

## Introducción

Para el desarrollo del proyecto, primeramente, se definirá cuál es su contexto inicial del problema tomando como centro la situación del cliente, en base a esto luego se utilizarán diferentes herramientas tales como encuestas, entrevistas, observación, tormenta de ideas e historias de usuario orientadas a tanto diferentes tipos de usuarios como a un conjunto de estos para así realizar un adecuado levantamiento de requerimientos abarcando así todas las necesidades presentes. Una vez teniendo esto, lo siguiente a realizar es la planificación de Sprint, lo cual será de vital importancia, puesto que definirá los tiempos y personal asignados para la realización de estos, generando así entregables que en consecuencia darán firmeza al proyecto a los ojos del cliente. Para continuar se crearán los diagramas UML de acuerdo con el paradigma 4 + 1 en donde se encontrarán el diagrama de casos de uso, el diagrama de secuencia, el diagrama de actividades, el diagrama de componentes, el diagrama de despliegue y por último el diagrama de clases. Para así dar fin con el prototipado de la interfaz de usuario enfocado al primer sprint, el cual corresponde al inicio de sesión.

## Objetivos

- Diseñar un sistema de soporte que permita la correcta gestión de tickets de una empresa de telefonía
- Efectuar levantamiento de los requerimientos
- Definir los instrumentos a utilizar para la establecer la prioridad, tareas y tiempo de cada historia de usuario en base a los requerimientos establecidos
- Generar los Sprint en base a las historias de usuario
- Diseñar la vista lógica, la vista de implementación, la vista de despliegue y la vista de procesos considerando la aplicación del paradigma 4+1
- Diseñar la interfaz gráfica de la aplicación considerando las necesidades del cliente y que esta posea una interfaz moderna y acorde a los estándares establecidos.

## **Definición del proyecto a resolver.**

Mesa de Ayuda con sistema de Tickets para empresa de Telefonía

La empresa de telefonía “Telecom” se encuentra en la necesidad de llevar la creación de los tiques de atención de su área de soporte a una plataforma que todos los usuarios puedan usar de manera más sencilla (una aplicación web).

Actualmente el proceso se ha realizado a través de atención telefónica, correo y gestionado en planillas Excel siendo el proceso de atención y tiempo de respuesta demasiado lento para brindar una correcta atención debido a la dificultad de organización que esto representa.

De las funcionalidades esperadas a ser desarrolladas en la aplicación web, se detallan las siguientes: 1. Creación de tiques a partir del cliente que lo necesite.

2. Los tiques creados a partir del cliente deberán tener las siguientes características (Aunque algunos datos como el Nombre del cliente, el rut, los datos de contacto y el modelo del dispositivo deben ir en la descripción.)

a. Nombre del cliente

b. RUT del cliente

c. Datos de contacto del cliente:

i. Teléfono

ii. Correo electrónico

e. Modelo del dispositivo

f. Tipo de tiques (consulta, reclamo, problema)

g. Descripción

h. Archivo adjunto (Imágenes-boletas)

3. Cada uno de los tiques creados podrá ser tomado por el técnico de área que corresponda 4. Una vez tomado el tique, el técnico deberá asignar los pasos a seguir, quedando el tique en estado: “Abierto” y Asignado al ejecutivo de área correspondiente.

5. Un técnico de área podrá ver el tique y cambiar el estado a “Cerrado” una vez solucionado el problema.

6. El técnico de área específica deberá agregar una observación y proceder a cerrar el tique, según el estado detallado en el punto anterior.

7. Se deberá conservar el tique y quienes lo cerraron como un historial para su posterior revisión por el supervisor (Administrador).

8. El Ejecutivo, Técnico de área, Supervisor deberá poder obtener el listado de tiques creados filtrando por:

Evaluación N°1 Levantamiento de Requerimientos y Diseño de Interfaces 4

a. Número de Ticket

b. Por categoría (consulta, reclamo, problema)

c. Por título

d. Por prioridad

f. Estado

g. Fecha Creación

h Fecha Asignación.

i. Fecha de Cierre

j. Soporte

9. El listado de tickets generados deberá ser presentados en una tabla donde se detalle: a. Número de Ticket

b. Por categoría (consulta, reclamo, problema)

c. Por título

d. Por prioridad

f. Estado

g. Fecha Creación

h Fecha Asignación.

i. Fecha de Cierre

j. Soporte

10. Todos los roles especificados (Ejecutivo, Técnico de área, Supervisor) podrán ingresar al sistema, se deberá considerar un usuario y contraseña para validar el ingreso. 11. No existen usuarios de múltiples perfiles, es decir, no existen usuarios que son Ejecutivo y Supervisor a la vez.

## **Herramientas por utilizar para la toma de Requerimientos**

### **Herramienta N°1 Encuesta**

Esta herramienta estará enfocada en los ejecutivos y supervisores, puesto que estos son los roles los cuales más uso darán del proyecto una vez finalizado.

¿Cómo calificaría el servicio de la empresa?

¿Por qué la calificación?

¿Qué dificultad ha encontrado a la hora de utilizar los servicios de la empresa?

¿Deberían tener un nuevo sistema de gestión o seguir con el mismo método?

¿Le gustaría una plataforma online?

¿Qué le gustaría que implementara la plataforma?

¿Por qué?

### **Herramienta N°2 Entrevistas**

Además, se trabajará con la herramienta de entrevistas a por lo menos un integrante de cada rol para así dejar en evidencia los problemas y las necesidades que se presentan por cada puesto y así poder entender mejor la situación que se quiere solucionar.

1. ¿Cuánto tiempo lleva trabajando en la empresa?

2. ¿En qué área se desempeña?

3. ¿Utilizan algún programa (computacional, software) en su área de trabajo?

4. ¿Cuáles? (según respuesta 3)

5. ¿Fue capacitado para el uso de estos programas o era requisito para el trabajo? 6.

¿Cuánto tiempo duró la capacitación, fue evaluada?

7. ¿Cuál sería lo más difícil de hacer en su trabajo actual que le dé más problemas? 8.

¿Qué le gustaría que se implementara en caso de una nueva plataforma para su trabajo? 9.

¿Cómo le gustaría que fuera el diseño de esta nueva aplicación?

10. ¿Prefiere una interfaz moderna y minimalista o una tradicional?

### **Herramienta N°3 Observación**

Ocupamos la herramienta de levantamientos de requerimientos fue la observación (de manera pasiva sin intervenir en su trabajo) observamos su entorno y cómo trabajaban normalmente observando sus dificultades y diferentes requerimientos en su trabajo los cuales se denotará la necesidad de una optimización.

1. La creación de ticket se lleva a cabo en Excel manualmente lo cual toma bastante tiempo y es ineficiente.
2. Puesto que se debe utilizar una plantilla por cada ticket, lo hace tedioso de procesar
3. Para la búsqueda de datos en las plantillas, no cuenta con sistema de búsqueda por lo que la búsqueda de ticket se realiza manualmente por fechas, los tipos de ticket y servicio que lo recibirá
4. Además se tiene escasez de información de parte de usuario solicitante, lo cual produce complicaciones al ejecutivo, ya que deberá tomar tiempo en analizar en profundidad para así hacer la derivación correspondiente.
5. Poco amigable con el usuario, es decir no se utiliza el concepto de usabilidad.
6. Los tickets no están compartidos en tiempo real por lo que puede producirse una duplicación del ticket. 7. Los datos no son respaldados de forma segura por lo que es propenso a pérdida de información.

### **Herramienta N°4 Tormenta de ideas**

Luego de haber implementado la encuesta, cuestionario, entrevistas e historias de usuario para el levantamiento de requerimientos, realizamos una mesa de trabajo en la cual también llevamos a cabo una tormenta de ideas como equipo y con parte del personal para centralizar las características más solicitadas y calificarlas según el criterio de aceptación para que sea solo lo necesario y cumpla con el objetivo.

