



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Comunicador para centrales de alarma

Autor:

Mariano Mondani

Director:

Nombre del Director (pertenencia)

Codirector:

John Doe (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 30 de abril de 2021 y el 18 de junio de 2021.*

Índice

Registros de cambios	3
Acta de constitución del proyecto.	4
Descripción técnica-conceptual del proyecto a realizar	5
Identificación y análisis de los interesados.	7
1. Propósito del proyecto.	7
2. Alcance del proyecto	8
3. Supuestos del proyecto.	8
4. Requerimientos	8
5. Historias de usuarios (<i>Product backlog</i>).	10
6. Entregables principales del proyecto	12
7. Desglose del trabajo en tareas	12
8. Diagrama de Activity On Node	13
9. Diagrama de Gantt.	15
10. Presupuesto detallado del proyecto	17
11. Matriz de asignación de responsabilidades	17
12. Gestión de riesgos	17
13. Gestión de la calidad	19
14. Procesos de cierre.	26

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento Se completa hasta el punto 8 inclusive, sin incluir el punto 5	30/04/2021
1	Se corrigen errores en la Descripción, Identificación de los interesados y Alcance. Se completa hasta el punto 12 inclusive.	09/05/2021
2	Se corrigen errores en Entregables del proyecto y Gestión de riesgos. Se completan los puntos 5, 13 y 14	16/05/2021

Acta de constitución del proyecto

Buenos Aires, 30 de abril de 2021

Por medio de la presente se acuerda con el Ing. Mariano Mondani que su Trabajo Final de la Maestría en Internet de las cosas se titulará “Comunicador para centrales de alarma”, consistirá esencialmente en la implementación de un sistema que permitirá comunicar una alarma domiciliaria con una aplicación móvil, y tendrá un presupuesto preliminar estimado de 600 hs de trabajo, con fecha de inicio 30 de abril de 2021 y fecha de presentación pública 25 de abril de 2022.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg
Director posgrado FIUBA

Claudio Bongiorno
X-28 Alarmas

Nombre del Director
Director del Trabajo Final

Descripción técnica-conceptual del proyecto a realizar

Desde hace más de una década es común encontrar en un sistema de alarma hogareño algún tipo de comunicador. Estos, generalmente, permiten recibir avisos cuando se produce un evento en la alarma como enviarle comandos para que se realice alguna acción en el sistema.

Con el paso de los años la forma de conexión entre el usuario y su alarma ha cambiado. Comenzó siendo a través de la línea telefónica, para luego, pasar a llevarse a cabo mediante la red celular.

Durante muchos años los usuarios percibieron como ágil y novedosa la comunicación a través de mensajes de texto (SMS). El comunicador le enviaba un mensaje cuando se producía algún evento en la alarma y el usuario podía realizar una acción enviando comandos sencillos. Sin embargo, la comunicación mediante SMS fue quedando obsoleta en la vida cotidiana y por lo tanto comenzó a ser considerado un medio poco confiable de interacción con un sistema de seguridad.

Impulsada por esta situación, la empresa comenzó a orientar sus esfuerzos a desarrollar nuevos productos que les permitan a los usuarios relacionarse con sus equipos de una forma más sencilla, consistente y robusta.

En este contexto surge el presente proyecto: un equipo que se conecta al sistema de alarma como cualquier otro dispositivo y que es acompañado mediante una aplicación que permite conocer el estado de la alarma, recibir eventos y enviar comandos.

A pesar de que actualmente la competencia de X-28 Alarmas ofrece comunicadores compatibles con las centrales de alarma de la marca, el principal diferencial que va a ofrecer este nuevo producto es la posibilidad de acceder a una gran variedad de funcionalidades del sistema de seguridad.

Esto se debe a que los comunicadores universales solo ofrecen un conjunto muy limitado de funciones, generalmente activar y desactivar la central, recibir notificaciones cuando el sistema está sonando y, en algunos casos, manejar una o dos cargas eléctricas. En cambio, el producto propuesto, busca cubrir una más amplia gama de necesidades:

- Asociar múltiples alarmas a un mismo usuario, con distintos niveles de acceso.
- Conocer el estado de cada uno de los sensores que componen la alarma.
- Permitir asignarle un nombre distintivo a cada: alarma, sensor, usuario y carga eléctrica.
- Recibir notificaciones no solo por disparos en la alarma, sino también por problemas en la red eléctrica, por activación o desactivación del sistema distinguiendo qué usuario lo hizo, por eventos personalizados, etc.
- Ser compatible con la línea de productos de automatización de X-28 Alarmas.

Además del equipo que va a ser conectado en la alarma y de la aplicación utilizada por el usuario, el sistema propuesto en este proyecto se completa con el desarrollo de un backend que permita la comunicación entre alarmas y usuarios.

En la Figura 1 se muestra el diagrama en bloques del sistema. Como puede verse, el comunicador se va a conectar a Internet mediante dos posibles vías: Wi-Fi o red celular. Al contemplar dos

formas de conexión, no solo se otorga mayor robustez a la comunicación, sino también se cubre una mayor variedad de casos de uso. Este equipo podría ser usado mediante Wi-Fi en el caso en el que la cobertura celular, en el lugar de la instalación, sea deficiente o en el caso en el que el usuario no quiera atarse a un abono mensual en relación al servicio celular. Por el contrario, podría ser utilizado a través de la red celular en aquellas situaciones en donde no hay una red Wi-Fi disponible.

