



Desarrollo de aplicación Hospitalaria con MQTT

Autor:

Ing. Gustavo Bastian

Director:

Mg.Ing. Ericson Joseph Estupiñan Pineda (Surix S.R.L)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 31 de Octubre de 2021 y el 7 de Diciembre de 2021.*

Índice

| | |
|---|----|
| 1. Descripción técnica-conceptual del proyecto a realizar | 5 |
| 2. Identificación y análisis de los interesados | 7 |
| 3. Propósito del proyecto | 7 |
| 4. Alcance del proyecto | 7 |
| 5. Supuestos del proyecto. | 8 |
| 6. Requerimientos | 8 |
| 7. Historias de usuarios (<i>Product backlog</i>). | 9 |
| 8. Entregables principales del proyecto | 11 |
| 9. Desglose del trabajo en tareas | 11 |
| 10. Diagrama de Activity On Node. | 13 |
| 11. Diagrama de Gantt | 15 |
| 12. Presupuesto detallado del proyecto | 20 |
| 13. Gestión de riesgos | 20 |
| 14. Gestión de la calidad | 22 |
| 15. Procesos de cierre | 25 |

Registros de cambios

| Revisión | Detalles de los cambios realizados | Fecha |
|----------|--|------------|
| 0 | Creación del documento | 31/10/2021 |
| 1 | Se completa hasta el punto 5 inclusive | 3/11/2021 |
| 2 | Se completa hasta el punto 9 inclusive | 11/11/2021 |
| 3 | Se corrigen historias de usuario y se completa hasta el punto 12 inclusive | 18/11/2021 |
| 4 | Se completa la totalidad del documento | 25/11/2021 |

Acta de constitución del proyecto

Buenos Aires, 31 de Octubre de 2021

Por medio de la presente se acuerda con el Ing. Ing. Gustavo Bastian que su Trabajo Final de la Carrera de Especialización en Internet de las Cosas se titulará “Desarrollo de aplicación Hospitalaria con MQTT”. Consistirá esencialmente en la implementación de un prototipo de una aplicación móvil para la llamada, gestión de enfermeras y consulta enfermera-médico mediante la utilización del protocolo MQTT, y tendrá un presupuesto preliminar estimado de 620 hs de trabajo y \$649400, con fecha de inicio 31 de Octubre de 2021 y fecha de presentación pública 15 de mayo de 2022.

Se adjunta a esta acta la planificación inicial.

Dr. Ariel Lutenberg
Director posgrado FIUBA

Ing. Sergio Starkloff
Surix S.R.L

Mg.Ing. Ericson Joseph Estupiñan Pineda
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

En la actualidad, el avance de la Internet de las Cosas (IoT de *Internet of Things*) y la disminución de costos asociados a la tecnología hacen factible su incorporación a distintos campos de la vida cotidiana. Uno de esos campos es el de infraestructuras hospitalarias inteligentes.

Por otra parte, dentro de las múltiples opciones para realizar la comunicación entre los dispositivos IoT, el protocolo Message Queuing Telemetry Transport (en adelante MQTT) se ha probado como un protocolo confiable y ampliamente utilizado.

Dentro del contexto, en este trabajo se desarrollará una aplicación multiplataforma que utilizará el protocolo MQTT para los distintos participantes de la actividad hospitalaria. El proyecto es una necesidad de la empresa Surix S.R.L. y se lleva a cabo como parte de la carrera Especialización de Internet de las Cosas.

Surix S.R.L. es una firma que se dedica al desarrollo, fabricación y comercialización de productos IP y sistemas hospitalarios de calidad. Posee una comprobada trayectoria dentro del mercado local e internacional. Se destaca por su compromiso con la industria nacional, la mejora continua de sus productos y el soporte que brinda a sus clientes. Este proyecto se enmarca dentro del segundo ítem de su misión, porque mejora y extiende capacidades a un sistema existente.

Surix S.R.L. fabrica un sistema IP de llamado a enfermera que está basado en el protocolo SIP. Este consiste en un servidor central y terminales que se encuentran en las habitaciones del hospital. La aplicación principal se ejecuta en una pc o bien en una tablet y monitorea el estado de las habitaciones.

El objetivo del proyecto es realizar un sistema con las ventajas que provee el protocolo MQTT, como es la posibilidad de agregar accesorios rápidamente con bajo costo de software, hardware e implementación.

MQTT es un protocolo open source liviano, hecho que permite aplicarlo en dispositivos con pocos recursos y baja velocidad de transmisión, ampliamente utilizado en dispositivos IoT. Está basado en la pila TCP/IP, se implementa en la capa de aplicación y sus mensajes se transmiten como colas de publicación/subscripción.

El desafío de este proyecto consiste en la programación de un sistema que contenga un servidor o broker MQTT, una base de datos donde alojar información de reportes de habitaciones/enfermeras y datos relevantes al paciente (incluyendo temporizadores para suministro de ciertos medicamentos y/o control), una página web para configuración y una aplicación multiplataforma donde se realice la gestión de datos e interacción con los clientes. La aplicación será capaz de identificar la cama correspondiente (mediante lectura de símbolos QR) y de transmitir mensajes de voz en caso de ser necesario.

La motivación que origina la realización del proyecto es generar las bases para poder incorporar otros dispositivos inteligentes al sistema principal a bajo costo. Por ejemplo, en un futuro se puede monitorear la temperatura de la habitación y saber si sufre un desperfecto el aire acondicionado, escuchar sonidos dentro de la sala en caso de que el paciente no pueda acceder al llamador, etc.

El proyecto completo consta de :

- Broker.
- Pantalla web de configuración.
- Cliente médico.
- Cliente enfermera.
- Cliente sistema.

En la figura 1 se presenta un diagrama de interconexión entre los dispositivos.

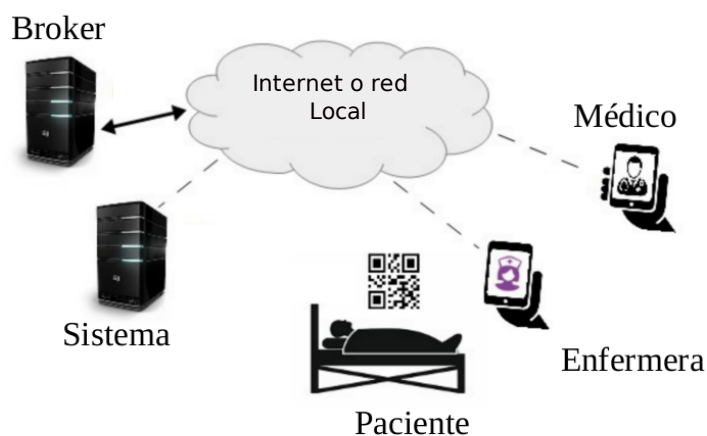


Figura 1. Diagrama en bloques del sistema.

