	Frecuencia	Salida
5	20mA	1 Hz	GGA

Cuadro 36. Bluetooth HC-05

Voltaje de alimentación	Corriente de alimentación	Conexión	Baud Rate
5	25mA	UART	9600

2. ACOUPLE

Figura 159: Acople de sensores

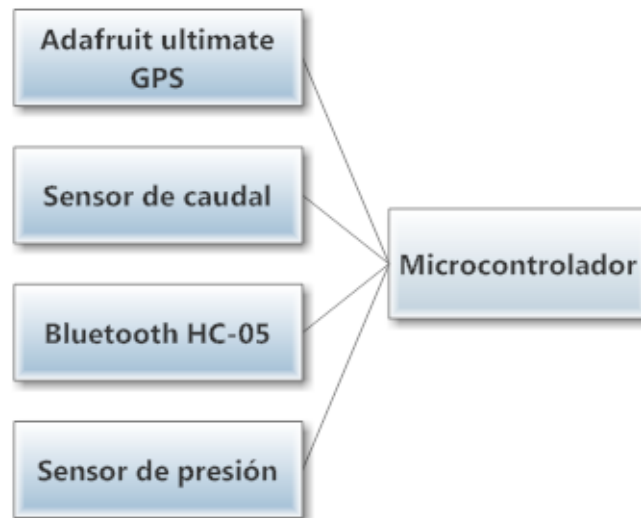


3. CARACTERÍSTICAS ELÉCTRICAS

Cuadro 37: Parámetros de operación

Voltaje máximo	30v
Voltaje mínimo	7v
Voltaje recomendado	9v
Corriente máxima	70mA
Corriente mínima	38mA
Corriente típica	58mA

Figura 160: Conexiones hacia el microcontrolador



4. PRUEBAS DE COMUNICACIÓN

Cuadro 38. Prueba de datos enviados desde el microcontrolador

Hora	Latitud	N/S	Longitud	E/ W	Presión	Caudal
000005.000	1436.3372	N	09029.3040	W	29	0
000006.000	1436.3382	N	09029.3038	W	26	0
000007.000	1436.3384	N	09029.3037	W	36	0
000008.000	1436.3386	N	09029.3033	W	41	11
000009.000	1436.3380	N	09029.3036	W	39	10
000010.000	1436.3374	N	09029.3039	W	45	12
000011.000	1436.3369	N	09029.3043	W	54	17

Cuadro 39. Prueba final de datos enviados desde el microcontrolador

Hora	Latitud	N/S	Longitud	E/ W	Presión	Caudal
161807.000	1437.2904	N	09027.4633	W	32	3
161808.000	1437.2867	N	09027.4634	W	59	8
161809.000	1437.2872	N	09027.4630	W	55	8
161810.000	1437.2874	N	09027.4628	W	51	8
161811.000	1437.2870	N	09027.4628	W	49	8
161812.000	1437.2866	N	09027.4628	W	47	7
161813.000	1437.2865	N	09027.4628	W	47	8
161814.000	1437.2864	N	09027.4628	W	53	10
161815.000	1437.2865	N	09027.4627	W	58	10
161816.000	1437.2866	N	09027.4627	W	60	10
161817.000	1437.2867	N	09027.4625	W	69	10
161818.000	1437.2839	N	09027.4627	W	70	11
161819.000	1437.2842	N	09027.4613	W	63	10
161820.000	1437.2843	N	09027.4612	W	59	8
161821.000	1437.2843	N	09027.4622	W	55	8
161822.000	1437.2818	N	09027.4627	W	52	7
161823.000	1437.2661	N	09027.4649	W	49	7
161824.000	1437.2659	N	09027.4649	W	47	7
161825.000	1437.2659	N	09027.4650	W	46	7

Cuadro 40. Prueba final datos que son enviados a la base de datos

finca	bomba	fecha	hora	latitud	longitud	presión	p_presión	caudal	p_caudal	escala
Finca.HaciendaReal	1	18/09/2014	08:31:29	14.62128	-90.4575	32	4.6	3	1.9	3.3
Finca.HaciendaReal	1	18/09/2014	08:31:30	14.62128	-90.4575	59	10	8	9.9	10
Finca.HaciendaReal	1	18/09/2014	08:31:31	14.62129	-90.4575	55	10	8	9.9	10
Finca.HaciendaReal	1	18/09/2014	08:31:32	14.62129	-90.4575	51	9.2	8	9.9	9.6
Finca.HaciendaReal	1	18/09/2014	08:31:33	14.62129	-90.4575	49	8.7	7	7.7	8.2
Finca.HaciendaReal	1	18/09/2014	08:31:34	14.6213	-90.4575	47	8.1	7	7.7	7.9
Finca.HaciendaReal	1	18/09/2014	08:31:35	14.62131	-90.4575	47	8.1	8	9.9	9
Finca.HaciendaReal	1	18/09/2014	08:31:36	14.62131	-90.4575	53	9.8	10	0	4.9
Finca.HaciendaReal	1	18/09/2014	08:31:37	14.62131	-90.4575	58	10	9	0	5
Finca.HaciendaReal	1	18/09/2014	08:31:38	14.6213	-90.4575	60	10	9	0	5
Finca.HaciendaReal	1	18/09/2014	08:31:39	14.6213	-90.4575	65	0	10	0	0
Finca.HaciendaReal	1	18/09/2014	08:31:40	14.62129	-90.4575	69	0	11	0	0
Finca.HaciendaReal	1	18/09/2014	08:31:41	14.62128	-90.4575	70	0	10	0	0
Finca.HaciendaReal	1	18/09/2014	08:31:42	14.62129	-90.4575	63	0	8	9.9	5
Finca.HaciendaReal	1	18/09/2014	08:31:43	14.62129	-90.4575	59	10	8	9.9	10
Finca.HaciendaReal	1	18/09/2014	08:31:44	14.62129	-90.4575	55	10	8	9.9	10
Finca.HaciendaReal	1	18/09/2014	08:31:45	14.62129	-90.4575	52	9.5	7	7.7	8.6
Finca.HaciendaReal	1	18/09/2014	08:31:46	14.62128	-90.4575	49	8.7	7	7.7	8.2
Finca.HaciendaReal	1	18/09/2014	08:31:47	14.62128	-90.4575	47	8.1	7	7.7	7.9

5. GRÁFICAS DE PRUEBAS REALIZADAS

Figura 161: Primera prueba real realizada

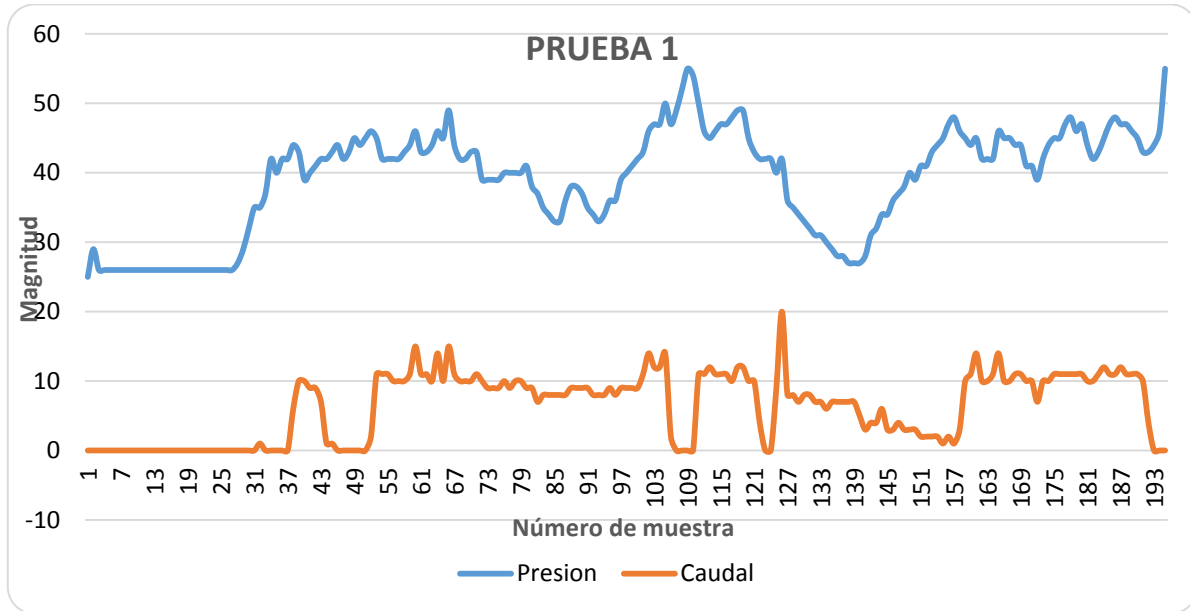
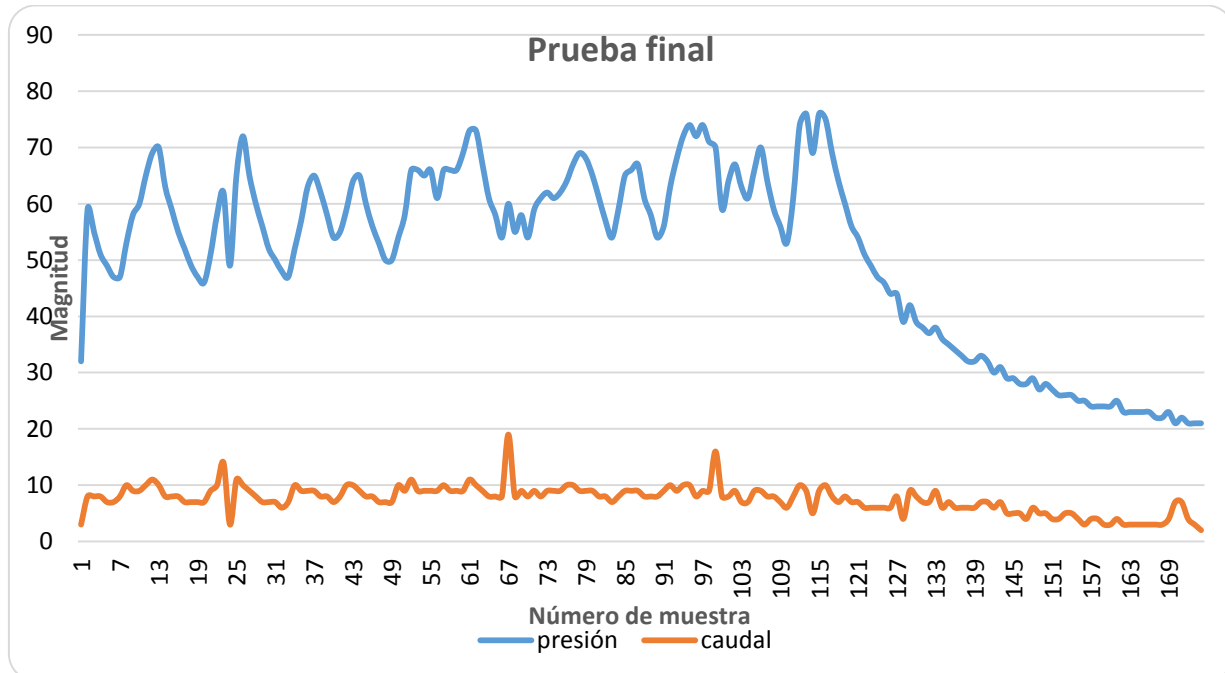


Figura 162: Prueba final con datos reales



6. MAPEO DE DATOS

Figura 163: Función para transformar los datos de la presión en una escala.

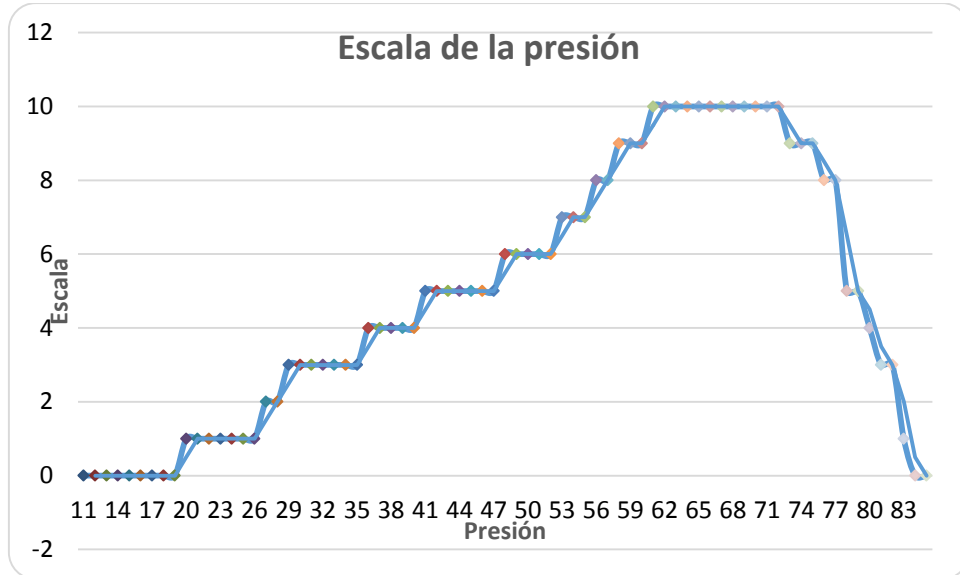


Figura 164: Primera parte de la función escala – presión

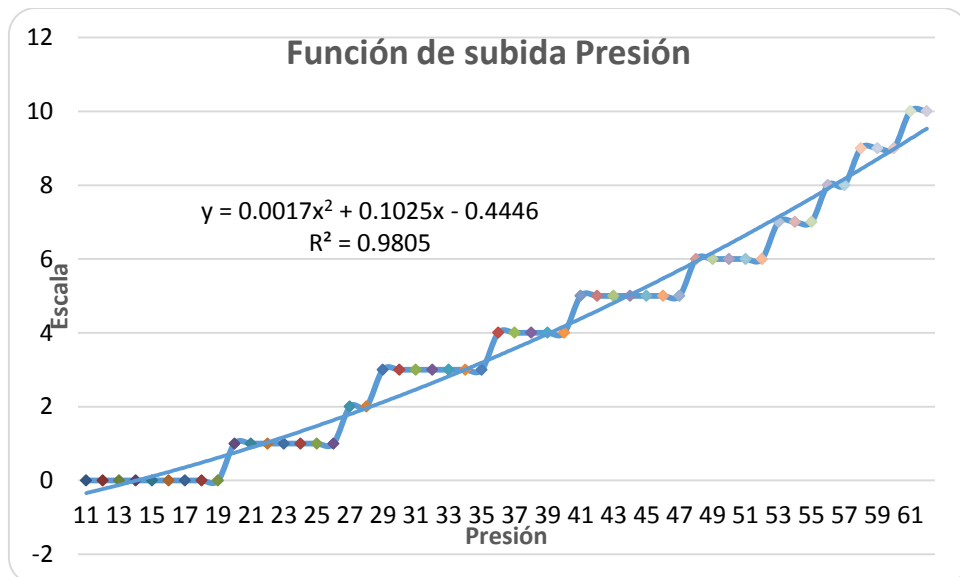


Figura 165: Segunda parte de la función escala – presión

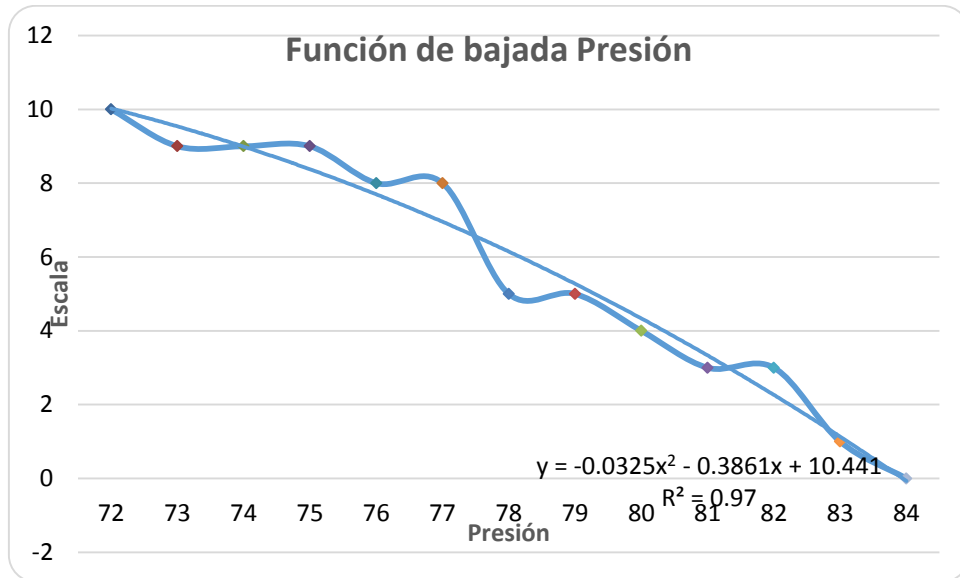


Figura 166: Función para transformar los datos de caudal en una escala

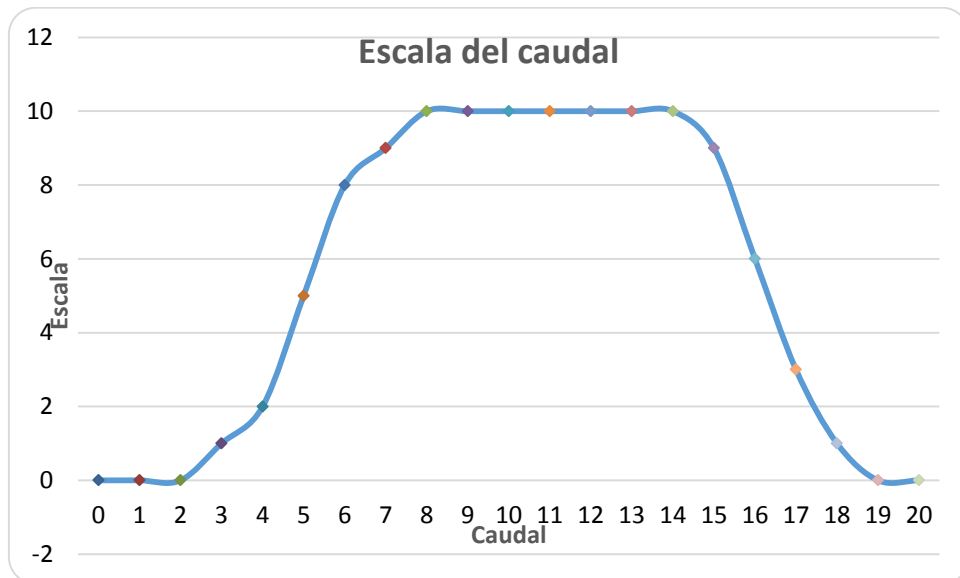


Figura 167: Primera parte función escala – caudal

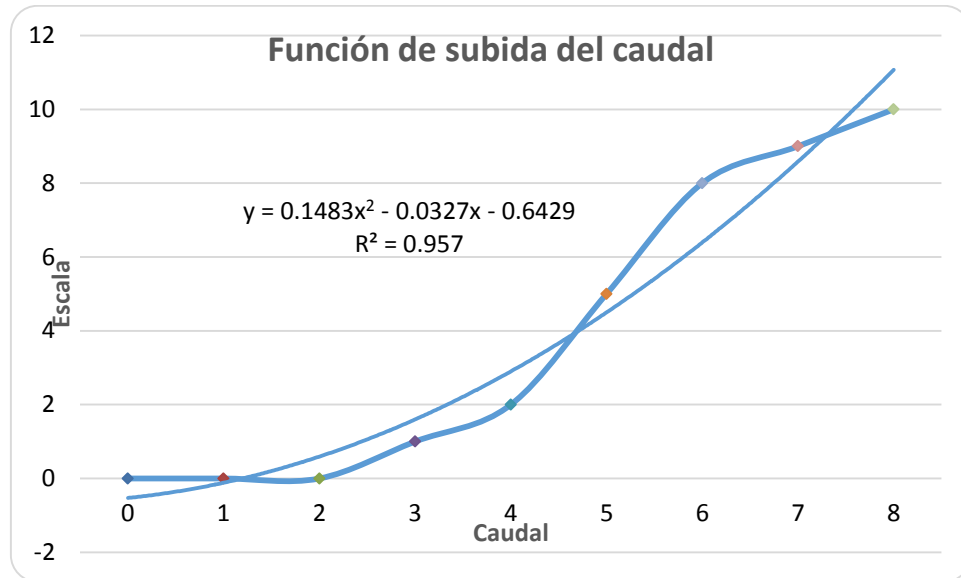
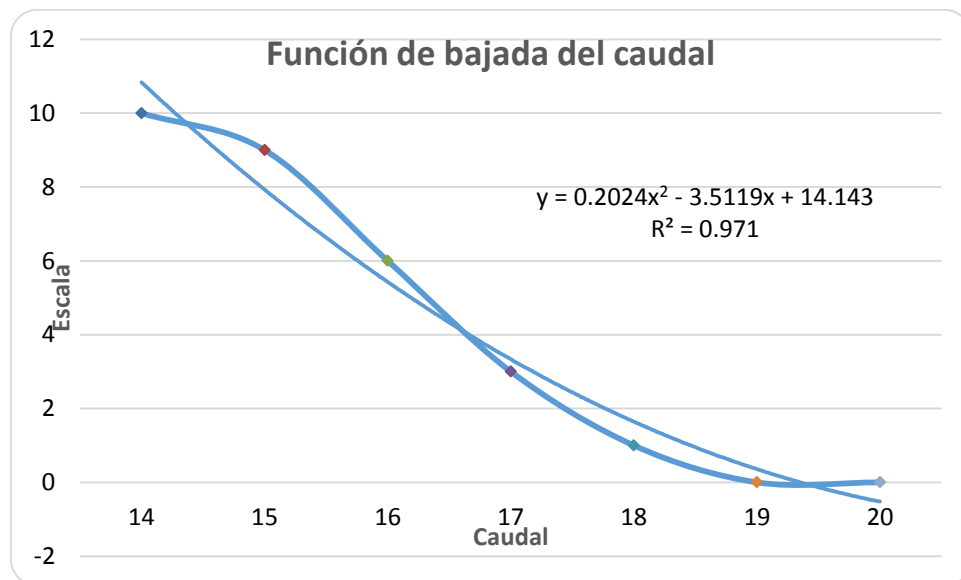


Figura 168: Segunda parte de la función escala – caudal



7. GRÁFICAS DE ESCALAS

Figura 169: Muestra la escala de la presión en el rango de 0 a 10

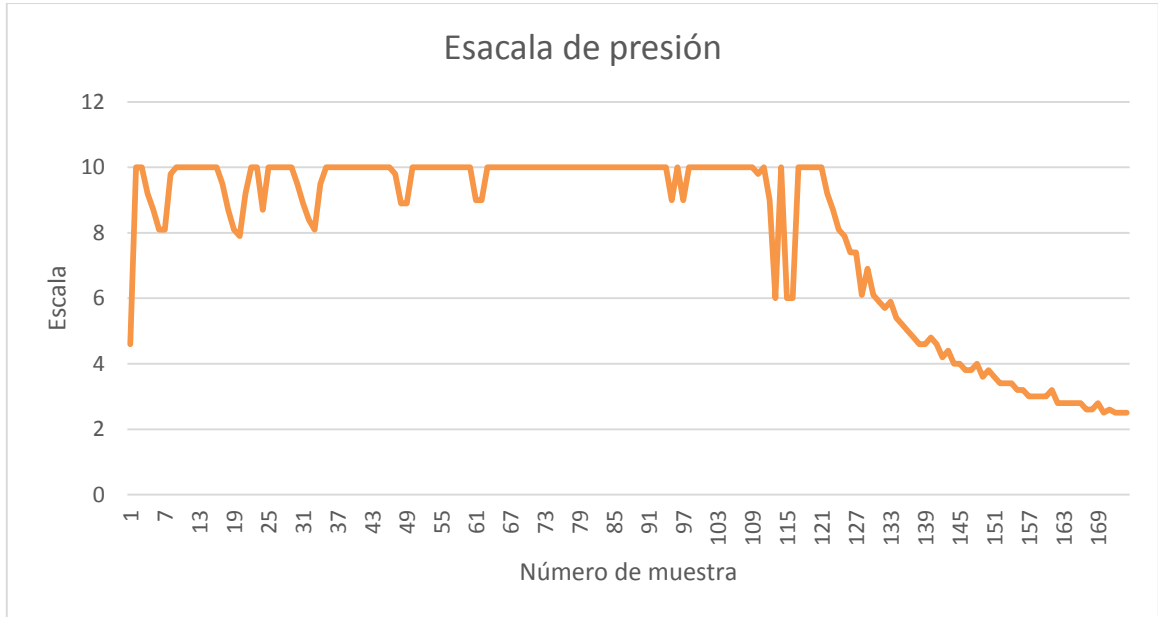


Figura 170: Escala del caudal

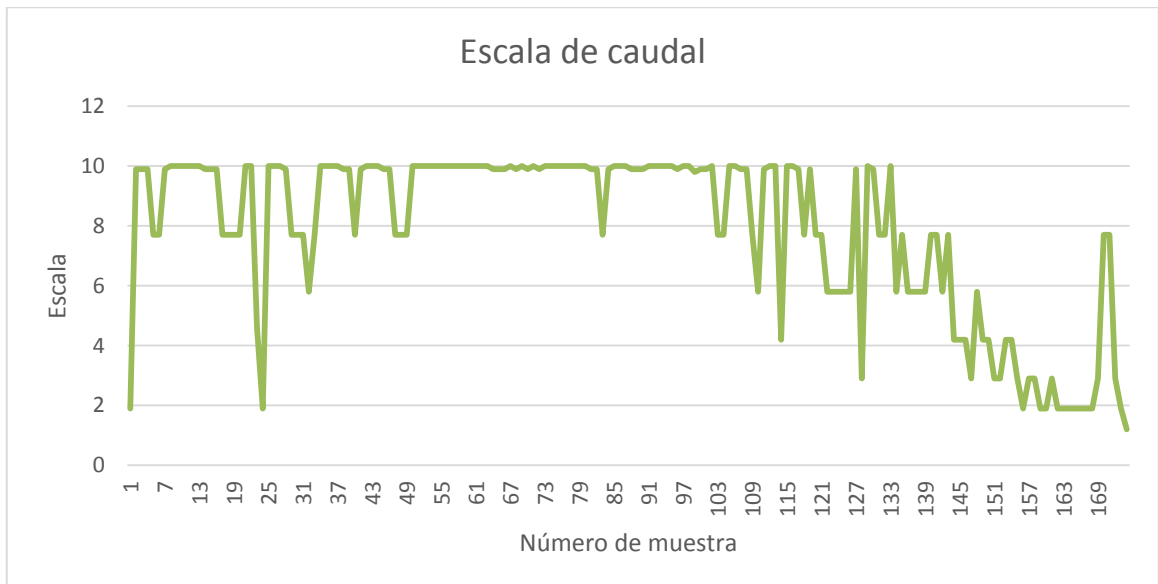
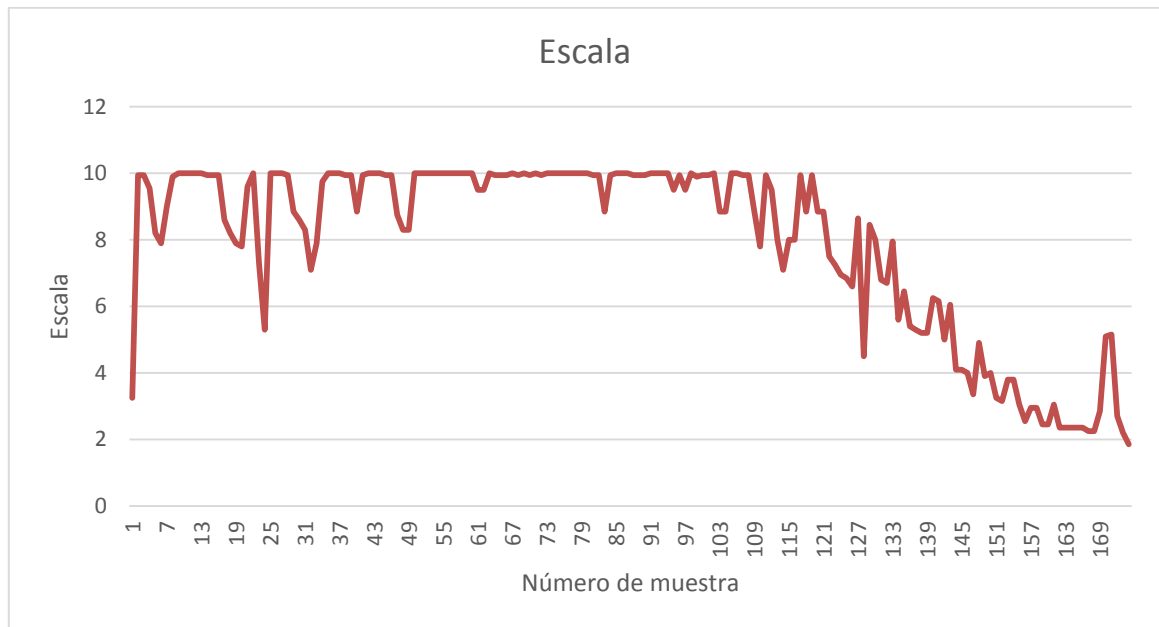


Figura 171: Escala después de analizar la escala de la presión y el caudal



D. ANÁLISIS DE RESULTADOS

Para determinar el sensor de presión se encontraron varios problemas porque no podía ser cualquier sensor, porque tiene que ser resistente al agua y soportar una presión de 100psi que es la presión máxima de la bomba. Inicialmente se probó el sensor Honeywell que por sus dimensiones más pequeñas se prefirió ante el Sensor 100psi ver “Cuadro 33. Sensor de presión”. Pero se presentaron problemas al querer amplificar la señal puesto que la salida variaba desde 2v hasta 2.1v por lo que se tenía que amplificar la señal, pero antes se tuvo que rectificar colocando los 2v que se tenía como el mínimo voltaje de salida en 0v. Para esto se utilizaron amplificadores operacionales con alimentación +5 y -5. Ya con el mínimo voltaje de salida en 0 se amplificó por 50 para que la salida estuviese entre 0 y 5v. Pero se llegó a la conclusión de cambiar el sensor Honeywell al Sensor 100psi ya que la fase de rectificación iba a complicar el módulo de alimentación, siendo necesario tener +5v y -5v (el Sensor 100psi ya cuenta con una salida rectificada y amplificada alimentándolo solo con 5v).