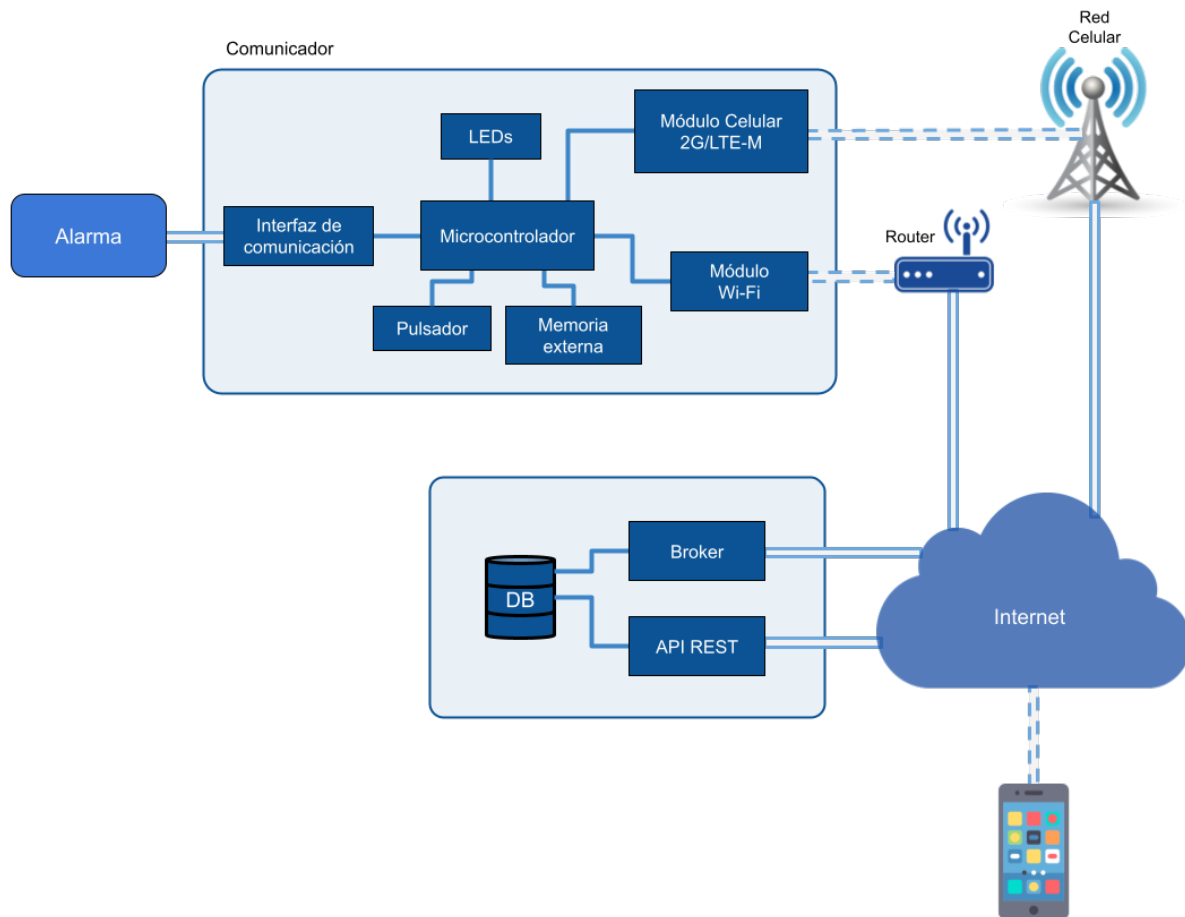


Figura 1. Diagrama en bloques del sistema

Finalmente, en la Figura 1 se observa que el backend se compone de:

- Un broker que se encarga de la comunicación entre las apps de los usuarios finales y los comunicadores
- Una API de tipo REST utilizada por las apps para: login, obtener información de la alarma, enviar comandos al comunicador, etc.
- Una base de datos en donde se lleva un registro de los comunicadores, los usuarios de las apps y el estado y configuración de cada uno de los sistemas de alarma.

Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Auspiciante	Norberto Vergani	X-28 Alarmas	Director
Cliente e impulsor	Claudio Bongiorno	X-28 Alarmas	Jefe de área
Responsable	Mariano Mondani	FIUBA	Alumno
Colaboradores	Pablo Marchant Docentes CEIoT	X-28 Alarmas FIUBA	Jefe de área -
Orientador	Nombre del Director	pertenencia	Director Trabajo final
Usuario final	Clientes de X-28 Alar- mas	-	-

- **Auspiciante:** busca completar la línea de productos de seguridad comercializados por la empresa. Está interesado en que el proyecto se complete en el tiempo establecido y está dispuesto a invertir para ayudar a lograrlo. Además tiene una amplia experiencia en el diseño de hardware de sistemas embebidos y en la programación de firmware. Se deben aprovechar especialmente sus conocimientos en este último tema.
Debido a ser una empresa que vende productos de seguridad, se pone especial atención en mantener el prestigio y la confianza que los clientes depositan en la empresa, por lo que es exigente en cuanto a la calidad y confiabilidad del producto.
- **Cliente e impulsor:** es exigente en cuanto a la integración de los nuevos dispositivos con la línea actual de productos. Su objetivo es que los nuevos equipos sean confiables y robustos en cuanto a su usabilidad. Además cuenta con conocimientos en el diseño de hardware y a participado en el diseño e implementación de gran parte de los productos actuales de la empresa. Sus sugerencias van a ayudar en el diseño del circuito impreso.
- **Colaboradores:** Pablo Marchant es el jefe del área de Comercio exterior, y va a ser el encargado de conseguir muchos de los componentes necesarios para el hardware. Su colaboración va a permitir encontrar la forma de reducir los costos en los componentes electrónicos.
- **Usuario final:** los clientes de X-28 Alarmas están acostumbrados a que los productos de la marca sean sencillos de utilizar. Además, buscan que los nuevos productos que se ofrecen sean compatibles con los ya existentes, permitiéndoles agregarlos sin problemas a sus sistemas de alarma.

1. Propósito del proyecto

El propósito de este proyecto consiste en diseñar e implementar un comunicador, una aplicación y un backend que le permita a los usuarios de las alarmas de X-28 comunicarse con sus sistemas de seguridad. Se busca lograr, por un lado, un producto que no solo sea sencillo de utilizar sino también robusto y seguro. Por otro lado, a lo largo del proyecto, se buscarán introducir prácticas de programación y periféricos de hardware que puedan ser reutilizados en otros proyectos y por lo tanto aumenten la base de conocimiento del sector de Investigación y Desarrollo.

2. Alcance del proyecto

El proyecto incluye:

- El desarrollo de un sistema embebido capaz de conectarse a un sistema de alarma y a Internet, mediante Wi-Fi y red celular. Se incluye tanto hardware como firmware.
- El desarrollo del prototipo de una aplicación híbrida (probada en Android) para poder controlar las alarmas
- El diseño del modelo de datos utilizado en la base de datos del backend
- El desarrollo de un broker que permita la comunicación entre las aplicaciones y los comunicadores.
- El desarrollo de una API REST que le permita a las aplicaciones obtener datos de las alarmas y enviarles comandos

El proyecto no incluye:

- Diseño del gabinete para el comunicador.
- Las implementaciones particulares para que la aplicación funcione en iOS.
- Consideraciones de diseño para el despliegue del broker y la API en un entorno cloud, contemplando alta disponibilidad y escalamiento.

3. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se podrá diseñar e implementar el producto, logrando su correcto funcionamiento.
- No habrá dificultades para conseguir los componentes electrónicos necesarios.
- Se contará con los conocimientos necesarios para desarrollar la aplicación híbrida y el backend. En caso de no contar con ciertos conocimientos, se los podrá investigar y aplicar satisfactoriamente.
- Dentro de la lista de proyectos actualmente en curso del área de Investigación y Desarrollo, se le dará la dedicación suficiente al presente proyecto y no se lo suspenderá en favor del desarrollo de otros productos.

4. Requerimientos

1. Requerimientos del hardware

- 1.1. Incluir una interfaz sencilla con el usuario. Debe existir un único pulsador para que el usuario realice acciones relacionadas con la red Wi-Fi.