## Requerimientos Funcionales

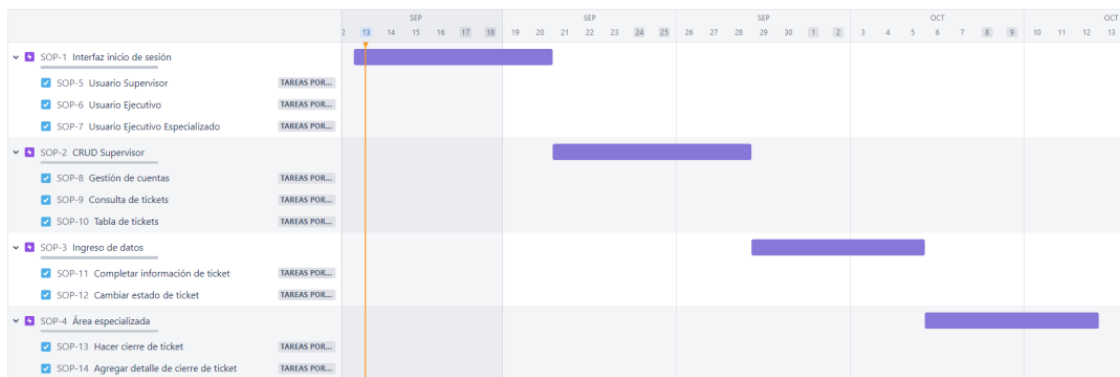
RECOPILACIÓN DE HISTORIAS DE USUARIO				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	Condiciones
RF1	Como supervisor/ejecutivo/técnico de área	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de ingresar al sistema a través de un login	Dicha cuenta no puede tener mas de un perfil asignado (ejecutivo, técnico de área o supervisor)
RF2	Como Supervisor	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de realizar gestiones referente a personal	Se debe de ingresar con usuario (rut ej:xxxxxx-x) y contraseña.
RF3	Como Supervisor	Necesito poder gestionar y modificar los roles de cada usuario	Con la finalidad de poder gestionar cuentas de demás supervisores, técnicos y ejecutivos	Podrá además editar áreas, tipos de ticket, estados. No podrá modificar los tickets si ya existen tickets con especificaciones antiguas
RF4	Como supervisor/ejecutivo/técnico de área	Necesito poder obtener listado de ticket	Con la finalidad de obtener un historial	Puede ser filtrado por número de Ticket, por categoría, por título, por prioridad, estado, fecha de creación, fecha de asignación, fecha de cierre, ejecutivo de área
RF5	Como supervisor/ejecutivo/técnico de área	Necesito poder visualizar los tickets que ya fueron solucionados	Con la finalidad de obtener información respecto al proceso que se llevo a cabo	No se puede editar un ticket ya cerrado.
RF6	Como ejecutivo	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de realizar gestiones referente a tickets	Se debe de ingresar con usuario (rut ej:xxxxxx-x) y contraseña.
RF7	Como ejecutivo	Necesito poder crear tickets	Con la finalidad de realizar gestiones correspondientes para los problemas de los usuarios	Se debe poder ingresar información como Título, Categoría, Subcategoría, Prioridad, Adjuntar documentos adicionales (fotos-archivos de texto)
RF8	Como ejecutivo	Necesito poder asignar un ticket creado	Con la finalidad de realizar gestiones para los problemas de los usuarios	Cada ticket creado podrá ser asignado al técnico de área correspondiente visualizándose así en la tabla al ejecutivo al cual se le asigno
RF9	Como ejecutivo	Necesito poder cambiar el estado	Con la finalidad de realizar gestiones para los problemas de los usuarios	El ejecutivo podrá cambiar el estado "Abierto" o "Cerrado" dependiendo de la gestión que se este llevando a cabo
RF10	Como técnico de área	Necesito poder agregar una solución y cambiar estado cuando el problema este resuelto	Con la finalidad de realizar gestiones para los problemas de los usuarios	El técnico de área podrá dar cierre del ticket junto con una observación
RF11	Como supervisor	Necesito poder crear nuevos ejecutivos/técnico de área/supervisores	Con la finalidad de administrar el sistema	Se deberá dar una opción para así administrar los permisos de cada rol
RF12	Como supervisor	Necesito que al eliminar un usuario no se eliminen los tickets creados	Con la finalidad de llevar un registro de todo el sistema	Se deberá solo poder desactivar el acceso al sistema
RF13	Como ejecutivo	Necesito hacer ingreso de una descripción del problema	Con la finalidad de ingresar datos correspondientes y relevantes para una correcta solución	El nombre, los datos de contacto, el modelo del dispositivo y el problema deben ir en la descripción del ticket.
RF14	Como ejecutivo	Necesito hacer ingreso del tipo de tique	Con la finalidad de establecer un orden	Se debe ingresar un tipo de tique de manera obligatoria
RF15	Como ejecutivo	Necesito hacer ingreso del área a derivar	Con la finalidad de establecer un orden y una correcta derivación	El área a derivar puede ser ("Hardware" o "Software")

## Requerimientos No Funcionales

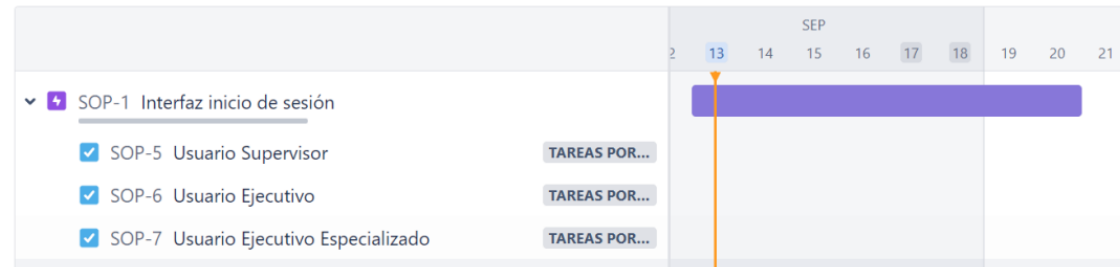
RECOPIACION DE HISTORIAS DE USUARIO				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	Criterios de Aceptación
RFN1	Como tecnico de area/ejecutivo/supervisor	La interfaz grafica debe ser de facil comprensión (Intuitiva y minimalista)	Mejorar la usabilidad y accesibilidad del Sistema	Agregar animaciones al momento de generar y cerrar tickets
RFN2	Como tecnico de area/ejecutivo/supervisor	Necesito poder contar con un apartado donde se indique como funciona el sistema	Para resolver dudas de una manera rapida	Implementar una pagina de ayuda exclusiva para visualizar como funciona el sistema. Debe contar con tutoriales de video
RFN3	Como tecnico de area/ejecutivo/supervisor	Necesito que los colores sean acordes a la imagen de la empresa	Para que aquella persona que este haciendo uso del sistema se sienta parte de la empresa	Los colores deben ser acordes a la empresa
RFN4	Como tecnico de area/ejecutivo/supervisor	El Sistema debe soportar el manejo de gran cantidad de informacion	Mejorar el rendimiento y la utilización del sistema	Optimizar el funcionamiento de la pagina
RFN5	Como tecnico de area/ejecutivo/supervisor	El sistema debe permitir controlar que solo usuarios registrados ingresen al sistema	Mantener la seguridad de la informacion	No puede acceder un usuario que no se haya registrado en el sistema
RFN6	Como tecnico de area/ejecutivo/supervisor	Que el ingreso al sistema sea de facil acceso	Para así garantizar una atencion mucho mas rapida	El acceso al sistema debe estar claramente definido al ingresar a una pagina

## Definición de Sprint

A continuación, se presentan los sprints creados a partir de los levantamientos de requerimientos.