Para el sensor de caudal primero se realizaron pruebas con un sensor que tenía un diámetro de entrada de 1/2” porque se encontraba en el mercado local y presentaba un caudal mínimo de 0.3l por minuto lo cual era excelente para nuestra aplicación. Pero en la práctica presentó una salida en la señal muy restringida, ya que la frecuencia de salida estaba por debajo de los 4Hz. Esta señal no presentó una buena resolución por lo que se decidió por cambiar el sensor a uno que tuviese la entrada de 3/8” ver “Cuadro 34. Sensor de caudal”. Con esto se logró obtener una señal con una mejor resolución llegando hasta los 40Hz.

Cuando se realizaron las primeras pruebas con el GPS ver “

Cuadro 35. Adafruit ultimate GPS” se tuvo la complicación de que manda todos los formatos simultáneamente, pero solo se necesita uno, el GGA, por lo que se aisló de manera que los demás formatos no se guardan en ningún lugar. Otro problema que se tuvo fue que de la lista guardada solo se necesitó la latitud, longitud y hora. Entonces se determinó que la mejor forma de realizarlo es guardar la lista de datos y luego recolectar los datos que se necesitan, para no perder datos en la recepción.

Cuando se comenzó a investigar para realizar la comunicación inalámbrica, desde la bomba hacia la central, se tenían 2 opciones, por medio de RF o por bluetooth. Pero por su simpleza, su precio y para tener una transferencia de datos más segura se optó por probar la comunicación bluetooth ver “Cuadro 36. Bluetooth HC-05”. En las primeras pruebas el objetivo era que se conectaran de forma automática sin necesidad de presionar un botón para que comenzara la transmisión de datos, pero no se tuvo éxito con las pruebas realizadas así que lo que se concluyó fue de colocar un botón para apagar el GPS y encender el modulo bluetooth para transferir datos.

El módulo HC-05 tiene el problema que la salida TX y entrada RX solo soporta 3.3v por lo que se colocó un divisor de voltaje en la entrada RX utilizando una resistencia de 2.2k Ω y otra en serie de 1k Ω con esto se logró que el voltaje de 5v que sale del microcontrolador bajara a 3.3v. Para la salida TX se tuvo que realizar algo distinto ya que se tiene que subir de 3.3 a 5v, por lo que se colocaron 2 transistores en configuración NOT en serie. Con esto se logró un acople de voltajes desde el módulo HC-05 hacia el microcontrolador.

Cuando se probó la comunicación del bluetooth y el GPS se tuvo el problema que ambos utilizaban transmisión serial por UART y el microcontrolador PIC18F45K22 solo cuenta con un módulo UART por lo que si se conecta RX y TX de ambos módulos podría existir una colisión de datos entonces se colocaron relés de estado sólido para separar las terminales de los módulos, estos se activaban por medio del microcontrolador cuando desactiva el GPS y enciende el Bluetooth.

Antes de poder ensamblar los sensores se probó el acople, que se puede observar en la “Figura 161: Primera prueba real realizada”. Se tuvo el problema de que la impresora 3D no puede realizar modelos resistentes al agua. Cuando se habla de resistencia al agua quiere decir que a presiones arriba de 10psi se tienen fugas en el modelo. Por lo anterior dicho se investigaron varias formas para impermeabilizar el acople por lo que el mejor resultado que se obtuvo fue utilizar TAPAPORO. También se encontró con un problema cuando se realizaron las pruebas. Las mangueras conectadas al acople se soltaban saliendo disparadas por la presión del agua, por lo que se re-diseñó el acople con orificios más pequeños creando una sujeción más fuerte de tal manera que para colocar los sensores se tuvo que calentar el acople a 80 °C. De esta forma se aseguraron los sensores y la manguera de forma segura.

Ya todo ensamblado se inicializaron las pruebas, los resultados de la primera prueba están en el “Figura 161: Primera prueba real realizada” en la gráfica se puede observar cómo fueron cambiando los datos del

sensor de presión y el sensor de caudal. Esta primera prueba nos ayudó a darnos una idea de cuales iban a ser los datos reales para inicializar con los análisis.

En el “Figura 162: Prueba final con datos reales” se puede observar los datos obtenidos del sensor de caudal y presión, también si se comparan la “Figura 161” con “Figura 162” se puede observar que en la “Figura 161” el flujo cambia repentinamente y en la “Figura 162” no. Esto se debe a que cuando se realizaron las pruebas finales se fumigó con un flujo constante por lo que se bombeaba mientras se fumigaba.

En el “Cuadro 38” se puede observar una serie de datos, en ese formato se guardaron los datos en la SD. No se guardaron todos los datos de entrada para optimizar el tiempo para guardar ya que para un bloque de datos se necesita un tiempo de 480ms, si se hubieran guardado todos los datos de entrada el tiempo de guardado hubiera incrementado considerablemente quitando tiempo para tomar datos de los sensores. Cuando se está guardando se quitaron las interrupciones para no interferir.

En el “Cuadro 39. Prueba final de datos enviados desde el microcontrolador” se puede observar algunos datos de la prueba final con la finalidad de comparar lo que llega a la central y los datos que salen ver Cuadro 40.

Ya con las pruebas realizadas se determinó el rango óptimo de operación para la presión y el caudal, y realizando una gráfica que se puede observar en “Figura 166: Función para transformar los datos de caudal en una escala” Para la presión y “Figura 166: Función para transformar los datos de caudal en una escala” para el caudal. Ya que se necesitaba crear una función para determinar la escala de la presión se probó insertando la función a un programa de matemática llamado “MatLAB” pero este no logró encontrar una función que representara a la gráfica por lo que se tuvo que separar en partes la primera se puede observar en “Figura 164: Primera parte de la función escala – presión” y la otra parte ver “Figura 165: Segunda parte de la función escala – presión” ya con 2 graficas se utilizó el programa Excel para encontrar una función. De forma similar se realizó con el caudal solo que las gráficas son distintas ver “Figura 167: Primera parte función escala – caudal” y “Figura 168: Segunda parte de la función escala – caudal”. Para determinar la mejor aproximación de las curvas se probaron distintas líneas de tendencia siendo la polinómica de orden 2 la que mejor aproximación dio y menos compleja, siempre se buscó estar arriba de 95% de exactitud. Las formulas obtenidas de los gráficos (las formulas se pueden observar en Figura 164, Figura 165, Figura 167y Figura 168) se programaron en la central teniendo el cuidado de que como son aproximaciones no siempre se tendrá un valor entre 0 y 10 por lo que se realizó un procedimiento para que no sea mayor que 10 y no menor de 0.

Se puede observar en los “Figura 169”, “Figura 170” y “Figura 171” la influencia de la escala de la presión y del caudal. En el “Figura 171: Escala después de analizar la escala de la presión y el caudal” se puede notar la escala final, está se determino realizando un promedio entre la presión y el caudal.

E. CONCLUSIONES

1. Se diseñó y creó un sistema para recolectar información de la operación de la bomba y el sistema de transmisión de esta información a los servidores informáticos.
2. Se logró crear un acople para el sensor de caudal y presión.
3. Se guardaron 6 datos por segundo, con esto se logró crear una muestra fiable de datos.
4. Los datos fueron guardados de una forma efectiva.
5. Se logró determinar los sensores necesarios para realizar el monitoreo en la bomba.
6. Se creó un programa para recolectar y procesar los datos.
7. Se logró establecer una comunicación inalámbrica para comunicar la bomba y el módulo receptor.

F. RECOMENDACIONES

- Se recomienda realizar una conexión inalámbrica que sea automática para evitar posibles daños y pérdida de datos al no presionar el botón.
- Conectar un módulo cargador de baterías dentro del sistema para evitar el cambio de las mismas.
- Impermeabilizar la caja en donde va el microcontrolador evitando daños por agua.
- Crear una verificación de redundancia cíclica (CRC) para tener control de la integridad de los datos al ser transmitidos inalámbricamente.

IX. MÓDULO FUMOGUA DISEÑO E IMPLEMENTACIÓN DE BASE DE DATOS Y VISUALIZACIÓN EN PÁGINA WEB

A. METODOLOGÍA

El desarrollo de la base de datos y página web se realizará por medio de la obtención de información por medio de una investigación, que permitirá profundizar sobre los temas a emplear en este módulo. El centro de la investigación que se realizará será en el framework web Django, el cual permitirá la manipulación de datos, para poderlos desplegar en la página web.

Otros temas a profundizar serán los APIs que ofrece Google, para poder implementar el área de visualización de datos en Google Maps y Google Chart, ya que se desplegará la información en un mapa, tabla y gráfico. Se obtendrá información sobre la base de datos Postgresql y su módulo geoespacial PostGIS, para conocer cómo se relaciona esta base de datos con Django y poder tener los datos necesarios para que la visualización en los mapas sea la correcta. Por último se investigará cual es el hosting web más económico, para poder montar página web.

Teniendo más conocimiento sobre los temas a utilizar en el módulo, se instalarán todas las herramientas necesarias en el host y se creará un proyecto de Django, después se creará la base de datos y se conectará con Django para obtener datos de prueba. Con esta conexión se realizarán las páginas Web que desplegarán la información de las bases de datos.

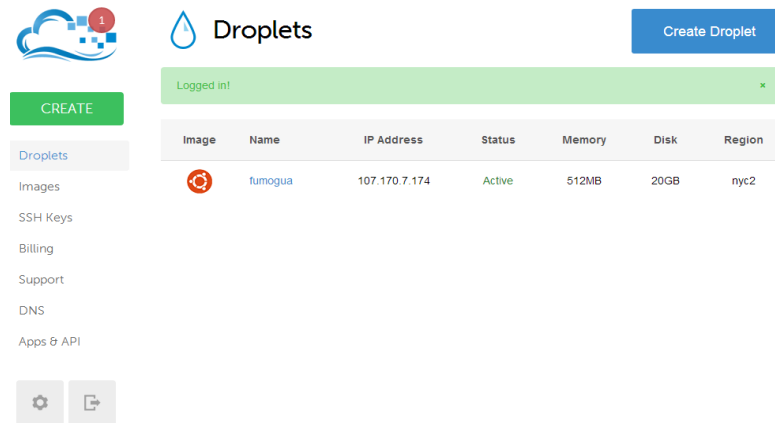
Se incorporarán las APIs de Google Maps para la visualización de datos, para que los usuarios puedan ver los recorridos que realizaron sus empleados y las zonas donde existe roya. Además se realizará un mapa para Anacafé para que tengan un control de los puntos, donde las fincas que cuenten con nuestro servicio, tengan roya.

Se crearán usuarios con el Administrador de Django, para que únicamente las personas registradas, puedan tener acceso a la información de su finca. Para los trabajadores de Anacafé, se contará con un usuario para que tengan acceso a la información de las zonas con roya de las fincas.

B. DESARROLLO

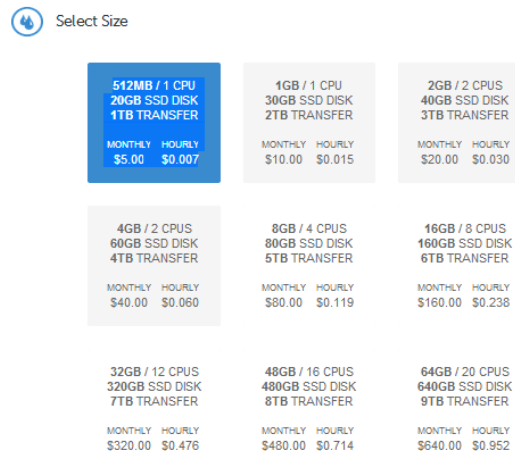
1. Configuración del host y dominio. El host que se utilizó fue Digital Ocean, este ofrece soporte para Django, además tiene un precio bajo respecto a otros host. Se creó un droplet, al cual se le colocó el nombre de fumogua y sistema operativo Ubuntu 13.04, debido a que las pruebas con django se habían hecho en este sistema operativo.

Figura 172: Droplets en DigitalOcean



Se eligió una forma de pago, como se muestra en la Figura 173, en el cual el cobro es de \$5.00 mensuales obteniendo 20GB de disco duro sólido y 512MB de RAM. Esta opción se eligió debido a que es la más económica, y tiene espacio suficiente para almacenar todos los archivos del proyecto.

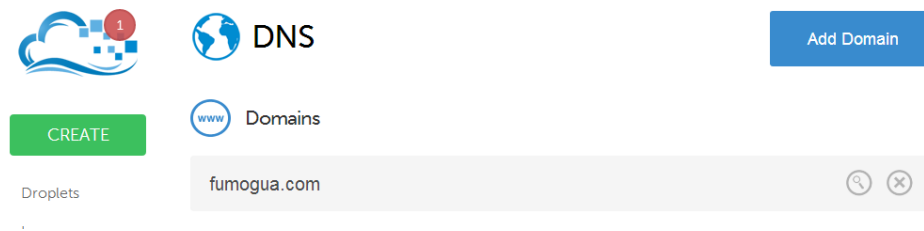
Figura 173: Selección de tamaño de Droplet



Una vez creado el droplet se realizó la compra del dominio en GO Daddy, el nombre del dominio que se eligió fue fumogua.com. Luego se actualizaron los DNS del dominio, para que cuando se

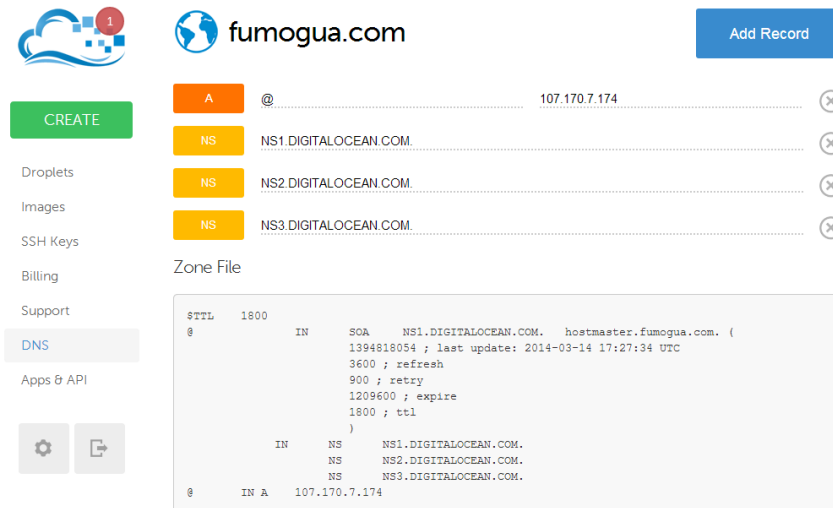
ingrese a fumogua.com, este apunte hacia el host de DigitalOcean. Para obtener el DNS, se ingresó a DNS en DigitalOcean, se añadió un dominio.

Figura 174: Añadir dominio



Cuando se añadió el dominio, se hizo clic sobre la lupa que aparece a un lado de fumogua.com, este despliega los DNS que se muestran en la Figura 175.

Figura 175: DNS en DigitalOcean



Conociendo los DNS, se ingresó a los detalles del dominio fumogua.com en GoDaddy.com, en las opciones del dominio, se hizo clic sobre manage de NameServers. En el cuadro de dialogo de estas opción, se escogió Custom y se añadieron NameServers que corresponden a los DNS obtenidos en DigitalOcean.

Figura 176: Nameserver Settings en GoDaddy

Nameserver Settings

FUMOGUA.COM

Nameservers point your domain to where it is located.

Setup type:

Standard
Go Daddy hosting, forwarding, and parked domains.

Custom
Customizable nameserver settings.

Nameservers: [Edit Nameservers](#)

#	Nameserver	Status
1	NS1.DIGITALOCEAN.COM	Active
2	NS2.DIGITALOCEAN.COM	Active
3	NS3.DIGITALOCEAN.COM	Active

[Add Nameserver](#)

[Save](#) [Cancel](#)

2. Acceso al host. Se utilizó PuTTY para conectarse al servidor remoto, el cual es el droplet que se creó anteriormente. Este programa se conecta por medio de SSH, para realizar el enlace se requiere conocer el Host Name o IP Address, esta información es obtenida por medio de un correo, enviado a la dirección con la cual se registró en DigitalOcean.com, cuando se crea el droplet. El puerto que se utiliza es el 22, debido a que este es el recomendado para utilizar con SSH.

Figura 177: Configuración de PuTTY

PuTTY Configuration

Category:

- Session
 - Logging
- Terminal
 - Keyboard
 - Bell
- Features
- Window
 - Appearance
 - Behaviour
 - Translation
 - Selection
 - Colours
- Connection
 - Data
 - Proxy
 - Telnet
 - Rlogin
 - SSH
 - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

Connection type:

Raw Telnet Rlogin SSH Serial

Load, save or delete a stored session

Saved Sessions

Fumogua

Default Settings [Load](#)

Fumogua [Save](#) [Delete](#)

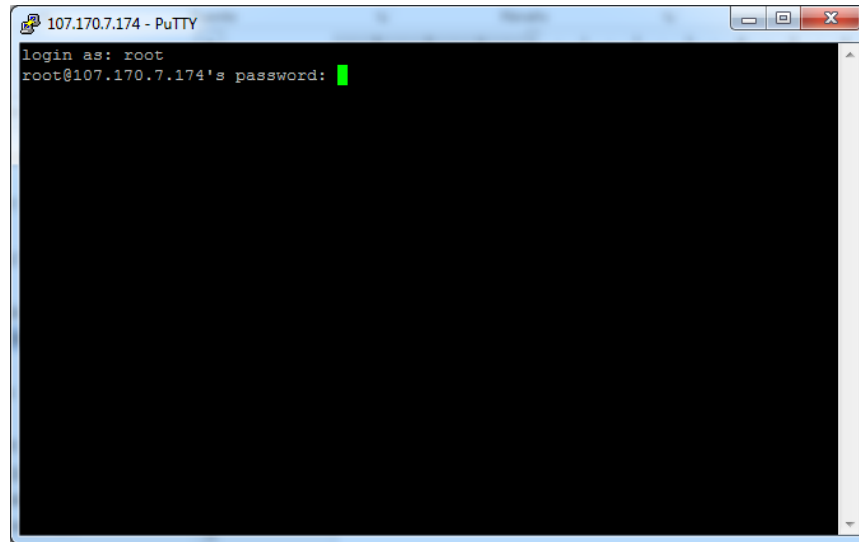
Close window on exit:

Always Never Only on clean exit

[About](#) [Open](#) [Cancel](#)

Al hacer clic en Open, se abre una ventana en donde se pide el usuario (login as) y password, ésta información se encuentra también en el correo mencionado anteriormente. En este caso el usuario que se utilizó fue root y la contraseña generada por DigitalOcean.

Figura 178: ventana PuTTY, ingreso usuario y password



3. Instalación y cómo crear un nuevo proyecto en Django. Realizado el Log In en PuTTY, se procedió a instalar Django, por medio del comando “pip install Django = 1.6.4” para obtener la versión 1.6.4. Como pip no se encontraba instalado, se realizó la instalación del mismo, por medio del comando “sudo apt-get install python-pip” con esto se escribió de nuevo el comando de Django y la instalación fue correcta.

Se creó un proyecto, en la dirección /opt/megaproyecto/geodjango/, para tener todos los archivos en un lugar específico. Por medio del comando `django-admin.py startproject geodjango`, se crea un nuevo proyecto el cual contiene una carpeta, esta contiene las configuraciones del proyecto. Para crear la aplicación, la cual contiene los modelos y las vistas, se utiliza el comando `python manage.py startapp world`, world es el nombre de la aplicación que se creó. Se debe tener en cuenta que para ejecutar el comando anterior, se debe estar dentro de la carpeta en donde se encuentra el archivo `manage.py`.

Se realizó la prueba para observar que la aplicación estaba funcionando, por medio de `python manage.py syncdb`, el cual pide un user (fumogua) y un password, este es el usuario de django que sirve como administrador del proyecto. Luego se utilizó `python manage.py runserver`, entrando a `fumogua.com`, se obtuvo la Figura 179, en donde se muestra un mensaje que el proyecto está funcionando.

Figura 179: página obtenida cuando funciona Django



4. Instalación de GUNICORN, NGINX, POSTGRESQL y SUPERVISOR. Se instaló Gunicorn por medio del comando `pip install gunicorn`, así como Nginx con el comando `sudo apt-get install nginx`. Se realizó la configuración por medio de los pasos que se muestran a continuación.

1. `sudo nano /etc/nginx/sites-available/geodjango`
2. Se añadió en el documento el código

```
server {
    listen 80;
    server_name fumogua.com;
    access_log /var/log/nginx/example.log;
    location /static/ {
        alias /opt/megaproyecto/geodjango/world/static/;
    }
    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

3. `cd /etc/nginx/sites-enabled`
4. `sudo ln -s ../sites-available/geodjango`
5. `sudo service nginx start`

6. `sudo service nginx restart`

Se instaló y configuró supervisor por medio de los siguientes pasos.

1. `apt-get install supervisor`
2. `service supervisor restart`
3. `sudo nano /etc/supervisor/conf.d/fumogua.conf`
4. Se añadió el código.

```
[program:fumogua]
command=python manage.py runserver
directory = /opt/megaproyecto/geodjango
autostart=true
autorestart=true
stderr_logfile=/var/log/long.err.log
stdout_logfile=/var/log/long.out.log
```

5. `supervisorctl reread`
6. `supervisorctl update`

La base de datos que se utilizó fue PostgreSQL junto con PostGIS, estos paquetes se instalaron por medio del comando `sudo apt-get install postgresql postgis`. Se creó la base de datos por el comando `createdb prueba`, `prueba` fue el nombre que se le dio a la base de datos del proyecto. Para añadirle la extensión de PostGIS a la base de datos, se accedió a ella por medio de `psql prueba`, luego se utilizó `CREATE EXTENSION postgis;` y `CREATE EXTENSION postgis_topology;` es importante agregar el punto y coma luego de cada instrucción para que postgresql lo reconozca.

5. Configuración del proyecto en Django. Con la base de datos creada, se realizaron los cambios a la configuración del proyecto en django, este archivo es `settings.py` que se encuentra en la carpeta `geodjango`. Las modificaciones que se realizaron a este archivo, fueron las siguientes:

- Se añadieron las instrucciones que permiten enviar EMAIL desde Django.

```
EMAIL_USE_TLS = True
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_HOST_USER = 'fumogua@gmail.com'
EMAIL_HOST_PASSWORD = '*****'
EMAIL_PORT = 587
```

- Se añadió la información de la base de datos para que Django la reconozca.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.contrib.gis.db.backends.postgis',
        'NAME': 'prueba',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
```

- Se agregó la dirección en donde están colocados los archivos estáticos del proyecto.

```
import os
PROJECT_ROOT = os.path.realpath(os.path.dirname(__file__))
STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
STATIC_URL = '/static/'
```

- Se agregaron las aplicaciones que se iban a utilizar, como es postgis, world y gunicorn.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.admin',
    'django.contrib.admindocs',
    'django.contrib.gis',
    'world',
    'gunicorn' )
```

6. **Uso de modelos en Django.** Con la configuración del proyecto terminada, se realizaron los modelos de la aplicación, para realizar estos, se modificó el archivo `models.py` que se encuentra en la carpeta `world`. Los modelos son las tablas que se crearán en la base de datos, para almacenar la información que se desea. Las tablas que se crearon en la base de datos son las siguientes:

- Tabla `world_fincas_roya`, esta se creó con el objetivo de almacenar las coordenadas, fecha y nombre de la imagen donde se encontró roya. Esto para poder desplegar un marcador con la imagen de la hoja, tomada en ese punto.
 - Finca: este campo se colocó para que se pueda desplegar la información dependiendo del usuario que está viendo su información.
 - Fecha: este campo se colocó para que se pueda desplegar la información dependiendo de la fecha que desea ver el usuario.
 - Latitud: este campo es necesario para conocer la latitud en donde se encontró el punto de roya.
 - Longitud: este campo junto con el anterior, da la ubicación en donde se encontró el punto de roya para poder desplegar la información.
 - Nombre de la imagen: este campo sirve como un identificador para desplegar la imagen correcta, en el punto en donde se encontró roya.
- Tabla `world_datos_fincas`, esta se creó para almacenar el número de bomba, fecha, hora, coordenada, presión, caudal y escala del recorrido que realizó el jornalero en el día.
 - Finca: este campo se utilizó para poder identificar a que finca pertenecen los datos.
 - Bomba: este campo se utilizó para poder identificar la bomba que hizo el recorrido trazado.
 - Fecha: este campo es para identificar la fecha en el que fue tomado el dato.
 - Hora: este campo es para conocer la hora en que el dato fue obtenido y que el caficultor pueda tener conocimiento de la hora en la que trabajan mejor sus jornaleros.
 - Latitud: este campo es para conocer la latitud del punto en donde estuvo el jornalero.
 - Longitud: este campo es para conocer la longitud y junto con la latitud poder dar la ubicación en donde fueron recolectados los datos.
 - Presión: este campo es para conocer la presión promedio que tenía la bomba.

- P_presion: este campo es para tener una escala de presión que se obtiene transformando el dato de presión a una escala de 1 a 10. Este proceso se realiza en el módulo de sensores que sube datos al host.
- Caudal: este campo sirve para conocer el caudal promedio al que estaba bombeando el jornalero.
- P_caudal: este campo es para tener una escala de caudal que se obtiene transformando el dato de caudal a una escala de 1 a 10. Este proceso se realiza en el módulo de sensores que sube datos al host.
- Escala: este campo se obtiene a partir del promedio de la escala presión y caudal para dar un número entre 1 a 10, para dar a conocer como uso la bomba el jornalero. Este proceso se realiza en el módulo de sensores que sube datos al host, esta escala es la misma que se gráfica como uso del equipo.
- Distancia: este campo es para almacenar el cálculo de la distancia entre dos puntos.
- Tabla world_poligono_fincas, esta tiene el nombre de la finca y las coordenadas que la delimitan.
 - Finca: este campo sirve como identificador para conocer de qué finca es la coordenada.
 - Latitud: este campo es necesario para tener registrada la latitud de los puntos que delimitan la finca.
 - Longitud: este campo junto con el anterior, da la ubicación en donde se delimita la finca.
- Tabla centro_fincas, esta tiene el nombre de la finca y la coordenada que sirve como centro de la finca, para poder hacer zoom respectivo desde google maps en la finca que el usuario desea ver.
 - Finca: este campo sirve como un identificador para conocer de qué finca es la coordenada guardada.
 - Latitud: este campo es necesario para conocer la latitud del punto que se tomó como referencia en la finca.
 - Longitud: este campo junto con el anterior, da la ubicación del punto de referencia de la finca.
 - Nombre: este campo sirve para desplegar el nombre de la finca en el mapa de ANACAFE.

Después de crear los modelos, se crearon las vistas, estas sirven para realizar consultas a la base de datos y redireccionar a las páginas que el usuario quiere ver.

7. **Uso de vistas y url.py en Django.** Se realizó una definición por cada página que se creó. Se realizaron queries en las vistas para poder acceder a la base de datos, obteniendo la información que se desea desplegar en la página web, estos se envían en un contexto, que es un diccionario esto tiene la forma `{'variable_en_html':variable_en_vista}`, en el `return render_to_response`, donde también se coloca el nombre del template a desplegar.

Cuando se accede a un url, se debe hacer referencia a la definición que se programó en las vistas, por lo cual se debe realizar esta referencia en el archivo `url.py` que se encuentra en la carpeta `geodjango`. Cada url debe apuntar a la definición que se debe llamar, cuando el usuario acceda a ella. Para lograr que se muestren los archivos estáticos se colocaron dos líneas para hacer referencia a la carpeta donde estos se encuentran.