- 1.2. Incluir un led bicolor (rojo y verde) para señalar el estado de funcionamiento y de error.
- 1.3. Incluir una interfaz que permita conectar el comunicador al bus del sistema de alarma.
- 1.4. Tener una memoria EEPROM para almacenar información de configuración.
- 1.5. Tener un módulo Wi-Fi.
- 1.6. Tener un módulo celular pueda conectarse a la red celular de 2G y de 4G Cat M1.
- 1.7. El PCB debe respetar las dimensiones del gabinete en el que va a ser comercializado el comunicador.
2. Requerimientos del software embebido
 - 2.1. Configuración de la red Wi-Fi a la que conectarse mediante WPS.
 - 2.2. Configuración de la red Wi-Fi a la que conectarse mediante la app.
 - 2.3. Interpretar los comandos AT del módulo celular.
 - 2.4. Implementar el protocolo definido para comunicar los comunicadores y el broker.
 - 2.5. La conexión con el broker debe ser implementada con TLS.
 - 2.6. La conexión con el broker se debe mantener abierta permanentemente.
 - 2.7. Ser compatible con el protocolo de comunicación de las alarmas de X-28.
 - 2.8. Guardar en la memoria EEPROM la configuración para el correcto funcionamiento del comunicador.
 - 2.9. Implementar la lógica necesaria para elegir prioritariamente la conexión con el broker mediante Wi-Fi y en caso de fallar, hacerlo mediante la red celular.
3. Requerimientos del protocolo de comunicación entre dispositivos y el broker
 - 3.1. Definir la forma en la que se van a identificar los dispositivos a nivel lógico con el broker.
 - 3.2. Permitir autenticar al dispositivo.
 - 3.3. Incluir una lógica de keepalive.
 - 3.4. Permitir solicitar fecha y hora en función de la región en la que esté el comunicador.
 - 3.5. Implementar comandos para solicitarle información al comunicador.
 - 3.6. Permitir el envío de eventos cuando ocurre un suceso en el sistema de alarma.
 - 3.7. Permitir enviar comandos al comunicador para realizar acciones en la alarma.
4. Requerimiento de la aplicación híbrida
 - 4.1. Incluir un login.
 - 4.2. Proceso guiado para ayudar a configurar la red Wi-Fi a la que se va a conectar el comunicador.
 - 4.3. Cada usuario de la aplicación puede agregar múltiples alarmas para gestionirlas.
 - 4.4. Mostrar el estado de la alarma (activada, desactivada, sonando) y problemas que pueda presentar (batería baja, red eléctrica cortada, error de comunicación, etc).
 - 4.5. Permitir enviar comandos a la alarma: activar, desactivar, cambiar su modo, disparar manualmente, incluir y excluir sensores, etc.
 - 4.6. Mostrar el estado de cada uno de los sensores.
 - 4.7. Mostrar el estado de cada una de las cargas eléctrica de la alarma y permitir encenderlas o apagarlas.

- 4.8. Incluir el CRUD de los sensores que tiene la alarma.
- 4.9. Incluir el CRUD de las cargas eléctricas que maneja la alarma.
- 4.10. Incluir el CRUD de los usuarios que tiene la alarma.
5. Requerimientos del broker
 - 5.1. Permitir múltiples comunicadores conectados simultáneamente.
 - 5.2. Interpretar la información que llega desde los comunicadores para actualizar la información en la base de datos.
 - 5.3. Reenviar los mensajes que llegan desde las aplicaciones móviles a los comunicadores.
 - 5.4. Enviar notificaciones push a las aplicaciones cuando se produzcan eventos en los comunicadores que tienen asociados.
6. Requerimientos de la API REST
 - 6.1. Todos los endpoints deben incluir un token exclusivo para cada aplicación.
 - 6.2. Implementar un endpoint para que las aplicaciones puedan autenticarse.
 - 6.3. Incluir endpoints para solicitar información de los comunicadores.
 - 6.4. Incluir endpoints para realizar el CRUD de comunicadores.
 - 6.5. Incluir endpoints para realizar el CRUD de los sensores de una alarma.
 - 6.6. Incluir endpoints para realizar el CRUD de los usuarios de una alarma.
 - 6.7. Incluir endpoints para realizar el CRUD de las cargas eléctricas de una alarma.
 - 6.8. Incluir endpoints para enviar comandos a los comunicadores.
7. Requerimientos de la base de datos
 - 7.1. Persistir los identificadores de los comunicadores que fueron fabricados.
 - 7.2. Persistir la información de los usuarios de las aplicaciones y qué comunicadores tienen asociados.
 - 7.3. Persistir la información del estado de cada alarma: estado de cada sensor, estado de la red eléctrica, estado de la batería; nombres de los sensores, usuarios y cargas eléctricas.

5. Historias de usuarios (*Product backlog*)

Para este proyecto, la ponderación de las historias de usuario se realizará a través de *story points*. El valor se va a determinar a partir de tres aspectos: cantidad de trabajo a realizar, complejidad del trabajo a realizar e incertidumbre del trabajo a realizar. La escala para cada uno de estos aspectos es:

- Cantidad de trabajo:
 - Alta: 13
 - Media: 5
 - Baja: 1
- Complejidad del trabajo:

- Alta: 8
 - Media: 3
 - Baja: 1
- Incertidumbre del trabajo:
 - Alta: 3
 - Media: 2
 - Baja: 1

Para determinar los *story points* de cada historia, se sumará el puntaje asignado a cada aspecto y se lo aproximará al siguiente número de la serie de Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, 55)

1. Como usuario de la alarma quiero recibir notificaciones en el celular cuando se produzca un disparo en mi alarma.
 - Cantidad de trabajo: Media (5)
 - Complejidad del trabajo: Media (3)
 - Incertidumbre del trabajo: Baja (1)
 - *Story points*: 13
2. Como usuario de la alarma quiero activar o desactivar mi central de alarma.
 - Cantidad de trabajo: Media (5)
 - Complejidad del trabajo: Baja (1)
 - Incertidumbre del trabajo: Baja (1)
 - *Story points*: 8
3. Como usuario de la alarma quiero poder ver el estado general de mi alarma y los dispositivos que la integran (sensores, cargas eléctricas, etc)
 - Cantidad de trabajo: Alta (13)
 - Complejidad del trabajo: Alta (8)
 - Incertidumbre del trabajo: Media (2)
 - *Story points*: 34
4. Como usuario de la alarma quiero poder asociar mi usuario de la app a varios equipos de alarma para poder gestionarlos.
 - Cantidad de trabajo: Baja (1)
 - Complejidad del trabajo: Media (3)
 - Incertidumbre del trabajo: Media (2)
 - *Story points*: 8
5. Como usuario de la alarma quiero poder asignarle un nombre a cada sensor y carga eléctrica que compone mi sistema.
 - Cantidad de trabajo: Media (5)
 - Complejidad del trabajo: Media (3)
 - Incertidumbre del trabajo: Baja (1)
 - *Story points*: 13

6. Entregables principales del proyecto

- Comunicador según requerimientos y diagrama esquemático.
- Prototipo funcional de la aplicación híbrida funcionando en Android.
- Implementación del modelo de datos en la base de datos.
- Implementación del broker.
- Implementación de la API REST.
- Informe de avance.
- Memoria escrita del proyecto.
- Presentación publica y defensa del trabajo ante el jurado.

7. Desglose del trabajo en tareas

1. Investigación preliminar (65 horas)
 - 1.1. Investigación acerca de los posibles módulos Wi-Fi y elección de uno de ellos (10 horas)
 - 1.2. Investigación acerca de los posibles módulos celulares y elección de uno de ellos (10 horas)
 - 1.3. Elección del microcontrolador a usar y familiarización con el mismo (15 horas)
 - 1.4. Desarrollo del protocolo de comunicación entre los comunicadores y el broker (30 horas)
2. Desarrollo del hardware (60 horas)
 - 2.1. Diseño del diagrama esquemático y determinación del BOM (20 horas)
 - 2.2. Diseño del PCB y fabricación de un prototipo (30 horas)
 - 2.3. Verificación del prototipo (10 horas)
3. Desarrollo del firmware (230 horas)
 - 3.1. Implementación del driver para el módulo Wi-Fi (30 horas)
 - 3.2. Testeo del driver para el módulo Wi-Fi (10 horas)
 - 3.3. Implementación del driver para el módulo celular (30 horas)
 - 3.4. Testeo del driver para el módulo celular (10 horas)
 - 3.5. Implementación de la capa de comunicación con el sistema de alarma (40 horas)
 - 3.6. Testeo de la capa de comunicación con la alarma (10 horas)
 - 3.7. Implementación del protocolo de comunicación con el broker (20 horas)
 - 3.8. Testeo del protocolo de comunicación con el broker (5 horas)
 - 3.9. Programación de la aplicación principal e integración de los drivers (60 horas)
 - 3.10. Testeo del firmware (15 horas)
4. Desarrollo del backend (85 horas)