El broker recibe mensajes llamados “eventos”, de distintos publicadores y los reenvía a los subscriptores que correspondan según una política asignada previamente. El cliente sistema puede gestionar la base de datos con información relevante para los pacientes y también puede observar el estado de las habitaciones. Mientras que el cliente enfermera solo recibe asignaciones o respuestas de un médico, y publica finalización o consulta por medio de la aplicación. Por último, el cliente médico solo recibe consultas y publica respuestas por medio de la aplicación.

Con la aplicación web se puede cargar la base de datos con información de enfermeras, pacientes y médicos.

2. Identificación y análisis de los interesados

| Rol | Nombre y Apellido | Organización | Puesto |
|---------------|--|--------------|------------------------|
| Cliente | Ing. Sergio Starkloff | Surix S.R.L | Propietario |
| Responsable | Ing. Gustavo Bastian | FIUBA | Alumno |
| Colaboradores | – | – | – |
| Orientador | Mg.Ing. Ericson Joseph Estupiñan Pineda | Surix S.R.L | Director trabajo final |
| Usuario final | personal de salud, administradores de sistemas | – | – |

- Orientador: Ericson va a poder ayudar mucho con la definición de los requerimientos.
- Usuario final: Todos los usuarios del sistema como ser, administradores de redes hospitalarias, médicos y enfermeras, que deseen observar y/o controlar el proceso.

3. Propósito del proyecto

El propósito de este proyecto es desarrollar un sistema para gestión de enfermeras basado en el protocolo MQTT compuesto por un broker, una web de configuración y una o varias aplicaciones, que agilice el desarrollo de funcionalidades futuras.

4. Alcance del proyecto

El presente proyecto incluye:

- Confección del plan de trabajo.
- Investigación y estudio del protocolo MQTT, bases de datos SQL y NoSQL, programación de aplicaciones web y programación de aplicaciones multiplataforma.
- Desarrollo local del broker que registre en un log las actividades realizadas y su horario de realización.
- Desarrollo local de una página web de configuración.
- Desarrollo local de una aplicación cliente médico con envío/recepción de mensajes de audio/texto/alarmas.
- Desarrollo local de una aplicación cliente enfermera con envío/recepción de mensajes de audio/texto/alarmas y escaneo de QR para identificar paciente.
- Desarrollo local de una aplicación que asigna enfermeras a pacientes de acuerdo a la indicación de un médico. Por ejemplo, todos los días a las 17 hs, medir la presión arterial del paciente “x”, independientemente de la enfermera. Esta aplicación permite monitorizar el estado de las habitaciones.
- Documentación de las aplicaciones desarrolladas.

El presente proyecto NO incluye:

- Manuales de las distintas aplicaciones desarrolladas.
- Traducciones a idiomas extranjeros de las aplicaciones y/o de la página web.
- Sistema llamador del paciente.
- Análisis en profundidad de tráfico en la red.
- Análisis en profundidad de Seguridad.
- Contratación de base de datos remota.
- Contratación e instalación de servidores remotos.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Existe un llamador paciente.
- Las aplicaciones moviles usaran iOS y Android como sistema operativo.
- La base de datos de los pacientes sólo puede ser afectada por medio de la aplicación sistema (el cliente médico sólo puede hacer consultas/modificaciones puntuales a una situación).
- Se dispondrá al menos de una pc para instalación del servidor web, la base de datos, y del broker MQTT.
- Se dispondrá de un router para la prueba funcional del sistema.

6. Requerimientos

Los principales requerimientos relevados son los siguientes:

1. Requerimientos funcionales del servidor:

- 1.1. El servidor debe tener instalado Eclipse Mosquitto como broker MQTT.
- 1.2. El servidor debe tener endpoints para interactuar con una aplicación Web, para consultas y envío de datos hacia dispositivos clientes.
- 1.3. El servidor debe tener endpoints para interactuar con la base de datos.

2. Requerimientos funcionales de la base de datos:

- 2.1. El sistema debe poseer una base de datos relacional.
- 2.2. La base de datos debe poseer datos cargados por default.
- 2.3. La base de datos a utilizar debe poseer las siguientes tablas:
 - Habitaciones
 - Pacientes

- Médicos
- Enfermeras
- Eventos

3. Requerimientos funcionales de la aplicación web de configuración:

- 3.1. La aplicación web debe ser cliente del broker MQTT.
- 3.2. La aplicación web debe poseer funciones de consulta o modificación de la base de datos.
- 3.3. La aplicación web debe permitir visualización de estadísticas de pacientes o enfermeras.
- 3.4. La aplicación debe contener acceso con usuario y contraseña para cada persona.

4. Requerimientos funcionales de la aplicación enfermera:

- 4.1. La aplicación enfermera debe tener 3 modos de uso: usuario médico, usuario enfermera y usuario sistema.
- 4.2. Al logearse el usuario, y contrastarse con la base de datos del sistema, se activa el modo correspondiente.
- 4.3. La aplicación en modo enfermera debe permitir leer códigos QR.
- 4.4. La aplicación en modo enfermera debe poder descargar información relevante del paciente (medicamentos a suministrar, controles a realizar, contacto del médico a cargo del tratamiento, etc).
- 4.5. La aplicación en modo sistema debe mostrar las habitaciones sin atención, según una tabla de prioridades y en caso de igualdad de prioridades, mostrár según el orden de llamada.
- 4.6. El modo usuario médico y el modo usuario enfermera deben poder enviar mensajes de textos o sonido comprimido en 32kbps (mp3).

5. Requerimientos de documentación:

- 5.1. Confección de documento con información relativa a la base de datos: detalles de la misma y de API para acceder.
- 5.2. Memoria del proyecto con diagramas UML de aplicación modo sistema, aplicación modo enfermera, aplicación modo médico y página web.

6. Requerimientos de integración de sistema:

- 6.1. El sistema debe integrar el funcionamiento del servidor con base de datos , aplicación web y aplicaciones móviles.