```
urlpatterns+=patterns("(r'^archivos/(?P<path>.*);$','django.views.static.serve',{'document_root':settings.MEDIA_ROOT}),(r'^static/(?P<path>.*);$','django.views.static.serve',{'document_root':settings.STATIC_ROOT});)
```

8. **Programación de templates.** Por último se realizaron las plantillas o templates, los cuales son HTML, esto sirve para desplegar toda la información que se desea proporcionar al usuario. Estos templates se componen por código HTML, CSS, Javascript, JQuery y variables de Django.

El lenguaje de programación Javascript se utilizó para realizar el despliegue de los datos, a través de Google Maps y Google Chart. En esta parte fue necesario utilizar las variables de Django, que contienen la información enviada desde las vistas, junto con variables de Javascript. Esto con el fin de poder desplegar los datos de la base de datos en el mapa, tabla y gráfico. El mapa se programó con un filtro para que se desplegará el recorrido, las zonas de desempeño bajo y las zonas con roya, para desplegar la información por fechas se realizó un calendario con JQuery, el cual permite escoger la fecha de los datos que desea ver el dueño de la finca. En el siguiente código se observa como fue el uso de las etiquetas de django junto con el Javascript, para desplegar el mapa en la página web.

```
var mapOptions = {
    zoom: 15,
    center: new google.maps.LatLng({{centro.latitud}},{{centro.longitud}}),
    mapTypeId: google.maps.MapTypeId.SATELLITE
};

var map = new google.maps.Map(document.getElementById('map-canvas'),
    mapOptions);
{% for info in info_finca %}
    var latlng = new google.maps.LatLng({{info.latitud}},{{info.longitud}});
```

```

        coordenadas_finca.push(latlng);
    {% endfor %}
    setPoligonoFinca(map, coordenadas_finca);

```

El despliegue de datos se logró a prueba y error, cambiando los queries en las vistas para obtener la información necesaria a desplegar, también se realizó una función para mostrar las distancias que recorrió cada jornalero.

```

    indice=user_fincas.index(usuario_ingreso.username)
    info = fincas[indice].objects.all()
    centro = centro_fincas.objects.get(id=(indice+1))
    roya = fincas_roya.objects.all()
    poligono=poligono_fincas.objects.all().filter(finca=usuario_ingreso.username)
    bandera = 0
    for contador in range(1, info.count()):
if bandera == 0:
    distancia.append(0)
    bandera=1
elif fincas[indice].objects.get(id=contador).fecha==fincas[indice].objects.get(id=contador
-1).fecha and (bandera == 1):
    dist=fincas[indice].objects.get(id=(contador+1))
    target=fincas[indice].objects.all().filter(id=contador).distance(dist.punto).order_b
y('distance')
    for target in target:
        distancia.append(int(target.distance.m))
else:
    distancia.append(0)

```

Para darle estilo a la página se utilizó CSS, el cual contiene las imágenes de fondo que se muestran en la página, así como tener estilos de fuente y tener un menú más agradable al usuario. CSS también se utilizó para hacer el slide de fotos en la página principal. Se hizo también en la página de Contacto, un formulario donde el usuario puede mandar sus datos y un mensaje para que se le pueda proporcionar la información que él requiera. Los datos del formulario son enviados a fumogua@gmail.com, esto se logra por la configuración que se hizo anteriormente en settings.py.

Cuando no se ha permitido el envío de correo desde este servidor, aparece un error de SMTP, por lo que se debe ir a <https://accounts.google.com/DisplayUnlockCaptcha>, para asegurar que no es una tercera persona la que está tratando de ingresar a la cuenta.

9. **Administrador de Django.** Para tener un ingreso restringido a los datos, se realizaron usuarios con el admin de Django, al cual se accede por medio del url `fumogua.com/admin`. En esta sección se accede con el administrador creado en el proyecto, luego se crearon usuarios sin privilegios con nombre, apellido y contraseña. Esta verificación se realizó en las vistas donde hay una definición que comprueba que los datos ingresados por el usuario, son los correctos, sino son correctos, se hizo una plantilla que despliega ingreso incorrecto y vuelve a la página de ingreso.

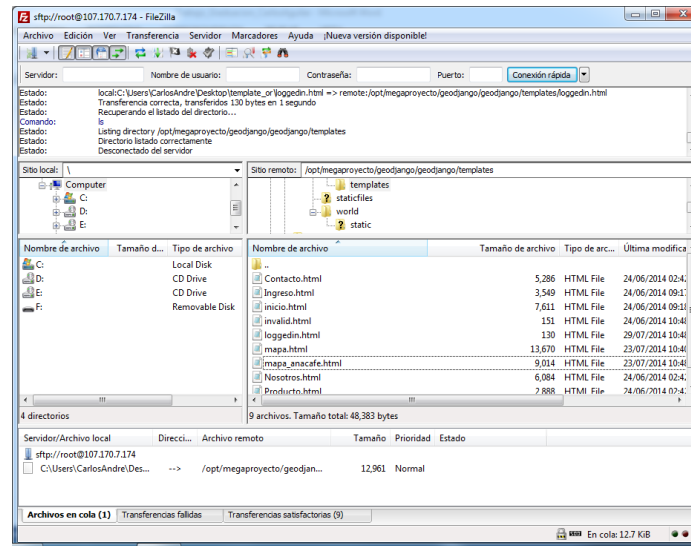
```
def auth_view(request):
    global accedio
    accedio=0
    username = request.POST.get('username',"")
    password = request.POST.get('password',"")
    user = auth.authenticate(username=username, password=password)
    if user is not None:
        accedio=1
        auth.login(request, user)
        return HttpResponseRedirect('/ingreso/loggedin/')
    else:
        return HttpResponseRedirect('/ingreso/invalid/')
```

Como existirán varios clientes, se requirió que cada uno pudiera tener acceso a su información, se realizaron dos listas, la primera sirve para llamar el modelo que contiene la información del usuario que ingresó a la página y la segunda contiene el nombre del usuario de la finca. Comparando el usuario ingresado, con el usuario de la finca almacenado en la segunda lista, con el índice se llama al modelo adecuado.