- 4.1. Definición del modelo de datos para la base de datos (5 horas)
- 4.2. Implementación del broker (30 horas)
- 4.3. Testeo del broker (10 horas)
- 4.4. Implementación de la API REST (20 horas)
- 4.5. Testeo de la API REST (10 horas)
- 4.6. Integración entre el comunicador y el broker (10 horas)
- 5. Desarrollo de la aplicación híbrida (160 horas)
 - 5.1. Profundización de los conocimientos para el desarrollo de la aplicación (25 horas)
 - 5.2. Maquetado de la aplicación (20 horas)
 - 5.3. Implementación de la aplicación (90 horas)
 - 5.4. Testeo de la aplicación (10 horas)
 - 5.5. Integración entre la aplicación y el backend (15 horas)
- 6. Presentación del proyecto (70 horas)
 - 6.1. Escritura del informe de avance (15 horas)
 - 6.2. Escritura del informe final (40 horas)
 - 6.3. Escritura de la presentación pública (15 horas)

Cantidad total de horas: 670 horas

8. Diagrama de Activity On Node

Los tiempos están expresados en horas.

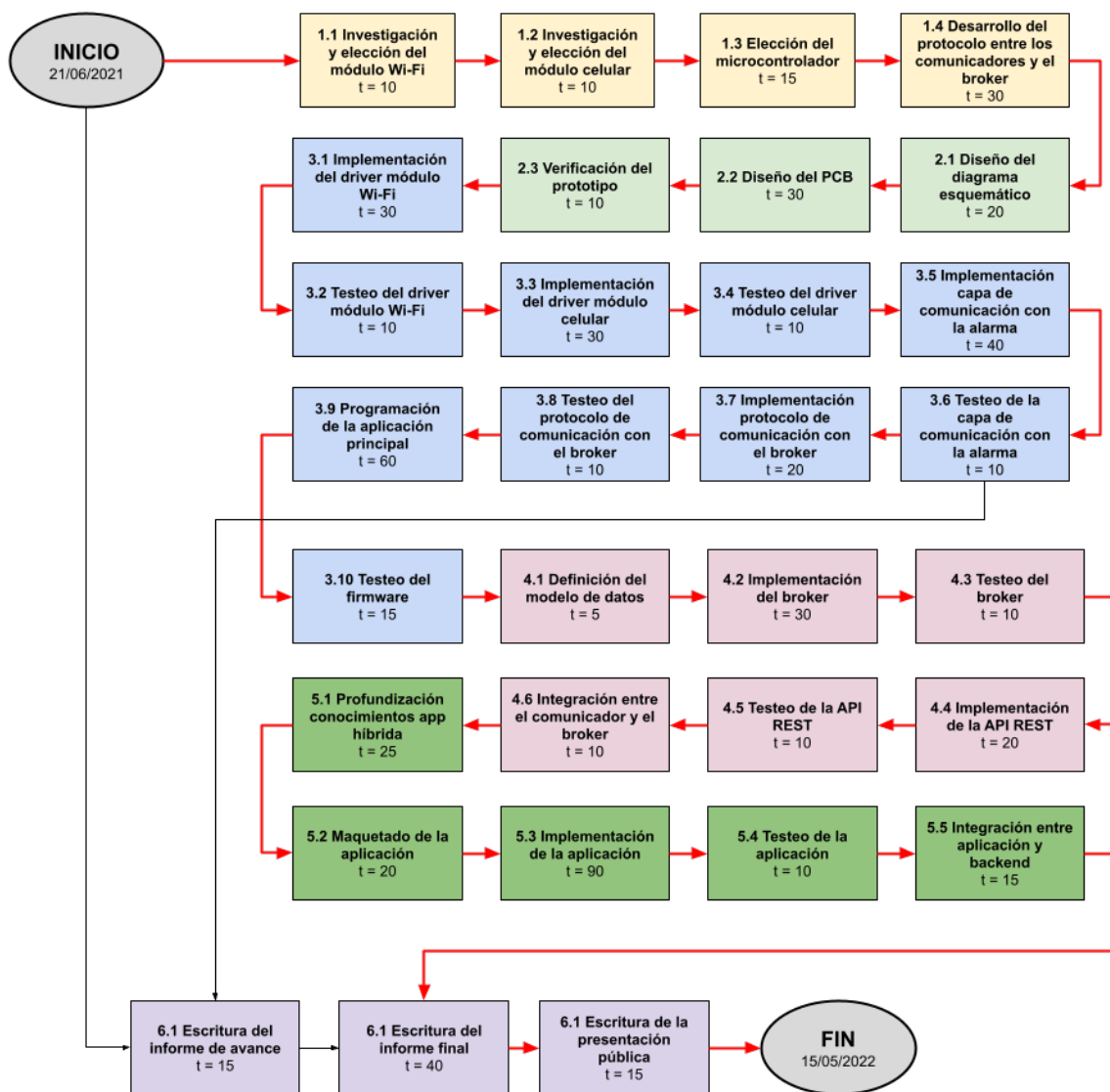


Figura 2. Diagrama en *Activity on Node*

9. Diagrama de Gantt

Nombre	Fecha de inicio	Fecha de fin
1 Investigación preliminar	21/06	07/07
1.1 Investigación y elección módulo Wi-Fi	21/06	22/06
1.2 Investigación y elección módulo celular	23/06	24/06
1.3 Elección y familiarización microcontrolador	25/06	29/06
1.4 Desarrollo de protocolo de comunicación del broker	30/06	07/07
2 Desarrollo del hardware	08/07	22/07
2.1 Diseño del diagrama esquemático	08/07	13/07
2.2 Diseño del PCB y fabricación del prototipo	14/07	20/07
2.3 Verificación del prototipo	21/07	22/07
3 Desarrollo del firmware	23/07	30/09
3.1 Implementación del driver del módulo Wi-Fi	23/07	30/07
3.2 Testeo del driver del módulo Wi-Fi	02/08	06/08
3.3 Implementación del driver del módulo celular	09/08	16/08
3.4 Testeo del driver del módulo celular	17/08	18/08
3.5 Implementación de la capa de comunicación con la alarma	19/08	30/08
3.6 Testeo de la capa de comunicación con la alarma	31/08	01/09
3.7 Implementación del protocolo de comunicación con el broker	02/09	07/09
3.8 Testeo del protocolo de comunicación el broker	08/09	09/09
3.9 Programación de la aplicación principal	10/09	27/09
3.10 Testeo del firmware	28/09	30/09
4 Desarrollo del backend	01/10	25/10
4.1 Definición del modelo de datos	01/10	01/10
4.2 Implementación del broker	04/10	11/10
4.3 Testeo del broker	12/10	13/10
4.4 Implementación de la API REST	14/10	19/10
4.5 Testeo de la API REST	20/10	21/10
4.6 Integración entre el comunicador y el broker	22/10	25/10
5 Desarrollo de la aplicación híbrida	26/10	07/12
5.1 Profundización de los conocimientos para la aplicación híbrida	26/10	01/11
5.2 Maquetado de la aplicación	02/11	05/11
5.3 Implementación de la aplicación	08/11	01/12
5.4 Testeo de la aplicación	02/12	03/12
5.5 Integración entre la aplicación y el backend	06/12	07/12
6 Presentación del proyecto	02/09	22/12
6.1 Escritura del informe de avance	02/09	06/09
6.2 Escritura del informe final	08/12	17/12
6.3 Escritura de la presentación pública	20/12	22/12

