**INFORME PROYECTO**

**RIEGO AUTOMATIZADO**

**Elaborado por:**

**Jeferson David Llantén Córdoba**

**Código: 104616021116**

**Fabian David Marin**

**Código:**

**Hector Fernando Dorado Sapuyes**

**Código:** **104616020854**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Ingeniería de Sistemas**

**Arquitectura Computacional**

**Popayán, Cauca, Colombia**

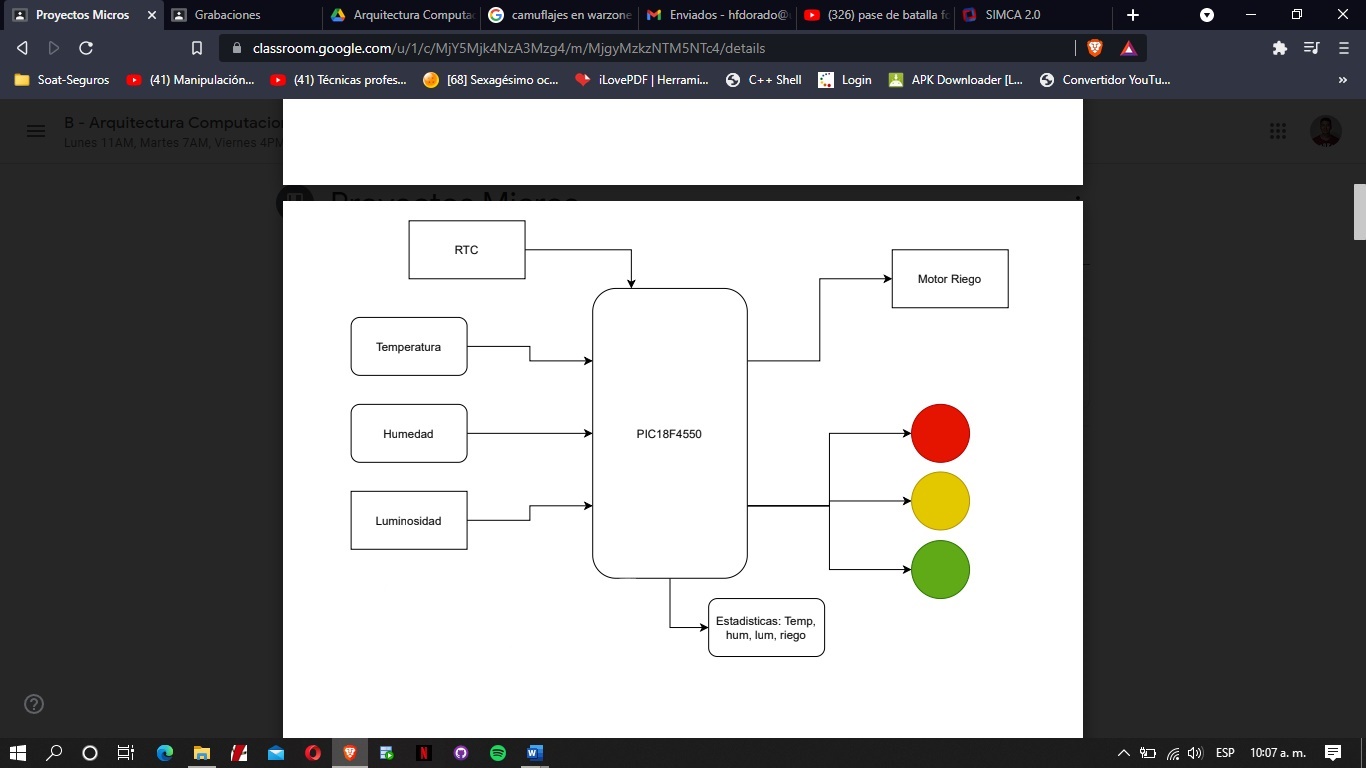
**08-10-2021**

**RIEGO AUTOMATIZADO**

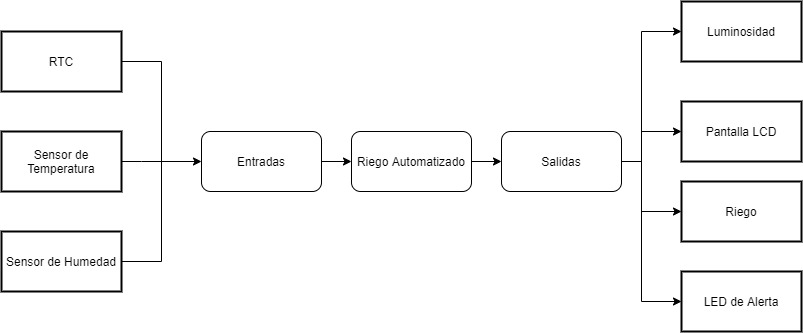
### Planteamiento del problema

La función principal de un invernadero es la de crear y mantener un espacio específico, con condiciones adecuadas y controladas de luz, humedad, temperatura, bióxido de carbono, y pesticidas entre otras, para cultivar plantas destinadas a diversos propósitos. Sin embargo, la fumigación de pesticidas es potencialmente dañinas para