```
fincas = [datos_finca1,datos_finca2]
user_fincas = ["Finca.Fumogua", "Finca.ElPotrero"]
if usuario_ingreso.username != "anacafe":
    indice=user_fincas.index(usuario_ingreso.username)
    info = fincas[indice].objects.all()
```

10. **Uso de FileZilla.** Para hacer modificaciones a los archivos, se hizo una carpeta en el computador donde se estaba trabajando y se descargaron todos los archivos del proyecto a la carpeta, por medio de FileZilla. Con este programa, se modificaron los archivos y se subieron al Host nuevamente para tener mejores resultados en las páginas.

Figura 180: Uso de FileZilla



Por último se creó un script para copiar automáticamente los datos del CSV a la base de datos. Además se calculó, por medio de queries de PostgreSQL y PostGIS, la distancia entre los puntos, para poder conocer cuánto se había desplazado el jornalero entre cada punto.

C. RESULTADOS

Figura 181: Página Inicio



Figura 182: Página Nosotros

Quiénes Somos?



Misión

Dar respuesta a los requerimientos de fumigación de las fincas cafetaleras de Guatemala, por medio de la obtención de datos del proceso de fumigación por jornada, permitiendo a los encargados de la finca acceder a los reportes desde cualquier ubicación; evitando la contratación innecesaria de personas que supervisen el trabajo de los jornaleros.

Visión

Ser la empresa líder en el mercado guatemalteco, capaz de resolver las necesidades de los caficultores, ofreciendo servicios de monitoreo que se adaptan a los requerimientos de fumigación para la prevención de roya, contribuyendo al continuo desarrollo de la industria cafetalera.

El café requiere de un procedimiento de fumigación efectivo para combatir la roya, el proceso en algunos casos es llevado a cabo de manera incorrecta, promoviendo el desarrollo de infecciones indeseadas en el café, así como un desperdicio de fungicida lo cual significa una pérdida monetaria para el caficultor. También existe el mal uso del equipo de fumigación, ya que los jornaleros forzan las bombas, provocando que los sellos se rompan.

Ante estos problemas surge FUMOGUÁ, como un sistema de monitoreo eficaz para la fumigación del café y control del equipo, brindando un dispositivo tecnológico adaptable a diferentes tipos de bombas manuales destinadas a la fumigación.



Figura 183: Página Producto

¿Qué Ofrecemos?

Mapa de Recorrido y Roya

En este mapa usted podrá visualizar gráficamente el recorrido realizado por el jornalero, así como los puntos en donde tuvo una fumigación deficiente y los puntos de roya que se han encontrado en su finca. Podrá ver las jornadas de los días que desee, desde cualquier ubicación mientras cuente con acceso a internet.




Tabla de Resultados

En esta tabla podrá ver todos los puntos del recorrido en la jornada, cuál fue la presión y caudal en cada punto. Además puede observar la distancia recorrida entre cada punto por el jornalero, para poder conocer si el jornalero se está desplazando de manera efectiva.

Latitud	Longitud	Presión	% presión	Caudal	% caudal	Escala	Distancia Recorrida (metros)
14.52182511	-90.78525424	139	69.5	28	56	4	0
14.52178495	-90.78628421	170	85	25	50	4	114.71
14.52121333	-90.78610182	172	86	38	75	7	69.9
14.52161039	-90.78452468	149	74.5	41	82	6	181.64
14.52125488	-90.78434229	143	71.5	22	44	3	45.47
14.52081886	-90.78389397	162	81	49	98	8	180.3
14.52059017	-90.78380141	149	74.5	20	40	3	28.39

Gráfico de Desempeño

Por medio de una escala de uso de equipo, tomando en cuenta presión y caudal de la bomba, usted podrá conocer cuál fue el desempeño del jornalero durante la jornada. Esto le permitirá tomar acciones para tener una fumigación más efectiva y evitar daños a su equipo de fumigación.

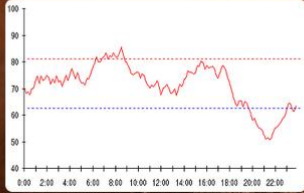


Figura 184: Página de Contacto

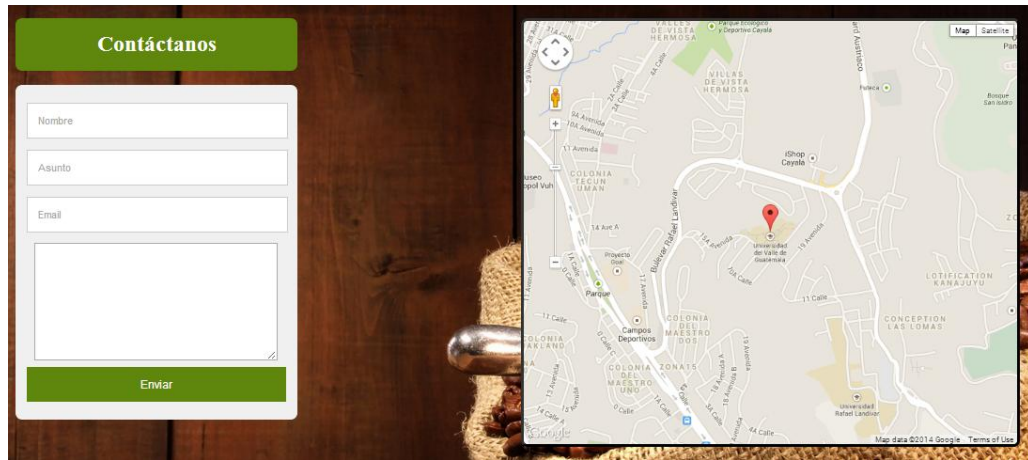


Figura 185: Formulario de Acceso a información



Figura 186: Usuarios para los clientes

Select user to change

Q Search

Action: Go 0 of 3 selected

<input type="checkbox"/> Username	Email address	First name	Last name	Staff status
<input type="checkbox"/> anacafe		Anacafe		⊖
<input type="checkbox"/> Finca.Fumogua		Carlos Aguilar	Finca Fumogua	⊖
<input checked="" type="checkbox"/> fumogua	fumogua@gmail.com			⊕

3 users

Figura 187: Calendario para filtrar información



1. RESULTADOS CON DATOS NO REALES

Figura 188: Mapa con filtros

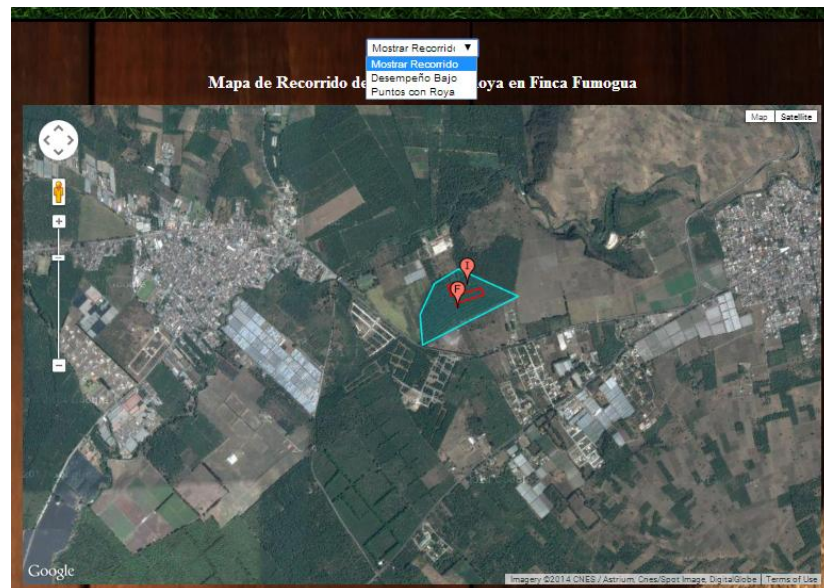


Figura 189: Puntos con Roya



Figura 190: Gráfica del desempeño de los Jornaleros

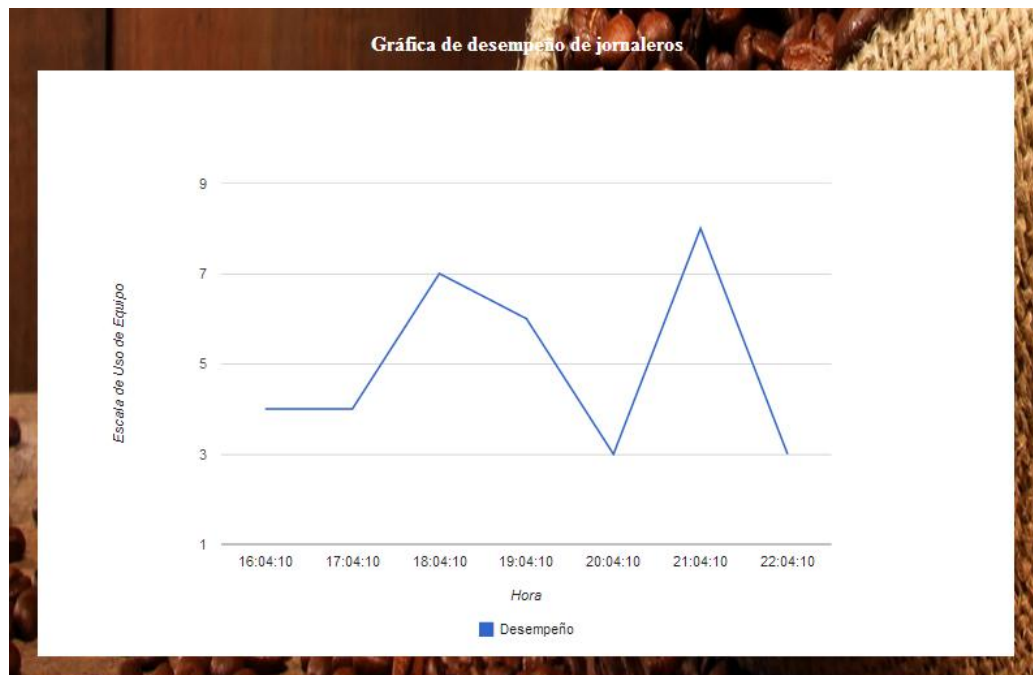
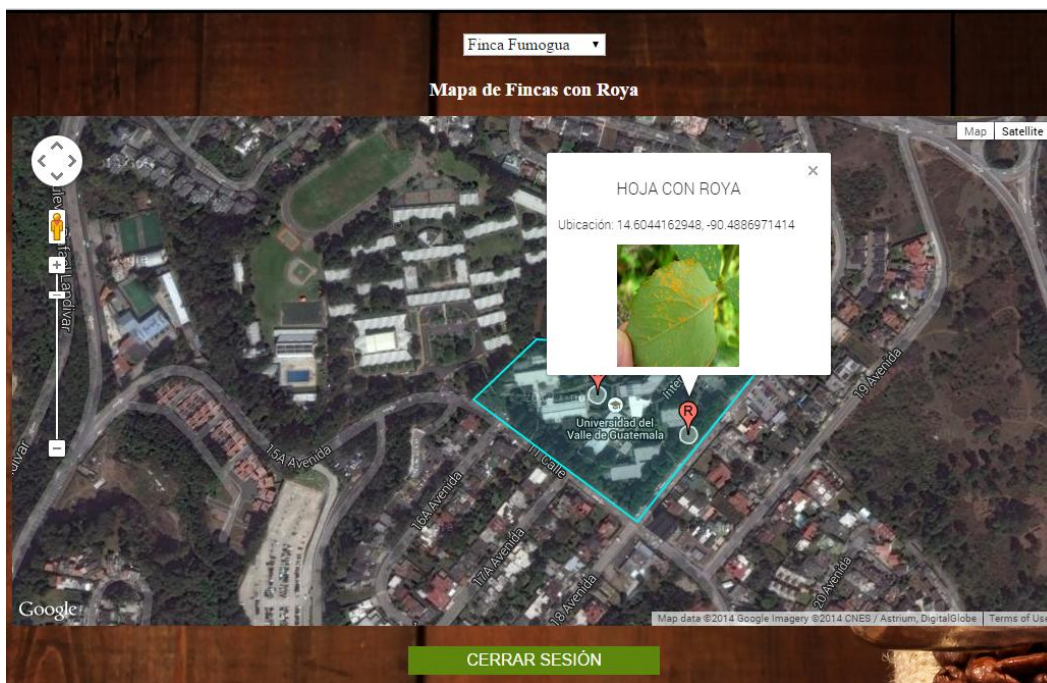


Figura 191: Mapa para ANACAFE



Cuadro 41: Tabla desempeño de jornaleros

#Bomba	Fecha	Hora	Latitud	Longitud	Presion	% presion	Caudal	% caudal	Escala	Distancia Recorrida (metros)
1	1 26/07/2014	16:04:10	14.52182611	-90.78525424	139	69.5	28	56	4	0
2	1 26/07/2014	17:04:10	14.52179495	-90.78628421	170	85	25	50	4	114.71
3	1 26/07/2014	18:04:10	14.52121333	-90.78610182	172	86	38	76	7	69.9
4	1 26/07/2014	19:04:10	14.52161839	-90.78452468	149	74.5	41	82	6	181.64
5	1 26/07/2014	20:04:10	14.52125488	-90.78434229	143	71.5	22	44	3	46.47
6	1 26/07/2014	21:04:10	14.52081866	-90.78589797	162	81	49	98	8	180.3
7	1 26/07/2014	22:04:10	14.52059017	-90.78580141	149	74.5	20	40	3	28.39

Cuadro 42: Tabla world_datos_fincas de la Base de Datos

id	fecha	hora	latitud	longitud	presion	p_presion	escala	p_caudal	escala	punto	distancia
1	30/07/2014	09:04:10	14.5218	-90.7863	174	87	43	86	7	0101000020E6100000EBD3017B52B256C098EF2AB5280B2D40	120.848
2	30/07/2014	09:05:10	14.5218	-90.7853	191	95.5	23	46	4	0101000020E61000001AF6FF9A41B256C03C4CB9CA2C0B2D40	114.712
3	30/07/2014	09:06:10	14.5212	-90.7861	139	69.5	38	76	5	0101000020E6100000780E027E4FB256C0B32D3D79DC0A2D40	181.64
4	30/07/2014	09:07:10	14.5208	-90.7859	191	50.5	36	72	4	0101000020E610000050E000A735B256C09815CC90110B2D40	50.7413
5	30/07/2014	09:08:10	14.5216	-90.7845	180	90	22	44	4	0101000020E610000050E000A735B256C09815CC90110B2D40	84.6541
6	30/07/2014	09:09:10	14.5213	-90.7843	188	94	41	82	8	0101000020E6100000DD1A01AA32B256C0F8F56CEBE10A2D40	180.296
7	30/07/2014	09:10:10	14.5206	-90.7858	118	59	36	72	4	0101000020E61000007246FF914AB256C0C58975CB8A0A2D40	179.515

2. RESULTADOS CON DATOS REALES

Figura 192: Recorrido con dispositivo ensamblado a la bomba

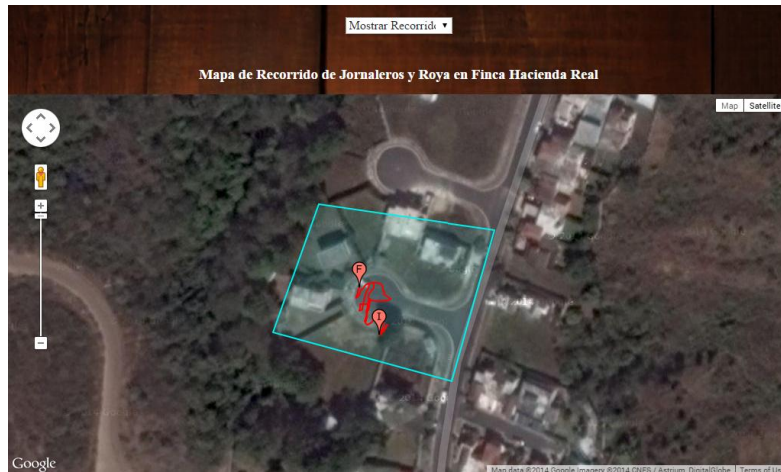


Figura 193: Desempeño bajo en mapa

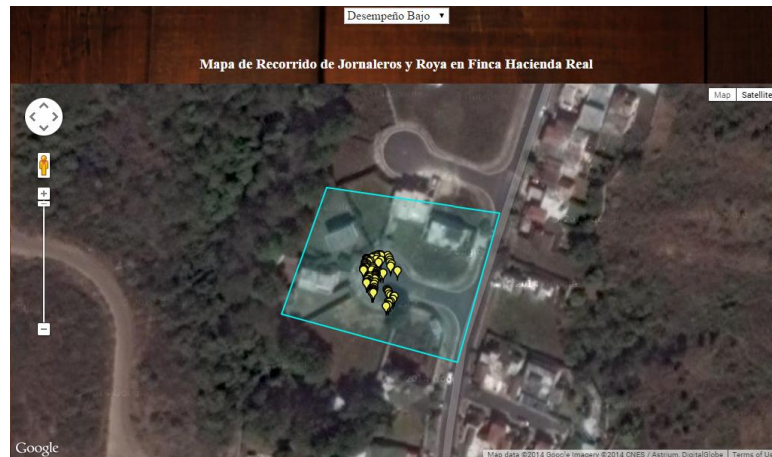
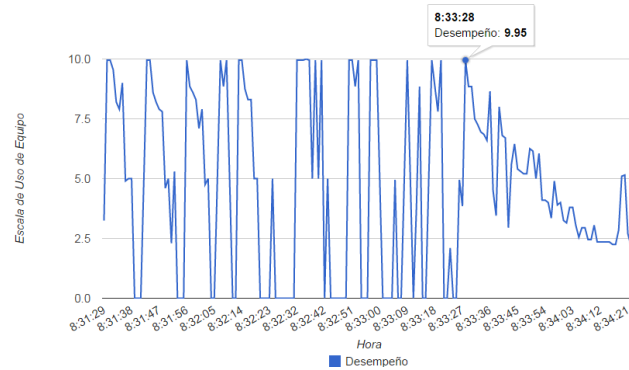


Figura 194: Gráfico de uso del equipo



Cuadro 43: Tabla de desempeño de los jornaleros

#Bomba	Fecha	Hora	Latitud	Longitud	Presion	Escala presion	Caudal	Escala caudal	Escala Total	Distancia Recorrida (metros)
1	1 18/09/2014	8:31:29	14.6212783333	-90.4575483333	32	4.6	3	1.9	3.25	0
2	1 18/09/2014	8:31:30	14.6212833333	-90.4575466667	59	10	8	9.9	9.95	0.6
3	1 18/09/2014	8:31:31	14.621285	-90.4575466667	55	10	8	9.9	9.95	0.19
4	1 18/09/2014	8:31:32	14.62129	-90.457545	51	9.2	8	9.9	9.55	0.6
5	1 18/09/2014	8:31:33	14.62129	-90.4575316667	49	8.7	7	7.7	8.2	1.48
6	1 18/09/2014	8:31:34	14.6212983333	-90.4575283333	47	8.1	7	7.7	7.9	1.03
7	1 18/09/2014	8:31:35	14.6213066667	-90.4575233333	47	8.1	8	9.9	9	1.11
8	1 18/09/2014	8:31:36	14.6213116667	-90.4575183333	53	9.8	10	0	4.9	0.8
9	1 18/09/2014	8:31:37	14.6213133333	-90.45752	58	10	9	0	5	0.27
10	1 18/09/2014	8:31:38	14.6213033333	-90.4575233333	60	10	9	0	5	1.21
11	1 18/09/2014	8:31:39	14.621295	-90.45753	65	0	10	0	0	1.21
12	1 18/09/2014	8:31:40	14.6212916667	-90.4575366667	69	0	11	0	0	0.84
13	1 18/09/2014	8:31:41	14.6212833333	-90.4575416667	70	0	10	0	0	1.11
14	1 18/09/2014	8:31:42	14.6212933333	-90.45754	63	0	8	9.9	4.95	1.17
15	1 18/09/2014	8:31:43	14.62129	-90.4575383333	59	10	8	9.9	9.95	0.43
16	1 18/09/2014	8:31:44	14.6212883333	-90.4575383333	55	10	8	9.9	9.95	0.19
17	1 18/09/2014	8:31:45	14.6212866667	-90.4575383333	52	9.5	7	7.7	8.6	0.19
18	1 18/09/2014	8:31:46	14.62128	-90.45754	49	8.7	7	7.7	8.2	0.79
19	1 18/09/2014	8:31:47	14.6212816667	-90.45754	47	8.1	7	7.7	7.9	0.19

Cuadro 44: Tabla world_datos_fincas de la base de datos

id	finca	bomba	fecha	hora	latitud	longitud	presion	p_presior	caudal	p_caudal	escala	punto	distancia
377	Finca.HaciendaReal	1	18/09/2014	8:33:04	14.62136	-90.4576	74	0	10	0	0	0101000020E6100000A7F7182499D56C0DAEDCA1D233E2D40	0.675913
378	Finca.HaciendaReal	1	18/09/2014	8:33:05	14.62136	-90.4576	72	0	8	9.9	4.95	0101000020E61000002ECA6C90499D56C0CA15DDE5223E2D40	0.417676
379	Finca.HaciendaReal	1	18/09/2014	8:33:06	14.62136	-90.4576	74	0	9	0	0	0101000020E6100000B5E4679E499D56C0B83DF1AD223E2D40	0.417676
380	Finca.HaciendaReal	1	18/09/2014	8:33:07	14.62136	-90.4576	71	0	9	0	0	0101000020E6100000B5E4679E499D56C0DAEDCA1D233E2D40	0.383492
381	Finca.HaciendaReal	1	18/09/2014	8:33:08	14.62137	-90.4576	70	0	16	9.8	4.9	0101000020E61000002ECA6C90499D56C0596E6935243E2D40	1.02801
382	Finca.HaciendaReal	1	18/09/2014	8:33:09	14.62138	-90.4576	59	10	8	9.9	9.95	0101000020E61000002ECA6C90499D56C0F69EE1BC253E2D40	1.342198
418	Finca.HaciendaReal	1	18/09/2014	8:33:45	14.62141	-90.4576	34	5	6	5.8	5.4	0101000020E610000041D058C1499D56C04A24D1CB283E2D40	0.266815
419	Finca.HaciendaReal	1	18/09/2014	8:33:46	14.6214	-90.4576	33	4.8	6	5.8	5.3	0101000020E6100000C64F59C8499D56C0384CE493283E2D40	0.266807
420	Finca.HaciendaReal	1	18/09/2014	8:33:47	14.6214	-90.4576	32	4.6	6	5.8	5.2	0101000020E6100000C64F59C8499D56C0384CE493283E2D40	0
421	Finca.HaciendaReal	1	18/09/2014	8:33:48	14.62141	-90.4576	32	4.6	6	5.8	5.2	0101000020E610000041D058C1499D56C04A24D1CB283E2D40	0.266807
422	Finca.HaciendaReal	1	18/09/2014	8:33:49	14.62141	-90.4576	33	4.8	7	7.7	6.25	0101000020E6100000C64F59C8499D56C04A24D1CB283E2D40	0.185525
423	Finca.HaciendaReal	1	18/09/2014	8:33:50	14.62141	-90.4576	32	4.6	7	7.7	6.15	0101000020E6100000C64F59C8499D56C04A24D1CB283E2D40	0
424	Finca.HaciendaReal	1	18/09/2014	8:33:51	14.62141	-90.4576	30	4.2	6	5.8	5	0101000020E610000041D058C1499D56C05AFCBD03293E2D40	0.266807
425	Finca.HaciendaReal	1	18/09/2014	8:33:52	14.62141	-90.4576	31	4.4	7	7.7	6.05	0101000020E610000041D058C1499D56C082F8A93B293E2D40	0.191734
426	Finca.HaciendaReal	1	18/09/2014	8:33:53	14.62141	-90.4576	29	4	5	4.2	4.1	0101000020E61000003F355EBA499D56C0A1A883AB293E2D40	0.426016
427	Finca.HaciendaReal	1	18/09/2014	8:33:54	14.62142	-90.4576	29	4	5	4.2	4.1	0101000020E61000003F355EBA499D56C0D97C5C182A3E2D40	0.38348
428	Finca.HaciendaReal	1	18/09/2014	8:33:55	14.62142	-90.4576	28	3.8	5	4.2	4	0101000020E61000003E9A6083499D56C01051358B2A3E2D40	0.426006
429	Finca.HaciendaReal	1	18/09/2014	8:33:56	14.62142	-90.4576	28	3.8	4	2.9	3.35	0101000020E6100000891A63AC499D56C02022C32A3E2D40	0.266807
430	Finca.HaciendaReal	1	18/09/2014	8:33:57	14.62142	-90.4576	29	4	6	5.8	4.9	0101000020E6100000891A63AC499D56C0570FDF322B3E2D40	0.38348

D. ANÁLISIS DE RESULTADOS

Cuando se inició el proyecto se eligió entre dos opciones, la primera era realizar la página web utilizando una Raspberry Pi como servidor y ésta opción se comparó con comprar un host. Utilizar la Raspberry Pi se vio como una opción económica, debido a que no se había tomado en cuenta que para realizar este tipo de proyecto se debía comprar una IP pública. Teniendo el dato del costo de una IP pública, el cual es \$5.99 en GoDaddy, se determinó que el host era la opción más viable. También se tuvo en cuenta la disponibilidad que debe tener el servidor, debido a que en el país existen fallas en el sistema eléctrico o el servicio de internet es irregular.

De la Figura 181 a la Figura 185 se muestran las páginas principales del sitio web, estas páginas fueron creadas con un CSS de base, con esto se logra asegurar de que todos los elementos que comparten las páginas como es el caso del menú esté en el mismo sitio. Todas las páginas fueron creadas con responsive web, lo que permite que el sitio se acomode dependiendo de la resolución del dispositivo en donde se esté accediendo a la página.

Anteriormente se realizó el sitio con diferentes CSS para cada página, lo cual resultó ineficiente, debido a que existían 5 archivos CSS a los cuales había que acceder cada vez que se realizaba un cambio al menú. El responsive web se estaba realizando con ajustes a los estilos con Javascript, pero esto era ineficiente porque se debía cambiar el tamaño de cada elemento, en cambio con Dreamweaver CC se realiza de manera gráfica el ajuste para cada rango de resolución y sobre una sola hoja de estilos.

Al utilizar responsive web y Google Maps, se tuvo el problema que el panel de control del mapa aparecía distorsionado. Este problema es debido a que existe un conflicto por la configuración en la hoja de estilos. Debido a que el max-width que tiene automáticamente el CSS tiene 100%, lo que crea el inconveniente con Google Maps, por lo que se debe cambiar a none. Con esto se arregló Google Maps, pero perjudicó el acomodamiento de las imágenes, ya que al cambiar de resolución las imágenes seguían del mismo tamaño y no se ajustaban. Para quitar este problema a las etiquetas div que contienen a las imágenes se les agregó max-width:100%, que en el CSS se había eliminado.

En la página contacto que se observa en la Figura 184, se realizó un formulario en el cual todos los campos son obligatorios, al presionar el botón enviar se recibe un correo en fumogua@gmail.com. Al colocar la configuración del correo en el settings.py de django, Gmail detecta host como una persona desconocida que puede ser maliciosa, por lo que impide el acceso y envía un correo a fumogua@gmail.com. Ingresando a la cuenta de Gmail, se abrió este correo y se colocó al host como persona confiable para ingresar al correo y así poder recibir los formularios de la página.

En el acceso para los usuarios que han contratado el servicio, se ingresa un usuario y contraseña como se observa en la Figura 185. Este usuario es creado por el administrador del proyecto de django, en donde

se introduce un username, nombre, apellido y contraseña del usuario. Con esto se logra crear y administrar todos los usuarios que estén activos. En la Figura 186 se muestran los usuarios que han sido creados, Finca.Fumogua es el usuario que se creó para poder realizar las pruebas para corroborar el correcto funcionamiento del despliegue de información. Para poder desplegar la información del usuario que está ingresando, se realizaron queries con filtros en donde se busca que el nombre de la finca sea el mismo que el usuario. Con esto se logró tener múltiples usuarios y desplegar la información correcta.

En la Figura 187 se observa que hay un calendario, este fue incorporado a la página ya que se desea que el usuario pueda tener acceso a todas las jornadas de fumigación. Anteriormente se había elegido desplegar solo 6 días antes de la fecha en la que se estaba revisando la información, pero esto era ineficiente porque la información anterior a esos 6 días se perdía. Con este calendario el usuario puede observar si el desempeño ha mejorado a través del tiempo. La librería utilizada para realizar el calendario fue datepicker, al cual se le agregó código para que este se desplegara en español.

Para desplegar la información según la fecha que el usuario ha elegido, se realizó una comparación directamente en el HTML utilizando tags de django y javascript, debido a que no se encontró la manera de enviar el dato de la fecha a la definición en el view. Por lo que esto realiza la comparación con todos los datos que existen dentro de la base de datos.

En la Figura 188 se observa que existe un filtro, este se implementó con el objetivo de que el mapa no estuviera saturado por toda la información que se debía desplegar. En el caso de mostrar recorrido, solo se muestra el trayecto que tuvo el jornalero; cuando se elige desempeño bajo, se despliegan pequeños marcadores de color amarillo, el desempeño bajo consiste en los puntos en donde el jornalero utilizó mal la bomba, es decir cuando obtuvo un desempeño menor a 7. Este desempeño es una escala que toma en cuenta la presión y caudal que tenía la bomba en ese punto. Por último, están los puntos de roya, en el cual se muestran marcadores con una letra R en donde se tomó una foto a la hoja con roya como se observa en la Figura 189.

En la Figura 190 se observa un gráfico, este tiene la finalidad de mostrar al caficultor en que horas se obtienen desempeños más altos. Con esto se logra que el dueño de la finca conozca si los jornaleros están forzando las bombas cerca de la hora de almuerzo para terminar rápido con el fungicida que tienen. Esto es útil debido a que en varias fincas este es un problema que aqueja a los dueños, ya que los jornaleros dañan las bombas al querer ir a almorzar más rápido.

En el mapa de ANACAFE mostrado en la Figura 191, únicamente se incluyeron las áreas en donde se encontró roya, por lo que es igual al código desarrollado anteriormente en el filtro de la Figura 188. En este caso desde el view.py se realiza una consulta a la base de datos sin filtros, debido a que en esta página si se despliegan todas las fincas que hay en el sistema.

Debido a que se requería delimitar la finca, se realizó una tabla que se llamó polígono_fincas, esto para tener todas las coordenadas de las fincas que cuentan con el dispositivo. Estos polígonos son desplegados por un queryset con filtro, en donde se busca que el nombre del usuario sea igual al nombre de la finca, para poder delimitar únicamente la finca del usuario que ingresó. Este procedimiento de filtro no se realiza en el mapa para ANACAFE, debido a que en ese caso se deben desplegar todas las fincas que tienen el dispositivo.

En Google Maps, se utiliza un centro que es para ubicar el mapa en esa coordenada, por lo que se realizó una tabla de centros. Se realizó el mismo procedimiento que polígono_fincas para poder ubicar la finca. Esta tabla se creó principalmente por el mapa para ANACAFE, debido a que se necesitaba que hubiera un filtro con los nombres de las fincas que se podían visualizar, como se muestra en la Figura 191. Por lo que al elegir la finca que se desea observar, se cambia el centro del mapa de google, esto hace más eficiente el proceso de cambiar de finca. Antes de tener este filtro se debía desplazar manualmente hasta la otra finca. Además este filtro permitió que los polígonos que delimitan a las fincas no se entrelazaran entre sí, ya que al tener un ciclo for que dibujara todos los polígonos tomaba todas las fincas como un solo polígono.

Utilizando Google Chart se creó el Cuadro 41, el cual contiene la información de cada punto tomado por el dispositivo en el recorrido del jornalero. Esto con la finalidad de que el dueño de la finca pueda tener los datos de las presiones y caudales que está manejando el jornalero, y no solo ver la escala que se le dio por uso del equipo. Además puede ver cuáles son las distancias en metros que recorrió el jornalero entre cada punto, para poder determinar si el jornalero está avanzando como debe ser.

Cuadro 42 se observa la tabla world_datos_finca, la cual tiene todos los datos provenientes de un CSV que es subido por la Raspberry Pi que se encuentra en la finca. Estos datos del CSV son copiados a la tabla, pero el campo punto y distancia, fueron creados automáticamente por queries geoespaciales. En el caso del campo punto se utilizó un SRID 4326 que utiliza la referencia WSG84, debido a que este permite ubicar puntos sin necesitar otro de referencia.

Se necesitó realizar una transformación de SRID, debido a que cuando se realizaba el cálculo de distancia entre puntos se obtenían datos erróneos, por lo que se transformó a un SRID 900913 este es el que usa Google como sistema de referencia espacial. Anteriormente no se contaba con el campo distancia, debido a que se estaban calculando por medio de queries en django, pero al tener que procesar una cantidad considerable de datos, se obtenía un error 504 de Gateway Time-out, por lo que se decidió cambiar la forma de obtener las distancias.

Debido a que los datos estarían subiendo a internet a diferente hora, dependiendo de la finca se realizó el script de copiado y llenado de los campos mencionados anteriormente. Se tuvo el inconveniente que los datos no eran subidos siempre o bien, la distancia no la calculaba por lo que al desplegar la tabla de datos (Cuadro 41) se obtenía error.

Se intentó solucionar este error colocando todos los campos de distancia en 0, pero este script se ejecuta cada minuto por lo que cada minuto llenaba las distancias en 0 y las volvía a calcular. Esto provocaba que la tabla tuviera las distancias calculadas y otras veces en 0, por lo que se realizó una comparación para corroborar que el archivo fincal.csv existiera en el directorio asignado. Así se aseguró que solo se actualizara la tabla únicamente cuando existiera nueva información.

De la Figura 188 a la Figura 191 se presentan pruebas con datos inventados y coordenadas obtenidas en un sitio web que al realizar clic sobre el mapa devuelve la coordenada. Esto sirvió para poder tener la seguridad de que los datos al momento de provenir del dispositivo, estos se iban a desplegar de la manera correcta. Por lo que de la Figura 192 a la Figura 194 se muestra el resultado de los datos reales obtenidos por el dispositivo dando recorridos en Hacienda Real. En donde se despliegan correctamente los datos provenientes del dispositivo y el sistema funciona automáticamente al detectar el archivo fincal.csv en el directorio que fue establecido.

En el Cuadro 43 se puede observar que existe la Escala Presión, Escala Caudal y Escala Total. La escala presión y escala caudal, provienen de los datos obtenidos de presión y caudal, transformando estos datos a una escala de 1 a 10 dependiendo del rango en el que se encuentren los datos. Estos inicialmente iban a ser un porcentaje, es por ello que en el Cuadro 41 se llaman %presión y %caudal, pero se decidió cambiarle el nombre, debido a que los datos solo iban a ser mapeados de 1 a 10. La escala total es un promedio de la escala presión y escala caudal, el cual indica el uso del equipo que se observa en la Figura 190 y Figura 194.

En el Cuadro 44 se puede observar la tabla datos_fincas, en donde se copiaron automáticamente los datos del archivo fincal.csv, los puntos y distancia entre ellos también se calculó de manera correcta. De la misma manera se aseguró esta comunicación automática con el dispositivo que toma y analiza fotos, pero este proceso no se pudo probar en conjunto.

E. CONCLUSIONES

1. Se logró implementar un mapa que muestra el recorrido recolectado por el GPS, los puntos con desempeño bajo y zonas con roya en Google Maps.
2. Se logró implementar tablas y gráficas en donde se muestra el desempeño de los trabajadores en cada punto.
3. Se implementó correctamente la base de datos geoespacial con PostGIS, para poder medir distancia entre puntos con referencia 900913 que utiliza Google.

4. Se implementó el diseño responsive web en la página para que se ajuste automáticamente a la resolución de pantalla.
5. Se implementó correctamente un script para el copiado a la base de datos y cálculo de distancias entre puntos.
6. Se automatizó el proceso de copiado de datos y cálculo de distancia entre puntos, por medio de crontab.
7. Se logró realizar un mapa para ANACAFE, para desplegar todas las fincas que poseen el dispositivo e indicar las zonas en donde existe roya.
8. Se implementó la base de datos para los accesos a los usuarios, por medio del administrador de Django, permitiendo el acceso a varios usuarios.

F. RECOMENDACIONES

- Se recomienda que se realicen más funciones con PostGIS, ya que en este caso solo se encontró la distancia entre puntos.
- Una mejora al despliegue de datos sería, colocar filtros para desplegar la información dependiendo del número de bomba. Esto para poder visualizar únicamente los datos del jornalero que se desea, ya que en este caso solo se implementó para una bomba.
- Implementar una forma más adecuada para desplegar los puntos en donde existe bajo desempeño, ya que puede llegar a saturar el mapa al ser marcadores.
- Mejorar el despliegue del recorrido, debido a que no se conoce cuando el jornalero empezó un nuevo recorrido en el mismo día, esto genera una línea entre el fin del primer recorrido y el inicio del siguiente.
- Realizar estadísticas que contengan la presión y caudal promedio que mantuvo el jornalero, para que el caficultor sepa con cuál de estos está teniendo problemas el jornalero.
- Buscar por una solución alternativa para encontrar las distancias entre puntos, ya que en el query realizado, cuando se tienen tablas con más datos, el proceso se vuelve lento y tarda en desplegar la página con la información.

X. CONCLUSIONES

1. Se diseñó un sistema de control y monitoreo de temperatura y humedad, para las secadoras estáticas y tipo guardiola.
2. Se logró la comunicación entre módulo de sensores y recolección de datos para el monitoreo de variables.
3. Se diseñó un mecanismo capaz de homogenizar la temperatura en las secadoras estáticas, con la velocidad y materiales adecuados para movilizar el grano de café.
4. Se logró diseñar y construir un sistema que mide y recolecta la información de los sensores.
5. Se logró enviar los datos de los sensores a un servidor informático, por medio de una Raspberry Pi.
6. Se logró diseñar e implementar una base de datos que contiene la información de las jornadas de fumigación.
7. Se diseñó una página web en donde se muestran los datos de la base de datos, por medio de Google Maps y Google Chart.
8. Se implementó un mapa en donde se muestran los sectores de las fincas en donde se encuentra la roya.

XI. RECOMENDACIONES

Se recomienda:

- Hacer un análisis de costos para determinar la validez de la implementación en la finca y conocer el tiempo de retorno de la inversión.
- Realizar más pruebas para tener un mejor monitoreo y control de los sistemas electrónicos y mecánicos.
- Llevar a cabo más pruebas del dispositivo de monitoreo de fumigación en fincas, para observar el comportamiento de los sensores y sobretodo el GPS. Al cambiar la topografía, puede que los datos sean más inexactos a los medidos.
- Para una futura etapa del proyecto, la integración de todos los módulos en una finca para determinar la funcionalidad de cada uno de ellos.
- Incluir una prueba de velocidad de fumigación, para poderlo incorporar al desempeño del jornalero, debido a que solo se tomó en cuenta la presión y caudal con la cual fumigaba.

XII. BIBLIOGRAFÍA

- Advanced HMI. (2014). *Software de expansión*. Recuperado el 18 de 09 de 2014, de http://www.advancedhmi.com/index.php?main_page=product_info&cPath=2&products_id=3
- Agencia. (2013). *La roya y la inseguridad amenazan a los pequeños cafetaleros de Guatemala*. Obtenido de <http://www.crhoy.com/la-roya-y-la-inseguridad-amenazan-a-los-pequenos-cafetaleros-de-guatemala/>
- Anacafe. (s.f.). *¿Qué es la roya del cafeto?* . Obtenido de http://www.anacafe.org/glifos/index.php/12PRIN:Que_es_la_Roya
- AnaCafé, A. N. (s.f.). *El beneficiado húmedo*. Recuperado el 26 de Enero de 2014, de https://www.anacafe.org/glifos/index.php/Caficultura_BeneficiadoHumedo#Secamient_o_del_café_lavado
- AnaCafé, A. N. (s.f.). *El beneficiado húmedo*. Recuperado el 23 de Enero de 2014, de https://www.anacafe.org/glifos/index.php/Caficultura_BeneficiadoHumedo#Secamient_o_del_café_lavado
- Andrei, V. (2014). *Impresión 3D: Cómo va a cambiar el mundo*. Editorial Baikal.
- Andreu, J. (2011). *Gestión de servidores web (Servicios en red)*. Editex.
- Arduino. (2014). *What is Arduino?* Recuperado el 16 de Septiembre de 2014, de <http://arduino.cc/en/Guide/Introduction>
- ASIDEK. (2014). Recuperado el Septiembre de 2014, de <http://www.asidek.es/index.php/industria-y-fabricacion/autodesk-inventor/>
- Budynas, R. G. (2008). *Diseño en ingeniería mecánica de Shigley* (Novena ed.). México: McGraw Hill. Recuperado el Julio de 2014
- Canto Q., C. (s.f.). *Microcontroladores*. México.
- CAP, S.A. (2000). *¿Qué es el acero?* Recuperado el Septiembre de 2014, de http://www.infoacero.cl/acero/que_es.htm
- Carrero, D. (2005). *Descripción de un programa gratuito que permite convertir nuestro ordenador en un servidor FTP*. Obtenido de <http://www.desarrolloweb.com/articulos/1943.php>

- Cavallé Oller, N. (2004). *Medición del caudal en sistemas de extracción localizada*. España.
- Cengel, Y. A., & Cimbala, J. M. (2010). *Fluid Mechanics: Fundamentals and applications* (2nd Edition ed.). McGraw-Hill.
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.
- Cytron Technologies. (2013). *Manual de usuario HC-RS04*. Recuperado el 18 de 09 de 2014, de https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit
- de Dios, C. A. (1996). *Secado de granos y secadoras*. Recuperado el 06 de Julio de 2014, de <http://www.fao.org/docrep/x5028s/X5028S00.htm#Contents>
- de Vandelvira, A. (2011). *Motores de corriente alterna*. Sistemas automáticos de producción alimentaria.
- DirectIndustry. (s.f.). *Rodamientos con una sola hilera*. Recuperado el Septiembre de 2014, de <http://www.directindustry.es/prod/rbc-bearings/rodamientos-perlas-perlas-sola-hilera-radiales-16837-1132359.html>
- Django. (2014). *Geodjango Tutorial*. Obtenido de <https://docs.djangoproject.com/en/1.6/ref/contrib/gis/tutorial/>
- ElecFreaks. (2012). *Manual de usuario HC-SR04*. Recuperado el 18 de 09 de 2014, de http://www.electfreaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf
- Electroindustria. (2013). *Interfaces Hombre Máquina*. Recuperado el 30 de Agosto de 2014, de Las "pantallas de operador" crecen con nuevas funcionalidades: <http://www.emb.cl/electroindustria/articulo.mvc?xid=2020#>
- Espinoza, J. (s.f.). *Medición de nivel y caudal por ultrasonido*. (U. T. María, Ed.) Obtenido de <http://profesores.elo.utfsm.cl/~jgb/AVILES.pdf>
- Estrella, J. (2013). *Arquitectura de una aplicación Sapns*. Obtenido de <http://www.ender.es/2013/07/arquitectura-de-una-aplicacion-sapns/>
- Flanagan, D. (2010). *jQuery Pocket Reference*. California, EEUU: O'Reilly Media Inc.
- Flores, F. (2013). *¿Qué es el servicio de web hosting?* Obtenido de <http://blog.compuclub.com.gt/2013/08/que-es-web-hosting/>
- Floyd, T. (2000). *Fundamentos Digitales*. Prentice Hall.

- Fonseca, V. J. (s.f.). *Transmisión de calor*. Colombia.
- Fridman, L. (2013). *DigitalOcean, un nuevo y asombroso hosting VPS*. Obtenido de <http://altoros.com.ar/blog/digitalocean-un-nuevo-y-asombroso-hosting-vps/>
- Fumigaciones Bay-Were C.A. (2014). *Tipos de fumigación o técnicas de fumigación en control de plagas*. Obtenido de http://www.fumigacionesbw.com/fumigacion_servicios.html
- Garcia, A. (2013). *¿Qué es Django Framework?* Obtenido de <http://vanuta.com/articulos/que-es-django-framework/>
- Gary Shelly, D. W. (2008). *HTML: conceptos y técnicas introductorias* (5 ed.). Cengage Learning.
- González Duque, R. (2008). *Python Para todos*. España.
- Gonzalez, A. (2008). *Tecnología Bluetooth*. México: Instituto Politécnico Nacional.
- goodellgroup. (2004). *Learning HTML for kids*. Recuperado el 18 de 09 de 2014, de <http://www.goodellgroup.com/tutorial/chapter2.html>
- GoogleMaps. (2005). *¿Qué es Google Maps?* Obtenido de http://www.googlemaps.es/?page_id=3
- Gutiérrez, J. (2006). *¿Qué es un framework web?* Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Gutierrez, O. (Diciembre de 2008). *Especificaciones de rodamientos*. Recuperado el Agosto de 2014, de <http://es.scribd.com/doc/13258786/TIPOS-Y-ESPECIFICACIONES-DE-RODAMIENTOS>
- Harper, G. E. (2000). *El libro práctico de los generadores, transformadores y motores eléctricos*. Editorial Limusa.
- Henares, A. (2013). *¿Qué es un script?* Obtenido de <http://www.ubuntufacil.com/2013/03/que-es-un-script/>
- Huerta, E., Mangiaterra, A., & Noguera, G. (2005). *GPS: Posicionamiento satelital*. Argentina: UNR Editora - Universidad Nacional de Rosario.
- Infante, S. (2012). *Curso Django: entendiendo como trabaja django*. Obtenido de <http://www.maestrosdelweb.com/editorial/curso-django-entendiendo-como-trabaja-django/>
- Instruments, N. (29 de Marzo de 2011). *PID Theory Explained*. Recuperado el 16 de Marzo de 2014, de <http://www.ni.com/white-paper/3782/en/>

- Instruments, P. (s.f.). *Transductor de velocidad y caudal del aire EE 75*. Recuperado el 14 de Septiembre de 2014, de <http://www.pce-iberica.es/medidor-detalles-tecnicos/sistemas/transductor-caudal-ee-75.htm>
- Inzunza, J. (s.f.). *Mecanismos de transferencia de calor*.
- Iparraguirre, C. B. (2012). *Autodesk Inventor 2012*. Recuperado el Septiembre de 2014, de http://es.slideshare.net/jhonathan21/manualautodeskinventor2012?next_slideshow=1
- J. B. Andriulli, A. E. (1999). *Advanced power generation systems for the 21st century: Market survey and recommendations for a design philosophy*. Oak Ridge National Laboratory.
- JTP, I. y. (s.f.). *Nomenclatura para diagramas de tuberías e instrumentación (P&ID)*.
- Junta de Andalucía. (2013). *Google Chart*. Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/886>
- Junta de Andalucía. (2013). *PostGIS*. Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/12-0>
- Kernighan, B. W., & Ritchie, D. M. (1991). *El lenguaje de programación C*. México: Person Education.
- LASKA. (s.f.). *Mezcladoras Industriales*. Recuperado el Julio de 2014, de http://www.laska.at/uploads/tx_laskaproducts/ES_LASKA_Mezcladoras.pdf
- Lenticular, R. (2009). *Introducción a Google Chart API*. Obtenido de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=googleChart>
- Liu, T. (s.f.). *Digital relative humidity & temperature sensor AM2302/DHT22*.
- Lucain. (2010). *Cron & crontab, explicados*. Obtenido de <http://blog.desdelinux.net/cron-crontab-explicados/>
- Manzur, S. (2013). *¿Qué es jQuery?* . Obtenido de <http://www.mexired.com/blog/que-es-jquery/#.U9GYzvldWrS>
- Martinez, R. (2010). *Sobre PostgreSQL*. Obtenido de http://www.postgresql.org.es/sobre_postgresql
- Matt Richardson, S. W. (2012). *Getting Started with Raspberry Pi*. O'Reilly Media, Inc.
- Mazzone, V. (Marzo de 2002). *Controladores PID*. Recuperado el 15 de Marzo de 2014, de <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>

- Mediawiki. (2012). *Filosofía Ubuntu*. Obtenido de http://doc.ubuntu-es.org/Filosof%C3%ADa_Ubuntu
- Mendoza, E. Y. (2003). *Control de una planta generadora de energía eléctrica*. Oaxaca, Mexico: Universidad tecnológica de la mixteca.
- Microsoft Windows. (2014). *Visual Studio*. Recuperado el 18 de 09 de 2014, de <http://www.visualstudio.com/>
- Morales, H. (2004). *La comunicación serial*. México.
- Mott, R. L. (2006). *Diseño de elementos de máquinas* (Cuarta ed.). México: Pearson Educación.
- NationalInstruments. (28 de Octubre de 2011). *Texas Instruments Circuit Experiments for NI myDAQ—Experiment 2: R-2R Resistive DAC*. Recuperado el 18 de Septiembre de 2014, de <http://www.ni.com/white-paper/13237/en/>
- NationalSemiconductor. (1995). *LM34/35 Precision Monolithic Temperature Sensors*. Europa.
- Nessys. (2014). *Transmisión de archivos mediante SFTP*. Obtenido de <http://www.nessys.es/knowledgebase.php?action=displayarticle&id=10>
- Nixon, R. (2012). *Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites*. California: O'Reilly Media Inc.
- NKE Bearings. (s.f.). *Rodamientos de rodillos cilíndricos*. Recuperado el Septiembre de 2014, de <http://www.nke.at/es/rodamientos-aplicaciones/rodamientos-estandar/rodamientos-de-rodillos-cilindricos/>
- Oleaga, J. (2013). *Los 10 mejores sitios para crear una web gratis*. Recuperado el 18 de 09 de 2014, de ABC: <http://www.abc.es/tecnologia/redes/20130219/abci-alojamiento-gratis-201302181850.html>
- Olimex. (2013). *Olimexino-328 User's manual*.
- Ordinas, J. M. (2008). *Protocolos y aplicaciones Internet*. Editorial UOC.
- Palau, S. &. (s.f.). *Circulación de aire por conductos*.
- PCE. (s.f.). *Pirómetros*. Recuperado el 14 de Septiembre de 2014, de <http://www.pce-iberica.es/instrumentos-de-medida/metros/pirometros.htm>
- Perez, D. (2007). *Base de Datos*. Obtenido de <http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/>

- Perez, D. (2007). *Javascript*. Obtenido de <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>
- Platea. (s.f.). *Tornillo Sinfin*. Obtenido de http://platea.pntic.mec.es/dlopez/mecanismos/tornillo_sin_fin.html
- QuimiNet. (25 de Octubre de 2006). *Los tipos de mezcladores*. Recuperado el 16 de Septiembre de 2014, de <http://www.quiminet.com/articulos/los-tipos-de-mezcladores-16423.htm>
- Ramirez, C. (2014). *¿Qué es PuTTY y para qué sirve?* Obtenido de <http://www.vozidea.com/que-es-putty-y-para-que-sirve>
- Raspbian*. (2012). Recuperado el 10 de Agosto de 2014, de <http://www.raspbian.org/>
- Rodriguez, M. (28 de 03 de 2013). *Tipos de sistemas de control*. Recuperado el 20 de Enero de 2014, de http://prof.usb.ve/mirodriguez/control/Introduccion_sistemas_control/tipos_de_sistemas_de_control.html
- SabienTechnology. (s.f.). *What are modulating burners?* Recuperado el 26 de Octubre de 2014, de <http://www.sabien-tech.co.uk/products/m2g/what-are-modulating-burners>
- SD Association*. (2014). (SD Association) Recuperado el 09 de Agosto de 2014, de https://www.sdcard.org/about_sda
- SKF. (s.f.). *Thrust ball bearings*. Recuperado el Septiembre de 2014, de <http://www.skf.com/group/products/bearings-units-housings/ball-bearings/thrust-ball-bearings/index.html>
- Sosa, L. (2008). *Definición HTML*. Obtenido de <http://definicion.de/html/>
- SutoCom. (2013). *Setting up Django with Ubuntu 12.10, SQLite, Gunicorn, and Nginx on a Digital Ocean droplet*. Obtenido de <http://markwingerd.wordpress.com/2013/09/01/setting-up-django-with-ubuntu-12-10-sqlite-gunicorn-and-nginx-on-a-digital-ocean-droplet/>
- Systems, A. I. (1998). *ACS 300 AC Drives for Speed Control if 1/2 Hp to 15 Hp Squirrel Cage Motors*.
- TECAM. (2012). *Mezclador de paleta*. Recuperado el Julio de 2014, de http://www.tecam.cl/equipo_paleta.html
- Techotopia. (2009). *Designing Forms in Visual Studio*. Recuperado el 18 de 09 de 2014, de http://www.techotopia.com/index.php/Designing_Forms_in_Visual_Studio

- TecnoMixer, S.A. (2010). *Mezcladores doble cinta o helicoidales*. Recuperado el Julio de 2014, de <http://ciudad-buenos-aires.all.biz/mezcladores-doble-cinta-o-helicoidal-g3020#.VBkXXFfONRs>
- TexasInstruments. (2013). *LM35 Precision Centigrade Temperature Sensors*.
- The PHP Group. (2014). *¿Qué es PHP?* Recuperado el 18 de 09 de 2014, de <http://php.net/manual/es/intro-whatis.php>
- TopCeller. (Septiembre de 2014). *ebay*. Recuperado el 09 de Septiembre de 2014, de http://www.ebay.com/itm/Pressure-transducer-sender-sensor-100-psi-stainless-steel-for-oil-fuel-air-water-/191272130627?pt=Motors_Car_Truck_Parts_Accessories&hash=item2c88b51843&vxp=mtr
- Trevi. (2007). *Mezcladores Helicoidales*. Recuperado el 16 de Septiembre de 2014, de <http://www.trevisin.com/mezcladorhelicoidal.html>
- Universidad Nacional de Quilmes. (2011). *Ingeniería en automatización y control industrial: Introducción a HMI*. Recuperado el 30 de Agosto de 2014, de <http://iaci.unq.edu.ar/materias/laboratorio2/HMI%5CIntroduccion%20HMI.pdf>
- Uxcell. (septiembre de 2014). *Amazon*. Recuperado el 9 de Septiembre de 2014, de http://www.amazon.com/HQ-A68-1-Effect-Sensor-Counter-Heater/dp/B00DNOXZZQ/ref=sr_1_6?ie=UTF8&qid=1410837204&sr=8-6&keywords=Water+Flow+Sensor
- Vaisala. (2012). *Cómo elegir el instrumento correcto para medir humedad y punto de rocío*.
- Ventura Nava, I. C. (2008). *Sistemas de control de motores eléctricos industriales*. Veracruz.
- Viera, M. (2013). *Gunicorn*. Obtenido de <http://blog.manuelviera.es/gunicorn>
- Viera, M. (2013). *Nginx*. Obtenido de <http://blog.manuelviera.es/nginx>
- W3C. (2014). *Guía Breve de CSS*. Obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- Weishaupt. (2014). *Quemadores industriales WK*.
- Wong, L. (2002). *1-Wire(R) (Protocol)*.

XIII. ANEXOS

A. CÓDIGO FUENTE PARA MONITOREO DE TEMPERATURA Y HUMEDAD