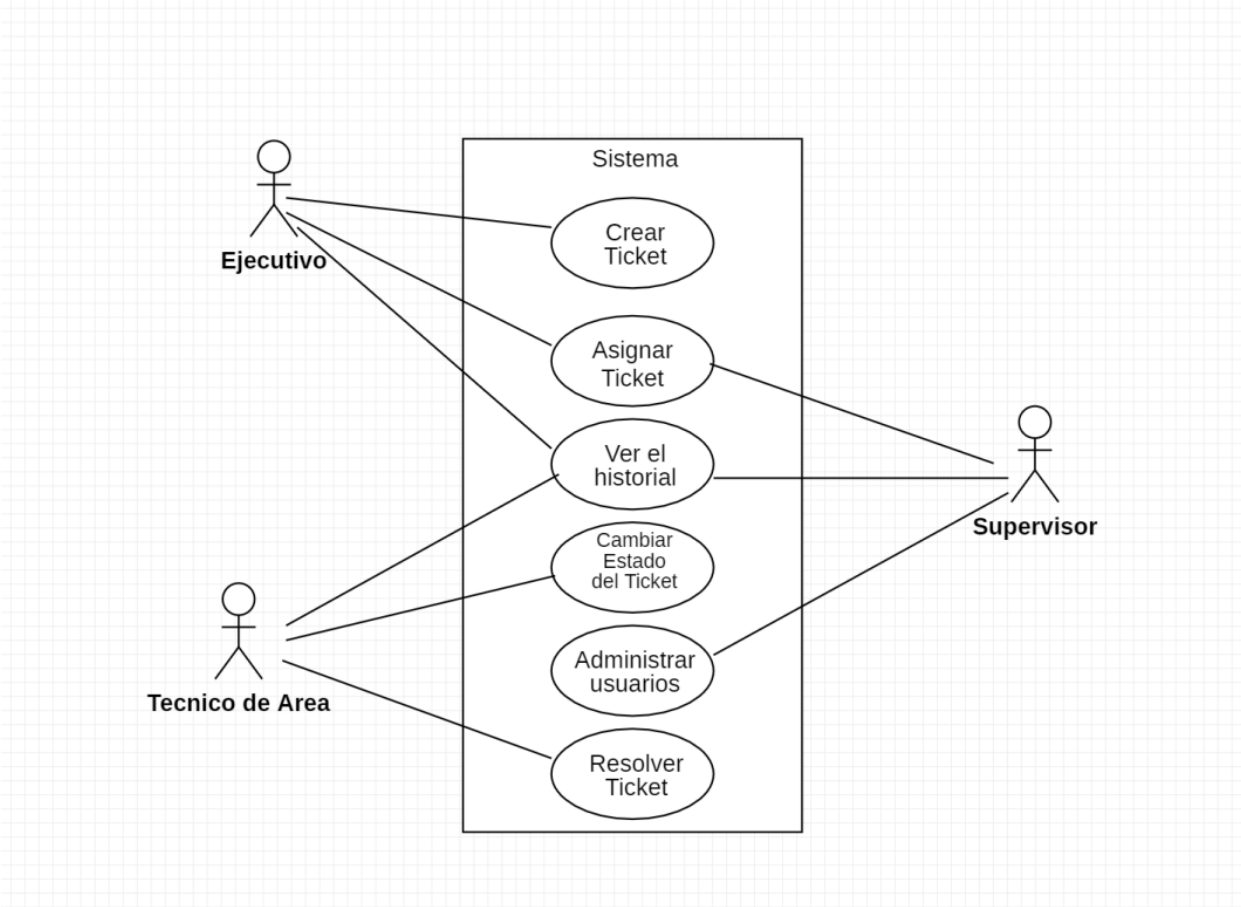


Haciendo especial énfasis en el primero de este.



Diseño de diagramas UML según las distintas vistas de acuerdo con el paradigma 4 + 1.

Diagrama de Casos de Uso





## Diagrama de Clases

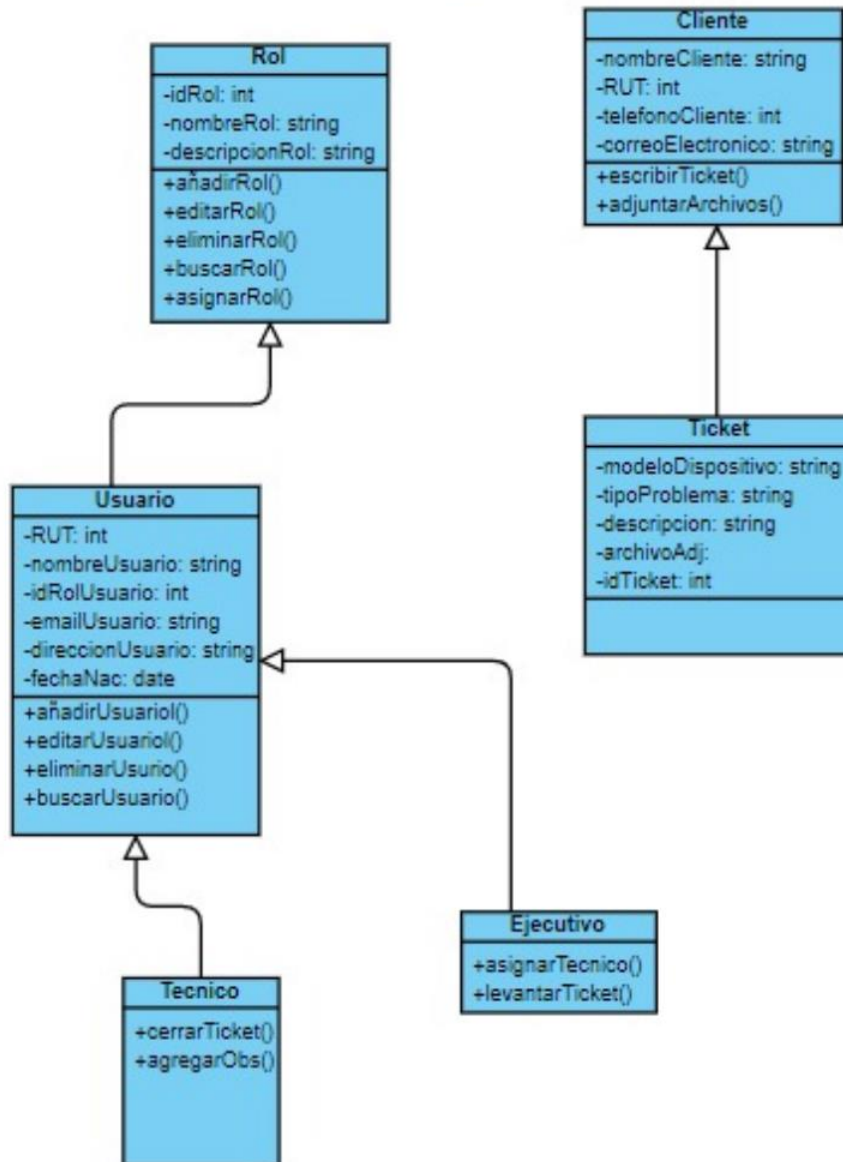
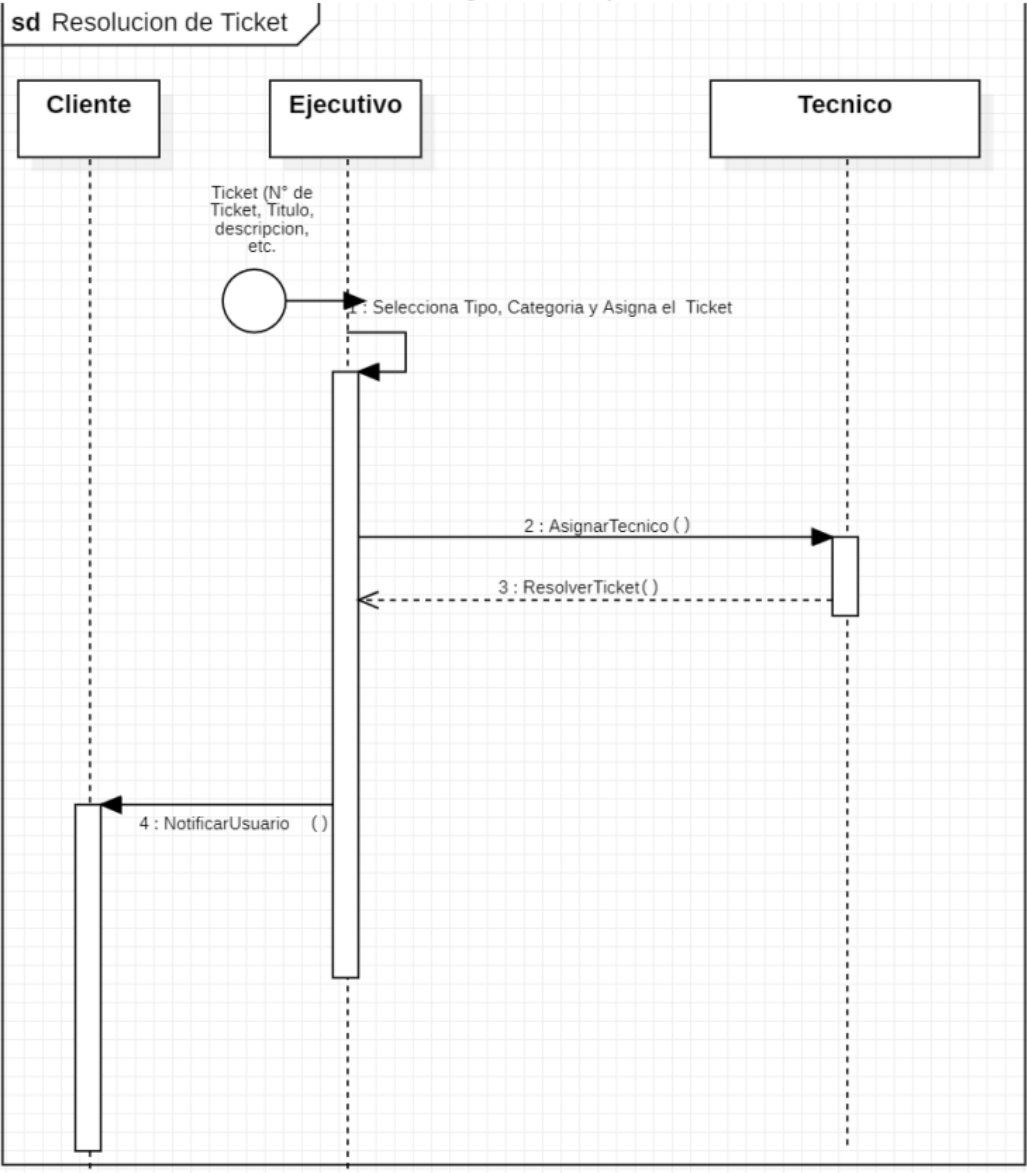
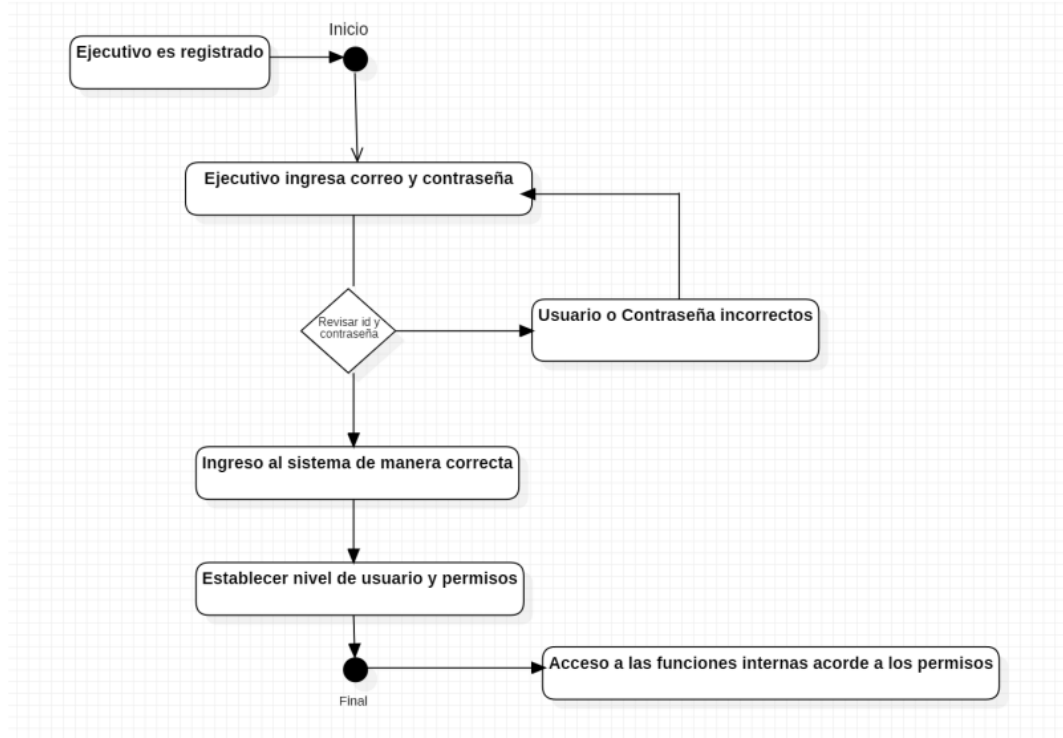