los seres humanos que laboran en invernaderos, por lo que se busca dar una solución. Una alternativa para solucionar este problema ha sido la automatización de los invernaderos con el fin de mejorar cada proceso, aplicando de manera adecuada condiciones fisicoquímicas óptimas para el adecuado cuidado y desarrollo que requiere cada producción.

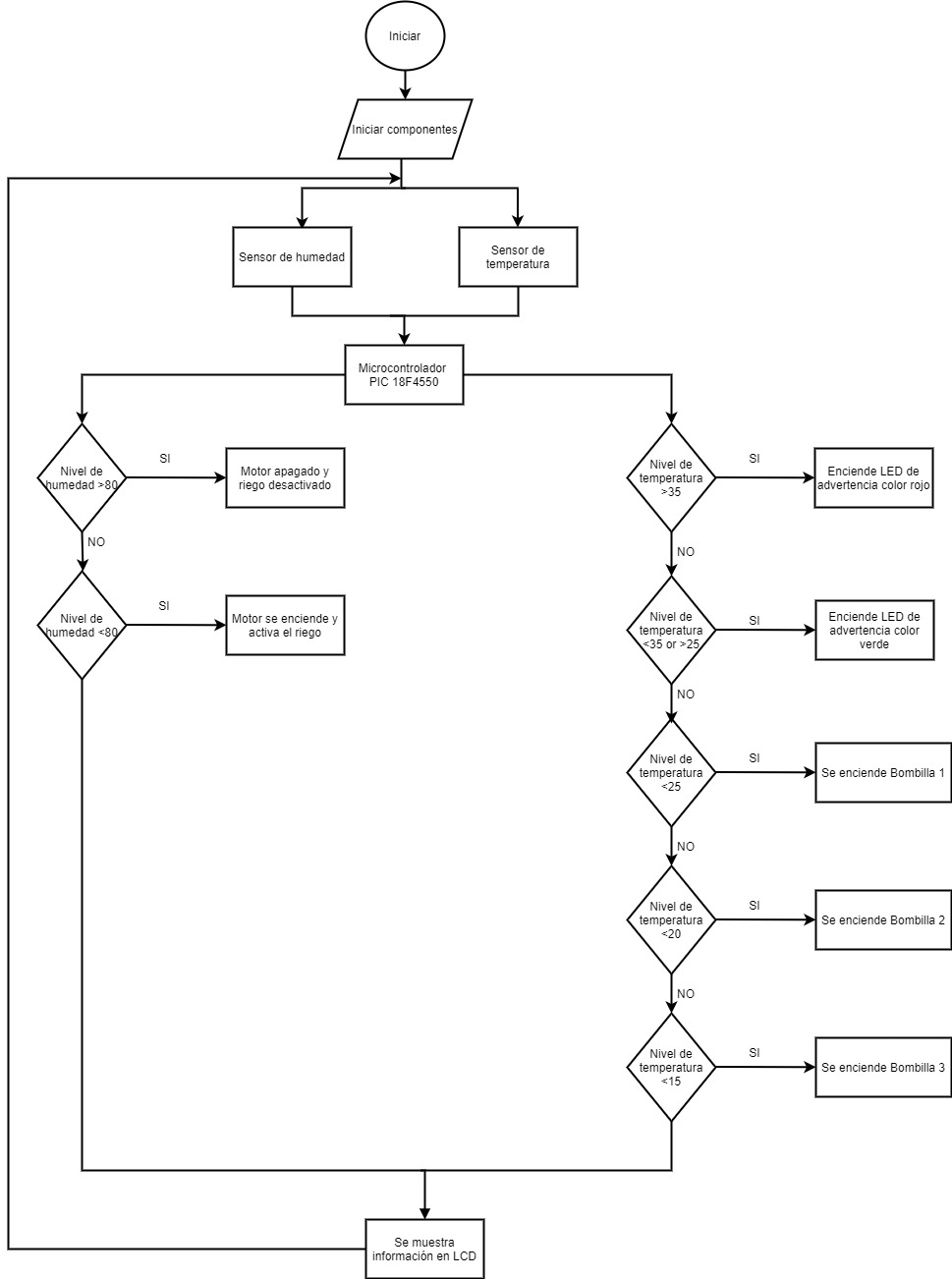
**Diagrama General del proyecto**



### Descripción general software



### El siguiente diagrama presenta el flujo del software de manera general



### Herramientas de desarrollo

Para el desarrollo del proyecto las herramientas que se están usando son:

|  |  |
| --- | --- |
| Proteus. Simulación de circuitos y programación en Arduino. |  |
| Repositorio en github. Manejo de versiones de código fuente. |  |
| MPLAB X IDE. | MPLAB® X IDE | Microchip Technology |