```
//Universidad del Valle de Guatemala
//Trabajo de Megaproyecto
//Módulo de sensores
//Trabajo presentado por Paola García Sandoval

#include <SHT1x.h>
#include <dht.h>

constint sensorTempG1 = A0; //LM35 Gua1
constint sensorTempG2 = A1; //LM35 Gua2
constint sensorTempE1 = A2; //LM35 Est1
constint sensorTempE2 = A3; //LM35 Est2

constint muchaTempE1 = 8; //indicador luminico de mucha temperatura para Est 1
constint muchaTempE2 = 9; //indicador luminico de mucha temperatura para Est 2
constint muchaTempG1 = 2; //indicador mucha temp G1 y G2
constint pocaTempE1 = 3; //indicador poca temp E1
constint pocaTempE2 = 4; //indicador poca temp E2
constint pocaTempG1 = 5; //indicadorpoca temp G1 y G2

constint sensorHumTempE1 = 6; //DHT22 Est1
constint sensorHumTempE2 = 7; //DHT22 Est2

//guardiola1, temp lm35, guardiola2, temp lm35, estatica1, temp DHT22, humedad, temp lm35, estatica2,
temp DHT22, humedad, temp lm35
//String enviar = "0,0,0,0,0,0,0,0";
String enviarE1 = "0";
String enviarE2 = "0";
String enviarG1 = "0";
String enviarG2 = "0";

int coma; //posicion
Stringvector[4];
charraspi;

//para el control en secadoras estaticas
double entrada1, salida1, referencia1, entrada12, salida12, diametro2, resistencia2;

//para el control en secadoras tipo guardiola
```

```

double entrada2, salida2, diametro, referencia2, entrada22, salida22;

dht miSensorE1;
dht miSensorE2;

//Configuración inicial
void setup(){

Serial.begin(9600);

referencia1 = 50; //50°C
referencia2 = 60; //60°C

//Modo de los pines a utilizar
pinMode(sensorTempG1, INPUT);
pinMode(sensorTempG2, INPUT);
pinMode(sensorTempE1, INPUT);
pinMode(sensorTempE2, INPUT);

pinMode(sensorHumTempE1, INPUT);
pinMode(sensorHumTempE2, INPUT);

pinMode(muchaTempG1, OUTPUT);
pinMode(muchaTempE1, OUTPUT);
pinMode(muchaTempE2, OUTPUT);
pinMode(pocaTempG1, OUTPUT);
pinMode(pocaTempE1, OUTPUT);
pinMode(pocaTempE2, OUTPUT);

doubletemp_amb = 24; //temperatura ambiente (°C)

}

//FUNCION para convertir de double a String
String doubleToString(double input,intdecimalPlaces){
if(decimalPlaces!=0){
String string = String((int)(input*pow(10,decimalPlaces)));
if(abs(input)<1){
if(input>0)
string = "0"+string;
else if(input<0)
string = string.substring(0,1)+"0"+string.substring(1);
}
return string.substring(0,string.length()-decimalPlaces)+"."+string.substring(string.length()-decimalPlaces);
}
else {
return String((int)input);
}
}
}

```

```

voidloop(){

// -----
//           Recepcion de datos desde Raspberry
// -----

if (Serial.available()){

raspi = Serial.read();

    // e = estatica 1
    if (raspi == 'e'){
    Serial.print(enviarE1);
    Serial.flush();
    }

    // f = estatica 2
    if (raspi == 'f'){
    Serial.print(enviarE2);
    }

    // g = guardiola 1
    if (raspi == 'g'){
    Serial.print(enviarG1);
    }

    // h = guardiola 2
    if (raspi == 'h'){
    Serial.print(enviarG2);
    }
    }

// -----
//           Control de temperatura en flujo de aire - Secadoras tipo Guardiola 1
// -----

    //LM35
    float temp = analogRead(sensorTempG1);
    entrada2 = temp*0.48828125; //Leer temperatura del aire

    //G1, temperatura del aire
    enviarG1 = doubleToString(entrada2, 2);

    //si esta muy caliente, prender led ROJO
    if(entrada2 > referencia2){
    digitalWrite(muchoTempG1, HIGH);
    digitalWrite(pocaTempG1, LOW);
    }
}

```

```

    }
else{
    //si esta muy frio, prender led VERDE
    if(entrada2 < referencia1){
    digitalWrite(pocaTempG1, HIGH);
    digitalWrite(muchaTempG1, LOW);
    }

else{
digitalWrite(muchaTempG1, LOW);
digitalWrite(pocaTempG1, LOW);
    }

    }

// -----
//          Control de temperatura en flujo de aire - Secadoras tipo Guardiola 2
// -----

//LM35
float temp2 = analogRead(sensorTempG2);
entrada2 = temp2*0.48828125+1; //Leer temperatura del aire

    enviarG2 = doubleToString(entrada2, 2);

// -----
//          Control de temperatura por medio de un inyector de gas - Secadora estatica 1
// -----

//DHT22
intchk = miSensorE1.read22(sensorHumTempE1);

float tempE1 = miSensorE1.temperature;
float humedadE1 = miSensorE1.humidity; //Leer a cuanto se encuentra la humedad del cafe (0 - 100%)
(sensor SHT1x)

//LM35
float tempE1 = analogRead(sensorTempE1);
entrada1 = tempE1*0.48828125; //Leer temperatura del aire

    enviarE1 = doubleToString(tempE1, 2);
    enviarE1 += ",";
    enviarE1 += doubleToString(humedadE1, 2);
    enviarE1 += ",";
    enviarE1 += doubleToString(entrada1, 2);
    enviarE1 += ",";

    //si esta muy caliente, prender led ROJO
    if(entrada1 > referencia2){

```

```

digitalWrite(muchaTempE1, HIGH);
digitalWrite(pocaTempE1, LOW);
}
else{
    //si esta muy frio, prender led VERDE
    if(entrada1 < referencia1){
        digitalWrite(pocaTempE1, HIGH);
        digitalWrite(muchaTempE1, LOW);
    }

    else{
        digitalWrite(muchaTempE1, LOW);
        digitalWrite(pocaTempE1, LOW);
    }
}

// -----
//          Control de temperatura por medio de un inyector de gas - Secadora estatica 2
// -----

//DHT22
int chk1 = miSensorE2.read22(sensorHumTempE2);

float tempE2 = miSensorE2.temperature;
float humedadE2 = miSensorE2.humidity; //Leer a cuanto se encuentra la humedad del cafe (0 - 100%)
(sensor SHT1x)

//LM35
float tempE2 = analogRead(sensorTempE2);
entrada12 = tempE2*0.48828125 ; //Leer temperatura del aire

    enviarE2 = doubleToString(tempE2, 2);
    enviarE2 += ",";
    enviarE2 += doubleToString(humedadE2, 2);
    enviarE2 += ",";
    enviarE2 += doubleToString(entrada12, 2);

    //si esta muy caliente, prender led ROJO
    if(entrada12 > referencia2){
        digitalWrite(muchaTempE2, HIGH);
        digitalWrite(pocaTempE2, LOW);
    }
    else{
        //si esta muy frio, prender led VERDE
        if(entrada12 < referencia1){
            digitalWrite(pocaTempE2, HIGH);
            digitalWrite(muchaTempE2, LOW);
        }
    }

```

```
else {  
  digitalWrite(muchaTempE2, LOW);  
  digitalWrite(pocaTempE2, LOW);  
}  
  
}  
  
}
```

B. CÓDIGO DE PROGRAMA DE MEDICIÓN Y REGULACIÓN DE PLANTA AUXILIAR DE GENERACIÓN.

```

#include <Servo.h>

Servo myservo; // creacion de objeto servo

//Definicion de variables
double Input, Setpoint, Output, Window, Temperatura, largo_tanque, nivel;
char raspi, control;
int trigPin = 13;
int echoPin = 12;
String data;

//Inicializacion de variables
void setup()
{
  myservo.attach(9); //
  Window = 0.1; //
  Setpoint = 0;
  Output = 500;
  control = '0';
  nivel = 0;
  largo_tanque = 50;
  Serial.begin(9600);
  myservo.write(0);
  pinMode(trigPin, OUTPUT); //trigger
  pinMode(echoPin, INPUT); // echo
}
//Ciclo principal
void loop()
{
  if (Serial.available()){ //se revisa si la raspberry envia algun comando

  raspi = Serial.read();
  // medicion y envio de temperatura, nivel de combustible y voltaje del generador
  if (raspi == 's'){
    data = "";
    // medicion de temperatura
    Temperatura = analogRead(A0);
    Temperatura = Temperatura * 0.48828125;
    data = ""+String(int(Temperatura));
    // medicion de nivel del tanque
    long duration, distance;
    digitalWrite(trigPin, LOW); //
    delayMicroseconds(2); //
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10); // Added this line
    digitalWrite(trigPin, LOW);
  }
}

```

```

duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
nivel = 100 - (distance*100/largo_tanque)
data = data + '+' + nivel;
// medicion y envio de voltaje generado
Input = analogRead(A1)*5.0/1023.0;
data = data + '+' + String(int(Input));
Serial.println(data);

}
// asignacion de setpoint
if (raspi == 'a'){
  Setpoint = 1;
}

if (raspi == 'b'){
  Setpoint = 2;
}
if (raspi == 'c'){
  Setpoint = 0;
}
// ciclo de control de servomotor
}
delay(10);
int a = 0;
while(a == 0){
  Input = analogRead(A1)*5.0/1023;
  if (Input < (Setpoint - Window)){
    Output = Output + 1;
    myservo.writeMicroseconds(Output); // Se realiza correccion a la posicion del servo
  }
  if (Input > (Setpoint + Window)){
    Output = Output - 1;
    myservo.writeMicroseconds(Output); // Se realiza correccion a la posicion del servo
  }
  delay(15);
  if (Input < (Setpoint + Window) && Input > (Setpoint - Window)){
    a = 1;
  }
}

```

C. CÓDIGO DE PROGRAMA DE HMI.

```

// LIBRERIAS UTILIZADAS
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Net;

using System.Net.Sockets;

using System.Threading;

using System.Windows.Forms.DataVisualization.Charting;

using Excel = Microsoft.Office.Interop.Excel;

using System.IO;

namespace prueba
{
    public partial class Form1 : Form
    {
        // DEFINICION DE VARAIBLES GLOBALES

        int lastRow, lastRow2;

        int value2;

        ThreadStart job;

        Thread thread;

        // METODO DE CONEXION FTP CON SERVIDOR WEB

```

```

private void FTP_connect(String str){

    FtpWebRequest request =
(FtpWebRequest)WebRequest.Create("ftp://server34.000webhost.com/public_html/archivo.txt");

    request.Method = WebRequestMethods.Ftp.UploadFile;

    request.Credentials = new NetworkCredential("user", "password");

    byte[] fileContents = new byte[str.Length * sizeof(char)];
    System.Buffer.BlockCopy(str.ToCharArray(), 0, fileContents, 0, fileContents.Length);
    request.ContentLength = fileContents.Length;

    Stream requestStream = request.GetRequestStream();
    requestStream.Write(fileContents, 0, fileContents.Length);
    requestStream.Close();

    FtpWebResponse response = (FtpWebResponse)request.GetResponse();
    Console.WriteLine("Upload File Complete, status {0}", response.StatusDescription);
    response.Close();
}

// METODO DE CONEXION FTP CON RASPBERRY

private void FTP_Download()
{
    // Get the object used to communicate with the server.

    FtpWebRequest request =
(FtpWebRequest)WebRequest.Create("ftp://192.168.2.14/Desktop/FINCA_LA_FAMA/data_secadoras.tex
t");

    request.Method = WebRequestMethods.Ftp.DownloadFile;

    request.Credentials = new NetworkCredential("user", "password");

    FtpWebResponse response = (FtpWebResponse)request.GetResponse();

```

```
Stream responseStream = response.GetResponseStream();

StreamReader reader = new StreamReader(responseStream);

int index_fila = 2;
// TABULACION DE DATOS EN ARCHIVO EXCEL

while (reader.Peek() >= 0)
{
    string linea_leida = reader.ReadLine();

    char[] delimitadores = {','};

    string[] linea_split = linea_leida.Split(delimitadores);

    if (linea_split.Length >= 15)
    {
        for (int index_col = 0; index_col < 16; index_col++)
        {
            //Console.WriteLine(linea_split[index_col] +"index" +index_col);

            try
            {
                writeExcelString(index_fila, index_col + 1, linea_split[index_col]);
            }

            catch
            {
                writeExcelString(index_fila, index_col + 1, "000");
            }
        }

        index_fila++;
    }
}

Console.WriteLine("Download Complete, status {0}", response.StatusDescription);

reader.Close();

response.Close();
```

```

    }

// METODO DE CONEXION FTP CON RASPBERRY PARA SEGUNDO ARCHIVO

private void FTP_Download2()

{

    // Get the object used to communicate with the server.

    FtpWebRequest request =
(FtpWebRequest)WebRequest.Create("ftp://192.168.2.14/Desktop/FINCA_LA_FAMA/archivo2.text");

    request.Method = WebRequestMethods.Ftp.DownloadFile;

    request.Credentials = new NetworkCredential("user", "password");

    FtpWebResponse response = (FtpWebResponse)request.GetResponse();

    Stream responseStream = response.GetResponseStream();

    StreamReader reader = new StreamReader(responseStream);

    int index_fila2 = 4;
// TABULACION DE DATOS EN ARCHIVO EXCEL

    while (reader.Peek() >= 0)

    {

        //Console.WriteLine("fila="+index_fila2);

        string linea_leida = reader.ReadLine();

        char[] delimitadores = {','};

        char[] delimitadore2 = {' '};

        string[] linea_split = linea_leida.Split(delimitadores);

        linea_split = linea_leida.Split(delimitadores2);

        if (linea_split.Length >= 11)

        {

            for (int index_col = 0; index_col < 11; index_col++)

            {

                //Console.WriteLine(linea_split[index_col] +"index" +index_col);

                try

                {

```

```

        writeExcel(index_fila2, index_col + 1, Convert.ToInt32(linea_split[index_col]));
    }
    catch
    {
        writeExcel(index_fila2, index_col + 1, 0);
    }
}
index_fila2++;
}
}
Console.WriteLine("Download Complete, status {0}", response.StatusDescription);
reader.Close();
response.Close();
}

```

// CREACION DE VENTANA PRINCIPAL E INICIALIZACION DE PROCESOS

```

public Form1()
{
    InitializeComponent();

    comboBox2.Items.Add("Voltaje de trabajo");
    comboBox2.Items.Add("Temperatura");
    comboBox2.Items.Add("Nivel de Combustible");

    comboBox1.Items.Add("Guardiola 1");
    comboBox1.Items.Add("Guardiola 2");
    comboBox1.Items.Add("Est-tica 1");
    comboBox1.Items.Add("Est-tica 2");
}

```

```
chart2.Series.Add("Voltaje");
chart2.Series.Add("Temperatura");
chart2.Series.Add("Combustible");
chart2.Series.RemoveAt(0);

chart1.Series.Add("Humedad");
chart1.Series.Add("Temperatura");
chart1.Series.RemoveAt(0);

// Excel para Datos de secadora y motor
MyApp = new Excel.Application();
MyApp.Visible = false;
MyBook = MyApp.Workbooks.Open("C:\\Datos_Secadoras1.xlsx");
MySheet = (Excel.Worksheet)MyBook.Sheets[1]; // Explicit cast is not required here
lastRow = ultimaFila();

// Excel para Secadoras
MyApp2 = new Excel.Application();
MyApp2.Visible = false;
MyBook2 = MyApp2.Workbooks.Open("C:\\Datos_Secadoras2.xlsx");
MySheet2 = (Excel.Worksheet)MyBook2.Sheets[1]; // Explicit cast is not required here
lastRow2 = ultimaFila2();
clearExcel(lastRow,lastRow2);
lastRow = ultimaFila();
Console.WriteLine(lastRow);

try
{
    FTP_Download();
```

```

    }

    catch { Console.WriteLine("Error de SECADORAS"); }

    try

    {

        FTP_Download2();

    }

    catch { Console.WriteLine("Error de DATOS"); }

    try

    {

        lastRow = ultimaFila();

        Console.WriteLine(lastRow);

        FTP_connect(readExcel(lastRow, 7) + "," + readExcel(lastRow, 8) + "," +
readExcel(lastRow, 9) + "," + readExcel(lastRow, 10) + "," + readExcel(lastRow, 1) + "," +
readExcel(lastRow, 2) + "," + readExcel(lastRow, 3) + "," + readExcel(lastRow, 4) + "," +
readExcel(lastRow, 5) + "," + readExcel(lastRow, 6) + "," + readExcel(lastRow, 13) + "," +
readExcel(lastRow, 11) + "," + readExcel(lastRow, 12));

    }

    catch { Console.WriteLine("Error de INTERNET"); }

    comboBox1.SelectedIndex = 0;

    comboBox2.SelectedIndex = 0;

    // Thread

    job = new ThreadStart(refrescar);

    thread = new Thread(job);

    thread.Start();

    this.FormClosed += new FormClosedEventHandler(Form1_FormClosed);

}

// SE GUARDAN LOS DATOS EN LOS ARCHIVOS EXCEL Y SE CIERRAN

void Form1_FormClosed(object sender, FormClosedEventArgs e)

```

```

{
    // Cerrar thread
    //Thread.Sleep(10000);
    thread.Abort();
    // Cerrar Archivo Excel 1
    MyBook.Save();
    MyBook.Close();
    MyApp.Quit();
    // Cerrar Archivo Excel 2
    MyBook2.Save();
    MyBook2.Close();
    MyApp2.Quit();
    Console.WriteLine("se cerraron los archivos");
}

// METODO LLAMADO EN SEGUNDO HILO DE PROCESO
public void refrescar()
{
    while (true){
        Thread.Sleep(60000);
        this.Invoke((MethodInvoker)delegate() { refrescar2(); });
        Console.WriteLine("Other thread: {0}");
    }
}

// METODO PARA LECTURA DE EXCEL
public int readExcel(int fila, int col)

```

```
{
    value2 = Convert.ToInt32(MySheet.Cells[fila,col].Value);
    return value2;
}

// METODO PARA ESCRITURA DE UN VALOR INT DE EXCEL

public void writeExcel(int fila, int columna, int valor)
{
    MySheet.Cells[fila, columna].Value = valor;
}

// METODO PARA BORRADO DE EXCEL

public void clearExcel(int lastrow, int lastrow2)
{
    for (int o = 4; o <= lastrow; o++)
    {
        for (int u = 1; u <= 11; u++)
        {
            MySheet.Cells[o, u].ClearContents();
        }
    }
    for (int y = 2; y <= lastrow2; y++)
    {
        for (int v = 1; v <= 16; v++)
        {
            MySheet2.Cells[y, v].ClearContents();
        }
    }
}
```

```
}

// METODO PARA ESCRITURA DE UN STRING DE EXCEL

public void writeExcelString(int fila, int columna, string valor)
{
    MySheet2.Cells[fila, columna].Value = valor;
}

// METODO PARA DETERMINAR LA ULTIMA FILA UTILIZADA DE EXCEL

public int ultimaFila()
{
    int index_ultima = 0;

    lastRow = MySheet.Cells.SpecialCells(Excel.XlCellType.xlCellTypeLastCell).Row;

    while (MySheet.Cells[lastRow, 1].Value == null)
    {
        lastRow--;
    }

    index_ultima = lastRow;

    return index_ultima;
}

public int ultimaFila2()
{
    int index_ultima = 0;

    lastRow2 = MySheet2.Cells.SpecialCells(Excel.XlCellType.xlCellTypeLastCell).Row;

    while (MySheet2.Cells[lastRow2, 1].Value == null)
    {
        lastRow2--;
    }
}
```

```

    index_ultima = lastRow2;

    return index_ultima;
}

```

// METODO PARA LLENADO DE DATOS PARA LAS GRAFICAS

```

private void fill_charts(string serie, int columna, int No_chart)
{
    lastRow = ultimaFila();
    if (No_chart==2){
        for (int i = 3; i <= lastRow; i++)
        {
            chart2.Series[serie].Points.AddXY(i-3, readExcel(i, columna));
        }
    }
    if (No_chart == 1)
    {
        for (int i = 3; i <= lastRow; i++)
        {
            chart1.Series[serie].Points.AddXY(i-3, readExcel(i, columna));
        }
    }
}

```

// CAMBIO DE GRAFICOS

```

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    lastRow = ultimaFila();
    if (comboBox2.SelectedIndex == 0)
    {

```

```

label3.Text = "Voltaje actual = " + readExcel(lastRow,11) + " V";
label10.Text = "Voltaje Promedio = " + readExcel(2, 22) + "V";

chart2.Series["Voltaje"].Points.Clear();

chart2.Series["Temperatura"].Points.Clear();

chart2.Series["Combustible"].Points.Clear();

fill_charts("Voltaje", 11, 2);

chart2.Series["Voltaje"].ChartType = SeriesChartType.FastLine;

chart2.Series["Voltaje"].Color = Color.Blue;
}

if (comboBox2.SelectedIndex == 1)
{
    label3.Text = "Temperatura actual = " + readExcel(lastRow, 9) + " ºC";
    label10.Text = "Temperatura Promedio = " + readExcel(2, 20) + " ºC";

    chart2.Series["Voltaje"].Points.Clear();

    chart2.Series["Temperatura"].Points.Clear();

    chart2.Series["Combustible"].Points.Clear();

    fill_charts("Temperatura", 9, 2);

    chart2.Series["Temperatura"].ChartType = SeriesChartType.FastLine;

    chart2.Series["Temperatura"].Color = Color.Orange;
}

if (comboBox2.SelectedIndex == 2)
{
    label3.Text = "Nivel de combustible actual = " + readExcel(lastRow, 10) + " %";
    label10.Text = "";

    chart2.Series["Voltaje"].Points.Clear();

    chart2.Series["Temperatura"].Points.Clear();

    chart2.Series["Combustible"].Points.Clear();
}

```

```

        fill_charts("Combustible", 10, 2);

        chart2.Series["Combustible"].ChartType = SeriesChartType.FastLine;

        chart2.Series["Combustible"].Color = Color.Red;
    }

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    lastRow = ultimaFila();

    if (comboBox1.SelectedIndex == 0)
    {
        label4.Text = "Temperatura actual = " + readExcel(lastRow, 1) + " ºC";

        label5.Text = "";

        label6.Text = "";

        label7.Text = "Temperatura Promedio = " + readExcel(2, 12) + " ºC";

        label8.Text = "";

        label9.Text = "";

        chart1.Series["Temperatura"].Points.Clear();

        chart1.Series["Humedad"].Points.Clear();

        fill_charts("Temperatura", 1, 1);

        chart1.Series["Temperatura"].ChartType = SeriesChartType.FastLine;

        chart1.Series["Temperatura"].Color = Color.Orange;
    }

    if (comboBox1.SelectedIndex == 1)
    {

```

```

label4.Text = "Temperatura actual = " + readExcel(lastRow, 2) + " [C";
label5.Text = "";
label6.Text = "";
label7.Text = "Temperatura Promedio = " + readExcel(2, 13) + " [C";
label8.Text = "";
label9.Text = "";

chart1.Series["Temperatura"].Points.Clear();
chart1.Series["Humedad"].Points.Clear();
fill_charts("Temperatura", 2, 1);
chart1.Series["Temperatura"].ChartType = SeriesChartType.FastLine;
chart1.Series["Temperatura"].Color = Color.Orange;

}

if (comboBox1.SelectedIndex == 2)
{
    label4.Text = "Temperatura actual = " + readExcel(lastRow, 3) + " [C";
    label5.Text = "Temperatura de Flujo = " + readExcel(lastRow, 5) + " [C";
    label6.Text = "Humedad = "+readExcel(lastRow,4)+"%";
    label7.Text = "Temperatura Promedio = " + readExcel(2, 14) + " [C";
    label8.Text = "Temperatura Prom. Flujo = " + readExcel(2, 16) + " [C";
    label9.Text = "Humedad Promedio = " + readExcel(2, 15) + "%";

    chart1.Series["Humedad"].Points.Clear();
    chart1.Series["Temperatura"].Points.Clear();
    fill_charts("Humedad", 4, 1);
    chart1.Series["Humedad"].ChartType = SeriesChartType.FastLine;
    chart1.Series["Humedad"].Color = Color.Blue;

}

```

```

if (comboBox1.SelectedIndex == 3)
{
    label4.Text = "Temperatura actual = " + readExcel(lastRow, 6) + " [C";
    label5.Text = "Temperatura de Flujo = " + readExcel(lastRow, 8) + " [C";
    label6.Text = "Humedad = " + readExcel(lastRow, 7) + "%";
    label7.Text = "Temperatura Promedio = " + readExcel(2, 17) + " [C";
    label8.Text = "Temperatura Prom. Flujo = " + readExcel(2, 19) + " [C";
    label9.Text = "Humedad Promedio = " + readExcel(2, 18) + "%";

    chart1.Series["Humedad"].Points.Clear();

    chart1.Series["Temperatura"].Points.Clear();

    fill_charts("Humedad", 7, 1);

    chart1.Series["Humedad"].ChartType = SeriesChartType.FastLine;

    chart1.Series["Humedad"].Color = Color.Blue;

}

}

// ACTUALIZACION DE DATOS

private void refrescar2() {

    Console.WriteLine("Other thread: {1}");

    lastRow = ultimaFila();

    lastRow2 = ultimaFila2();

    clearExcel(lastRow,lastRow2);

    lastRow = ultimaFila();

    Console.WriteLine(lastRow);

    try

    {

        FTP_Download();

```

```

    }

    catch { Console.WriteLine("Error de SECADORAS"); }

    try

    {

        FTP_Download2();

    }

    catch { Console.WriteLine("Error de DATOS"); }

    try

    {

        lastRow = ultimaFila();

        Console.WriteLine(lastRow);

        FTP_connect(readExcel(lastRow, 7) + "," + readExcel(lastRow, 8) + "," +
readExcel(lastRow, 9) + "," + readExcel(lastRow, 10) + "," + readExcel(lastRow, 1) + "," +
readExcel(lastRow, 2) + "," + readExcel(lastRow, 3) + "," + readExcel(lastRow, 4) + "," +
readExcel(lastRow, 5) + "," + readExcel(lastRow, 6) + "," + readExcel(lastRow, 13) + "," +
readExcel(lastRow, 11) + "," + readExcel(lastRow, 12));

    }

    catch { Console.WriteLine("Error de INTERNET"); }

    try

    {

        comboBox1.SelectedIndex = comboBox1.SelectedIndex + 1;

        comboBox1.SelectedIndex = comboBox1.SelectedIndex - 1;

        comboBox2.SelectedIndex = comboBox2.SelectedIndex + 1;

        comboBox2.SelectedIndex = comboBox2.SelectedIndex - 1;

    }

    catch

    {

        comboBox1.SelectedIndex = comboBox1.SelectedIndex - 1;

        comboBox1.SelectedIndex = comboBox1.SelectedIndex + 1;

```

```
        comboBox2.SelectedIndex = comboBox2.SelectedIndex - 1;
        comboBox2.SelectedIndex = comboBox2.SelectedIndex + 1;
    }

}

// DESCARGA DE ARCHIVO EXCEL CON DATOS DE SECADORA
private void button1_Click(object sender, EventArgs e)
{
    MyBook2.SaveAs("C:\\Users\\Fama\\Desktop\\Datos Secadoras.xlsx");
    // Cerrar Archivo Excel 2
    MyBook2.Save();
    MyBook2.Close();
    MyApp2.Quit();

    // Excel para Secadoras
    MyApp2 = new Excel.Application();
    MyApp2.Visible = false;
    MyBook2 = MyApp2.Workbooks.Open("C:\\Datos_Secadoras2.xlsx");
    MySheet2 = (Excel.Worksheet)MyBook2.Sheets[1]; //
}
}
```

D. CÓDIGO DE PROGRAMA DE RECOLECCIÓN DE DATOS EN RASPBERRY PI.