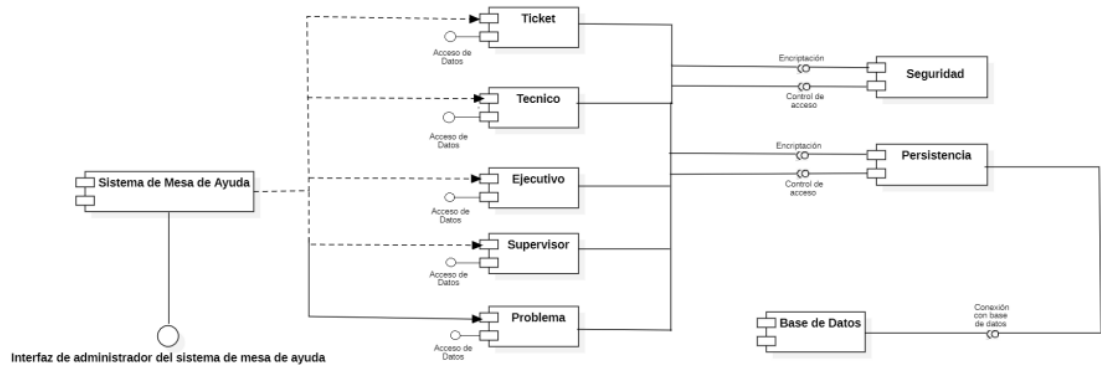
Diagrama de Secuencia



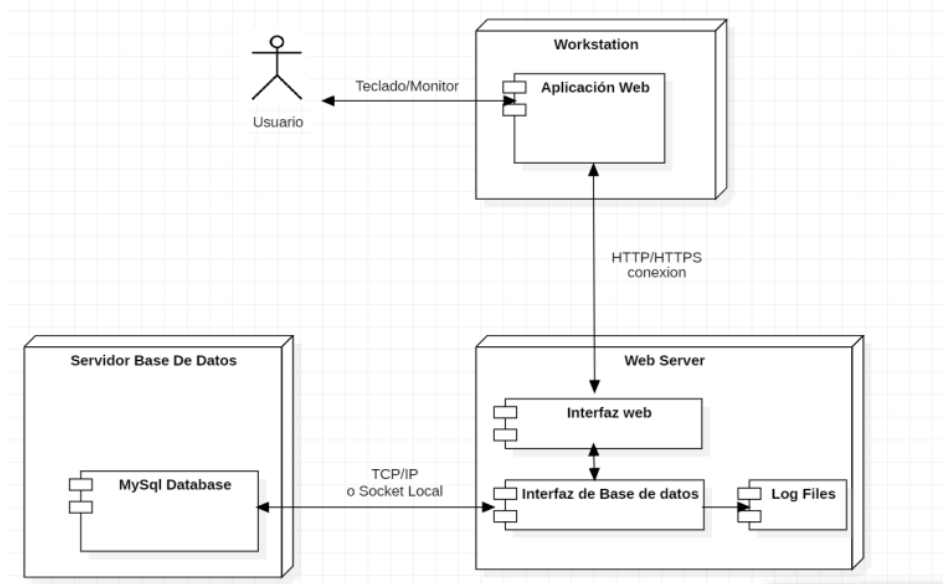
### Diagrama de Actividades




## Diagrama de Componentes



## Diagrama de Despliegue



## Interfaces de Usuario




### Acceso Ejecutivo/Supervisor

Email

Contraseña

Acceder

[Acceso Tecnico](#) [Cambiar Contraseña](#)