Figura 3. Diagrama de Gantt: lista de tareas

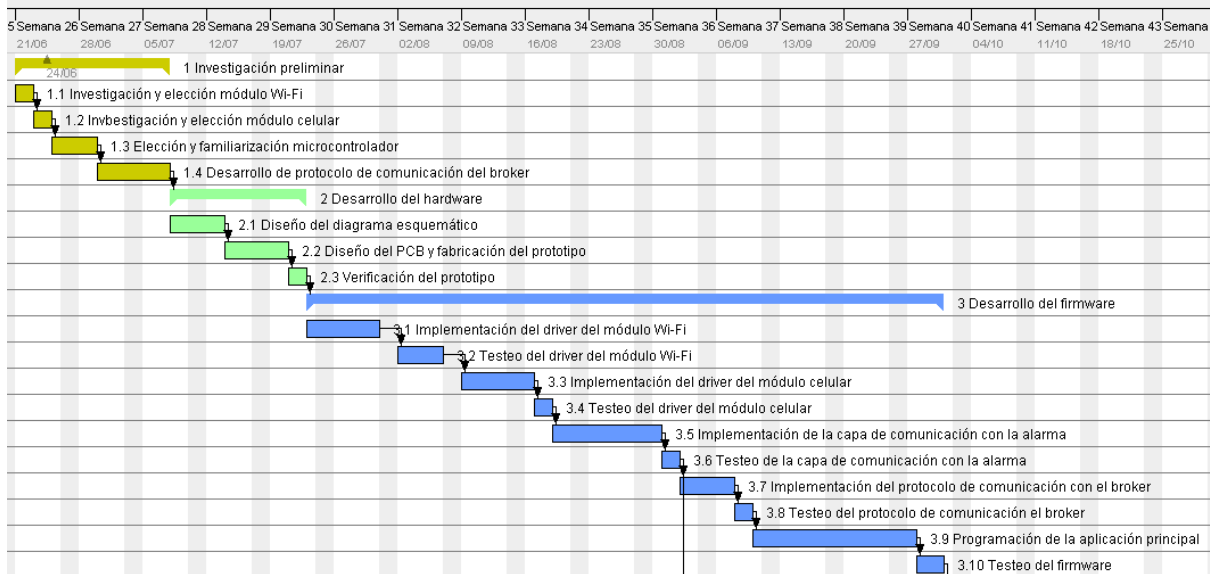


Figura 4. Detalle del diagrama de Gantt

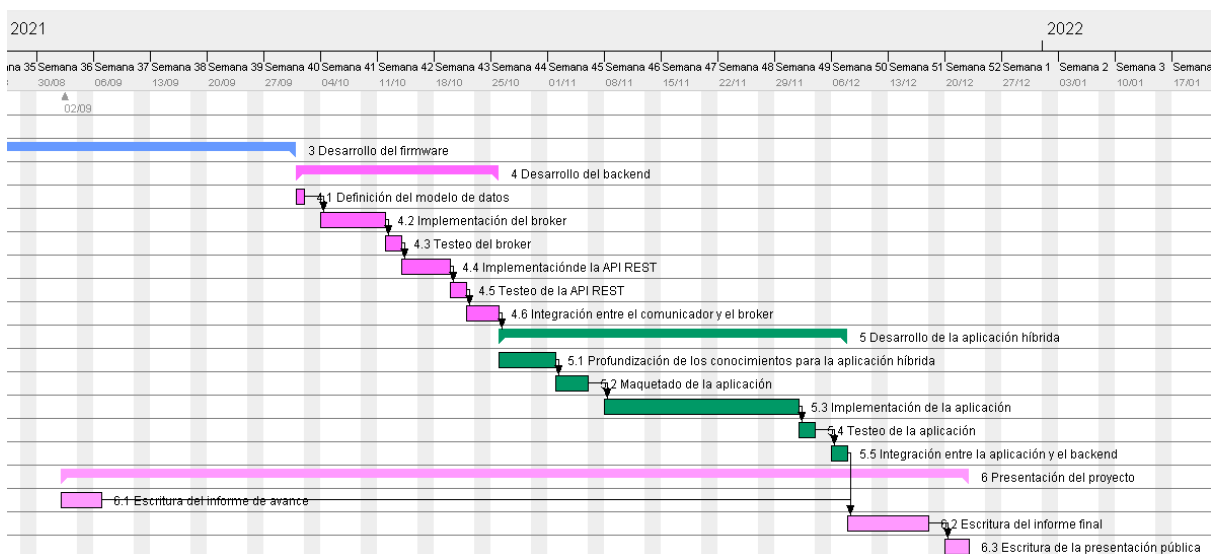


Figura 5. Detalle del diagrama de Gantt

10. Presupuesto detallado del proyecto

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Trabajo directo	670	\$700	\$469.000
Placa de desarrollo	1	\$7.000	\$7.000
Fabricación del PCB	1	\$5.000	\$5.000
SUBTOTAL			\$481.000
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Costos indirectos (30 % del trabajo directo)	-	-	\$140.700
SUBTOTAL			\$140.700
TOTAL			\$621.700

11. Matriz de asignación de responsabilidades

Código WBS	Nombre de la tarea	Listar todos los nombres y roles del proyecto			
		Responsable	Orientador	Auspiciante	Cliente
		Mariano Mondani	Nombre del Director	Norberto Vergani	Claudio Bongiorno
1	Investigación preliminar	P	I	C	A
2	Desarrollo del hardware	P	I	I	C/A
3	Desarrollo del firmware	P	C	A	I
4	Desarrollo del backend	P	C	I	A
5.1	Profundización de los conocimientos	P	C	-	-
5.2	Maquetado de la app	P	I	I	A
5.3	Implementación de la app	P	C	I	A
5.4	Testeo de la app	P	-	I	I
5.5	Integración	P	I	I	A
6	Presentación del proyecto	P	A	-	-

Referencias:

- P = Responsabilidad Primaria
- S = Responsabilidad Secundaria
- A = Aprobación
- I = Informado
- C = Consultado

12. Gestión de riesgos

Riesgo 1: pérdida del prototipo del comunicador.

- Severidad (7): al contar con un único prototipo, en caso de producirse un daño irreparable, el proyecto se detiene hasta que se construya uno nuevo.
- Probabilidad de ocurrencia (4): se toman los recaudos necesarios para evitar dañar el prototipo

Riesgo 2: imposibilidad de conseguir componentes electrónicos clave.

- Severidad (7): los componentes de mayor importancia son el módulo Wi-Fi y el módulo celular. En caso de no conseguir los componentes elegidos, tanto el PCB como el firmware se verán seriamente afectados.
- Ocurrencia (2): la empresa cuenta con varios proveedores de componentes, por lo que es improbable no conseguirlos.

Riesgo 3: retraso en la fabricación del PCB

- Severidad (6): un retraso en la fabricación retrasaría el desarrollo del firmware y por lo tanto el del backend y el de la app.
- Ocurrencia (2): de acuerdo a la experiencia que se tiene con el proveedor de PCB, se concluye que en raras ocasiones se produjo un retraso y, en caso de ocurrir, fue un retraso de unos pocos días.