```

import Tkinter as tk
import tkMessageBox
import serial
import threading
import time

root = tk.Tk()
voltaje = tk.StringVar()
voltaje.set("0")
temp = tk.StringVar()
temp.set("0")
nivel = tk.StringVar()
nivel.set("0")
dato_enviar = "0"
nuevo = tk.Tk()
nuevo.destroy()

def get_data():
    #-----
    # Se piden los datos medidos por los sensores
    #-----
    while True:
        try:
            # Configuracion de comunicacion serial
            arduino=serial.Serial('/dev/ttyUSB0',9600)
            arduino.open()
            archivo = open('motor_data.text','w')
            time.sleep(3)
            datos_motor = ""
            arduino.write('s')
            time.sleep(0.01)
            while arduino.inWaiting() > 0:
                datos_motor += arduino.read()
            print(datos_motor)
            if datos_motor != "":
                print datos_motor
                archivo.write(datos_motor+'\n') #*****
                archivo.close()
                datos_split = datos_motor.split(',')
                print(datos_split)
                global voltaje
                voltaje.set(datos_split[2]+" V")
                global temp
                temp.set(datos_split[0]+" C")
                global nivel
                nivel.set(datos_split[1]+" %")
                print("printing...")

            if dato_enviar == "1":
                arduino.write('a')

```

```

        time.sleep(0.01)
    except:
        print("Erro de connexion")

    if root.winfo_exists() != 1:
        t.stop()

    return

# \n ---->Enter
class Application(tk.Frame):

    def __init__(self, master=None):
        tk.Frame.__init__(self, master)
        self.pack()
        self.createWidgets()

    t = threading.Thread(target=get_data)
    t.start()

    def createWidgets(self):
#codigo para labels
        self.etiqueta = tk.Label(self)
        self.etiqueta["text"] = "Fecha de Recibido: "
        self.etiqueta.grid(row = 0,column = 1)

        self.etiqueta1 = tk.Label(self)
        self.etiqueta1["text"] = "Quintales Recibidos en UVA:"
        self.etiqueta1.grid(row = 1,column = 1)

        self.etiqueta2 = tk.Label(self)
        self.etiqueta2["text"] = "Secciones: "
        self.etiqueta2.grid(row = 2,column = 1)

        self.etiqueta3 = tk.Label(self)
        self.etiqueta3["text"] = "Partida #: "
        self.etiqueta3.grid(row = 3,column = 1)

        self.etiqueta4 = tk.Label(self)
        self.etiqueta4["text"] = "Tipo: "
        self.etiqueta4.grid(row = 4,column = 1)

        self.etiqueta5 = tk.Label(self)
        self.etiqueta5["text"] = "Secadora: "
        self.etiqueta5.grid(row = 5,column = 1)

        self.etiqueta6 = tk.Label(self)
        self.etiqueta6["text"] = "Fecha de Carga: "
        self.etiqueta6.grid(row = 6,column = 1)

        self.etiqueta7 = tk.Label(self)
        self.etiqueta7["text"] = "Hora de Carga: "
        self.etiqueta7.grid(row = 7,column = 1)

```

```
self.etiqueta8 = tk.Label(self)
self.etiqueta8["text"] = "Cantidad Cargada: "
self.etiqueta8.grid(row = 8,column = 1)
```

```
self.etiqueta9 = tk.Label(self)
self.etiqueta9["text"] = "Fecha de Descarga: "
self.etiqueta9.grid(row = 9,column = 1)
```

```
self.etiqueta10 = tk.Label(self)
self.etiqueta10["text"] = "Hora de Descarga: "
self.etiqueta10.grid(row = 10,column = 1)
```

```
self.etiqueta11 = tk.Label(self)
self.etiqueta11["text"] = "Cantidad Descargada: "
self.etiqueta11.grid(row = 11,column = 1)
```

```
self.etiqueta12 = tk.Label(self)
self.etiqueta12["text"] = "% de Humedad: "
self.etiqueta12.grid(row = 12,column = 1)
```

```
self.etiqueta13 = tk.Label(self)
self.etiqueta13["text"] = "HRS. Duracion en Secadora: "
self.etiqueta13.grid(row = 13,column = 1)
```

```
self.etiqueta14 = tk.Label(self)
self.etiqueta14["text"] = "CANT. Lena Utilizada/Tareas: "
self.etiqueta14.grid(row = 14,column = 1)
```

```
self.etiqueta15 = tk.Label(self)
self.etiqueta15["text"] = "Rendimiento: "
self.etiqueta15.grid(row = 15,column = 1)
```

#Codigo para text input

```
self.input_text0 = tk.Entry(self)
self.input_text0.grid(row = 0,column = 2)
```

```
self.input_text1 = tk.Entry(self)
self.input_text1.grid(row = 1,column = 2)
```

```
self.input_text2 = tk.Entry(self)
self.input_text2.grid(row = 2,column = 2)
```

```
self.input_text3 = tk.Entry(self)
self.input_text3.grid(row = 3,column = 2)
```

```
self.input_text4 = tk.Entry(self)
self.input_text4.grid(row = 4,column = 2)
```

```
self.input_text5 = tk.Entry(self)
self.input_text5.grid(row = 5,column = 2)
```

```
self.input_text6 = tk.Entry(self)
self.input_text6.grid(row = 6,column = 2)
```

```
self.input_text7 = tk.Entry(self)
self.input_text7.grid(row = 7,column = 2)
```

```

self.input_text8 = tk.Entry(self)
self.input_text8.grid(row = 8,column = 2)

self.input_text9 = tk.Entry(self)
self.input_text9.grid(row = 9,column = 2)

self.input_text10 = tk.Entry(self)
self.input_text10.grid(row = 10,column = 2)

self.input_text11 = tk.Entry(self)
self.input_text11.grid(row = 11,column = 2)

self.input_text12 = tk.Entry(self)
self.input_text12.grid(row = 12,column = 2)

self.input_text13 = tk.Entry(self)
self.input_text13.grid(row = 13,column = 2)

self.input_text14 = tk.Entry(self)
self.input_text14.grid(row = 14,column = 2)

self.input_text15 = tk.Entry(self)
self.input_text15.grid(row = 15,column = 2)

#boton de Guardar Datos
self.savedata= tk.Button(self)
self.savedata["text"] = "Guardar Datos"
self.savedata["command"] = self.Guardar_Datos
self.savedata.grid(row = 17,column = 0) # ***** cambiar row *****

#boton de Ver Ultimo Ingreso
self.ultimo_ingreso= tk.Button(self)
self.ultimo_ingreso["text"] = "Ver Ultimo Ingreso"
self.ultimo_ingreso["command"] = self.Ver_Ultimo_Ingreso
self.ultimo_ingreso.grid(row = 17,column = 1) # ***** cambiar row *****

#boton de Ver Ultimo Ingreso
self.modificar= tk.Button(self)
self.modificar["text"] = "Modificar Ultimo Ingreso"
self.modificar["command"] = self.Modificar_Ultimo
self.modificar.grid(row = 17,column = 2) # ***** cambiar row *****

#boton de Ver Datos Motor
self.datos_motor= tk.Button(self)
self.datos_motor["text"] = "Control de planta auxiliar"
self.datos_motor["command"] = self.Datos_Motor
self.datos_motor.grid(row = 17,column = 3) # ***** cambiar row *****

#boton de cerrar
self.QUIT = tk.Button(self, text="Cerrar", fg="red",
                      command=root.destroy)
self.QUIT.grid(row = 17,column = 4)

def Datos_Motor(self):

```

```

nuevo = tk.Toplevel()
nuevo.wm_title("Control de Planta Auxiliar")
nuevo.etiqueta_voltaje = tk.Label(nuevo, textvariable=voltaje)
nuevo.etiqueta_voltaje.grid(row = 1,column = 2)

nuevo.etiqueta_temp = tk.Label(nuevo, textvariable=temp)
nuevo.etiqueta_temp.grid(row = 2,column = 2)

nuevo.etiqueta_nivel = tk.Label(nuevo, textvariable=nivel)
nuevo.etiqueta_nivel.grid(row = 3,column = 2)

nuevo.etiqueta_voltaje1 = tk.Label(nuevo, text="voltaje = ")
nuevo.etiqueta_voltaje1.grid(row = 1,column = 1)

nuevo.etiqueta_temp1 = tk.Label(nuevo, text="Temperatura = ")
nuevo.etiqueta_temp1.grid(row = 2,column = 1)

nuevo.etiqueta_nivel1 = tk.Label(nuevo, text="Nivel = ")
nuevo.etiqueta_nivel1.grid(row = 3,column = 1)

#boton de 110
nuevo.uno= tk.Button(nuevo)
nuevo.uno["text"] = "110V"
nuevo.uno["command"] = self.volt_110
nuevo.uno.grid(row = 4,column = 0) # *****

#boton de 220
nuevo.dos= tk.Button(nuevo)
nuevo.dos["text"] = "220V"
nuevo.dos["command"] = self.volt_220
nuevo.dos.grid(row = 4,column = 1) # *****

#boton de 340
nuevo.tres= tk.Button(nuevo)
nuevo.tres["text"] = "Apagar"
nuevo.tres["command"] = self.volt_340
nuevo.tres.grid(row = 4,column = 2) # *****

def volt_110(self):
    print("setting 110")
    data_enviar = "1"

def volt_220(self):
    print("setting 220")
    data_enviar = "2"

def volt_340(self):
    print("setting 440")
    data_enviar = "3"

def Ver_Ultimo_Ingreso(self):
    archivo = open('data_secadoras.text','r')
    last = ""
    line2 = 0

```

```

#print "ultimo ingreso"
for line2 in archivo:
    pass
last = line2
archivo.close()
#print (last)
line = last.split(',')
tkMessageBox.showinfo(title="Ultimo Dato Guardado", message="Fecha de Recibido: "+line[0]+
"\nQuintales Recibidos en UVA:"+line[1]+"nSecciones: "+line[2]+"nPartida #: "+line[3]+"nTipo:
"+line[4]+"nSecadora: "+line[5]+"nSecadora: "+line[6]+"nHora de Carga: "+line[7]+"nCantidad
Cargada: "+line[8]+"nFecha de Descarga: "+line[9]+"nHora de Descarga: "+line[10]+"nCantidad
Descargada: "+line[11]+"n% de Humedad: "+line[12]+"nHRS. Duracion en Secadora:
"+line[13]+"nCANT. Lena Utilizada/Tareas: "+line[14]+"nRendimiento: "+line[15])

```

```

def Modificar_Ultimo(self):

    archivo = open('data_secadoras.txt','r')
    lineas_archivo = archivo.readlines()
    archivo.close()
    del lineas_archivo[len(lineas_archivo)-1]
    archivo = open('data_secadoras.txt','w+')
    archivo.truncate(0)
    archivo.close()
    archivo = open('data_secadoras.txt','a')
    for x in range(0, len(lineas_archivo)):
        archivo.write(lineas_archivo[x])
        archivo.write(self.input_text0.get()+',')
        archivo.write(self.input_text1.get()+',')
        archivo.write(self.input_text2.get()+',')
        archivo.write(self.input_text3.get()+',')
        archivo.write(self.input_text4.get()+',')
        archivo.write(self.input_text5.get()+',')
        archivo.write(self.input_text6.get()+',')
        archivo.write(self.input_text7.get()+',')
        archivo.write(self.input_text8.get()+',')
        archivo.write(self.input_text9.get()+',')
        archivo.write(self.input_text10.get()+',')
        archivo.write(self.input_text11.get()+',')
        archivo.write(self.input_text12.get()+',')
        archivo.write(self.input_text13.get()+',')
        archivo.write(self.input_text14.get()+',')
        archivo.write(self.input_text15.get()+'\n')
    archivo.close()

```

```

self.input_text0.delete(0, tk.END)
self.input_text1.delete(0, tk.END)
self.input_text2.delete(0, tk.END)
self.input_text3.delete(0, tk.END)
self.input_text4.delete(0, tk.END)
self.input_text5.delete(0, tk.END)
self.input_text6.delete(0, tk.END)
self.input_text7.delete(0, tk.END)
self.input_text8.delete(0, tk.END)
self.input_text9.delete(0, tk.END)

```

```

self.input_text10.delete(0, tk.END)
self.input_text11.delete(0, tk.END)
self.input_text12.delete(0, tk.END)
self.input_text13.delete(0, tk.END)
self.input_text14.delete(0, tk.END)
self.input_text15.delete(0, tk.END)

```

```

def Guardar_Datos(self):
    archivo = open('data_secadoras.text','a')
    archivo.write(self.input_text0.get()+',')
    archivo.write(self.input_text1.get()+',')
    archivo.write(self.input_text2.get()+',')
    archivo.write(self.input_text3.get()+',')
    archivo.write(self.input_text4.get()+',')
    archivo.write(self.input_text5.get()+',')
    archivo.write(self.input_text6.get()+',')
    archivo.write(self.input_text7.get()+',')
    archivo.write(self.input_text8.get()+',')
    archivo.write(self.input_text9.get()+',')
    archivo.write(self.input_text10.get()+',')
    archivo.write(self.input_text11.get()+',')
    archivo.write(self.input_text12.get()+',')
    archivo.write(self.input_text13.get()+',')
    archivo.write(self.input_text14.get()+',')
    archivo.write(self.input_text15.get()+'\n')
    archivo.close()

```

```

self.input_text0.delete(0, tk.END)
self.input_text1.delete(0, tk.END)
self.input_text2.delete(0, tk.END)
self.input_text3.delete(0, tk.END)
self.input_text4.delete(0, tk.END)
self.input_text5.delete(0, tk.END)
self.input_text6.delete(0, tk.END)
self.input_text7.delete(0, tk.END)
self.input_text8.delete(0, tk.END)
self.input_text9.delete(0, tk.END)
self.input_text10.delete(0, tk.END)
self.input_text11.delete(0, tk.END)
self.input_text12.delete(0, tk.END)
self.input_text13.delete(0, tk.END)
self.input_text14.delete(0, tk.END)
self.input_text15.delete(0, tk.END)

```

```

root.wm_title("Finca La Fama")
app = Application(master=root)
app.mainloop()

```

E. CÓDIGO DE SEGUNDO PROGRAMA DE RECOLECCIÓN DE DATOS EN RASPBERRY PI.

```

# Librerias utilizadas
import time
import os
import serial

#-----

#-----
# funcion de limpieza de pantalla

def cls():
    os.system(['clear','cls'][os.name == 'nt'])

# Archivo a guardar

archivo = open('saved_data.text','a')

# Configuracion de comunicacion serial
arduino=serial.Serial('/dev/ttyACM0',9600)
arduino.open()

# Inicializacion de variables

temp_hum = "no se recibieron mediciones"

#-----
# poll de datos
#-----
while True: # *****
    archivo = open('saved_data.text','a')

    time.sleep(15)

#-----
# Se piden los datos medidos por los sensores
#-----
    temp_hum = "
    arduino.write('s')
    time.sleep(0.01)
    while arduino.inWaiting() > 0:
        temp_hum += arduino.read()
    if temp_hum != "":
        print temp_hum
        archivo.write(temp_hum+'\n')
    archivo.close()

#-----

#-----

```

F. PROGRAMA PIC18F45K22 PARA TOMA DE DATOS DE LOS SENSORES Y ENVIO.

```
/*
```

```
Programa para manejar los sensores y transferir los datos.
```

```
Reloj interno de 16MHz X 4
```

```
*/
```

```
// LCD module connections
```

```
sbit LCD_RS at LATB4_bit;
```

```
sbit LCD_EN at LATB5_bit;
```

```
sbit LCD_D4 at LATB0_bit;
```

```
sbit LCD_D5 at LATB1_bit;
```

```
sbit LCD_D6 at LATB2_bit;
```

```
sbit LCD_D7 at LATB3_bit;
```

```
sbit LCD_RS_Direction at TRISB4_bit;
```

```
sbit LCD_EN_Direction at TRISB5_bit;
```

```
sbit LCD_D4_Direction at TRISB0_bit;
```

```
sbit LCD_D5_Direction at TRISB1_bit;
```

```
sbit LCD_D6_Direction at TRISB2_bit;
```

```
sbit LCD_D7_Direction at TRISB3_bit;
```

```
//Escribir SD
```

```
sbit Mmc_Chip_Select at LATC0_bit;
```

```
sbit Mmc_Chip_Select_Direction at TRISC0_bit;
```

```
//variables
```

```
unsigned int ADC_var1[50], TMR0_var[50];
```

```
unsigned long ADC, TMR0, size, a1, ADC_temp;
```

```
int i,j,k,g,t,h,contBUF,INT_ENVIAR,ON_BLUE,ingreso,comas,cont_T1;
```

```
char ADC_txt1[7], ADC_txt2[7],TMR0_txt[7], longitud[15], latitud[15], hora[15];
```

```
char UART1_Data_Ready();
```

```
char uart_rd, fla, F_cont, F_var, A_var;
```

```

char set_var, set_cont, buffer;
unsigned char set_output[100];
float F_longitud, F_latitud;

unsigned int error;

char caracter, caracter1[1];

char file_contents[50] = "Latitud, Longitud, Presion, Caudal";
unsigned short loop, loop2;

```

```

//*****INTERRUPCIONES*****

```

```

void interrupt() {

    if (TMR1IF_bit == 1) { //cada 0.125s.
        //cont_T1++;
        //if (cont_T1 == 2) {
            //TMR0_var[F_cont] = TMR0L;
            ADC_var1[F_cont] = ADC_Read(2);
            F_cont++;
            TMR0H = 0;
            TMR0L = 0; //para comenzar la cuenta
            TMR1H = 0;
            TMR1L = 0;
            TMR1IF_bit = 0;
            //fla=1;
            // cont_T1 = 0;
        //}
    }
}

```

```

//***** PROGRAMA PRINCIPAL *****

```

```

void main() {
    PORTA = 0;
    PORTB = 0;

```

```

PORTC = 0;
PORTD = 0;

//oscilador de 16MHz interno
OSCCON = 0b01110110;

ADC_Init(); //inicializa ADC con los parametros normales
ANSELA = 0x0C;      // AN2 y AN3 como analogico
ANSELB = 0;        // todos los demas pines se dejan como digitales
ANSELC = 0;
C1ON_bit = 0;      // Disable comparators
C2ON_bit = 0;

TRISA = 0xFF;      // PORTA como entrada
TRISC = 0b10000000; // PORTC como salida menos RC7 entrada por RX
TRISB = 0b00001100; // PORTB como salida RB2 COMO ENTRADA
TRISA0_bit = 1;

//UART
UART1_Init(9600);
Delay_ms(100);

//SPI
TRISC0_bit=1; //SS como entrada
TRISC3_bit=0; //SCK como salida
TRISC4_bit=1; //SDI1 entrada
TRISC5_bit=0; //SDO1 salida

RB7_bit = 0;

SPI1_Init(); //inicializando a Fosc/4
Delay_ms(100);

error = Mmc_Fat_Init();
if (error == 255){
    RB7_bit = 1;
}

```

```

}
Delay_ms(100);
Mmc_Fat_Assign("DATOS.txt", 0xA0);    // Find existing file or create a new one
Mmc_Fat_Append();                    // To clear file and start with new data

TMR1H = 0;
TMR1L = 0;
//interrupciones
INTCON = 0xC0;
GIE_bit = 0; //general
PEIE_bit = 1; //perifericos
TMR1IE_bit = 1; //Timer1
TMR3IE_bit = 0; //Timer3

//Lcd_Init();

//Tiemer 0 utilizado para contar la entrada del sensor de caudal
TMR0L = 0;
T0IE_bit = 0; //sin interrupcion
TRISA4_bit = 1; //como entrada
T0CON = 0b11101000; //sin pre entrada en RA4

//Timer 1
TMR1H = 0;
TMR1L = 0;
T1CON = 0b00110011; // para el tiemr1 16bit como TIMER pre de 1:8

//Timer 3
TMR3H = 0;
TMR3L = 0;
T3CON = 0b00110110;

/*
Lcd_Cmd(_LCD_CLEAR);    // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off

```

```

Lcd_Out(1, 1, "V1:   Caudal  ");
*/
F_cont = 1;
F_var = 1;
GIE_bit = 1;

i=0;
j=0;
INT_ENVIAR = 0;
TMR0_var[0]=0;
A_var = 1;
ADC_var1[0]=0;
TMR0=0;
ADC=0;
ON_BLUE=0;
caracter = 't';
contBUF = 0;
ingreso=0;
comas=0;
cont_T1=0;
F_cont = 1;
/*
g=14;
t=14;
for(i=0; i>14; i++){
    latitud[i]='-';
    longitud[i]='-';
}

i=0;
//$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65
*/
RB0_bit = 1; //control del GPS
RB1_bit = 0; //control del bluetooth
RB2_bit = 0;
RB7_bit = 0; //indica si existe algun error

```

```

TMR1IF_bit = 0;
Delay_ms(100);
//***** CICLO PRINCIPAL *****
while(1) {
// Ciclo para ENVIO de datos
if(RB2_bit==1){
    GIE_bit = 0;
    i=0;
    RB0_bit = 0;
    RB1_bit = 1;
    do{
        if (UART1_Data_Ready() == 1) {
            i++;
            if (i>70){
                i=1;
            }
            set_output[i] = UART1_Read();
            if (set_output[i]=='h'){
                //se apaga el GPS y se guarda el fin de la jornada
                i=1;
                a1=0;
                RB7_bit = 1;
                Delay_ms(100);
                Mmc_Fat_Write("**FIN**",8);
                Delay_ms(100);

                Mmc_Fat_Assign("DATOS.txt", 0);
                Mmc_Fat_Append();
                Mmc_Fat_Reset(&size);// To read file, procedure returns size of file
                //size = Mmc_Fat_Get_File_Size(); // se coloco debido a que la instruccion anterior
no funciona
                //size = 10;
                for (a1 = 1; a1 <= size; a1++) {
                    Mmc_Fat_Read(&caracter);
                    UART1_Write(caracter); // Escribir por UART
                }
                Mmc_Fat_Assign("DATOS.txt", 0xA0);

```

```

Mmc_Fat_Append();
i=0;
while (1){
    if(UART1_Data_Ready() == 1) {
        set_output[i] = UART1_Read();
        if (set_output[i]!='J'){
            Mmc_Fat_Delete(); //para eliminar el archivo
            Delay_ms(1000);
            PORTA=0;
            PORTB=0;
            PORTC=0;
            Delay_ms(100);
            asm sleep;
        }
    }
}

}
}
}while(1);
}
// Ciclo para RECOLECCION de datos
/*
if (fla == 1){
    GIE_bit = 0;
    ADC_var1[A_var] = ADC_Read(2)/4; // 10 bit de la conversión ADC para convertirlo a un
BYTE
    A_var++;
    if(A_var==10){
        ADC = 0;
        A_var = 1;
        for (A_var=1; A_var<10; A_var++){
            ADC = ADC_var1[A_var]+ADC;
        }
        ADC_var1[0]=ADC/10;
        A_var=1;
        IntToStr(ADC_var1[0], ADC_txt1); //pasa de int a string

```

```

    }

    F_cont++;
    //F_var = (TMR0_var + F_var)/2;
    if(F_cont== 3) {
        //TMR0_var = F_var/F_cont;
        F_var = 1;
        TMR0=0;

        for (F_var=1; F_var<F_cont; F_var++){
            TMR0= TMR0_var[F_var] + TMR0;
        }
        TMR0_var[0]= TMR0/F_cont;
        F_cont = 1;
        IntToStr(TMR0_var[0], TMR0_txt);
        //Lcd_Out(2, 10, TMR0_txt);    //lo coloca en la pantalla

    }

    fla = 0;
    GIE_bit = 1;
}
*/
//if (INT_ENVIAR == 5){
// INT_ENVIAR = 0;
// Delay_ms(100);
//}

//***** DETENER *****
if(PORTA.RA0 == 1){
    while (PORTA.RA0 == 1) {
        INT_ENVIAR = 0;
        RB7_bit = 0;
    }
}

//***** RECIVIR DATOS *****
if (UART1_Data_Ready() == 1) {

```

```

i++;
set_output[i] = UART1_Read();

if (i > 70){
    i=0;
    comas=0;
    ingreso=0;
}
if (set_output[i] == '$'){
    set_output[0]='$';
    i=0;    //para colocar al inicio
    comas=0;
    ingreso=0;

}
if (set_output[i]=='A' && set_output[i-1]=='G'){
    ingreso = 1;
    comas=0;

}
if (set_output[i]==','){
    comas++;
}
//UART1_Write(set_output[i]);

if (comas==14 && ingreso==1){
    GIE_bit = 0;
    //CONVERSION ADC
    ADC = 0;
    A_var = 1;
    for (A_var=1; A_var<F_cont; A_var++){
        ADC_temp = ADC_var1[A_var]/4;
        ADC = ADC_temp + ADC;
    }
    ADC_var1[0]=ADC/F_cont;
    IntToStr(ADC_var1[0], ADC_txt1);//pasa de int a string

```

```

//F_var = 1;
//TMR0 = 0;
//for (F_var=1; F_var<F_cont; F_var++){
//  TMR0= TMR0_var[F_var] + TMR0;
//}

TMR0_var[0]= TMR0L;
IntToStr(TMR0_var[0], TMR0_txt);
F_cont = 1;

k=0;
ingreso = 0;
comas =0;
/*
for (a1 = 0; a1 <= i; a1++) {
UART1_Write(set_output[a1]);    // Escribir por UART
}
*/
do{
  buffer = set_output[k];
  k++;
  if (buffer == '$'){
    contBUF=0;
    t=0;
    g=0;
    h=0;
  }

  if (contBUF == 1){
    hora[h] = buffer;
    h++;
  }
  if (contBUF == 2 ){
    latitud[t] = buffer;
    t++;
  }
}

```

```

if (contBUF == 3 ){
    latitud[t] = buffer;
    t++;
}
if (contBUF == 4){
    longitud[g] = buffer;
    g++;
}
if (contBUF == 5){
    longitud[g] = buffer;
    g++;
}
if (buffer == ','){
    contBUF++;
}
} while(contBUF!=14);
i=0;
//GIE_bit = 0;
//RB7_bit = 1;
if (h>2){
    Mmc_Fat_Write(hora,h-1);
}
else{
    Mmc_Fat_Write(" ", 1);
}
Mmc_Fat_Write(" ", 1);
if (t>2){
    Mmc_Fat_Write(latitud,t-1);
}
else{
    Mmc_Fat_Write(" ", 1);
}
Mmc_Fat_Write(" ", 1);
if (g>2){
    Mmc_Fat_Write(longitud,g-1);
}
else{

```

```

        Mmc_Fat_Write(",", 1);
    }
    Mmc_Fat_Write(",", 1);
    Mmc_Fat_Write(ADC_txt1, 7);
    Mmc_Fat_Write(",", 1);
    Mmc_Fat_Write(TMR0_txt, 7);
    Mmc_Fat_Write(",", 1);
    TMR0H = 0;
    TMR0L = 0;
    //RB7_bit = 0;
    GIE_bit = 1;
//***** ENCENDER BLUETOOTH *****
    /*
    UART1_Write_Text(longitud);
    UART1_Write_Text(latitud);
    F_longitud = atof(longitud);
    F_latitud = atof(latitud);
    F_longitud = F_longitud + 0.123;
    FloatToStr(F_longitud, longitud);
    FloatToStr(F_latitud, latitud);
    UART1_Write_Text(longitud);
    UART1_Write_Text(latitud);
    */
}

}

//$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65

//enviar datos por UART
/*
if (i==59) { //PARA Enviar datos
    GIE_bit = 0;
    i=0;
    Mmc_Fat_Write(latitud,t);
    Mmc_Fat_Write(longitud,g);
    Mmc_Fat_Write(",", 1);
    Mmc_Fat_Write(ADC_txt1, 7);

```

```

        Mmc_Fat_Write(",", 1);
        Mmc_Fat_Write(TMR0_txt, 7);
        GIE_bit = 1;

    }
    */
//***** ENVIAR DATOS POR UART *****
funciono
if(RA1_bit==1){
    GIE_bit = 0;
    i=1;
    a1=0;