7. Historias de usuarios (*Product backlog*)

Se identifican los siguientes roles:

- Usuario final: persona que utilizará el sistema. Existen usuarios finales bien definidos: usuario final modo enfermero, usuario final modo médico y usuario final modo operador del sistema.
- Cliente: persona que necesita el desarrollo. Sus intereses están orientados al negocio.

Story points: Se utilizarán 1, 2, 3, 5 y 8 para la estimación de las historias de usuario, según la serie de Fibonacci.

- Historias de 1 punto: son historias cuya implementación es rápida y con baja posibilidad de falla. Por ejemplo envío de un mensaje sencillo entre un nodo y el servidor de la red.
- Historias de 2 puntos: son historias cuya implementación no es rápida, pero tienen baja posibilidad de falla. Por ejemplo el diseño de la interfaz de usuario.
- Historias de 3 puntos: son historias cuya implementación no es rápida y tienen una posibilidad de falla media. Por ejemplo para la lectura de códigos QR es necesario investigar bibliotecas y evaluar el correcto funcionamiento en dispositivos con sistemas operativos distintos.
- Historias de 5 puntos: son historias cuya implementación es lenta y poseen una posibilidad de falla alta ya que interactúan varios elementos de la red. Por ejemplo el envío de un mensaje de audio entre distintos actores, debido a que hay que corroborar el funcionamiento de la captura de audio en ambos dispositivos, el correcto traslado por la red y la recepción perfecta del archivo.
- Historias de 8 puntos: son historias cuya implementación es compleja y poseen elevada posibilidad de falla porque interactúan distintos componentes del sistema. Por ejemplo en la solicitud de ayuda de una enfermera al sistema, donde se debe chequear disponibilidad de otras enfermeras, y dependiendo del tipo de ayuda necesaria, la capacidad de la misma.

Historias de Usuario:

- Como usuario en modo enfermero quiero que la aplicación me notifique si hay un pedido de una habitación para ir a atenderla. Dificultad baja y posibilidad de falla baja. (1 punto)
- Como usuario en modo enfermero quiero que la aplicación permita que notifique que finalicé la tarea para poder recibir otro pedido. Dificultad baja y posibilidad de falla baja. (1 punto)
- Como usuario en modo enfermero quiero que la aplicación me permita notificar que estoy atendiendo para no tener varios pedidos pendiente. Dificultad media y posibilidad de falla media. (3 puntos)
- Como usuario en modo enfermero quiero que la aplicación permita que consulte a un médico en caso de necesidad. Dificultad muy alta y posibilidad de falla muy alta. (8 puntos)
- Como usuario en modo enfermero quiero que la aplicación permita obtener información del paciente para entender el estado de la situación. Dificultad alta y posibilidad de falla alta. (5 puntos)
- Como usuario en modo enfermero quiero que la aplicación permita avisar que mi jornada terminó para no recibir más notificaciones. Dificultad alta y posibilidad de falla alta. (5 puntos)
- Como usuario en modo enfermero quiero que la aplicación permita que notifique que necesito ayuda de otro enfermero para atender al paciente. Dificultad muy alta y posibilidad de falla muy alta. (8 puntos)

- Como usuario en modo médico quiero que la aplicación posea un mecanismo sencillo para responder dudas de la enfermera. Dificultad media y posibilidad de falla media. (3 puntos)
- Como usuario en modo médico quiero que la aplicación permita acceder a la información del paciente en caso de consulta de un enfermero. Dificultad media y posibilidad de falla baja. (2 puntos)
- Como usuario en modo sistema quiero que la aplicación me permita asignar tareas temporizadas para atender rutinas preestablecidas por el médico. Dificultad alta y posibilidad de falla alta. (5 puntos)
- Como usuario en modo sistema quiero que la aplicación permita obtener datos estadísticos de cada enfermero para poder reportar. Dificultad media y posibilidad de falla baja. (2 puntos)
- Como usuario en modo sistema quiero que la aplicación móvil me muestre el estado de las habitaciones. Dificultad baja y posibilidad de falla baja. (1 punto)
- Como cliente quiero que el sistema sea de fácil instalación. Dificultad baja y posibilidad de falla baja. (1 punto)

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Plan de proyecto del trabajo final y memoria técnica.
- Servidor MQTT.
- Base de datos configurada.
- Aplicación Web responsive.
- Aplicación enfermera (con los 3 modos de funcionamiento).
- Código del sistema (utilizando dockerhub y github).

9. Desglose del trabajo en tareas

Desglose del trabajo:

1. Investigación preliminar (70 hs):
 - 1.1. Confección del plan de trabajo del proyecto (10 hs).
 - 1.2. Investigar funcionamiento del broker Mosquitto (10 hs).
 - 1.3. Investigar base de datos relacionales (10 hs).
 - 1.4. Investigar bibliotecas gráficas para implementar página web (20 hs).
 - 1.5. Investigar bibliotecas gráficas para implementar aplicación móvil (20 hs).
2. Implementación del servidor web (40 hs):

- 2.1. Instalación y configuración de Node.js (10 hs).
- 2.2. Instalación del broker MQTT (10 hs).
- 2.3. Programación de funciones de consulta para aplicación web /móvil (20 hs).
3. Implementación de la base de datos (110 hs):
 - 3.1. Instalación de base de datos (40 hs).
 - 3.2. Configuración de parámetros de seguridad y datos (30 hs).
 - 3.3. Creación y carga de datos en tablas (40 hs).
4. Implementación de página web (190 hs):
 - 4.1. Programación de funciones MQTT como cliente del broker (50 hs).
 - 4.2. Prueba de envío y recepción de datos al broker (10 hs).
 - 4.3. Programación de funciones de consulta de datos a la base de datos (40 hs).
 - 4.4. Programación de funciones de carga de datos a la base de datos (40 hs).
 - 4.5. Programación de lógica de acceso y seguridad de usuarios (40 hs).
 - 4.6. Prueba de acceso de usuarios (10 hs).
5. Implementación de aplicación enfermera (140 hs):
 - 5.1. Programación de interfaz de login de usuarios (8 hs).
 - 5.2. Testeo de lógica y acceso de seguridad de usuarios (6 hs).
 - 5.3. Programación de interfaces de usuario aplicación móvil (30 hs).
 - 5.4. Programación de funciones de acceso a la base de datos (40 hs).
 - 5.5. Prueba de envío y recepción de datos entre clientes modo enfermera y cliente modo médico (10 hs).
 - 5.6. Programación y prueba de lectura de códigos QR (6 hs):
 - 5.7. Programación de captura/compresión/envío de audio usuario modo enfermera (30 hs).
 - 5.8. Prueba de envíos de distintos audios (10 hs).
6. Pruebas de sistema (10 hs):
 - 6.1. Prueba del funcionamiento correcto con 2 enfermeras 2 médicos y 3 llamadores (10 hs).
7. Documentación técnica del proyecto (60 hs):
 - 7.1. Confección de informe de avance (10 hs).
 - 7.2. Redacción de la memoria técnica del proyecto(50 hs).