### Componentes software

● **LDC:** se implementó un LCD 16x2 para mostrar los valores de temperatura y humedad, adicionalmente, se muestra la fecha y hora actual.

● **DHT11:** Este sensor permite aumentar o disminuir los valores de temperatura y humedad.

● **RTC y I2C:** Se utilizan para poder saber la hora y fecha actual.

● **Tres leds**: Empleados para advertir al usuario sobre el estado de la temperatura, alta (rojo), ideal (verde) y baja (amarillo).

**● Tres lámparas:** cada una con 6 grados de temperatura.

#### Interfaz gráfica de usuario (GUI)

|  |  |  |  |
| --- | --- | --- | --- |
| **LIBRERÍA** | **FUNCIONES** | **DESCRIPCIÓN** | **ESTADO** |
| RTC | RTC\_Read\_Clock() | Método RTC para leer el tiempo. | Implementado  Probado  Habilitado |
| RTC\_Read\_Calendar() | Método RTC para leer el calendario. |
| RTC(); | Método para mostrar tiempo y fecha. |
|  |  |  |
| I2C\_Master\_File | I2C\_Ready(); | Verifica y espera si el I2C está listo o no | Implementado  Probado  Habilitado |
| I2C\_Init(); | Inicializa la configuración del registro I2C |
| I2C\_Start(char); | Envía pulso de inicio  junto con la dirección  del dispositivo esclavo |
| I2C\_Start\_Wait(char); | Envía el pulso de inicio junto con la dirección del dispositivo esclavo y espere si no funciona |
| I2C\_Repeated\_Start(char); | Envía pulsos de inicio repetidos junto con la dirección de escritura del dispositivo |
| I2C\_Stop(); | Envía pulso de parada |
| I2C\_Write(unsigned char); | Transmite / escribe  datos en dirección al dispositivo esclavo |
| I2C\_Ack(); | Envía Ack de recibo del maestro para otra lectura |
| I2C\_Nack(); | Envía reconocimiento negativo del maestro para detener la lectura |
| I2C\_Read(char flag); | Lee datos de dispositivos  esclavos |
|  | | |

#### Sensores

|  |  |  |  |
| --- | --- | --- | --- |
| **LIBRERÍA** | **FUNCIONES** | **DESCRIPCIÓN** | **ESTADO** |
| DHT11 | DHT11\_Start(); | Enviar pulso de inicio al módulo DHT11 | Implementado  Probado  Habilitado |
| DHT11\_CheckResponse(); | Esperar la respuesta del módulo DHT11 |
|  | DHT11\_ReadData(); | Leer valor de humedad y temperatura |
| imprimirTemHum(); | Imprime la temperatura leída en el LCD |
|  |  |  |

#### Actuadores

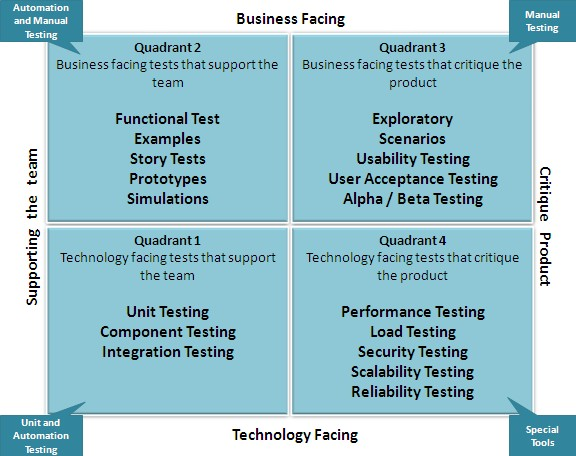
|  |  |  |  |
| --- | --- | --- | --- |
| **LIBRERÍA** | **FUNCIONES** | **DESCRIPCIÓN** | **ESTADO** |
| LCD\_16x2\_8-bit\_Header\_File | LCD\_Init() | Inicializar | Implementado  Probado  Deshabilitado |
| LCD\_Clear() | Limpiar LCD |
| LCD\_Command(char cmd ) |  |
| LCD\_Char(char dat) |  |
| LCD\_String(const char \*msg) | Escribe en el LCD |
| LCD\_String\_xy(char row,char pos,const char \*msg) | Coordenadas del LDC |
| MSdelay(unsigned int val) | Proporciona un retardo de 1ms para una frecuencia de 8 MHz |
| imprimirTemHum() | Imprime en el LCD los datos de la Temperatura y la Humedad |

#### Alarmas

En esta sección se detallan cada una de las alarmas que se definen para el buen funcionamiento del sistema.