Riesgo 4: el proyecto queda suspendido frente a otros proyectos más prioritarios.

- Severidad (10): una decisión por parte del directorio que implique suspender este proyecto debido a que hay otros más prioritarios puede provocar que no se puedan cumplir con los plazos establecidos en esta planificación.
- Ocurrencia (2): debido al fuerte interés de la empresa por completar su oferta de productos con este tipo de comunicadores, es difícil pensar que se vaya a dejar de lado este proyecto.

Riesgo 5: retrasos en la implementación de la app

- Severidad (8): una demora en la implementación de la app que acompaña al comunicador retrasaría la salida del producto.
- Ocurrencia (5): debido al bajo dominio que se tiene en relación al desarrollo de aplicaciones híbridas, pueden aparecer demoras debido a problemas con la implementación.

Tabla de gestión de riesgos: (El RPN se calcula como $RPN = S \times O$)

Riesgo	S	O	RPN	S*	O*	RPN*
1. Pérdida del prototipo	7	4	28	2	2	4
2. Imposibilidad de conseguir componentes	7	2	14			
3. Retraso en la fabricación del PCB	6	2	12			
4. Suspensión del proyecto	10	2	20			
5. Retrasos en la implementación de la app	8	5	40	4	2	8

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 25.

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: encargar varias muestras del PCB y armar varios prototipos

- Severidad (2): en caso de dañarse el primer prototipo, inmediatamente se lo puede reemplazar por otro. Sin embargo, se debe minimizar la repetición de esta situación para evitar que otros miembros del equipo de Desarrollo pierdan tiempo en armar más prototipos.
- Probabilidad de ocurrencia (2): es difícil que, debido a una mala manipulación, se dañe más de un prototipo.

Riesgo 5: anticiparse a la etapa del proyecto relacionada con la implementación de la app e ir haciendo pruebas de concepto de distintos aspectos de la aplicación, para profundizar los conocimientos necesarios.

- Severidad (4): las posibles demoras se van a reducir ya que se va a contar con un conocimiento más profundo del framework a usar.
- Probabilidad de ocurrencia (2): a pesar de que, tal vez, no se llegue a la etapa de implementar la app con todos los conocimientos necesarios, la cantidad de incertidumbres respecto al framework a usar va a ser menor.

13. Gestión de la calidad

- Requerimiento #1.1: Incluir una interfaz sencilla con el usuario. Debe existir un único pulsador para que el usuario realice acciones relacionadas con la red Wi-Fi.
 - Verificación: uno de las características que debe informar la hoja de datos del módulo WiFi elegido es la de WPS (WiFi Protected Setup). Es una función que permite conectar un dispositivo a una red WiFi de forma sencilla. Este proceso debe ser disparado en el equipo presionando un botón
 - Validación: disparar el proceso de configuración mediante WPS tanto en el comunicador como en el router WiFi y comprobar que el equipo se conecta satisfactoriamente a la red WiFi.
- Requerimiento #1.2: Incluir un led bicolor (rojo y verde) para señalar el estado de funcionamiento y de error.
 - Verificación: diseñar el firmware para, a través de un led bicolor, informar todos los estados de funcionamiento del equipo: desconectado, buscando una red, conectado, no hay red disponible, etc
 - Validación: realizar pruebas que hagan que el dispositivo atravesase todos los estados de funcionamiento por los que puede pasar y comprobar que las indicaciones a través del led bicolor sean claras.
- Requerimiento #1.3: Incluir una interfaz que permita conectar el comunicador al bus del sistema de alarma.
 - Verificación: el esquemático y el PCB del comunicador debe incluir el hardware necesario para poder conectarlo al bus de comunicación de la alarma.
 - Validación: realizar mediciones mediante un osciloscopio conectando el comunicador al bus de la alarma para determinar que las señales que llegan al microcontrolador tengan los niveles de tensión correctos.

- Requerimiento #1.4: Tener una memoria EEPROM para almacenar información de configuración.
 - Verificación: el esquemático y el PCB del comunicador debe incluir una memoria EEPROM.
 - Validación: se revisará el esquemático y el PCB para corroborar la conexión entre el microcontrolador y la memoria EEPROM.
- Requerimiento #1.5: Tener un módulo Wi-Fi.
 - Verificación: el esquemático y el PCB del comunicador debe incluir un módulo Wi-Fi.
 - Validación: se revisará el esquemático y el PCB para corroborar la conexión entre el microcontrolador y el módulo Wi-Fi.
- Requerimiento #1.6: Tener un módulo celular pueda conectarse a la red celular de 2G y de 4G Cat M1.
 - Verificación: el esquemático y el PCB del comunicador debe incluir un módulo celular. El módulo elegido debe poder conectarse a redes celulares de 2G y de 4G Cat M1.
 - Validación: se revisará el esquemático y el PCB para corroborar la conexión entre el microcontrolador y el módulo celular. Mediante un firmware de prueba se corroborará que el módulo puede conectarse a ambos tipos de redes.
- Requerimiento #1.7: El PCB debe respetar las dimensiones del gabinete en el que va a ser comercializado el comunicador.
 - Verificación: al momento de diseñar el PCB se tendrán en cuenta las dimensiones del gabinete elegido para comercializar el comunicador.
 - Validación: se colocará el PCB fabricado dentro del gabinete elegido para validar la su correcta sujeción.
- Requerimiento #2.1: Configuración de la red Wi-Fi a la que conectarse mediante WPS.
 - Verificación: el módulo WiFi elegido debe soportar esta funcionalidad
 - Validación: disparar el proceso de WPS a través del driver desarrollado para el módulo Wi-Fi y comprobar que el dispositivo se conecta a la red Wi-Fi.
- Requerimiento #2.2: Configuración de la red Wi-Fi a la que conectarse mediante la app.
 - Verificación: el firmware debe incluir un mecanismo para poder recibir los datos para conectar a una red Wi-Fi desde la app.
 - Validación: realizar una prueba en la que se simule el envío de los datos de conexión a una red Wi-Fi de igual forma que lo haría la app.
- Requerimiento #2.3: Interpretar los comandos AT del módulo celular.
 - Verificación: el firmware debe incluir un driver que permita interpretar los comandos AT que llegan desde el módulo celular y a su vez poder enviarle comandos para poder controlarlo.
 - Validación: conectar un conversor TTL a USB en las señales que comunican el microcontrolador con el módulo celular, para poder verificar la correcta comunicación usando una terminal serie en una PC.
- Requerimiento #2.4: Implementar el protocolo definido para comunicar los comunicadores y el broker.