Cantidad total de horas: (620 hs).

10. Diagrama de Activity On Node

En la figura 2 se identifican mediante distintos colores las tareas del proyecto:

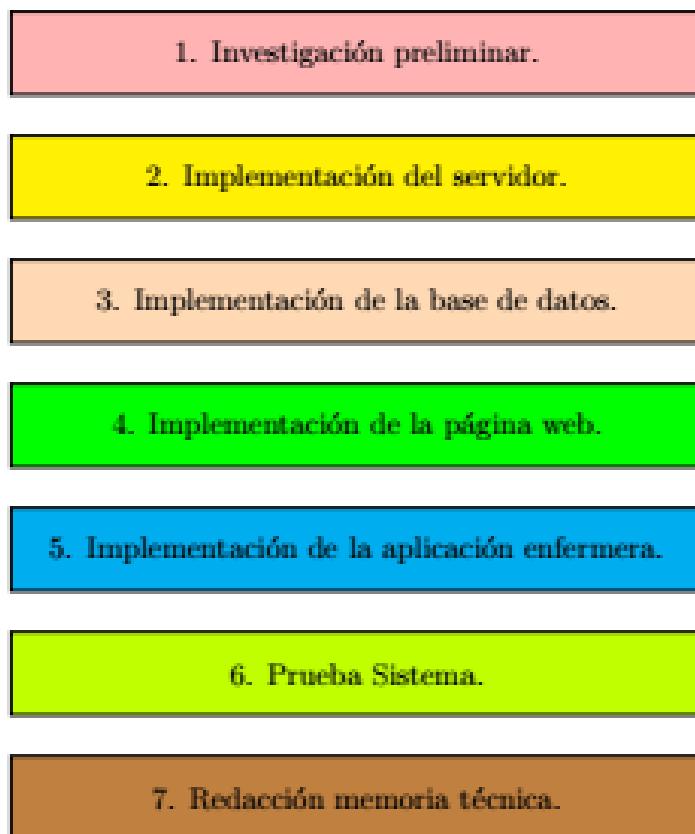


Figura 2. Colores del diagrama de *Activity on Node*.

En la figura 3 se presenta el diagrama de *Activity on Node*, donde el camino crítico es indicado mediante una línea de interconexión resaltada:

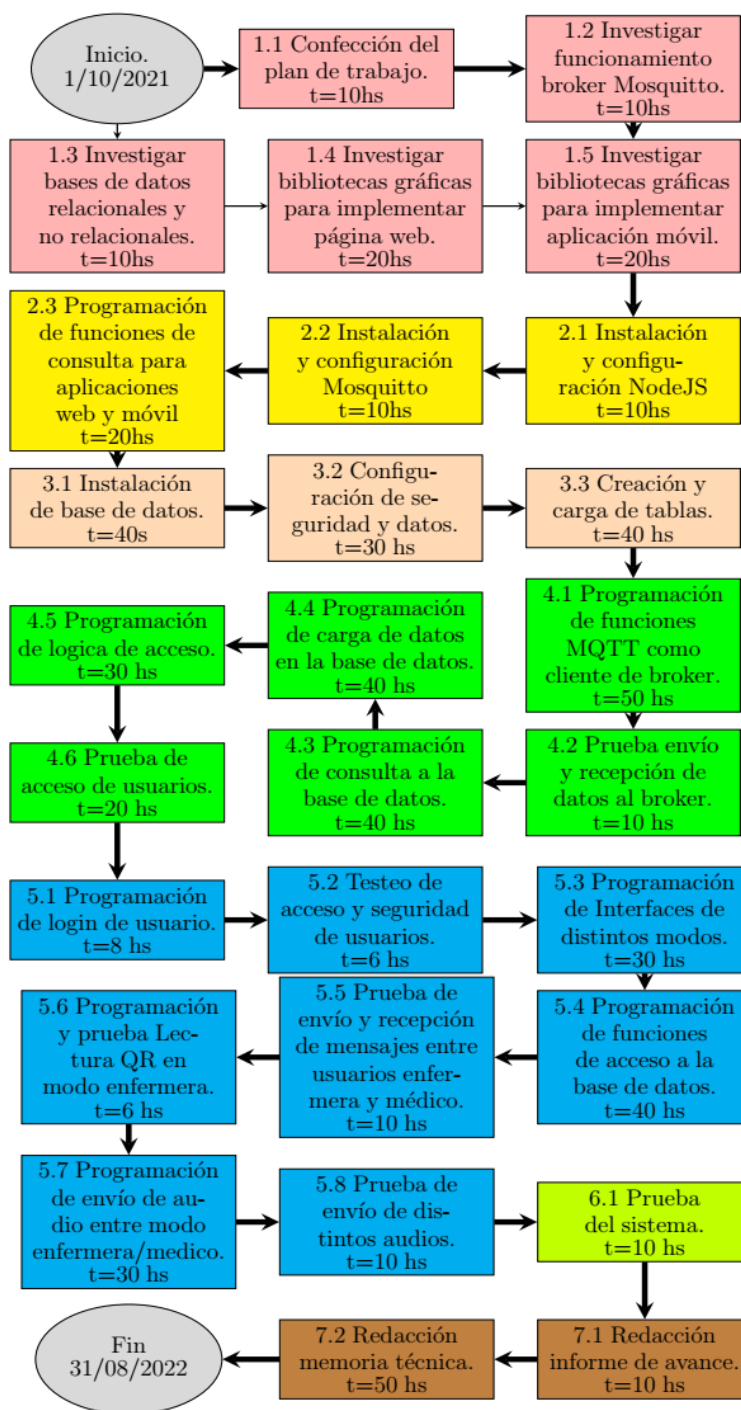


Figura 3. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

En la siguiente figura se presenta la lista de tareas ingresadas al sistema de gestión de tareas y el diagrama de Gantt con las tareas acordes con el diagrama de *Activity on Node*.