    Mmc_Fat_Assign("DATOS.txt", 0);
    Mmc_Fat_Append();
    Mmc_Fat_Reset(&size);      // To read file, procedure returns size of file
    //size = Mmc_Fat_Get_File_Size(); // se coloco debido a que la instruccion anterior no

    //size = 10;
    for (a1 = 1; a1 <= size; a1++) {
        Mmc_Fat_Read(&caracter);
        UART1_Write(caracter);    // Escribir por UART
    }
    Mmc_Fat_Assign("DATOS.txt", 0xA0);
    Mmc_Fat_Append();
    i=0;
//***** DETENER *****
if(PORTA.RA1 == 1){
    while (PORTA.RA1 == 1) {
        INT_ENVIAR = 0;
    }
}
if(PORTA.RA0 == 1){
    Mmc_Fat_Delete(); //para eliminar el archivo
    //Coloca un nuevo archivo y lo Inicializa
    Mmc_Fat_Assign("DATOS.txt", 0xA0);    // Find existing file or create a new one
    Mmc_Fat_Append();                    // To clear file and start with new data
    //Mmc_Fat_write("***Inicio**",10);

```

```
    }  
    GIE_bit = 1;  
  
    }  
    }  
  
}
```

G. PROGRAMA RASPBERRY PI

```

import sys
import bluetooth
import time
import subprocess
import commands
import datetime
from math import sin,cos,sqrt,asin,pi

print 'Esperando coneccion ...'

bd_addr = "00:14:03:11:07:36"
port = 1
data1 = '***,'
latitud = []
longitud = []
presion = []
p_presion = []
caudal = []
p_caudal = []
escala = []
hora = []
i=1
N_S = []
E_W = []
temp_presion = []
temp_caudal = []
num_presion = 0
num_caudal = 0
latitud1 = 0
longitud1 = 0
latitud2 = 0
longitud2 = 0
distancia_L=[]

##### DEFINICIONES#####
def crearcsv():

```

```

archi=open('datos/finca1.csv','w')
archi.close()

def escribircsv():
    archi=open('datos/finca1.csv','a')

archi.write('finca,bomba,fecha,hora,latitud,longitud,presion,p_presion,caudal,p_caudal,escala\n')

for i in range(len(longitud)):
    archi.write('Finca.Fumogua')
    archi.write(',')
    archi.write(' 1 ')
    archi.write(',')
    archi.write(fecha)
    archi.write(',')
    archi.write(hora[i])
    archi.write(',')
    archi.write(str(latitud[i]))
    archi.write(',')
    archi.write(str(longitud[i]))
    archi.write(',')
    archi.write(str(presion[i]))
    archi.write(',')
    archi.write(str(p_presion[i]))
    archi.write(',')
    archi.write(str(caudal[i]))
    archi.write(',')
    archi.write(str(p_caudal[i]))
    archi.write(',')
    archi.write(str(escala[i]))
    archi.write('\n')

archi.close()

def coordenada(lat,lon,c1,c2):
    lat2=""
    lat1=""

```

```

lon1=""
lon2=""
for i in range(len(lat)-7, len(lat)):
    lat2=lat2+lat[i]
for j in range(len(lat)-8,-1,-1):
    lat1=lat[j]+lat1
for i in range(len(lon)-7, len(lon)):
    lon2=lon2+lon[i]
for j in range(len(lon)-8,-1,-1):
    lon1=lon[j]+lon1
latitud = float(lat1)+(float(lat2)/60)
longitud = float(lon1)+(float(lon2)/60)
if c2 == "W":
    longitud=longitud*-1
if c1 == "S":
    latitud = latitud*-1
return latitud, longitud

def dist(la1, lo1, la2, lo2):
    c = pi/180
    la1 = la1*c
    la2 = la2*c
    lo1 = lo1*c
    lo2 = lo2*c
    distancia = 6378137*2*asin(sqrt(sin(c*(la2-la1)/2)**2 + cos(c*la1)*cos(c*la2)*sin(c*(lo2-
lo1)/2)**2))
    distancia = round(distancia, 2)
    return distancia

#resivir
while True:
    try:
        server_sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )
        server_sock.connect((bd_addr, port))
        break
    except:

```

```

time.sleep(1)

time.sleep(5)
#message = raw_input('send: ')
message = 'h'
server_sock.send(message)
while True:
    data = server_sock.recv(8)
    data1 = data1 + data
    if (data1.find('**FIN**') != -1) : break
server_sock.send('J')
server_sock.close()

data_lista = data1.split(',')
#*****
#para eliminar \x00 de las medidas en la presion y el caudal
#ya que no funcono el replace directo en data1
for i in range(len(data_lista)):
    r = data_lista[i]
    r = r.replace("\x00", "")
    data_lista[i]=r

print 'Lista Recivida: ', data_lista
print 'Tamano de Lista: ', len(data_lista)
i=8
# ciclo para separar la lista de entrada en hora, latitud, longitud,
# caudal y presion
while (len(data_lista) > (i+7)):

    try:
        sinc_lista = data_lista[i]
        if (sinc_lista[len(sinc_lista) - 4] == '.'):
            sinc = 1
        else:
            sinc = 0
        try:
            if (sinc_lista[len(sinc_lista) - 4] != '.'):

```

```

    print 'sincronizando...'
    while True:
        if (sinc_lista[len(sinc_lista) - 4] == '.'):
            sinc = 1
            break
        i = i+1
        sinc_lista = data_lista[i]
    except:
        sinc = 0
except:
    sinc = 0
    i = i+1

if (sinc ==1):
    try:
        prueba_pre = int(data_lista[i+5])
        prueba_cau = int(data_lista[i+6])

    if (data_lista[i+1]==""): #no hay dato de GPS
        i = i+5
        temp_presion.append(data_lista[i])
        i = i+1
        temp_caudal.append(data_lista[i])
        i= i+1
    else:
        hora.append(data_lista[i])
        i = i+1
        latitud.append(data_lista[i])
        i = i+1
        N_S.append(data_lista[i])
        i = i+1
        longitud.append(data_lista[i])
        i = i+1
        E_W.append(data_lista[i])
        i = i+1
    if (len(temp_presion) > 0):
        temp_presion.append(data_lista[i])

```

```

len_lista = len(temp_presion)
for b in range(len(temp_presion)):
    r = temp_presion[b]
    r = r.replace(" ", "")
    temp_presion[b]=r
    num_presion = num_presion + (int(temp_presion[b]))

presion.append(num_presion/len_lista)
temp_presion = []
i = i+1
else:
    for b in range(len(data_lista[i])):
        r = data_lista[i]
        r = r.replace(" ", "")
    presion.append(int(r))
    i = i+1

if (len(temp_caudal) > 0):
    temp_caudal.append(data_lista[i])
    len_lista = len(temp_caudal)
    for b in range(len(temp_caudal)):
        r = temp_caudal[b]
        r = r.replace(" ", "")
        temp_caudal[b]=r
        num_caudal = num_caudal + (int(temp_caudal[b]))

caudal.append(num_caudal/len_lista)
temp_caudal = []
i = i+1
else:
    for b in range(len(data_lista[i])):
        r = data_lista[i]
        r = r.replace(" ", "")
    caudal.append(int(r))
    i = i+1
except:
    i = i+7

```

```

if not (len(data_lista) > (i+7)):
    break

try:
    print 'ERROR !!!', i, 'hora ', data_lista[i]
except:
    print 'ERROR !!!'

print 'hora', len(hora)
print 'N/S', len(N_S)
print 'E/W', len(E_W)
print 'longitud', len(longitud)
print 'latitud', len(latitud)
print 'presion', len(presion)
print 'caudal', len(caudal)

#fase para fecha
ahora = datetime.datetime.now()
dia = str(ahora.day)
if (len(dia) == 1):
    dia = '0'+dia
mes= str(ahora.month)
if (len(mes) == 1):
    mes = '0'+mes

fecha = dia+'/'+mes+'/'+str(ahora.year)

#fase para hora
for i in range(len(hora)):
    h=hora[i]
    H_int = int(h[:2])
    if (H_int < 6):
        H_lista = ['18','19','20','21','22','23']
        H = H_lista[H_int]+'-'+h[2:4]+'-'+h[4:6]
    else:

```

```

H = str(H_int-6)+'h[2:4]'+h[4:6]

hora[i]=H
#fase para latitud y long
a=0
for i in range(len(latitud)):
    c1 = latitud[i]
    c2 = N_S[i]
    c3 = longitud[i]
    c4 = E_W[i]
    latitud[i], longitud[i] = coordenada(c1,c3,c2,c4)
    if (i>0 and i<(len(latitud))):
        distancia = dist(latitud[i-1],longitud[i-1],latitud[i],longitud[i])
        distancia_L.append(distancia)

#fase para presion
for i in range(len(presion)):
    if int(presion[i]) <= 61:
        x = int(presion[i])
        p_escala = (0.0017*x*x)+(0.1025*x)-0.4446
    elif int(presion[i]) > 61:
        x = int(presion[i])
        p_escala = (-0.0325*x*x)-(0.3861*x)+10.441
    if (p_escala > 10):
        p_presion.append(10)
    if (p_escala < 0):
        p_presion.append(0)
    if (p_escala>=0 and p_escala<=10):
        p_presion.append(round(p_escala,1))

#fase para caudal
for i in range(len(caudal)):
    if int(caudal[i]) <= 8:
        x = int(caudal[i])
        c_escala = (0.1483*x*x)-(0.0327*x)+0.6429

```

```

elif int(caudal[i]) > 8:
    x = int(caudal[i])
    c_escal = (0.2024*x*x)-(3.5119*x)+(14.143)
if (c_escal > 10):
    p_caudal.append(10)
if (c_escal < 0):
    p_caudal.append(0)
if (c_escal>=0 and c_escal<=10):
    p_caudal.append(round(c_escal,1))

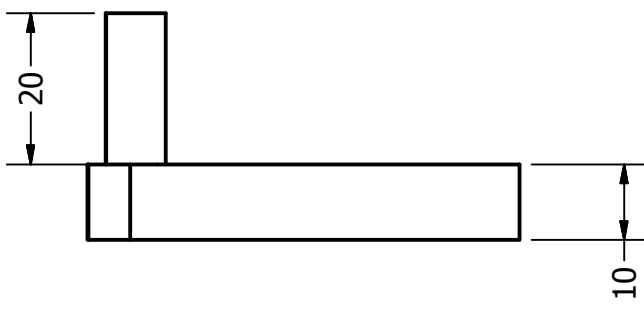
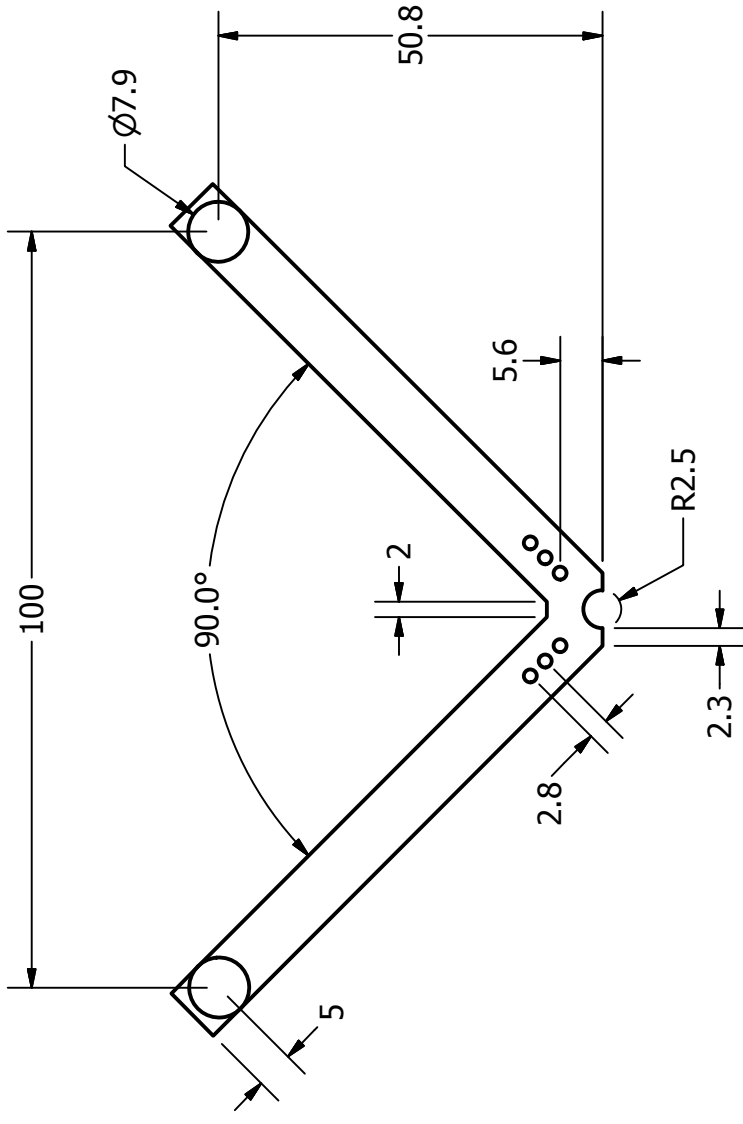
#fase para escala
for i in range(len(latitud)):
    escala.append(((p_presion[i])+(p_caudal[i]))/2)

print 'fecha', fecha
print 'hora ya arreglada', hora
print 'latitud', latitud
print 'N_S', N_S
print 'longitud', longitud
print 'E_W', E_W
print 'distancia', distancia_L
print 'presion ya arreglada', presion
print 'caudal ya arreglada', caudal
print 'ESCALA ', escala
print 'len p_presion', len(p_presion)
print 'len p_caudal', len(p_caudal)
print 'len escala', len(escala)