- Verificación: incluir en el firmware un módulo que implemente la lógica del protocolo de comunicación con el broker.
- Validación: comprobar que la comunicación entre el comunicador y el broker cumpla con lo establecido en el protocolo diseñado.
- Requerimiento #2.5: La conexión con el broker debe ser implementada con TLS.
 - Verificación: tanto el módulo Wi-Fi como el módulo celular deben soportar conexiones seguras mediante TLS.
 - Validación: comprobar que la conexión que se abre entre el comunicador y el broker utilice TLS.
- Requerimiento #2.6: La conexión con el broker se debe mantener abierta permanentemente.
 - Verificación: el firmware debe incluir la lógica que mantenga constantemente abierta la conexión con el broker y que en caso de cerrarse, la vuelva a abrir.
 - Validación: simular cortes en la conexión entre el comunicador y el broker y comprobar que se vuelve a establecer.
- Requerimiento #2.7: Ser compatible con el protocolo de comunicación de las alarmas de X-28.
 - Verificación: incluir en el firmware un módulo que se encargue de implementar el protocolo de comunicación propietarios de las alarmas de X-28.
 - Validación: corroborar recepción de mensajes enviados por otros dispositivos de la alarma y el envío de mensajes, por parte del comunicador, a otros equipos de la alarma.
- Requerimiento #2.8: Guardar en la memoria EEPROM la configuración para el correcto funcionamiento del comunicador.
 - Verificación: todos aquellos parámetros que el firmware necesite para funcionar deben ser guardados en la memoria EEPROM externa.
 - Validación: luego de hacer cambios en las configuraciones del comunicador, desconectar la alimentación y volver a conectarla. Comprobar que las configuraciones retienen los valores elegidos.
- Requerimiento #2.9: Implementar la lógica necesaria para elegir prioritariamente la conexión con el broker mediante Wi-Fi y en caso de fallar, hacerlo mediante la red celular.
 - Verificación: el comunicador debe mantenerse siempre conectado a Internet. El firmware debe elegir en primer lugar, conectarse a través de Wi-Fi y en caso de no poder hacerlo, pasar a la red celular.
 - Validación: simular situaciones en las cuales: deje de funcionar Wi-Fi, deje de haber conexión a Internet en el router y deje de haber cobertura celular.
- Requerimiento #3.1: Definir la forma en la que se van a identificar los dispositivos a nivel lógico con el broker.
 - Verificación: el protocolo de comunicación entre los comunicadores y el broker debe incluir algún tipo de identificador único para comunicador.
 - Validación: comprobar que cada comunicador tiene un identificador único y que el broker lo reconoce.

- Requerimiento #3.2: Permitir autenticar al dispositivo.
 - Verificación: el protocolo debe incluir algún mecanismo para que un comunicador se autentique y de esta forma pueda empezar a comunicarse.
 - Validación: comprobar que si el dispositivo cumple con el proceso de autenticación se pueda comunicar con el broker, y en caso contrario, el broker cierre la conexión.
- Requerimiento #3.3: Incluir una lógica de keepalive.
 - Verificación: debe incluir un mecanismo para que los comunicadores indiquen que están conectados y funcionando.
 - Validación: entre los mensajes que llegan al broker desde los comunicadores, se deben encontrar los mensajes de keepalive y en caso de faltar, el broker debe señalarlo.
- Requerimiento #3.4: Permitir solicitar fecha y hora en función de la región en la que esté el comunicador.
 - Verificación: en el protocolo se debe contemplar un mecanismo por medio del cual los comunicadores puedan solicitar la fecha y la hora al broker.
 - Validación: cuando desde un comunicador llegue la solicitud de la fecha y hora, el broker debe responderla.
- Requerimiento #3.5: Implementar comandos para solicitarle información al comunicador.
 - Verificación: en el protocolo se debe incluir un conjunto de comandos mediante los cuales se pueda solicitar información al comunicador.
 - Validación: enviar los comandos al comunicador y comprobar que la información que devuelva cumpla con lo indicado en el protocolo.
- Requerimiento #3.6: Permitir el envío de eventos cuando ocurre un suceso en el sistema de alarma.
 - Verificación: el protocolo debe incluir mensajes que sean usados por el comunicador para informar sobre eventos que ocurrieron en la alarma.
 - Validación: generar eventos en la alarma y comprobar que los mensajes enviados por el comunicador cumplan con lo indicado en el protocolo.
- Requerimiento #3.7: Permitir enviar comandos al comunicador para realizar acciones en la alarma.
 - Verificación: el protocolo debe incluir un conjunto de comandos mediante los cuales se le indique al comunicador que tiene que realizar alguna acción en la alarma.
 - Validación: enviar estos comandos al comunicador y comprobar que realice las acciones indicadas en la alarma.
- Requerimiento #4.1: Incluir un login.
 - Verificación: la app debe incluir una o más pantallas que permitan realizar el proceso de login, incluyendo la creación de una nueva cuenta y la recuperación de una contraseña olvidada.
 - Validación: realizar una prueba en la que se cree una nueva cuenta, se la verifique, se realice el login y se pida recuperar la contraseña.
- Requerimiento #4.2: Proceso guiado para ayudar a configurar la red Wi-Fi a la que se va a conectar el comunicador.

- Verificación: la app debe incluir una o más pantallas que guíen al usuario en el proceso de configurar una red Wi-Fi. En este proceso se le van a enviar los datos de la red Wi-Fi elegida al comunicador.
- Validación: recorrer el proceso de configuración de la red Wi-Fi y comprobar que el comunicador se conecta exitosamente a la red.
- Requerimiento #4.3: Cada usuario de la aplicación puede agregar múltiples alarmas para gestionarlas.
 - Verificación: incluir una pantalla en la que el usuario pueda ver todos los comunicadores que tiene asociados y pueda agregar más.
 - Validación: llevar a cabo el proceso de agregar múltiples comunicadores y comprobar que se visualicen todos ellos en la app.
- Requerimiento #4.4: Mostrar el estado de la alarma (activada, desactivada, sonando) y problemas que pueda presentar (batería baja, red eléctrica cortada, error de comunicación, etc).
 - Verificación: en la app se debe representar el estado de cada una de las alarmas, así como también información general de esta.
 - Validación:
- Requerimiento #4.5: Permitir enviar comandos a la alarma: activar, desactivar, cambiar su modo, disparar manualmente, incluir y excluir sensores, etc.
 - Verificación: se debe incluir en la app la posibilidad de cambiar el estado de la alarma (activada/desactivada), cambiar su modo (modo Estoy, modo Me Voy), dispararla manualmente y cambiar la inclusión de los sensores.
 - Validación: disparar estas acciones desde la app y comprobar que ocurra lo esperado en la alarma a la que se le están enviando los comandos.
- Requerimiento #4.6: Mostrar el estado de cada uno de los sensores.
 - Verificación: incluir una pantalla en la que se visualice el estado de cada uno de los sensores de la alarma: estado normal o anormal.
 - Validación: cambiar el estado de los sensores de la alarma y comprobar que se estos cambios se reflejen en la app.
- Requerimiento #4.7: Mostrar el estado de cada una de las cargas eléctrica de la alarma y permitir encenderlas o apagarlas.
 - Verificación: la alarma puede incluir dispositivos para controlar cargas eléctricas. La app debe incluir una pantalla en la que se visualicen estos dispositivos y permita enviarle a la alarma el comando de encenderlos o apagarlos.
 - Validación: enviar desde la app los comandos de encender y apagar los controladores de cargas eléctricas y comprobar que esto ocurre en la alarma.
- Requerimiento #4.8: Incluir el CRUD de los sensores que tiene la alarma.
 - Verificación: se deben incluir en la interfaz de usuario los controles necesarios para llevar a cabo las acciones de dar de alta, eliminar, modificar y leer los sensores que tiene la alarma.
 - Validación: realizar estas 4 acciones desde la app y ver los cambios se reflejen tanto en la app como en la base de datos.