|  | | | |
|---|---|-----------------|--------------|
| | Nombre | Fecha de inicio | Fecha de fin |
| ▲ | 1 Investigación preliminar. | 1/10/21 | 3/3/22 |
| ● | 1.1 Redacción plan de trabajo | 1/10/21 | 1/10/21 |
| ● | 1.2 Investigar funcionamiento broker Mosquitto. | 1/3/22 | 1/3/22 |
| ● | 1.3 Investigar bases de datos relacionales y no relacionales | 1/10/21 | 1/10/21 |
| ● | 1.4 Investigar bibliotecas para desarrollo página web. | 2/10/21 | 4/10/21 |
| ● | 1.5 Investigar bibliotecas para aplicación móvil. | 2/3/22 | 3/3/22 |
| ● | Herramientas definidas | 4/3/22 | 4/3/22 |
| ▣ | 2 Implementar servidor. | 4/3/22 | 8/3/22 |
| ● | 2.1 Instalación y configuración Node.js. | 4/3/22 | 4/3/22 |
| ● | 2.2 Instalación broker MQTT. | 5/3/22 | 5/3/22 |
| ● | 2.3 Programación de funciones de consulta para aplicación web/móvil. | 7/3/22 | 8/3/22 |
| ▣ | 3 Implementar base de datos. | 9/3/22 | 12/4/22 |
| ● | 3.1 Instalación base de datos. | 9/3/22 | 19/3/22 |
| ● | 3.2 Configuración de parámetros de seguridad y datos. | 21/3/22 | 30/3/22 |
| ● | 3.3 Creación y carga de datos en tabla. | 31/3/22 | 12/4/22 |
| ● | Plataforma preparada para desarrollar aplicaciones | 13/4/22 | 13/4/22 |
| ▣ | 4 Implementación de página web. | 13/4/22 | 3/6/22 |
| ● | 4.1 Programación de funciones MQTT como cliente del broker. | 13/4/22 | 23/4/22 |
| ● | 4.2 Prueba de envío y recepción de datos al broker. | 25/4/22 | 26/4/22 |
| ● | 4.3 Programación de consultas a base de dato. | 27/4/22 | 7/5/22 |
| ● | 4.4 Programación de carga de datos en la base de datos. | 9/5/22 | 19/5/22 |
| ● | 4.5 Programación de logica de acceso y seguridad de usuarios. | 20/5/22 | 1/6/22 |
| ● | 4.6 Prueba de acceso de usuarios. | 2/6/22 | 3/6/22 |
| ● | Finalización pagina web configuración | 4/6/22 | 4/6/22 |
| ▣ | 5 Implementación de aplicación enfermera. | 4/6/22 | 14/7/22 |
| ● | 5.1 Programación de interfaz de login de usuarios. | 4/6/22 | 6/6/22 |
| ● | 5.2 Testeo de lógica y acceso de seguridad de usuarios. | 7/6/22 | 7/6/22 |
| ● | 5.3 Programación de interfaces de usuario de aplicación móvil. | 8/6/22 | 16/6/22 |
| ● | 5.4 Programación de funciones de acceso a la base de datos. | 17/6/22 | 28/6/22 |
| ● | 5.6 Programación y prueba de lectura de códigos QR. | 29/6/22 | 29/6/22 |
| ● | 5.5 Prueba de envío y recepción de datos entre clientes modo enfermera y cliente modo médico. | 30/6/22 | 1/7/22 |
| ● | 5.7 Programación de captura/compresión/envío de audio modo enfermera. | 2/7/22 | 12/7/22 |
| ● | 5.8 Prueba de envio de distintos audios. | 13/7/22 | 14/7/22 |
| ● | Finalización primer prototipo aplicación enfermera | 15/7/22 | 15/7/22 |
| ▣ | 6 Pruebas de sistema. | 15/7/22 | 16/7/22 |
| ● | 6.1 Prueba de funcionamiento correcto con 2 enfermera 2 médico y 3 llamadores. | 15/7/22 | 16/7/22 |
| ▣ | 7 Memoria técnica del proyecto | 18/7/22 | 28/7/22 |
| ● | 7.1 Redacción informe de avance | 18/7/22 | 28/7/22 |
| ● | 7.2 Redacción de la memoria técnica del proyecto | 18/7/22 | 28/7/22 |

Figura 4. Planificación de tareas.