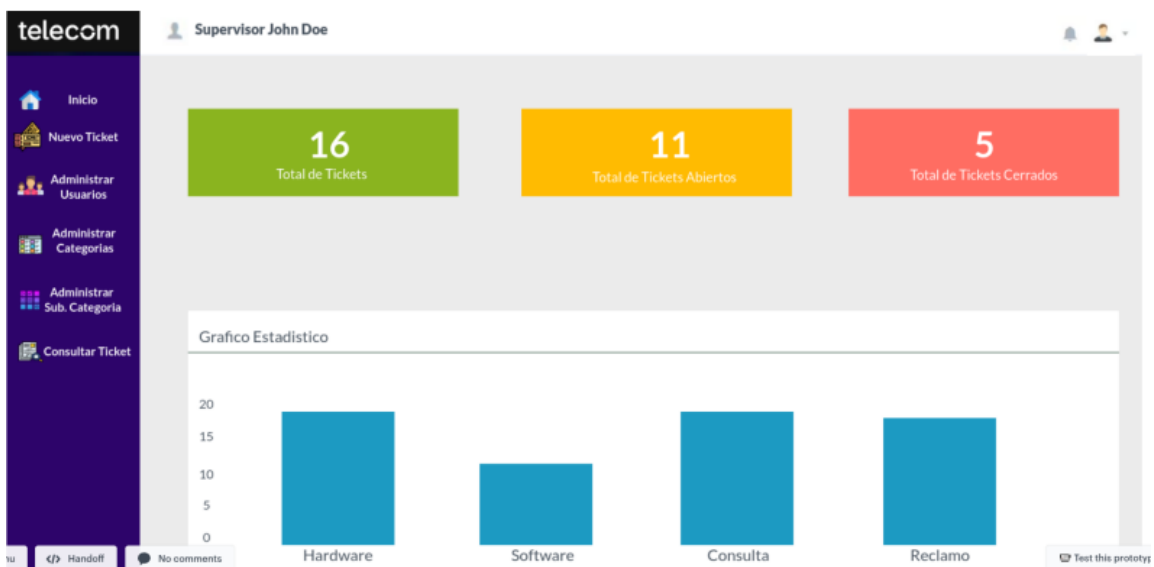
### Acceso Tecnico

Email

Contraseña

Acceder

[Acceso Ejecutivo](#) [Cambiar Contraseña](#)



## **Patrones de Arquitectura de la solución**

Debido a la creciente complejidad del software actual, las tareas asociadas de desarrollo, mantenimiento y mejora se han vuelto cada vez más tediosas y lentas. Como resultado, se han creado patrones de arquitectura de software para aliviar la carga de trabajo de los arquitectos de software, desarrolladores y operadores de este.

Los patrones de arquitectura son una solución reutilizable a un problema común dentro del campo del diseño de software. Los patrones de arquitectura de software son una descripción o plantilla para la organización estructural o el esquema de cómo resolver un problema que se puede utilizar en muchas situaciones diferentes. Puede pensar en esto como mejores prácticas formalizadas que los proyectos pueden usar para resolver problemas comunes cuando se diseña un producto de software. Los problemas recurrentes a los que se ha enfrentado la industria han dado como resultado en la creación de patrones de arquitectura comunes. Esto se debe a que estos patrones se pueden reutilizar para resolver problemas futuros similares dentro de diferentes dominios. Algunos ejemplos de cómo se han utilizado los patrones de arquitectura de software para resolver problemas comunes incluyen:

- Las solicitudes largas y complicadas de los usuarios pueden dividirse en partes más pequeñas y distribuirse a lo largo de varios servidores para completar el trabajo más rápido que si solo un servidor estuviera encargado de esto.
- Los protocolos de prueba complicados se van simplificando en función de nuestra estructura del software para que esto permita que los desarrolladores prueben individualmente los patrones del software sin tener que probar todo a la vez.

En general los patrones de arquitectura brindan el conocimiento y un conjunto de estrategias para crear un producto de software basado en un problema similar resuelto previamente. El principal desafío sin embargo es encontrar una manera de hacer que un patrón de arquitectura se ajuste al sistema de software particular del proyecto a crear.

Dicho esto, a través de la investigación realizada en conjunto como equipo hemos llegado a la conclusión que el patrón de arquitectura de software ideal para nuestra solución es el MVC (Modelo vista controlador).

## Modelo Vista Controlador

El patrón de arquitectura modelo-vista-controlador (MVC) implica la separación del modelo de datos, la capa de presentación y los aspectos de control de una aplicación de software. Las características clave del patrón son:

- **Modelo:** Contiene la funcionalidad principal y los datos de una aplicación
- **Vista:** Muestra los datos de la aplicación al usuario; sin embargo, la vista no puede influir en lo que el usuario hará con los datos. Los desarrolladores pueden definir más de una vista.
- **Controlador:** Actúa como bloque de construcción entre el modelo y la vista. Maneja la entrada del usuario y actúa adecuadamente en función de los eventos de activación de la vista.

Este patrón de arquitectura separa las representaciones internas de la información de cómo se presenta y cómo se acepta. Esto desacopla diferentes componentes de software, lo que permite una reutilización eficiente del código más adelante.

Las ventajas de utilizar este patrón de arquitectura son:

- Desacopla los componentes y permite la reutilización eficiente del código
- El equipo de desarrollo puede presentar múltiples vistas a los usuarios
- No da formato a los datos antes de presentarlos a los usuarios por lo que este patrón se puede usar con cualquier interfaz

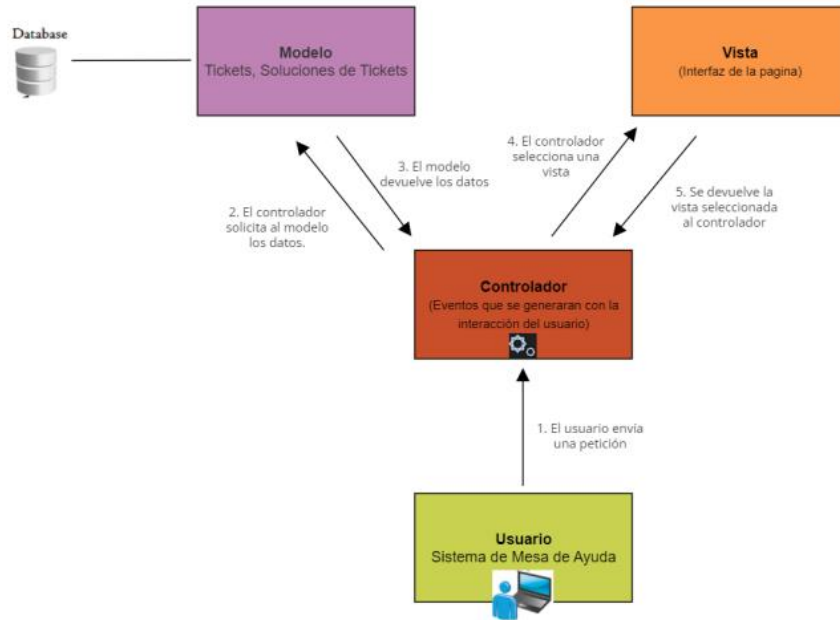
Este patrón se usa comúnmente para crear una serie de diferentes aplicaciones entre estas están:

- Interfaces gráficas de usuario de escritorio
- Aplicaciones Web
- Librerías para lenguajes de programación

En conclusión, el patrón de diseño de software MVC es ideal para la creación del proyecto de mesa de ayuda, pese a la alta complejidad que supone la separación de la vista y el modelo permite una reutilización del código eficiente, aunque hay una curva de aprendizaje pronunciada como equipo estamos dispuestos a asumir este desafío con la finalidad de hacer un producto que aporte valor a nuestro desarrollo como profesionales.

## Diagrama de Mesa de Ayuda (MVC)

Mesa de Ayuda (MVC)





## **Estándares de Programación segura.**

### **Roles y Perfiles**

En el diseño de nuestra solución se implementarán 4 tipos de roles con sus diferentes privilegios de acceso, en donde se apreciará el rol de usuario, operador, técnico de área y como supervisor. Luego de esto, como se mencionaba, cada rol tendrá privilegios diferentes, en donde tenemos que el usuario directamente no contara con acceso puesto que la solución está diseñada para un nivel administrativo, para continuar, el rol de operador contará con accesos de creación, visualización y edición de tickets, destacar que también podrá hacer derivación de ésta al siguiente rol el cual es técnico de área, rol que contará con privilegios de visualización y edición, con el fin de dar una observación de la solución del caso y efectuar cierre del ticket. Para finalizar se tiene al rol de supervisor, el cual tendrá todos los permisos mencionados, por otra parte, tendrá la posibilidad de dar gestión de tickets, creando, modificando o eliminando propiedades. Por otra parte, crear, visualizar, eliminar usuarios dentro del sistema, dando así los permisos correspondientes de cada rol, con la posibilidad de crear más usuarios administradores y hacer excepciones dependiendo del juicio o solicitud.