crearcsv()
escribircsv()
time.sleep(2)
envio=commands.getoutput("rsync -ogpEav --compress-level=9 -e 'sshpass -p gistphapwkap ssh -
o StrictHostKeyChecking=no -p 22' /home/pi/datos/*
root@107.170.7.174:/opt/megaproyecto/geodjango/datos")
print str(envio)

```

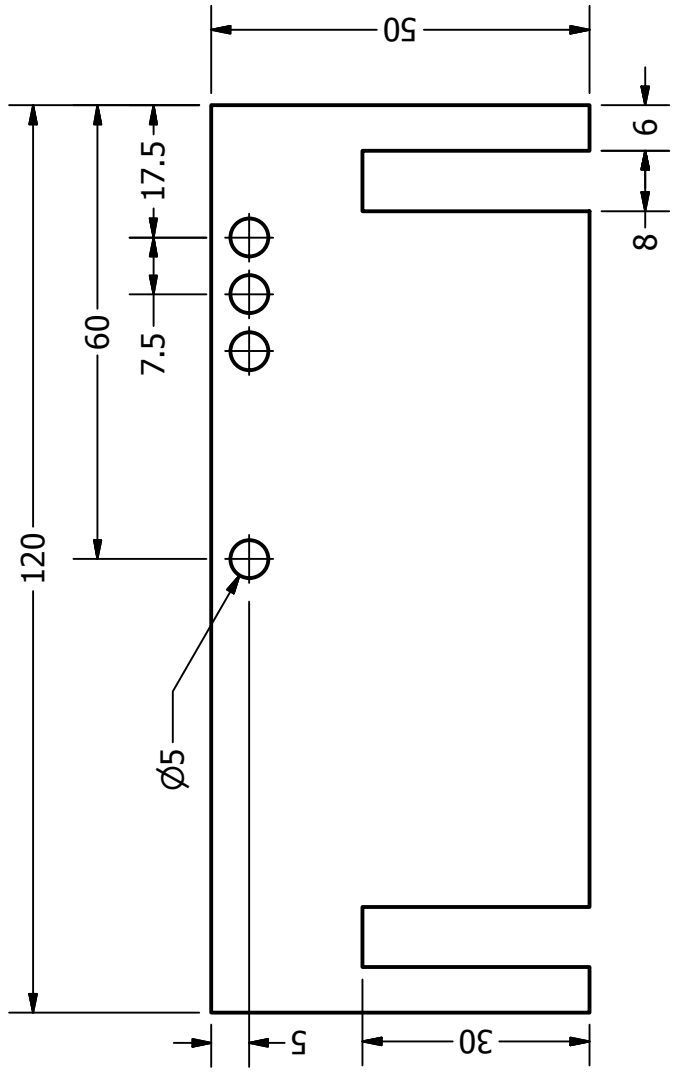
H. PLANO DE PIÑÓN DE ACELERADOR DE MOTOR DIESEL.



Nombre de parte: Acelerador_P1		Página: 1/1	Material Plástico
No. de parte: 001			
Unidades Milímetros	Escala 1:1		

A4

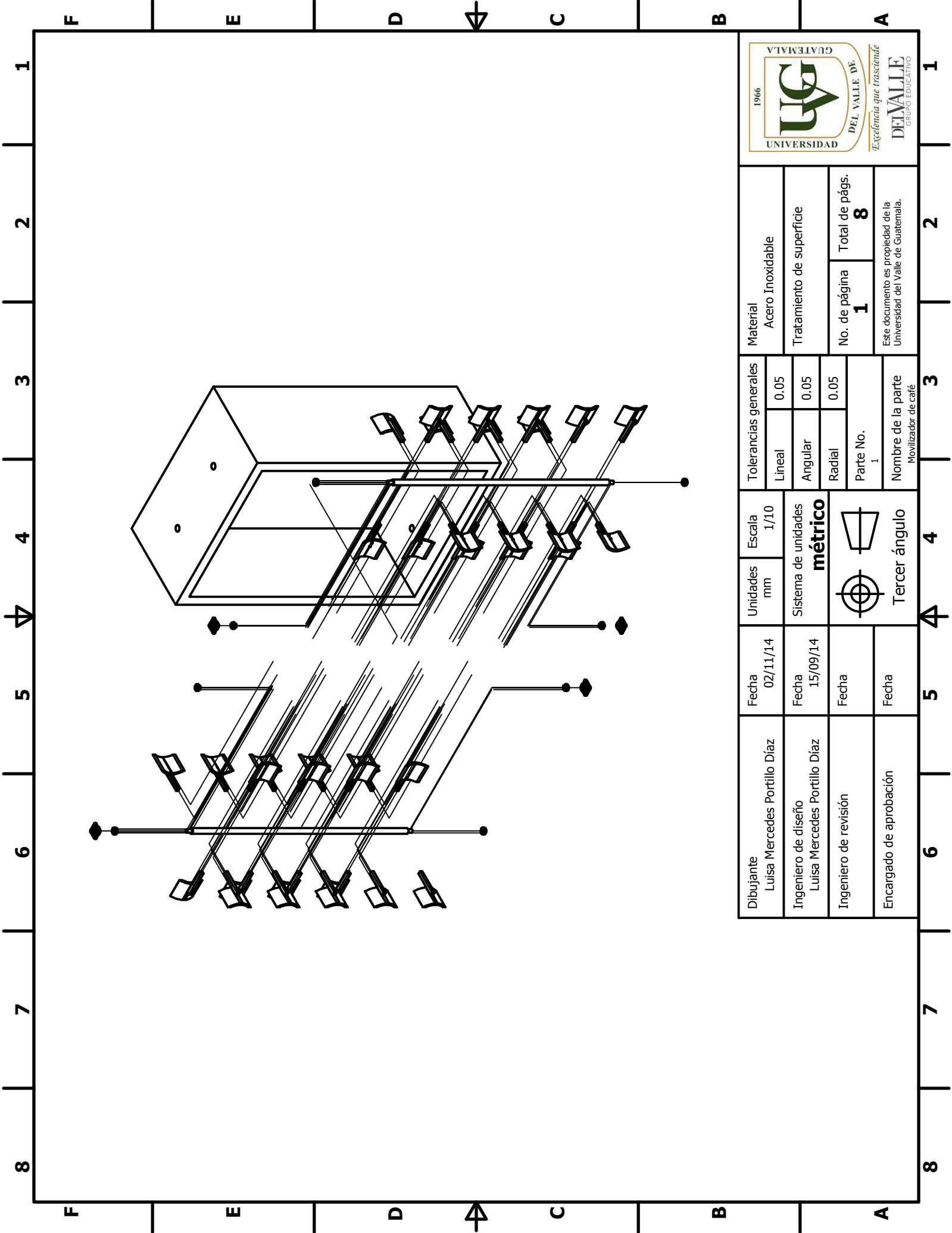
I. PLANO DE CREMALLERA DE ACELERADOR DE MOTOR DIESEL.



Nombre de parte: Acelerador_P2		Página: 1/1	
No. de parte: 002		Escala: 1:1	
Unidades: Milímetros	Material: Plástico		

A4

J. PLANOS DEL MECANISMO PARA HOMOGENIZAR LA
TEMPERATURA DEL CAFÉ



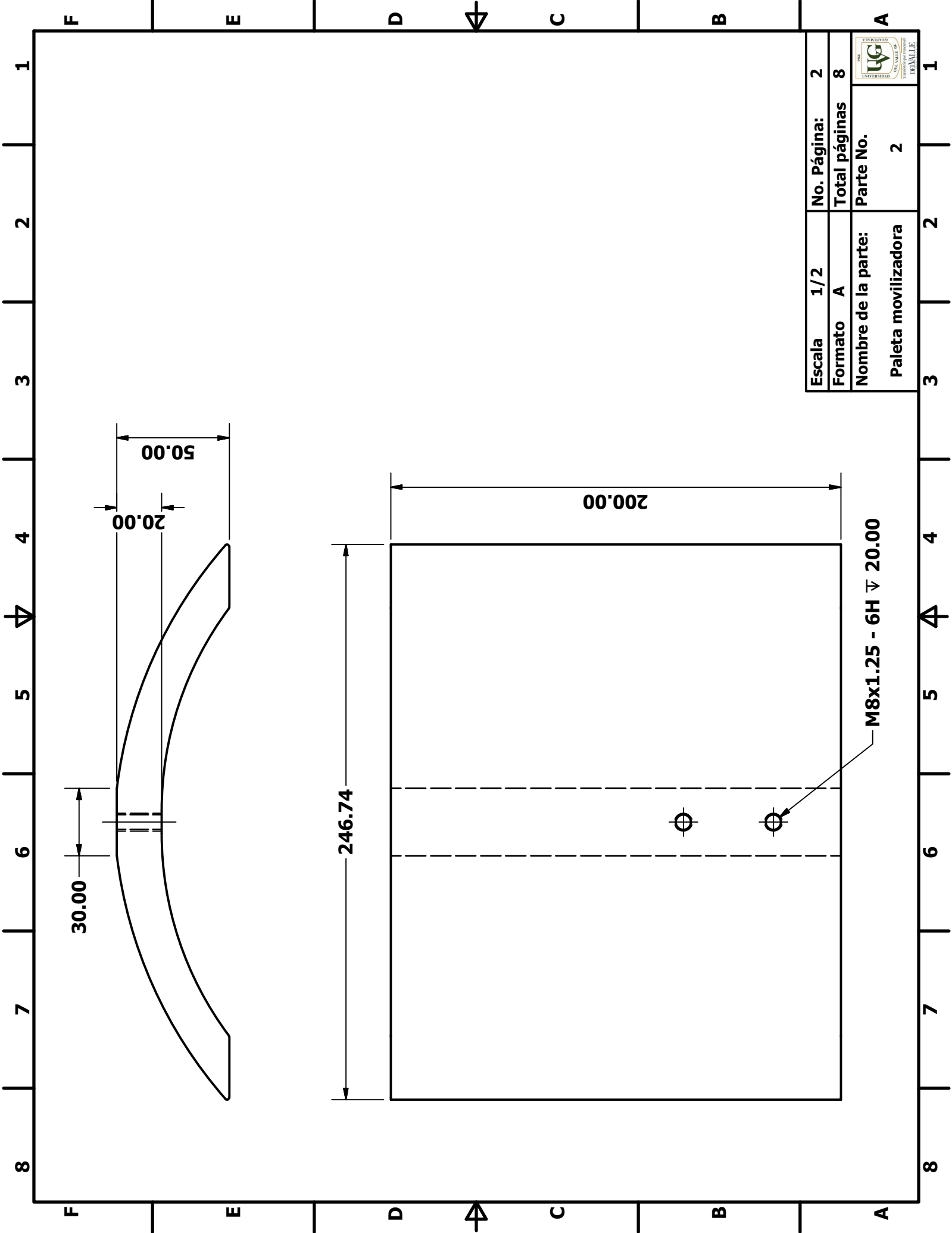
Material Acero Inoxidable	
Tratamiento de superficie	
No. de página 1	Total de págs. 8
Este documento es propiedad de la Universidad del Valle de Guatemala.	

Tolerancias generales	
Lineal	0.05
Angular	0.05
Radial	0.05
Parte No. 1	
Nombre de la parte Movilizador de caté	

Unidades mm	Escala 1/10
Sistema de unidades métrico	
	Tercer ángulo

Fecha 02/11/14	Fecha 15/09/14
Fecha	Fecha

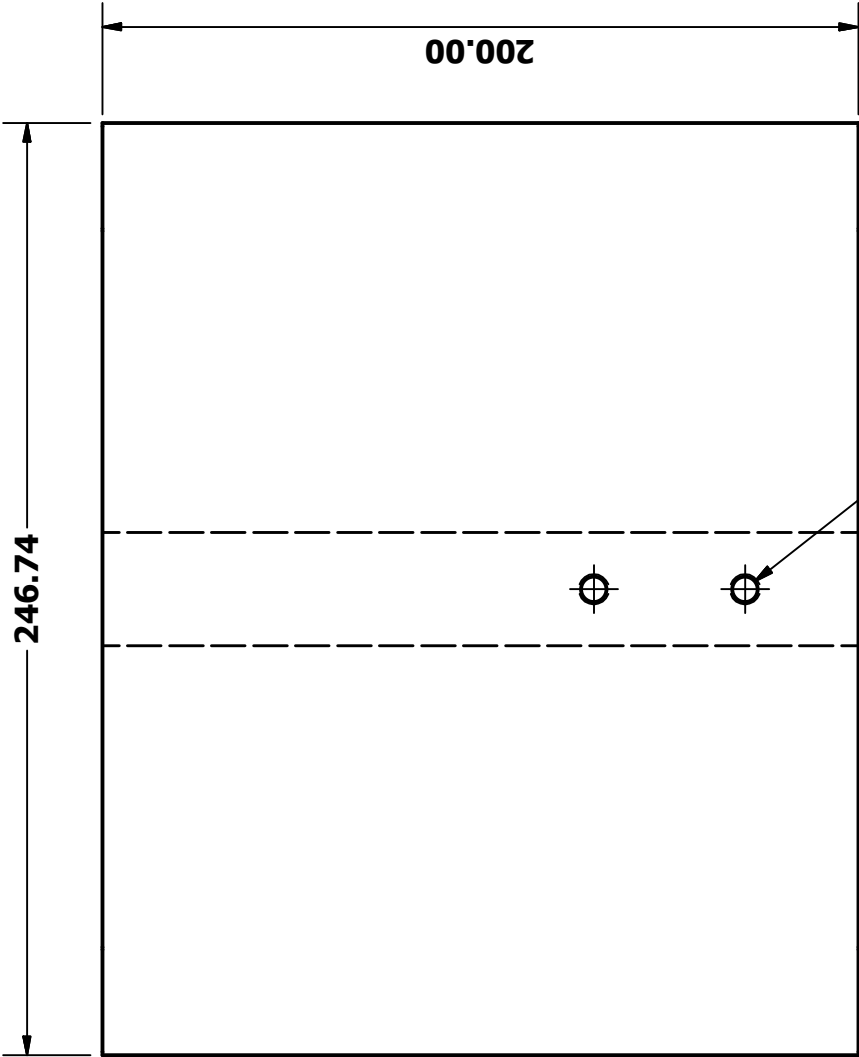
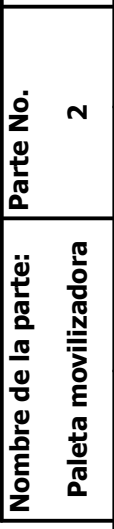
Dibujante Luisa Mercedes Portillo Díaz	Fecha 02/11/14
Ingeniero de diseño Luisa Mercedes Portillo Díaz	Fecha 15/09/14
Ingeniero de revisión	Fecha
Encargado de aprobación	Fecha



Escala	1/2	No. Página:	2
Formato	A	Total páginas	8
Nombre de la parte:	Paleta movilizador		
Paleta movilizador	Parte No.		2



M8x1.25 - 6H ∅ 20.00



F

E

D

C

B

A

1

2

3

4

5

6

7

8

F

E

D

C

B

A

1

2

3

4

5

6

7

8

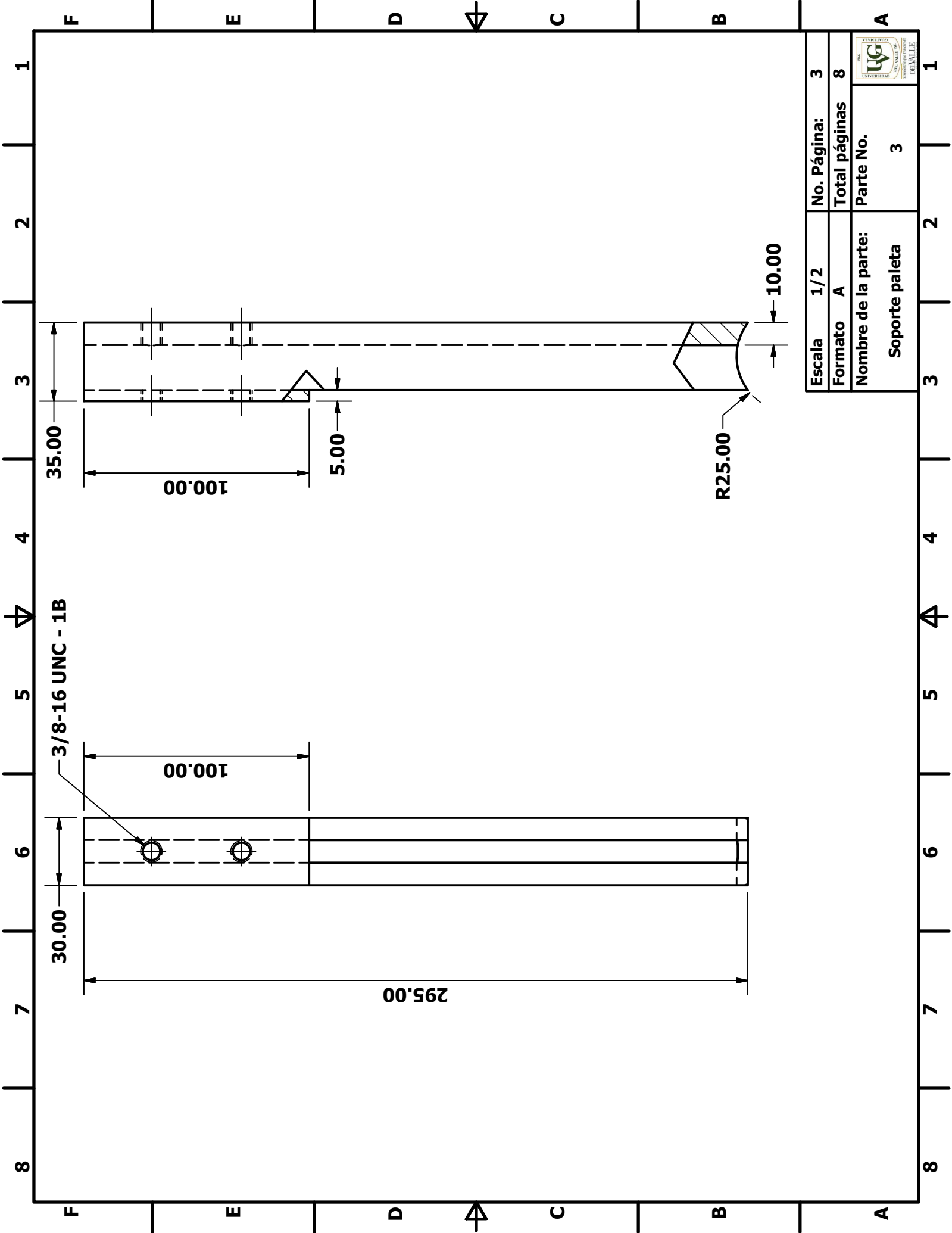
30.00

20.00

50.00

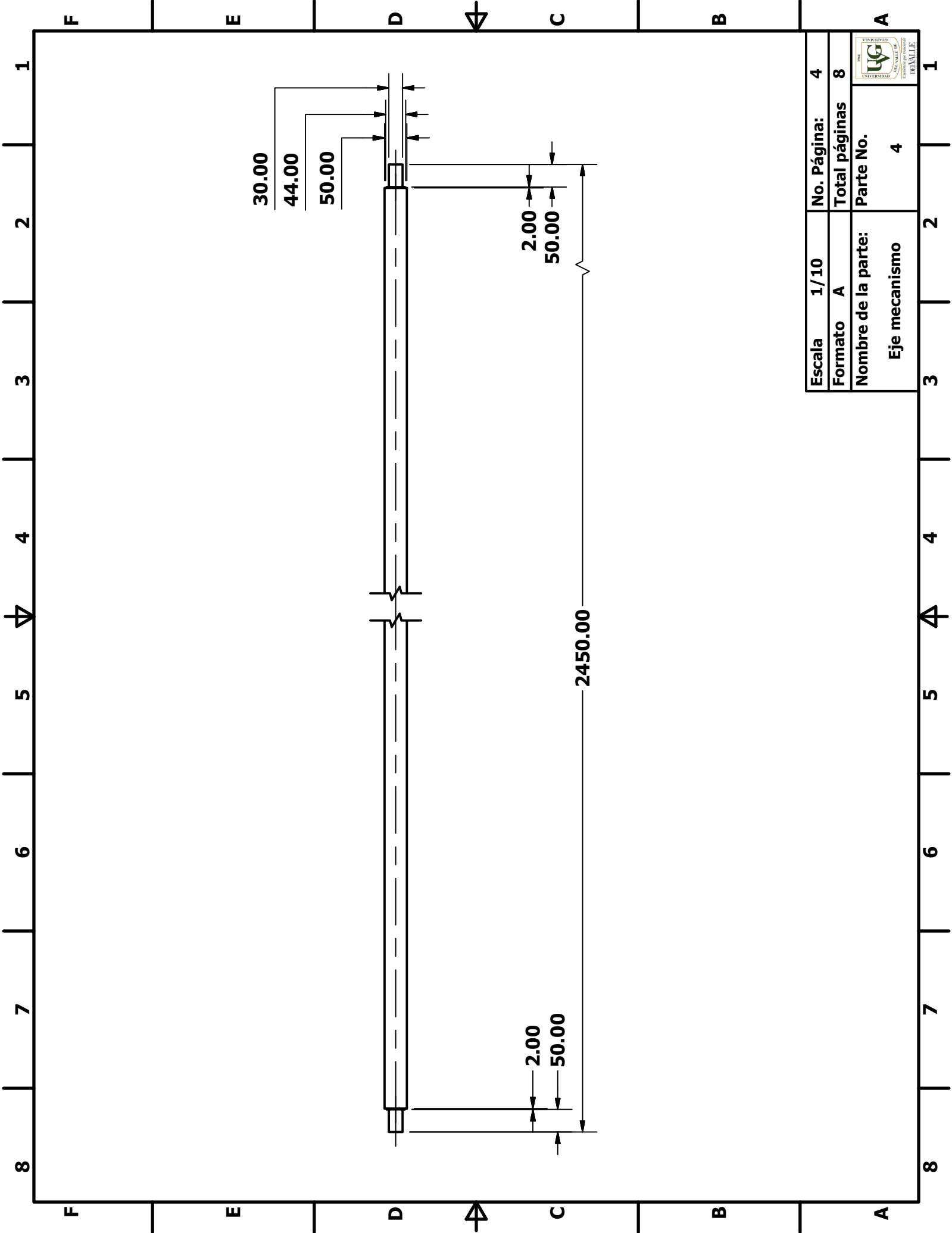
246.74

200.00



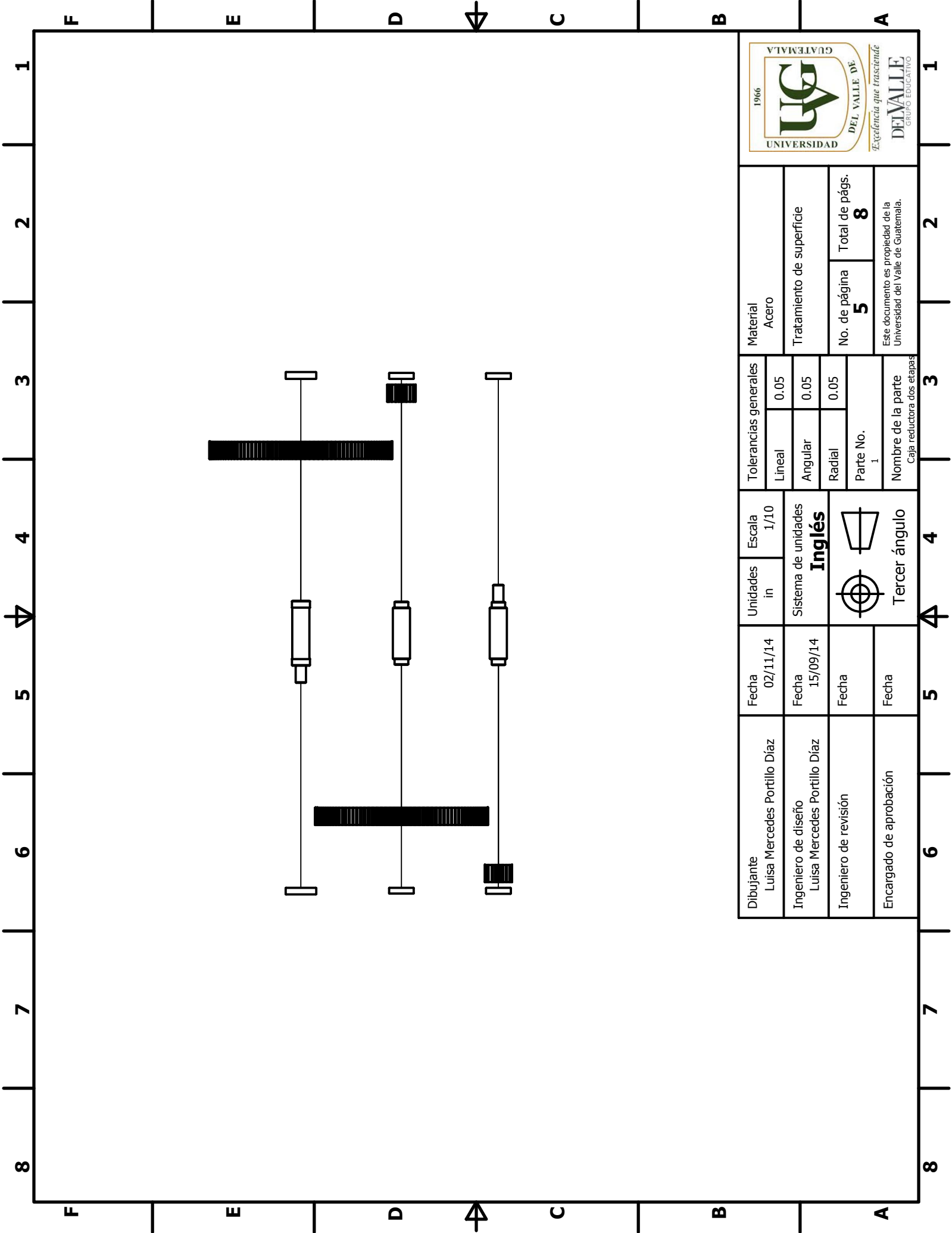
Escala	1/2	No. Página:	3
Formato	A	Total páginas	8
Nombre de la parte:	Soporte paleta		
Parte No.	3		





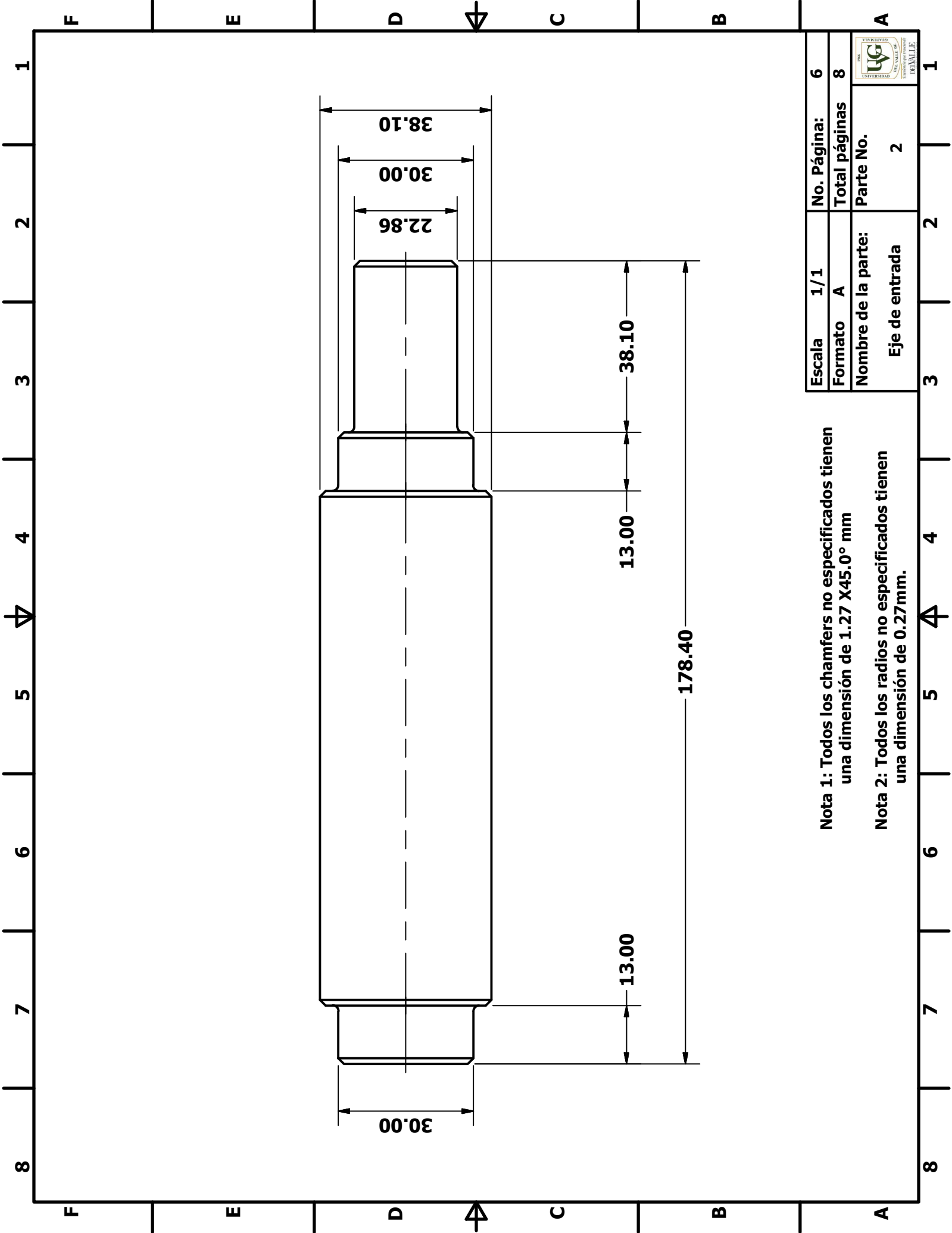


Escala	1/10	No. Página:	4
Formato	A	Total páginas	8
Nombre de la parte:	Eje mecanismo		
Parte No.	4		





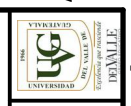
Dibujante Luisa Mercedes Portillo Díaz	Fecha 02/11/14	Unidades in	Escala 1/10	Tolerancias generales		Material Acero
	Ingeniero de diseño Luisa Mercedes Portillo Díaz		Fecha 15/09/14	Sistema de unidades Inglés	Lineal 0.05	
Ingeniero de revisión	Fecha	 Tercer ángulo	 Parte No. 1	Radial 0.05	No. de página 5	Total de págs. 8
Encargado de aprobación	Fecha			Nombre de la parte Caja reductora dos etapas	Este documento es propiedad de la Universidad del Valle de Guatemala.	

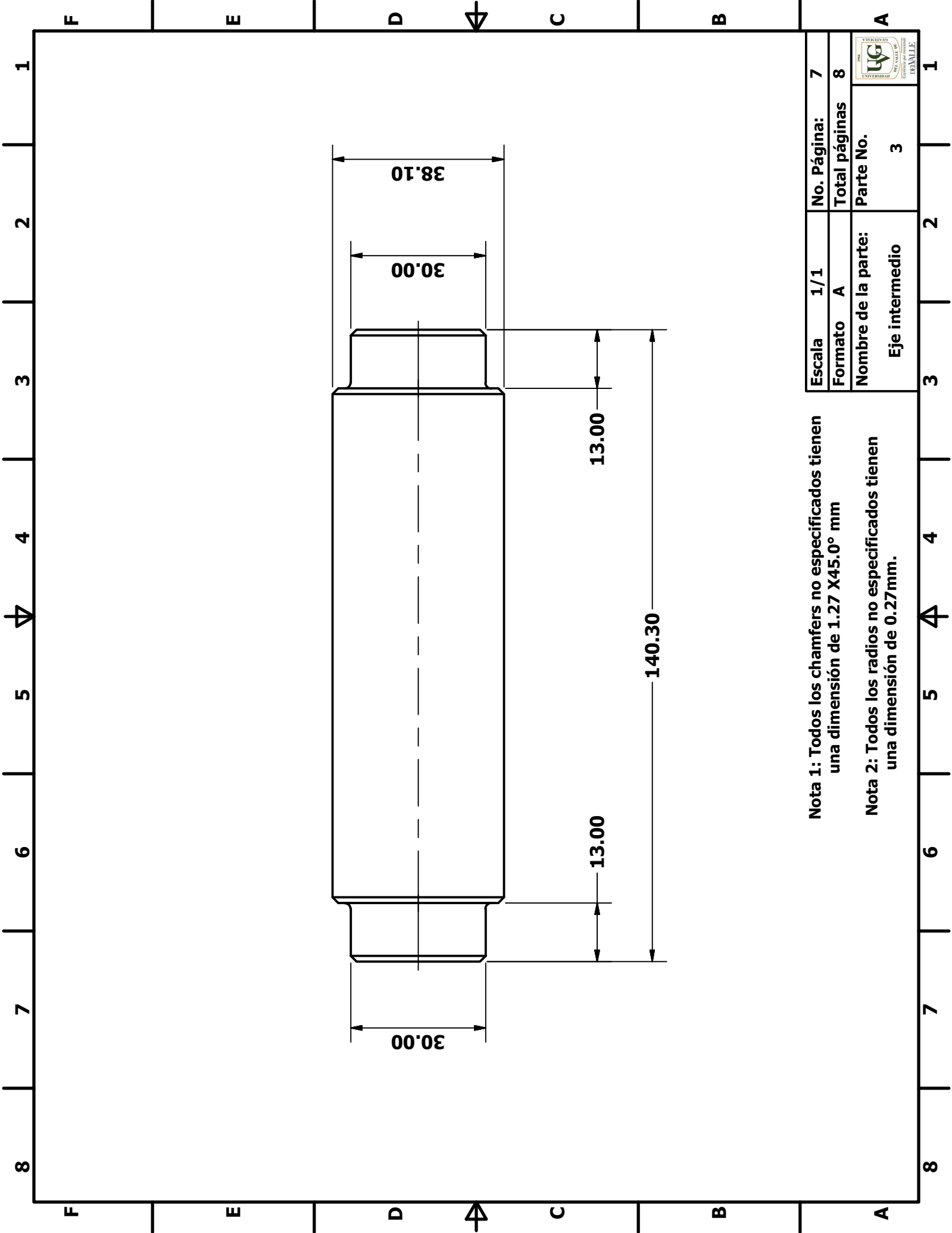


Nota 1: Todos los chamfers no especificados tienen una dimensión de 1.27 X45.0° mm

Nota 2: Todos los radios no especificados tienen una dimensión de 0.27mm.

Escala	1/1	No. Página:	6
Formato	A	Total páginas	8
Nombre de la parte:	Eje de entrada	Parte No.	2



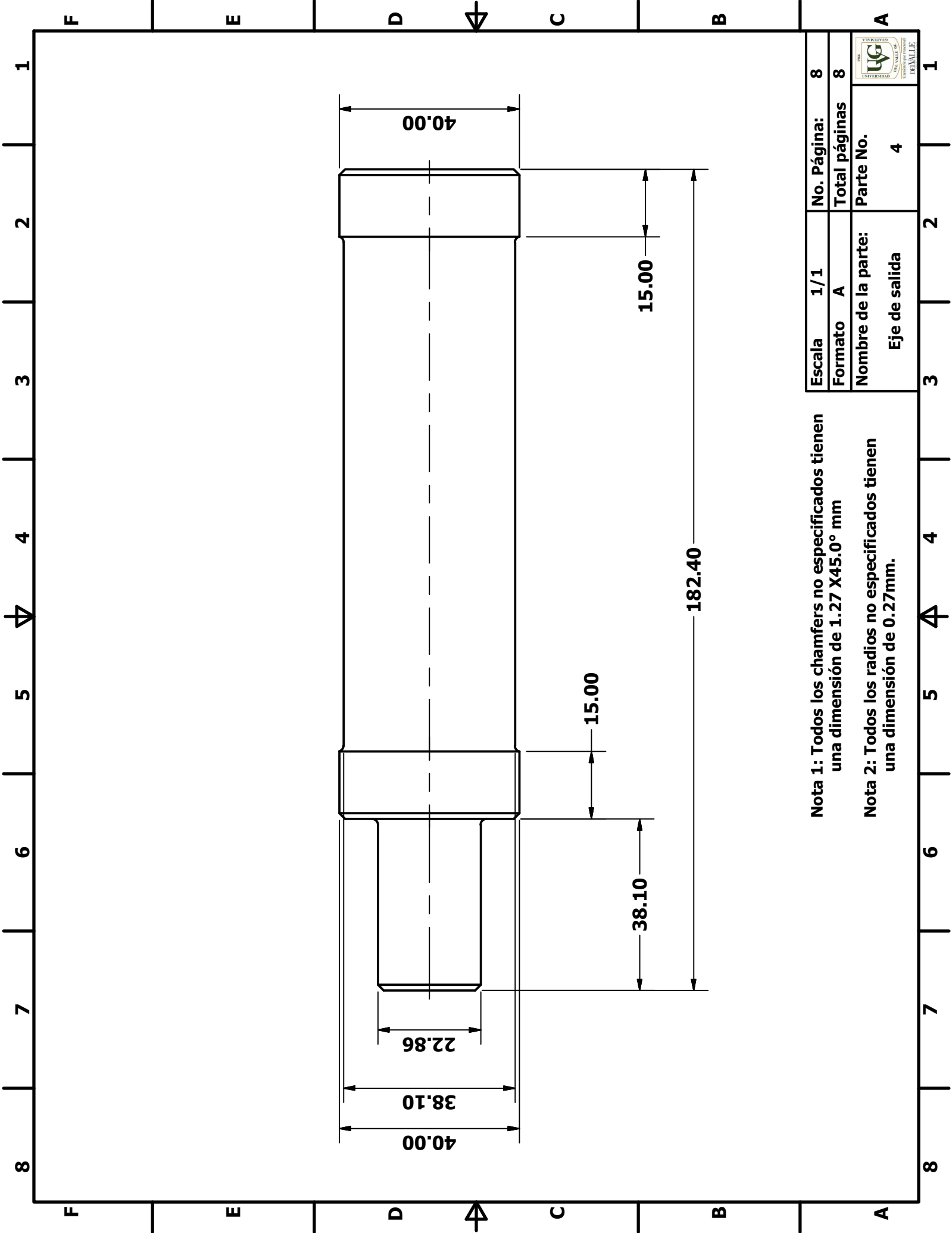


Nota 1: Todos los chamfers no especificados tienen una dimensión de 1.27 X45.0° mm

Nota 2: Todos los radios no especificados tienen una dimensión de 0.27mm.

Escala	1/1	No. Página:	7
Formato	A	Total páginas	8
Nombre de la parte:	A		
Eje intermedio	3		





Nota 1: Todos los chamfers no especificados tienen una dimensión de 1.27 X45.0° mm

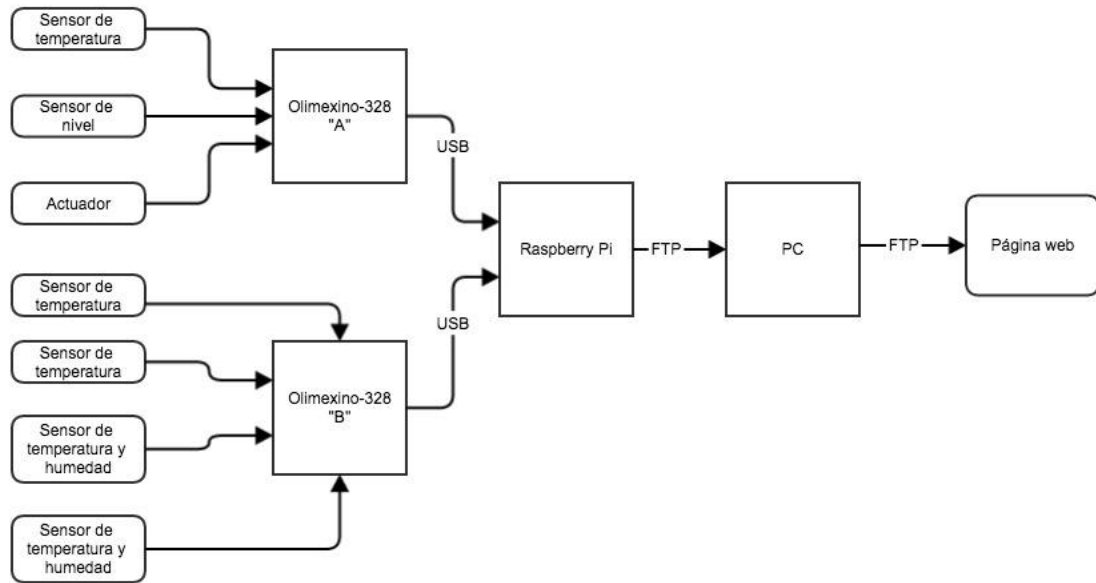
Nota 2: Todos los radios no especificados tienen una dimensión de 0.27mm.

Escala	1/1	No. Página:	8
Formato	A	Total páginas	8
Nombre de la parte:	A		
Eje de salida	Parte No. 4		



K. DIAGRAMA DE FLUJO DE DATOS DEL PROCESO DE SECADO DE CAFÉ.

Figura 195: Diagrama de flujo de datos del proceso de secado de café.



XIV. GLOSARIO

1. Quemadores modulantes: un quemador modulante tiene la capacidad de controlar la el tamaño de la llama de salida, con el fin de mantener la variable requerida, por medio del control de potencia.
2. Controlador PID: un PID utiliza un sistema de control de lazo cerrado (con retroalimentación) el cual tiene la capacidad de corregir el error en la salida comparándolo con una señal de referencia.
3. Pérdidas de carga: las pérdidas de carga en tuberías son las presiones que requiere la sustancia para vencer la fricción del medio por donde se mueven.
4. Sensor LM35: el sensor LM35 es un circuito integrado de precisión para medición de temperatura, éste proporciona una salida analógica de voltaje que representa la temperatura en grados centígrados, de manera lineal.
5. Pirómetro: un pirómetro es un instrumento de medición de temperatura de alta precisión, sin tener un contacto directo con la superficie.
6. DHT22/AM2302: el sensor DHT22, posee una tecnología que permite medir la humedad y temperatura del medio, de manera capacitiva y con un termistor respectivamente.
7. Diagrama P&ID: un diagrama P&ID tiene como fin la identificación de equipo industrial, identificación de procesos, condiciones de diseño, de una manera sencilla en proyectos de tuberías e instrumentación.
8. Variador de frecuencia: un variador de frecuencia, también conocido como variador de velocidad, es un dispositivo electrónico que controla la velocidad síncrona del motor trifásico.
9. Protocolo one-wire: el protocolo one-wire permite una comunicación serial por medio de un solo cable, ésta va definida por un maestro y un esclavo.
10. Olimexino-328: el Olimexino-328 es una tarjeta que puede ser utilizada como un Arduino UNO, con una tarjeta ATmega328 para uso industrial.
11. Tapaporo: Sellante formulado a base de resinas acrílicas, desarrollado especialmente para resanar imperfecciones en maderas, en especial si son muy porosas. Su buen relleno deja una superficie más apta para la aplicación de sistemas de sellado a base de nitrocelulosa o poliuretano, garantizando un mejor desempeño de los mismos. Cuenta con excelentes características como relleno, secado, lijabilidad y transparencia.

12. EEPROM: Por sus siglas en inglés Erasable Programmable Read-Only. Es un tipo de chip de memoria ROM no volátil.
13. Baudios: Medida que representa el número de símbolos por segundo en un medio de transmisión digital.
14. API: Una interfaz de programación representa la capacidad de comunicación entre componentes de software.
15. Archivos estáticos: Permiten cargar contenido al que pueda hacer referencia en una página, lo que incluye archivos (como archivos .zip y .jar), imágenes, hojas de estilo, programas JavaScript y otros archivos.
16. Dirección IP: Es un número único e irrepetible con el cual se identifica una computadora conectada a una red que corre el protocolo IP. Una dirección IP (o simplemente IP como a veces se les refiere) es un conjunto de cuatro números del 0 al 255 separados por puntos.
17. DNS: Es un sistema de nombres que permite traducir de nombre de dominio a dirección IP y viceversa.
18. Query: Se usa para obtener datos, modificarlos o bien borrarlos.
19. Script: Programas, usualmente pequeños o simples, para realizar generalmente tareas muy específicas.
20. Sistema operativo: Se encarga de crear el vínculo entre los recursos materiales, el usuario y las aplicaciones.
21. SMTP: Protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.
22. SRID: Es un identificador único asociado con un sistema de coordenadas específico, dato y la unidad de medición.