### Proceso de pruebas

La sección de pruebas del proyecto sigue el cuadrante que se muestra a continuación. En el cuadrante 1 tenemos: las pruebas unitarias que se realizan al momento de implementar un nuevo módulo software. Las pruebas de componentes nos ayudan a dar más funcionalidad y después las pruebas de integración de componentes reúne varios componentes para dar una funcionalidad específica.



En el cuadrante 2 ya pertenece al dominio del negocio, en este caso el funcionamiento del ventilador usando un prototipo.

En el cuadrante 3. Nos muestra el prototipo evaluado por un experto en el área. a partir de sus observaciones se configuran nuevos escenarios y mejoras en el proyecto a nivel de usabilidad, funcionalidad y desempeño.

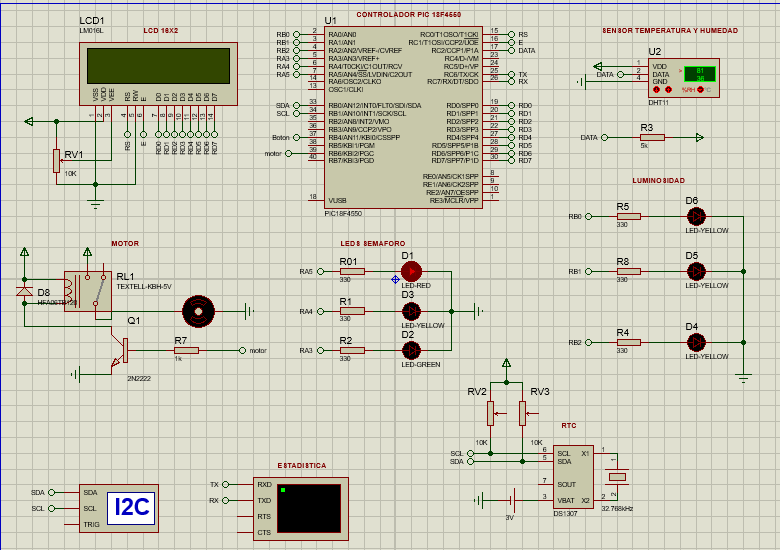
En el cuadrante 4. las pruebas de desempeño, carga y confiabilidad nos ayuda a orientar el proyecto a ser robusto ante cualquier falla y uso prolongado por horas y días.

Todas las pruebas están documentadas en la siguiente tabla llamada Test BackLog, el cual busca describir la prueba, conocer si pasó o no la prueba y dejar algunos comentarios u observaciones.

**Test Backlog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | PRUEBA | DESCRIPCIÓN | PASÓ PRUEBA (SI/NO) | OBSERVACIONES |
| 1 | Comprobar RTC | Visualizar que la hora y fecha sean coherentes | SI | Si muestra la hora y fecha actualizada |
| 2 | Encender leds | De acuerdo al valor de la temperatura, existen 3 leds, rojo, amarillo, verde, se encienden si: la temperatura está muy alta, si la temperatura es ideal y si la temperatura es muy baja, respectivamente. | Si |  |
| 3 | Medir la temperatura con el sensor DHT11 | Variar la temperatura manualmente | Si |  |
| 4 | Encender lámparas | De acuerdo el nivel de temperatura, las lámparas se deben encender, cada lampara tiene 6 grados de temperatura | Si | En la consola se imprime cuál de las cinco lampara está encendida. |
| 5 | Mostrar información en el LCD | Visualizar en el LCD los parámetros humedad y temperatura | Si |  |
| 6 | Mostrar información en la consola | Visualizar información en tiempo real | SI |  |
| 7 | Mostrar información en el LCD | Visualizar en el LCD la fecha y hora utilizando el RTC | Si |  |
| 8 | Mostrar cantidad de lámparas prendidas en el LCD | Visualizar la cantidad de lámparas a medida que se encienden | No |  |

### Diagrama final en proteus



### Link github

* <https://github.com/fernandorado?tab=repositories>
* <https://github.com/jdllanten?tab=repositories>
* <https://github.com/fdavidmarin?tab=repositories>