- Requerimiento #4.9: Incluir el CRUD de las cargas eléctricas que maneja la alarma.
 - Verificación: se deben incluir en la interfaz de usuario los controles necesarios para llevar a cabo las acciones de dar de alta, eliminar, modificar y leer los controladores de cargas eléctricas que tiene la alarma.
 - Validación: realizar estas 4 acciones desde la app y ver los cambios se reflejen tanto en la app como en la base de datos.
- Requerimiento #4.10: Incluir el CRUD de los usuarios que tiene la alarma.
 - Verificación: se deben incluir en la interfaz de usuario los controles necesarios para llevar a cabo las acciones de dar de alta, eliminar, modificar y leer los usuarios que tiene la alarma.
 - Validación: realizar estas 4 acciones desde la app y ver los cambios se reflejen tanto en la app como en la base de datos.
- Requerimiento #5.1: Permitir múltiples comunicadores conectados simultáneamente.
 - Verificación: el broker debe ser capaz de mantener múltiples conexiones abiertas con los comunicadores.
 - Validación: conectar varios comunicadores al broker y comprobar que el broker pueda gestionar correctamente las conexiones.
- Requerimiento #5.2: Interpretar la información que llega desde los comunicadores para actualizar la información en la base de datos.
 - Verificación: el broker va a recibir mensajes con información proveniente de los comunicadores. Debe incluir la lógica para procesar esta información y actualizar la base de datos, registrando así el estado de la alarma y de sus componentes.
 - Validación: generar cambios en la alarma (activarla/desactivarla, incluir/excluir sensores, apagar/encender cargas eléctricas, etc) y comprobar que el broker recibe los mensajes y actualiza la base de datos.
- Requerimiento #5.3: Reenviar los mensajes que llegan desde las aplicaciones móviles a los comunicadores.
 - Verificación: el broker debe reenviar los comandos que lleguen desde las apps destinados a los comunicadores.
 - Validación: disparar acciones desde la app y comprobar que llegan al broker y luego son reenviados al comunicador que corresponde.
- Requerimiento #5.4: Enviar notificaciones push a las aplicaciones cuando se produzcan eventos en los comunicadores que tienen asociados.
 - Verificación: cuando se produzcan ciertos eventos en la alarma, el comunicador lo va a informar al broker. Cuando el broker reciba estos mensajes, debe enviar notificaciones push a todas las apps que lo tengan asociado.
 - Validación: generar eventos en la alarma y comprobar que aquellas app que tengan asociado al comunicador en cuestión, reciban las notificación push correspondientes a los eventos.
- Requerimiento #6.1: Todos los endpoints deben incluir un token exclusivo para cada aplicación.

- Verificación: la API debe implementar la lógica para generar tokens que le permitan autenticar los requests que reciba desde las apps.
- Validación: obtener desde la API el token y utilizarlo en los requests que lo requieran para comprobar que si no se lo incluye, el request devuelve un error.
- Requerimiento #6.2: Implementar un endpoint para que las aplicaciones puedan autenticarse.
 - Verificación: la API debe tener un endpoint el cual sea usado por las apps en el proceso de login.
 - Validación: enviar la información de login y comprobar que, en caso de que la información sea correcta, la API devuelve un token válido para esa app.
- Requerimiento #6.3: Incluir endpoints para solicitar información de los comunicadores.
 - Verificación: se deben incluir endpoints mediante los cuales se le indique al broker que debe enviar mensajes al comunicador para obtener información.
 - Validación: realizar los requests a estos endpoint y corroborar que se envíen los mensajes correctos al comunicador.
- Requerimiento #6.4: Incluir endpoints para realizar el CRUD de comunicadores.
 - Verificación: la API debe incluir los endpoint que permitan realizar las operaciones de CRUD de los comunicadores.
 - Validación: ejecutar los requests y comprobar que la base de datos se actualiza correctamente.
- Requerimiento #6.5: Incluir endpoints para realizar el CRUD de los sensores de una alarma
 - Verificación: la API debe incluir los endpoint que permitan realizar las operaciones de CRUD de los sensores.
 - Validación: ejecutar los requests y comprobar que la base de datos se actualiza correctamente.
- Requerimiento #6.6: Incluir endpoints para realizar el CRUD de los usuarios de una alarma
 - Verificación: la API debe incluir los endpoint que permitan realizar las operaciones de CRUD de los usuarios de una alarma.
 - Validación: ejecutar los requests y comprobar que la base de datos se actualiza correctamente.
- Requerimiento #6.7: Incluir endpoints para realizar el CRUD de las cargas eléctricas de una alarma
 - Verificación: la API debe incluir los endpoint que permitan realizar las operaciones de CRUD de los controladores de cargas eléctricas.
 - Validación: ejecutar los requests y comprobar que la base de datos se actualiza correctamente.
- Requerimiento #6.8: Incluir endpoints para enviar comandos a los comunicadores.
 - Verificación: se deben incluir endpoints mediante los cuales se le indique al broker que debe enviar comandos al comunicador.

- Validación: realizar los requests a estos endpoint y corroborar que se envíen los comandos correctos al comunicador.
- Requerimiento #7.1: Persistir los identificadores de los comunicadores que fueron fabricados.
 - Verificación: en el proceso de fabricación de los comunicadores, se debe incluir una etapa en la que se den de alta en el backend los comunicadores.
 - Validación: comprobar que luego del proceso de alta de los comunicadores en el backend, queden registrados en la base de datos.
- Requerimiento #7.2: Persistir la información de los usuarios de las aplicaciones y qué comunicadores tienen asociados.
 - Verificación: la base de datos debe incluir las tablas necesarias para registrar los usuarios creados en las apps y los comunicadores que tienen vinculados.
 - Validación: al crear un usuario en una app y vincularlo a comunicadores, comprobar que esta información es agregada a la base de datos
- Requerimiento #7.3: Persistir la información del estado de cada alarma: estado de cada sensor, estado de la red eléctrica, estado de la batería; nombres de los sensores, usuarios y cargas eléctricas.
 - Verificación: la base de datos debe incluir las tablas necesarias para modelar toda la información de una alarma, desde su estado hasta información los dispositivos que la componen.
 - Validación: realizar cambios y generar eventos en la alarma. Comprobar que esta información queda reflejada en la base de datos.

14. Procesos de cierre

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original
 - Encargado: Mariano Mondani.
 - Se revisará el plan de trabajo y se revisará punto por punto el cumplimiento de los requerimientos y objetivos.
 - Se revisará el diagrama de Gantt y se analizarán las demoras que pudieran aparecer.
 - Se pedirá la opinión de los auspiciantes y del cliente para discutir el grado de satisfacción con el resultado del proyecto.
- Identificación de las técnicas y procedimientos útiles e inútiles que se utilizaron, y los problemas que surgieron y cómo se solucionaron
 - Encargado: Mariano Mondani.
 - Se realizará una reunión con los auspiciantes y los colaboradores de la empresa.
 - A partir de la memoria escrita se discutirán las técnicas usadas tanto en el desarrollo del hardware como del software.
 - Se analizará cuáles de estas técnicas puede ser incorporada a la base de conocimiento del sector de Desarrollo para aplicar en futuros proyectos.
 - Se plantearán los problemas que surgieron durante el proyecto y se discutirá si las soluciones implementadas fueron las más convenientes.

- Acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores
 - Encargado: Mariano Mondani.
 - Se escribirá un agradecimiento a los auspiciantes y a los colaboradores en la memoria del proyecto.
 - Se enviará la presentación y la memoria del proyecto a los auspiciantes y colaboradores, expresando el profundo agradecimiento hacia ellos.