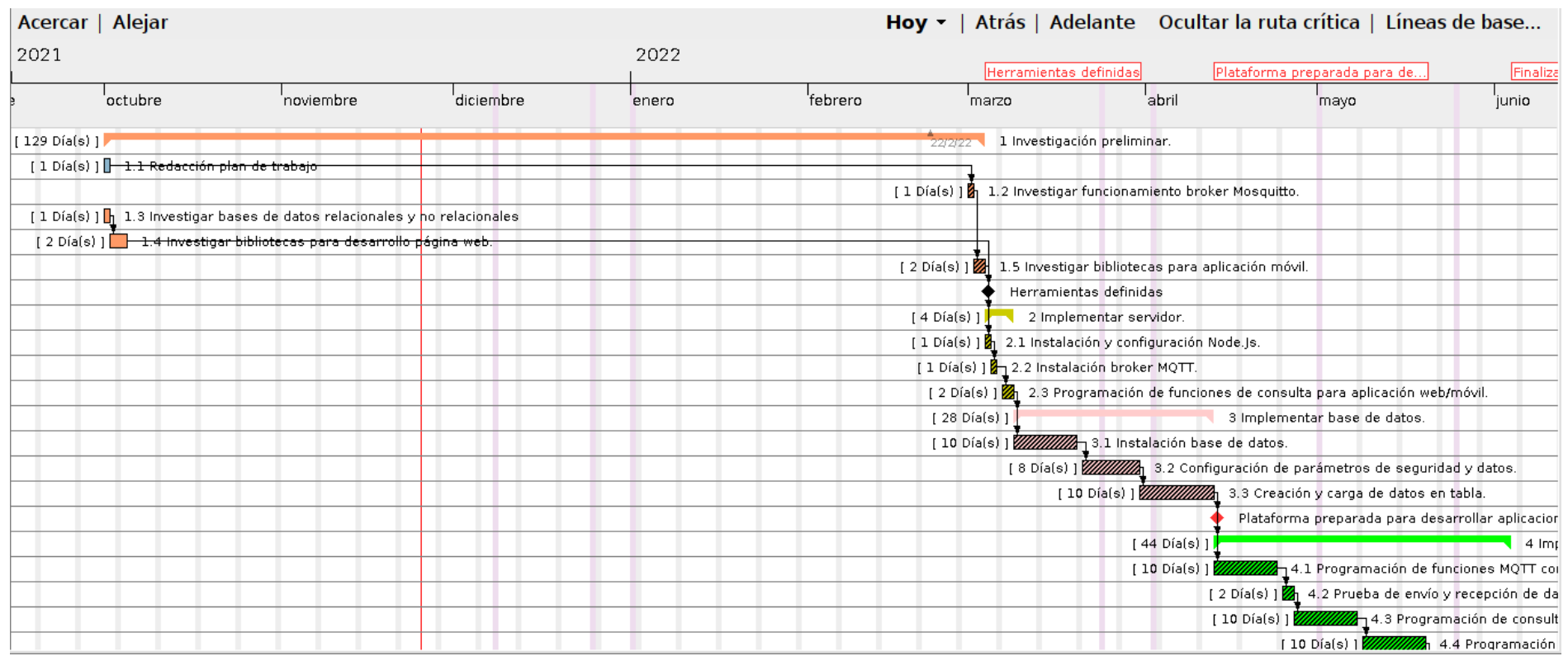


Figura 5. Diagrama de Gantt (continua en la página siguiente).

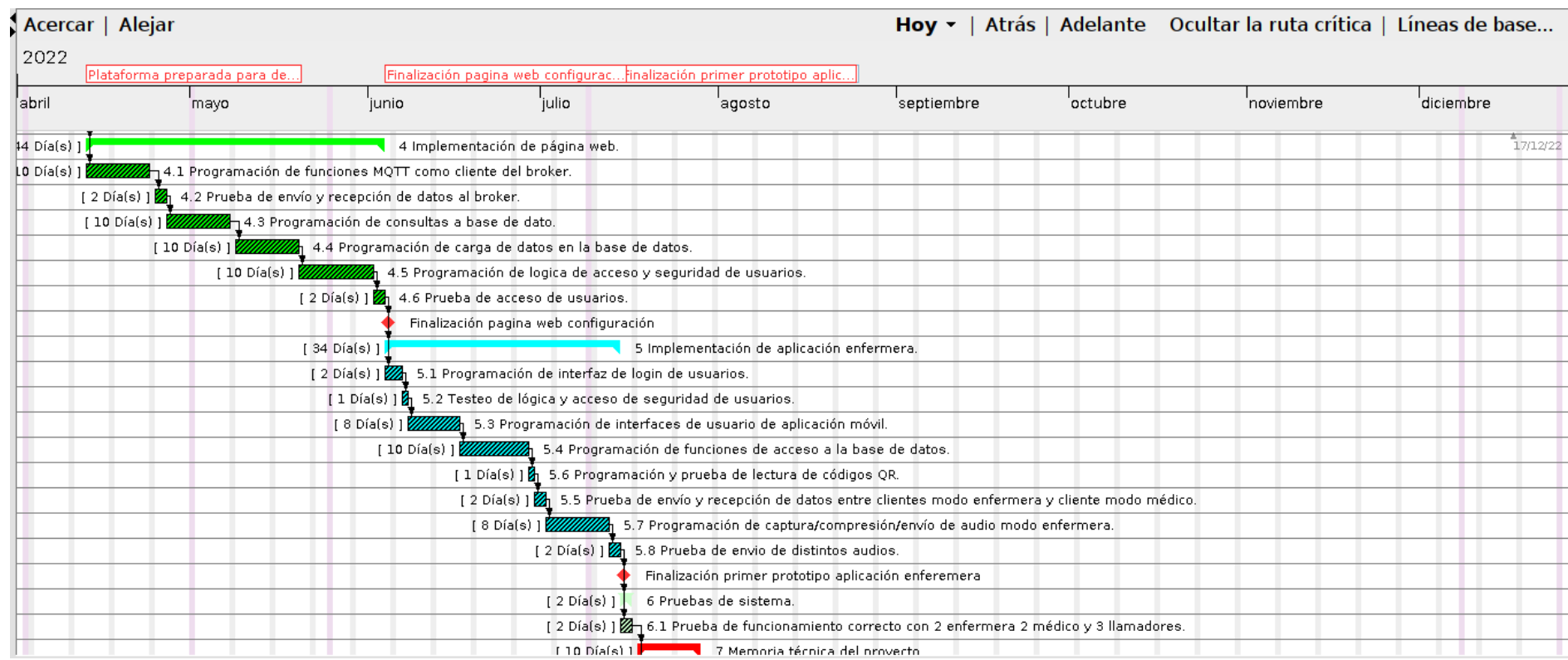


Figura 6. Diagrama de Gantt (continua en la página siguiente).