Luego de esto, se tomarán prácticas de seguridad tales como la autenticación de dos pasos, la cual consiste en que el usuario luego de hacer ingreso de sus credenciales, se solicitará un código de confirmación de identidad, este código podrá ser verificado bajo diferentes métodos tales como correo electrónico, mensaje sms, o la aplicación de Google Authenticator. Por otra parte, se tomarán medidas de tiempo de sesión activa, la cual mediante un periodo de tiempo la sesión estará activa y en cuyo caso se detecte inactividad de parte de usuario, se cerrará la sesión, por otra parte respecto a la simplicidad de la página se tomará como base diseños simples, minimalistas, agradables a la vista, también teniendo una estructura usable e intuitiva para el usuario.

## **Almacenamiento en la nube.**

Nuestros usuarios finales necesitarán de la disponibilidad de algún servicio de almacenamiento en la nube implementada en la aplicación para almacenar y recuperar información con relación a su negocio.

### **¿Qué son los servicios de almacenamiento en la nube?**

Son plataformas que almacenan datos en Internet a través de un proveedor informático el cual lo opera como servicio. Se ofrece bajo demanda y elimina la necesidad de administrar y comprar infraestructura de almacenamiento de datos, la cual puede llegar a tener un costo altamente elevado. Otorga seguridad, agilidad y durabilidad con acceso a los datos guardados en cualquier momento y lugar.

### **Servicios de Almacenamiento en la nube:**

#### **1. Amazon AWS Storage Gateway**

- Conjunto de servicios de almacenamiento en la nube que brinda acceso a las
- Instalaciones de manera prácticamente ilimitada.
- Ofrece 100 GB en almacenamiento de datos de forma gratuita durante 12 meses.
- Luego de los doce meses el cliente puede seguir utilizando el servicio pagando solo lo que utilice.
- Uso compartido entre usuarios.
- Copias de seguridad basadas en la nube para todos los archivos.

#### **2. Microsoft OneDrive para Empresas.**

- Carga de archivos de hasta 100 GB.
- Almacenamiento total de 1 TB por usuario.
- Acceso desde varios dispositivos a los archivos.
- Realiza copias de seguridad en caso de eliminaciones accidentales o ataques maliciosos.
- USD\$5.00 por usuario al mes.

### 3. Google Drive Business

- Almacenamiento total de 2TB por usuario.
- Unidades compartidas para el equipo de trabajo.
- Protección integrada contra software malicioso, spam y ransomware.
- Asistencia online 24 horas.

### **Servicio de almacenamiento en la nube a usar en el proyecto**

La mejor opción para el almacenamiento en la nube para nuestro proyecto es el servicio de Amazon AWS Storage Gateway. Esto debido a que es un servicio gratuito por 12 meses que brinda cantidad de almacenamiento suficiente para lo que el proyecto requiere, además luego del uso gratuito el usuario puede pagar solo lo que utilice sin tarifas mínimas.

### **Plan de Pruebas**

Un plan de prueba es un documento detallado que describe la estrategia de prueba, los objetivos, el cronograma, la estimación, los entregables y los recursos necesarios para realizar la prueba de un producto de software. El plan de prueba nos ayuda a determinar el esfuerzo necesario para realizar actividades de prueba de software como un proceso definido, que el administrador de pruebas supervisa y controla minuciosamente.

Según la definición del ISTQB: “El plan de prueba es un documento que describe el alcance, el enfoque, los recursos, y el cronograma de las actividades de pruebas previstas”

¿Cuál es la importancia del plan de pruebas?

- Ayuda a las personas ajenas al equipo de prueba, como desarrolladores, gerentes comerciales y clientes a comprender los detalles de las pruebas.
- El plan de prueba guía nuestro pensamiento. Es como un libro de reglas que debe seguirse.
- Los aspectos importantes tales como la estimación de la prueba, el alcance de la prueba y la estrategia de prueba

## **Criterios de la Aceptación de Historias de usuario**

Los criterios de aceptación son los criterios que se establecen para evaluar si una aplicación cumple con los requisitos que se le han impuesto y si es aceptada por el usuario. Los criterios de aceptación suelen estar relacionados con el funcionamiento correcto de la aplicación, la calidad de la interfaz de usuario, la seguridad de los datos, el rendimiento de la aplicación, etc.

Los criterios de aceptación se establecen en función de las necesidades y expectativas del usuario y se utilizan para evaluar si la aplicación cumple con estos requisitos y si es aceptada por el usuario. Los criterios de aceptación se establecen al inicio del desarrollo de la aplicación y se utilizan como guía para el desarrollo y las pruebas de la aplicación. Además, se utilizan para determinar si la aplicación está lista para su entrega y su puesta en producción.

A continuación, se muestra una tabla con las historias de usuario y sus criterios de aceptación:

Requerimientos funcionales con su criterio de Aceptación:

RECOPILACIÓN DE HISTORIAS DE USUARIO				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	Criterios de Aceptación
RF1	Como supervisor/ejecutivo/técnico de área	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de ingresar al sistema a través de un login	Dicha cuenta no puede tener mas de un perfil asignado(ejecutivo, técnico de área o supervisor)
RF2	Como Supervisor	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de realizar gestiones referente a personal	Se debe de ingresar con usuario(rut ej:xxxxxx-x) y contraseña.
RF3	Como Supervisor	Necesito poder gestionar y modificar los roles de cada usuario	Con la finalidad de poder gestionar cuentas de demas supervisores, tecnicos y ejecutivos	Podra ademas editar areas, tipos de ticket, estados. No podra modificar los tickets si ya existen tickets con especificaciones antiguas
RF4	Como supervisor/ejecutivo/técnico de área	Necesito poder obtener listado de ticket	Con la finalidad de obtener un historial	Puese ser filtrafo por numero de Ticket, por categoria, por titulo, por priodidad, estado, fecha de creacion, fecha de asignacion, fecha de cierre, ejecutivo de area
RF5	Como supervisor/ejecutivo/técnico de área	Necesito poder visualizar los tickets que ya fueron solucionados	Con la finalidad de obtener informacion respecto al proceso que se llevo acabo	No se puede editar un ticket ya cerrado.
RF6	Como ejecutivo	Necesito poder ingresar con una cuenta con permisos específicos al rol	Con la finalidad de realizar gestiones referente a tickets	Se debe de ingresar con usuario(rut ej:xxxxxx-x) y contraseña.
RF7	Como ejecutivo	Necesito poder crear tickets	Con la finalidad de realizar gestiones correspondientes para los problemas de los usuarios	Se debe poder ingresar informacion como Titulo, Categoria, Subcategoria, Prioridad, Adjuntar documentos adicionales (fotos-archivos de texto)
RF8	Como ejecutivo	Necesito poder asignar un ticket creado	Con la finalidad de realizar gestiones para los problemas de los usuarios	Cada ticket creado podra ser asignado al etécnico de área correspondiente visualizandose asi en la tabla al ejecutivo al cual se le asigno
RF9	Como ejecutivo	Necesito poder cambiar el estado	Con la finalidad de realizar gestiones para los problemas de los usuarios	El ejecutivo podra cambiar el estado "Abierto" o "Cerrado" dependiendo de la gestion que se este llevando acabo
RF10	Como técnico de área	Necesito poder agregar una solución y cambiar estado cuando el problema este resuelto	Con la finalidad de realizar gestiones para los problemas de los usuarios	El técnico de área podra dar cierre del ticket junto con una observacion
RF11	Como supervisor	Necesito poder crear nuevos ejecutivos/técnico de área/supervisores	Con la finalidad de administrar el sistema	Se debera dar una opcion para asi administrar los permisos de cada rol
RF12	Como supervisor	Necesito que al eliminar un usuario no se eliminen los tiques creados	Con la finalidad de llevar un registro de todo el sistema	Se debera solo poder desactivar el acceso al sistema
RF13	Como ejecutivo	Necesito hacer ingreso de una descripcion del problema	Con la finalidad de ingresar datos correspondientes y relevantes para una correcta solucion	El nombre, los datos de contacto, el modelo del dispositivo y el problema deben ir en la descripcion del ticket.
RF14	Como ejecutivo	Necesito hacer ingreso del tipo de tique	Con la finalidad de establecer un orden	Se debe ingresar un tipo de tique de manera obligatoria
RF15	Como ejecutivo	Necesito hacer ingreso del area a derivar	Con la finalidad de establecer un orden y una correcta derivacion	El area a derivar puede ser ("Hardware" o "Software")

Requerimientos No funcionales con su criterio de Aceptación:

