



Universidad Nacional de Río Negro

Sede Andina

Software Design Descriptions
Laboratorio de Sistemas Embebidos

Título: SDD InterHome

Año: 2024

Docente: Gustavo Ortiz Uriburu

Alumnos
Fioroni Tomás
Ceruse Candelaria
Piñero Iván
Renolfi Ezequiel

Índice

1	Introducción	3
1.1	Propósito	3
1.2	Alcance	3
1.3	Contexto	3
1.4	Resumen	3
2	Preocupaciones de Diseño	4
3	Puntos de vista del Diseño	4
4	Razonamiento del Diseño	5
5	Glosario	7
6	Referencias	7

1. Introducción

1.1. Propósito

Este documento especifica las decisiones de diseño tomadas durante el desarrollo del Software. Estas responden a los requisitos que fueron planteados en el documento de *Software Requirements Specifications*.

1.2. Alcance

Este documento expone la implementación del diseño propuesto especificado en el *SRS*. Más precisamente, abarca las decisiones de diseño tomadas para que;

1. El usuario pueda encender y apagar módulos de luz a través de la página web y/o un botón físico asociado al respectivo módulo.
2. El usuario pueda acceder a mediciones de temperatura y humedad tomadas por el respectivo módulo instalado.

1.3. Contexto

El diseño modular permitió desarrollar cada pieza de manera aislada, luego se desarrollaron los medios y protocolos de comunicación entre módulos.

1. Módulo sensores/luces: Se utilizó *Arduino IDE(C++) v.2.3.2* para programar el comportamiento de cada tipo particular de módulo.
2. Módulo Pagina Web: Se utilizó *CSS, JavaScript y HTML* para el desarrollo de la Página Web.
3. Módulo Central: Computadora central, donde se instalaron las librerías utilizadas, que funciona como centro de comunicaciones entre los distintos módulos y utiliza un modelo de publicador-suscriptor por *MQTT*.
4. Módulo Comunicación: El módulo sensores/luces se comunica con la computadora central a partir de un protocolo *MQTT*, mientras que la computadora central se comunica con la página web a través de un formato *JSON*.

1.4. Resumen

Este documento explica el diseño desde los distintos puntos de vista del proyecto, abarcando el punto de vista del usuario así como el de futuros desarrolladores.

2. Preocupaciones de Diseño

1. Este proyecto está destinado a personas interesadas en incluir en su hogar las comodidades de la domótica e *IoT*.
2. De igual manera, está destinado a personas que estén interesadas en replicar o continuar este proyecto.

3. Puntos de vista del Diseño

1. Punto de vista del cliente: El cliente percibe las siguientes representaciones del diseño.
 - a) Interfaz virtual
 - 1) Interruptores para cada módulo de luz.
 - 2) Etiqueta de nombre personalizable para cada módulo de luz.
 - 3) Gráfico de porcentaje de humedad, con aclaración numérica.
 - 4) Indicador numérico de temperatura, en grados Celsius.
 - 5) Etiqueta de nombre personalizable para cada módulo de temperatura y humedad.
 - b) Interfaz física
 - 1) Interruptor físico para cada módulo de luz.
2. Punto de vista del desarrollador: El desarrollador percibe las siguientes representaciones del diseño.
 - a) Modulo sensores/luces
 - 1) En este módulo se utilizaron *ESP8266 NodeMCU*.
 - 2) En este módulo se utilizaron *DHT11*.
 - 3) En este módulo se utilizaron módulos rele de un canal de 5V.
 - 4) En este módulo se utilizaron botones pulsadores.
 - 5) Para el desarrollo de este módulo se implementaron dos códigos específicos, uno para el control de luces y otro para el control del sensor *DHT11*.

b) Módulo página web

- 1) *HTML* se utilizó para generar, estructurar y definir la página web.
- 2) *CSS* se utilizó para darle estilo al cuerpo de la página web.
- 3) *JavaScript* se utilizó para cubrir la interacción entre *Flask* y el usuario.

c) Módulo central

- 1) Este módulo se basó en una *Raspberry Pi 3 modelo B+*, sobre la cual se instaló un *Raspberry Pi OS*.
- 2) Al sistema operativo se le instaló *MQTT Mosquitto*, *Paho-MQTT*, *Python*, *Flask v.3.0.23* y *RaspAP*.

d) Módulo comunicaciones

- 1) Se utilizó un protocolo publicador suscriptor basado en *MQTT* para la comunicación entre los módulos y la computadora central.
- 2) Se utilizó el formato de datos *JSON* para el pasaje de mensajes entre módulos.