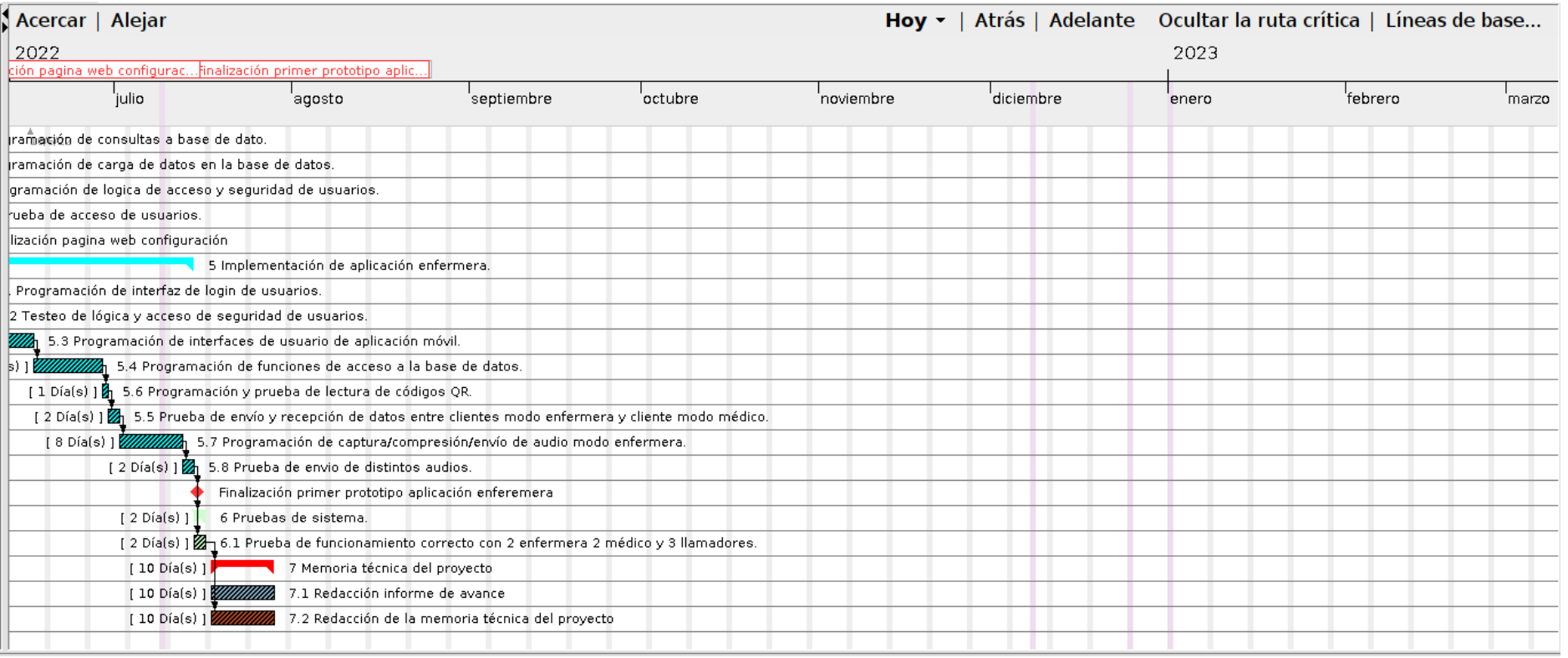


Figura 7. Diagrama de Gantt.

12. Presupuesto detallado del proyecto

En la siguiente tabla se presentan los gastos previstos del proyecto, expresados en pesos argentinos.

| COSTOS DIRECTOS | | | |
|-----------------------------|----------|--------------------|-----------------|
| Descripción | Cantidad | Valor unitario(\$) | Valor total(\$) |
| Investigación | 70hs | 1000 | 70000 |
| Programación | 424hs | 1000 | 424000 |
| Testing | 66hs | 1000 | 66000 |
| Documentación | 60hs | 1000 | 60000 |
| SUBTOTAL | | | 620000 |
| COSTOS INDIRECTOS | | | |
| Descripción | Cantidad | Valor unitario(\$) | Valor total(\$) |
| Raspberry pi 4(Server) | 1 | 27000 | 27000 |
| Costos de Consumo eléctrico | 600 | 4 | 2400 |
| SUBTOTAL | | | 29400 |
| TOTAL | | | 649400 |

13. Gestión de riesgos

Riesgo 1: no contar con un dispositivo de los necesarios para el testeo de las aplicaciones.

- Severidad (S): 10.
Justificación: al ser la aplicación multiplataforma, es menester contar con un dispositivo Android, un dispositivo IOS y una pc con navegador web.
- Probabilidad de ocurrencia (O): 5.
Justificación: en este momento no se posee un dispositivo IOS pero existen posibilidades de acceder a uno.

Riesgo 2: disminución de tiempo disponible para la realización del plan trazado.

- Severidad (S): 7.
Justificación: el disminuir el tiempo diario asignado a cada tarea, retrasa todo el calendario.
- Ocurrencia (O):3.
Justificación: la planificación es correcta y es muy poco probable que surjan actividades adicionales que ocupen al recurso humano.

Riesgo 3: daño en equipamiento utilizado en el proyecto por problemas en el suministro eléctrico.

- Severidad (S): 10.
Justificación: genera retrasos temporales y modificaciones al presupuesto ya que se deben adquirir nuevos recursos materiales, instalar el framework de desarrollo y acondicionarlo para continuar el desarrollo.

- Ocurrencia (O): 3.

Justificación: en estos momentos el equipamiento se encuentra protegido mediante un estabilizador de tensión.

Riesgo 4: no contar con colaboradores para realizar la validación del sistema.

- Severidad (S): 5.

Justificación: es necesario que los colaboradores sean idóneos.

- Ocurrencia (O): 8.

Justificación: debido a que los colaboradores poseen otras tareas en su trabajo puede que no puedan dedicarle el tiempo necesario a las pruebas.

Riesgo 5: surgimiento de una alternativa con mejores prestaciones a la sugerida en el plan de trabajo (nueva librería, framework, etc).

- Severidad (S): 1.

Justificación: el surgimiento de una nueva alternativa con mejores prestaciones a la elegida para el desarrollo puede dejar al producto en una posición de desventaja respecto a la competencia. No obstante, el utilizar tecnologías ya probadas asegura un nivel de confianza mayor en cuanto a seguridad y estabilidad a lo largo del tiempo.

- Ocurrencia (O): 10.

Justificación: en el rubro informático continuamente surgen nuevas alternativas para el desarrollo.

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

| Riesgo | S | O | RPN | S* | O* | RPN* |
|---|----|----|-----|----|----|------|
| 1.No contar con un dispositivo de los necesario para el testeado de las aplicaciones. | 10 | 5 | 50 | 5 | 1 | 5 |
| 2.Disminución de tiempo disponible para la realización del plan trazado | 7 | 3 | 21 | | | |
| 3.Daño en equipamiento utilizado en el proyecto por problemas en el suministro eléctrico | 10 | 3 | 30 | 1 | 1 | 1 |
| 4.No contar con colaboradores para realizar la validación del sistema. | 5 | 8 | 40 | 5 | 2 | 10 |
| 5.Surgimiento de una alternativa con mejores prestaciones a la sugerida en el plan de trabajo | 1 | 10 | 10 | | | |

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.

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

Riesgo 1: se buscarán contactos conocidos que posean un dispositivo de esas características.

-Severidad(S) : 5.

Justificación: impide la validación del sistema en una plataforma.

-Ocurrencia: 1.

Justificación: es muy improbable que al momento de la validación no se encuentre el recurso.

Riesgo 3: se incorporará un sistema de alimentación ininterrumpible (UPS, Uninterruptable Power Supply) y se mantendrá todo progreso del trabajo en servidores en la nube.

-Severidad(S) : 1.

Justificación: al tener un equipo de backup y todo lo trabajado en la nube, no existe posibilidad de demoras por este motivo.

-Ocurrencia: 1.

Justificación: al incorporar la UPS, los daños a los discos rígidos se anulan.

Riesgo 4: se gestiona una lista de contactos que se encargarán de validar el sistema y se genera un compromiso con los mismos.

-Severidad(S) : 5.

Justificación: no se modifica la severidad del mismo.