RECOPIACIÓN DE HISTORIAS DE USUARIO				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	Criterios de Aceptación
RFN1	Como tecnico de area/ejecutivo/supervisor	La interfaz grafica debe ser de facil comprensión (Intuitiva y minimalista)	Mejorar la usabilidad y accesibilidad del Sistema	Agregar animaciones al momento de generar y cerrar tickets
RFN2	Como tecnico de area/ejecutivo/supervisor	Necesito poder contar con un apartado donde se indique como funciona el sistema	Para resolver dudas de una manera rapida	Implementar una pagina de ayuda exclusiva para visualizar como funciona el sistema. Debe contar con tutoriales de video
RFN3	Como tecnico de area/ejecutivo/supervisor	Necesito que los colores sean acordes a la imagen de la empresa	Para que aquella persona que este haciendo uso del sistema se sienta parte de la empresa	Los colores deben ser acordes a la empresa
RFN4	Como tecnico de area/ejecutivo/supervisor	El Sistema debe soportar el manejo de gran cantidad de informacion	Mejorar el rendimiento y la utilización del sistema	Optimizar el funcionamiento de la pagina
RFN5	Como tecnico de area/ejecutivo/supervisor	El sistema debe permitir controlar que solo usuarios registrados ingresen al sistema	Mantener la seguridad de la informacion	No puede acceder un usuario que no se haya registrado en el sistema
RFN6	Como tecnico de area/ejecutivo/supervisor	Que el ingreso al sistema sea de facil acceso	Para asi garantizar una atencion mucho mas rapida	El acceso al sistema debe estar claramente definido al ingresar a una pagina

## Estrategias de Prueba

Una estrategia de pruebas es un plan que define cómo se llevarán a cabo las pruebas en una aplicación. La estrategia de pruebas especifica qué se va a probar, cómo se van a realizar las pruebas, cuándo se van a ejecutar y quién será responsable de realizarlas. Además, la estrategia de pruebas también define los criterios de aceptación y los objetivos que se quieren alcanzar con las pruebas.

Las estrategias de prueba para realizar son las siguientes:

Existen diferentes tipos de pruebas que se pueden utilizar en el desarrollo de sistemas y aplicaciones, cada una con diferentes objetivos y enfoques. Cuando se trata de una aplicación de mesa de ayuda en PHP, una estrategia de pruebas adecuada podría incluir los siguientes tipos de pruebas:

**Pruebas unitarias:** Las pruebas unitarias son un tipo de prueba que se realiza en el nivel más básico del código, y que se utiliza para validar si una función o método en particular cumple con lo que se espera que haga. Las pruebas unitarias son esenciales en el desarrollo de aplicaciones en PHP, ya que ayudan a asegurar que cada una de las funciones o métodos que se implementen en la aplicación de mesa de ayuda cumplan con sus requerimientos y se comporten de la manera esperada.

**Pruebas de integración:** Las pruebas de integración son un tipo de prueba que se realiza cuando ya se han implementado y probado las funciones o métodos individuales de la aplicación, y que se utiliza para validar si estos elementos trabajan correctamente juntos en el sistema completo. Las pruebas de integración son importantes en una aplicación de mesa de ayuda en PHP, ya que ayudan a asegurar que las diferentes funciones o métodos que se implementen en la aplicación trabajen correctamente juntos, sin errores ni interrupciones.

**Pruebas de sistema:** Las pruebas de sistema son un tipo de prueba que se realiza cuando ya se ha implementado y probado la aplicación completa, y que se utiliza para validar si el sistema cumple con los requerimientos funcionales y no funcionales establecidos. Las pruebas de sistema son esenciales en una aplicación de mesa de ayuda en PHP, ya que ayudan a asegurar que la aplicación cumple con todos los requerimientos especificados y se comporta de la manera esperada en diferentes escenarios y condiciones.

**Pruebas de usuario:** Las pruebas de usuario son un tipo de prueba que se realiza con el objetivo de validar si la aplicación cumple con las necesidades y expectativas de los usuarios que la utilizan. Las pruebas de usuario son importantes en una aplicación de mesa de ayuda en PHP, ya que ayudan a asegurar que la aplicación sea fácil de usar

**Pruebas de carga:** Las pruebas de carga son un tipo de prueba que se realiza con el objetivo de validar si el sistema es capaz de soportar el tráfico y la actividad esperada en condiciones normales de uso. Las pruebas de carga son importantes en una aplicación de mesa de ayuda en PHP, ya que ayudan a asegurar que la aplicación pueda manejar un elevado nivel de tráfico y actividad sin caerse o experimentar lentitud.

**Pruebas de seguridad:** Las pruebas de seguridad son un tipo de prueba que se realiza con el objetivo de validar si el sistema es seguro y resistente a ataques externos. Las pruebas de seguridad son importantes en una aplicación de mesa de ayuda en PHP, ya que ayudan a asegurar que la aplicación proteja la información confidencial y sensible de los usuarios, y que no sea vulnerable a ataques externos que puedan comprometer la seguridad del sistema.

En resumen, una estrategia de pruebas adecuada para una aplicación de mesa de ayuda en PHP podría incluir pruebas unitarias, pruebas de integración, pruebas de sistema, pruebas de usuario, pruebas de carga y pruebas de seguridad. Estos diferentes tipos de pruebas ayudarán a asegurar que la aplicación de mesa de ayuda cumpla con sus requerimientos y se comporte de la manera esperada en diferentes escenarios y condiciones. Además, las pruebas de carga y seguridad son especialmente importantes en una aplicación de este tipo, ya que ayudan a asegurar que la aplicación pueda manejar un elevado nivel de tráfico y actividad sin experimentar lentitud ni ser vulnerable a ataques externos.



## **Tipos de Pruebas Seleccionadas**

**Pruebas de equivalencia:** Las pruebas de equivalencia son un tipo de prueba unitaria que se utiliza para validar si una función o método cumple con lo que se espera que haga en diferentes escenarios y condiciones. Por ejemplo, si se está desarrollando una función para ingresar un nuevo usuario, se podrían realizar pruebas de equivalencia para validar si la función devuelve el resultado que es visible tanto en la base de datos como en el sistema.

**Pruebas de límites:** Las pruebas de límites son un tipo de prueba unitaria que se utiliza para validar si una función o método se comporta correctamente en los límites de sus entradas o parámetros. Por ejemplo, si se está desarrollando una función para ingresar un ticket se podrían realizar pruebas de límites para validar si la función se comporta correctamente cuando se le pasan valores mínimos, máximos y extremos como parámetros.

**Pruebas de excepciones:** Las pruebas de excepciones son un tipo de prueba unitaria que se utiliza para validar si una función o método maneja correctamente las excepciones o errores que puedan ocurrir durante su ejecución. Por ejemplo, si se está ingresando un ticket vacío se podrían realizar pruebas de excepciones para validar si la función maneja correctamente las excepciones que pueden ocurrir cuando se le pasan valores inválidos o no numéricos como parámetros.

**Pruebas de regresión:** Las pruebas de regresión son un tipo de prueba unitaria que se utiliza para validar si una función o método que ya ha sido probada y validada sigue cumpliendo con sus requerimientos después de haber sido modificada o actualizada. Por ejemplo, si se ha modificado una función de ingresar un ticket para corregir un error, se podrían realizar pruebas de regresión para validar si la función sigue cumpliendo con sus requerimientos y se comporta de la manera esperada después de la modificación.

## Pruebas de Sistema

**Pruebas de compatibilidad:** en estas pruebas, se verifica si el sistema es capaz de funcionar correctamente en diferentes entornos y plataformas. Por ejemplo, se pueden realizar pruebas para verificar que el sistema funciona correctamente en diferentes versiones del sistema operativo, en diferentes navegadores web o en diferentes dispositivos móviles.

**Pruebas de seguridad:** en estas pruebas, se evalúa la capacidad del sistema para protegerse contra posibles ataques o amenazas externas. Se verifica si el sistema es capaz de mantener la confidencialidad, la integridad y la disponibilidad de los datos en todo momento, y si cumple con los estándares de seguridad establecidos.

**Pruebas de aceptación:** Las pruebas de aceptación son un tipo de pruebas de sistema que se llevan a cabo para comprobar que el sistema cumple con los requisitos funcionales y no funcionales especificados en su diseño, y que es apto para ser utilizado por los usuarios finales. Estas pruebas se realizan simulando el uso del sistema por parte de los usuarios finales, es decir, ejecutando el sistema y utilizándolo como si fuera un usuario real. Se crean casos de prueba que cubren todas las funcionalidades y características del sistema, y se comprueba que el sistema responde de manera adecuada a cada caso de prueba.