4. Razonamiento del Diseño

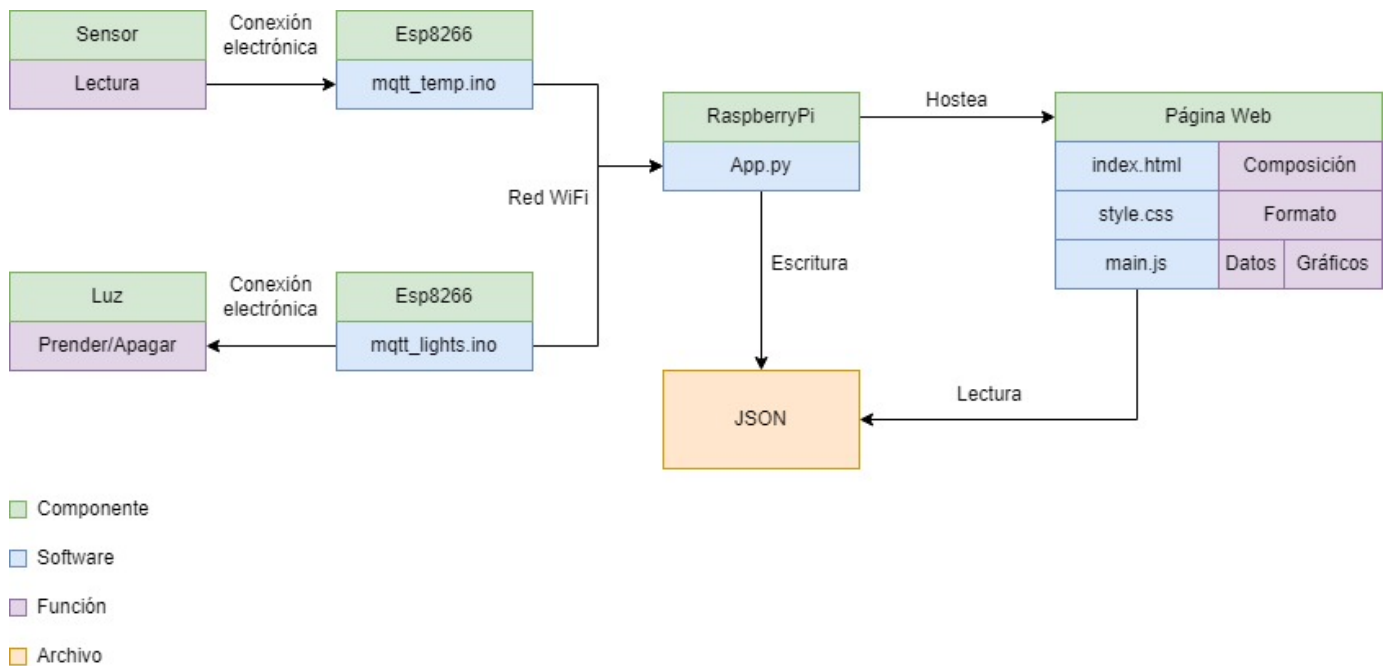


Figura 1: Esquema general

1. Interfaz virtual: Se eligieron los interruptores y gráficos, así como su distribución, de manera que el diseño sea simple y accesible para el usuario, sin importar el dispositivo ni su resolución específica.
2. Interfaz física: Se optó por un botón pulsador para el control manual de los módulos de luces para que al instalar el producto no se pierda la posibilidad de controlar físicamente la luz.
3. Modulo sensores/luces
 - a) Se optó por *ESP8266 NodeMCU* ya que permitió programar facilmente las terminales del proyecto. Este también tiene disponibilidad WiFi, lo que fue necesario para la comunicación inalámbrica con la computadora central.
 - b) Se eligió el sensor de temperatura y humedad *DHT11* debido a su simplicidad de funcionamiento, disponibilidad de mercado y precio asequible.
 - c) Se optó por módulos rele de 1 canal, debido a su simplicidad de funcionamiento, disponibilidad de mercado y precio asequible.
4. Módulo página web
 - a) *HTML*, *CSS* y *JavaScript* se eligieron para la creación de la página web porque son lenguajes ampliamente utilizados y que cuentan con mucho soporte e información para su uso y desarrollo. Se utilizó *Flask* para facilitar la interacción entre la página web y el software del módulo central.
5. Módulo central
 - a) Se utilizó la *Raspberry Pi 3 modelo B+* por la disponibilidad y su posible funcionamiento como punto de acceso.
 - b) Se utilizó *RaspAP* para configurar la *Raspberry Pi 3 modelo B+* como punto de acceso.
6. Módulo comunicaciones
 - a) Se utilizó *Eclipse Mosquitto* para *hostear* el servidor *MQTT* ya que provee lo necesario y es de código abierto.
 - b) Se utilizó *Paho-MQTT* para configurar el servidor *MQTT* desde la comodidad de *Python* para recibir y publicar los mensajes, así como generar ordenes a partir de estos.
 - c) Se utilizó el formato de datos JSON para el pasaje de mensajes por su simplicidad y eficiencia.

5. Glosario

- *IoT*: Internet de las Cosas.
- *MQTT*: *Message Queuing Telemetry Transport* es un protocolo de mensajería ligero.
- *JSON*: *JavaScript Object Notation*, formato de texto sencillo para el intercambio de datos.

6. Referencias

I. C. S. (2009). IEEE Std 1016-1998. Standard for Information Technology—Systems Design—Software Design Descriptions.