-Ocurrencia: 2.

Justificación: al haber varios contactos, la imposibilidad simultánea de todos es improbable.

14. Gestión de la calidad

- Req 1.1: el servidor debe tener instalado Eclipse Mosquitto como broker MQTT.
 - Verificación: se verificará que el servidor se encuentre listo para recibir conexiones y se generarán envíos a través de un software de envíos de paquetes TCP.
 - Validación: no aplica.
- Req 1.2: el servidor debe tener endpoints para interactuar con una aplicación Web, para consultas y envío de datos hacia dispositivos clientes.
 - Verificación: se verificará que el servidor responda correctamente a solicitudes desde dispositivos.
 - Validación: se generarán peticiones según las aplicaciones programadas y se analizarán las respuestas (se entregará informe al cliente).
- Req 1.3: el servidor debe tener endpoints para interactuar con la base de datos.
 - Verificación: se verificará la presencia de las funciones CRUD a la base de datos.
 - Validación: se validará que las funciones respondan a las aplicaciones correctamente y se entregará informe al cliente.
- Req 2.1: el sistema debe poseer una base de datos relacional.
 - Verificación: se verificará la presencia de la base de datos.
 - Validación: se cargarán tablas en las bases de datos y se validarán las funciones CRUD y se entregará informe al cliente.
- Req 2.2: la base de datos debe poseer datos cargados por default.
 - Verificación: se chequeará que la base de datos contenga datos.
 - Validación: se chequeará que los datos sean correctos.

- Req 2.3: la base de datos debe poseer datos cargados por default.
 - Verificación: se chequeará que la base de datos contenga datos.
 - Validación: se chequeará que los datos sean correctos.
- Req 2.4: la base de datos a utilizar debe poseer las siguientes tablas:
 - Habitaciones
 - Pacientes
 - Médicos
 - Enfermeras
 - Eventos
 - Verificación: se chequeará la estructura necesaria.
 - Validación: se chequeará que se puedan cargar los datos en la estructura y se entregará el diagrama de base de datos y datos ejemplo al cliente.
- Req 3.1: la página web debe ser cliente del broker MQTT.
 - Verificación: se verificará que la pagina sea cliente del servidor.
 - Validación: se confirmará que el broker es MQTT y no otro. _____
_____!!!
- Req 3.2: la página web debe poseer funciones de consulta o modificación de la base de datos.
 - Verificación: se verificará que desde la página se puedan cargar o modificar datos en la base de datos.
 - Validación: se validarán que desde la página se puedan cargar o modificar datos en la base de datos.
- Req 3.3: la página web debe ser permitir visualización de estadísticas de pacientes o enfermeras.
 - Verificación: se verificará que la aplicación pueda mostrar el número de visitas a un paciente en cierto día, y la cantidad de llamadas a las que acudió cierta enfermera en un día.
 - Validación: se validará que es el funcionamiento deseado por el cliente.
- Req 3.4: la aplicación debe contener acceso con usuario y contraseña para cada persona.
 - Verificación: se comprobará el funcionamiento con distintos usuarios, contraseñas y modos de uso.
 - Validación: se presenta al cliente la forma de funcionamiento y el mismo evalúa si es de su agrado.
- Req 4.1: la aplicación enfermera debe tener 3 modos de uso: usuario médico, usuario enfermera y usuario sistema.
 - Verificación: se comprobará que se puede logear con los distintos modos.
 - Validación: se presentará al cliente los distintos modos de funcionamiento y se consultará si es del agrado.

- Req 4.2: al logearse el usuario , y contrastarse con la base de datos del sistema, se activa el modo correspondiente.
 - Verificación: se comprobará que se puede logear con los distintos modos y que hay una comunicación efectiva con la base de datos.
 - Validación: se consultará al cliente si es del agrado..
- Req 4.3: la aplicación en modo enfermera debe permitir leer códigos QR.
 - Verificación: se comprobará que se puede escanear el código QR de la cama del paciente cuando la aplicación está logeada en modo enfermera, se comunica con el sistema y se accionan las actividades correspondientes.
 - Validación: se presenta el funcionamiento al cliente y se consulta si es del agrado.
- Req 4.4: la aplicación en modo enfermera debe poder descargar información relevante del paciente (medicamentos a suministrar, controles a realizar, contacto del médico a cargo del tratamiento, etc).
 - Verificación: se verifica que se pueda descargar de la base de datos la información necesaria al presentarse el código qr.
 - Validación: se consulta al cliente si es del agrado. Puede ser afectado por el tipo de dispositivo(tamaño de visualización).
- Req 4.5: la aplicación en modo sistema debe mostrar las habitaciones sin atención, según una tabla de prioridades y en caso de igualdad de prioridades, mostrár según el orden de llamada.
 - Verificación: se loguea en la aplicación como usuario sistema y se verifica la visualización correcta.
 - Validación: se presenta al cliente y se consulta si es de su agrado.
- Req 4.6: el modo usuario médico y el modo usuario enfermera deben poder enviar mensajes de textos o sonido comprimido en 32kbps (mp3).
 - Verificación: se hace una demostración de envío de audio entre dos usuarios enfermera y médico.
 - Validación: se muestra al cliente y se consulta si es de su agrado.
- Req 5.1: confección de documento con información relativa a la base de datos: detalles de la misma y de API para acceder.
 - Verificación: se verifica la presencia de la documentación.
 - Validación: se presenta al cliente.
- Req 5.2: memoria del proyecto con diagramas UML de aplicación modo sistema, aplicación modo enfermera, aplicación modo médico y página web.
 - Verificación: se verifica la presencia de la documentación.
 - Validación: se presenta al cliente.
- Req 6.2: el sistema debe integrar el funcionamiento del servidor con base de datos , aplicación web y aplicaciones móviles.
 - Verificación: se realiza un video con una demostración del sistema.
 - Validación: se presenta al cliente el video de la funcionalidad. Se genera una simulación real con colaboradores.

15. Procesos de cierre

Al finalizar el proyecto se realizará una evaluación final que contemple las siguientes actividades:

- El responsable del proyecto analizará el cumplimiento del plan del proyecto con sus tiempos indicados y se evaluará los requerimientos no cumplidos.
- Se evaluarán los riesgos previstos que efectivamente se hayan manifestado, y el procedimiento de mitigación planificado versus el real. Se evaluarán demoras causadas por eventos no previstos.
- El responsable del proyecto realiza una presentación pública del resultado del proyecto así como del historial de su desarrollo.