## Pruebas de Usuario

**Pruebas de funcionalidad:** Estas pruebas se centran en verificar si la aplicación cumple con los requisitos funcionales especificados, es decir, si puede realizar las tareas y acciones que se esperan de ella.

## Pruebas de Seguridad

**Escaneo de vulnerabilidades:** Esta prueba consiste en utilizar herramientas especializadas para buscar e identificar posibles vulnerabilidades en la aplicación, como errores de programación o configuraciones inseguras.

**Pruebas de autenticación y autorización:** Estas pruebas se enfocan en verificar que la aplicación implementa mecanismos de autenticación y autorización adecuados para garantizar que solo usuarios autorizados puedan acceder a la información y funcionalidades protegidas.

**Implementar el plan de pruebas diseñado en las etapas anteriores.**

- Efectuar y documentar las pruebas funcionales, según los criterios de aceptación.

ID Prueba	ID Requisito	Criterio de aceptación	Datos de Entrada	Resultado obtenido	Fecha de Prueba	Responsable de la Prueba	Observaciones	Estado
TEST 1	RF1	Dicha cuenta no puede tener más de un perfil asignado (ejecutivo, tecnico de area)	Se realiza la creación de un perfil con el mismo correo	No puede tener 2 roles	06/12/22	Matias Osorio	N/A	Completado
TEST2	RF2	Se debe de ingresar con usuario(rut ej:xxxxxxx-x) y contraseña.	Se realiza el ingreso con rut	No se puede ingresar	06/12/22	Fabian Barra	Durante la fase de desarrollo se cambio para que solo se ingresará con correo	Incompleto
TEST3	RF3	Podrá además editar áreas, tipos de ticket, estados. No podrá modificar los tickets si ya existen tickets con especificaciones antiguas	Se realizó la prueba de esas funcionalidades	Todo funciona bien	06/12/22	Matias Osorio	N/A	Completado

TEST 4	RF4	Puede ser filtrado por número de Ticket, por categoría, por título, por prioridad, estado, fecha de creación, fecha de asignación, fecha de cierre, ejecutivo de área	Se comprobaron los filtros	Funcionan bien	06/12 /22	Matias Osorio	N/A	Completado
TEST 5	RF5	No se puede editar un ticket ya cerrado.	Se intento editar un ticket ya cerrado	No fue posible hacerlo	06/12 /22	Matias Osorio	N/A	Completado
TEST 6	RF6	Se debe de ingresar con usuario(rut ej:xxxxxx-x) y contraseña.	Se realiza el ingreso con rut para comprobar que ingresa con los permisos de rol	No fue posible hacerlo por los cambios realizados durante el desarrollo	06/12 /22	Nicolas Cona	Durante la fase de desarrollo se cambio para que solo se ingresará con correo	Incompleto
TEST 07	RF7	Se debe poder ingresar información como Título, Categoría, Subcategoría, Prioridad, Adjuntar documentos adicionales (fotos-archivos de texto)	Se realizó la creación de un Ticket para comprobar que funcionaba	El ticket fue ingresado con éxito	06/12 /22	Matias Osorio	N/A	Completado
TEST 8	RF8	Cada ticket creado podrá ser tomado por el técnico de área correspondiente visualizandose así en la tabla al ejecutivo al cual se le asignó	Se ingresó con un perfil de tecnico de area y se comprobó la funcionalidad	El ticket fue tomado por el técnico de manera exitosa	06/12 /22	Matias Osorio	N/A	Completado
TEST 9	RF9	El ejecutivo podrá cambiar el estado "Abierto" o "Cerrado" dependiendo de la gestión que se esté llevando a cabo	Se realizó un cierre de ticket por parte del ejecutivo	El ticket fue cerrado de manera exitosa	06/12 /22	Matias Osorio	N/A	Completado
TEST	RF10	El técnico de área podrá	Se realizó un	El ticket fue	06/12	Matias	N/A	Completado

10		dar cierre del ticket junto con una observación	cierre de ticket por parte del técnico de área	cerrado de manera exitosa	/22	Osorio		
TEST 11	RF11	Se deberá dar una opción para así administrar los permisos de cada rol	Se inició sesión con la cuenta con los permisos correspondientes	El apartado de gestión de usuarios funciona correctamente	06/12/22	Matias Osorio	N/A	Completado
TEST 12	RF12	Se deberá solo poder desactivar el acceso al sistema	Se comprobó eliminar el acceso a un usuario al sistema	El usuario ya no puede acceder	06/12/22	Matias Osorio	El estado del usuario cambio a 0 en la base de datos	Completado
TEST 13	RF13	El nombre, los datos de contacto, el modelo del dispositivo y el problema deben ir en la descripción del ticket.	Se ingresó un ticket	Se pueden agregar los suficientes caracteres como para incorporar los valores antes mencionados	06/12/22	Matias Osorio	N/A	Completado
TEST 14	RF14	Se debe ingresar un tipo de tique de manera obligatoria	Se comprobó esta funcionalidad	El sistema no deja ingresar un ticket sin antes seleccionar el tipo	06/12/22	Matias Osorio	N/A	Completado
TEST 14	RF15	El área a derivar puede ser ("Hardware" o "Software")	Se comprobaron las áreas	Existe otra llamada "Otros"	06/12/22	Matias Osorio	Fue cambiado durante el proceso de desarrollo y revisión para mejorar así el sistema	Completado

- Efectuar y documentar las pruebas integrales.

cero de Prueba	Descripción	Módulos Involucrados	Resultado Esperado	Resultado Obtenido
1	Prueba de integración del módulo de creación de tickets con la base de datos	Creación de tickets, base de datos	Al crear un nuevo ticket, este debe almacenarse correctamente en la base de datos para que pueda ser accedido y tratado por los agentes de soporte	Éxito
2	Prueba de integración del módulo de seguimiento de tickets con la base de datos	Seguimiento de tickets, base de datos	Al cambiar el estado de un ticket, este cambio debe reflejarse correctamente en la base de datos para que pueda ser visto por los agentes de soporte y por el usuario que ha creado el ticket	Éxito
3	Prueba de integración del módulo de tickets con el módulo de respuestas automatizadas	Tickets, respuestas automatizadas	Al cerrar un nuevo ticket, se deben realizar una	Éxito
4	Comprobar que el usuario puede asignarse la solicitud de ticket así mismo	Asignación de solicitudes de ticket	El usuario debe asignarse la solicitud de un ticket.	Éxito
5	Comprobar que el sistema puede recibir y procesar una nueva solicitud de registro de usuario	Recepción y procesamiento de solicitudes de registro de usuario	El sistema debe recibir y procesar la solicitud de registro de usuario sin errores	Éxito
6	Comprobar que el sistema puede autenticar al usuario cuando intenta iniciar sesión	Autenticación de usuarios	El sistema debe autenticar al usuario correctamente cuando intenta iniciar sesión	Éxito
7	Comprobar que el sistema puede aplicar los filtros especificados en la solicitud de filtrado de tickets	Aplicación de filtros de tickets	El sistema debe aplicar los filtros especificados en la solicitud de filtrado de tickets sin errores	Éxito
8	Comprobar que el sistema puede mostrar los tickets filtrados en una interfaz de usuario adecuada	Visualización de tickets filtrados	El sistema debe mostrar los tickets filtrados en una interfaz de usuario clara y fácil de usar	Éxito

- Efectuar y documentar las **pruebas de sistema**.

Número de Prueba	Descripción de la prueba	Entorno de prueba	Resultado esperado	Resultado obtenido
1	Comprobar la compatibilidad del sistema con diferentes navegadores web	Chrome, Firefox, Safari, Edge	El sistema debe funcionar de manera adecuada en todos los navegadores web especificados	Éxito
2	Comprobar la compatibilidad del sistema con diferentes sistemas operativos	Windows, MacOS	El sistema debe funcionar de manera adecuada en todos los sistemas operativos especificados	Éxito
3	Comprobar la compatibilidad del sistema con diferentes dispositivos móviles	iPhone, Android	El sistema debe funcionar de manera adecuada en todos los dispositivos móviles especificados	Éxito
4	Comprobar la compatibilidad del sistema con diferentes resoluciones de pantalla	1024x768, 1280x720, 1920x1080	El sistema debe funcionar de manera adecuada en todas las resoluciones de pantalla especificadas	Éxito
5	Comprobar la compatibilidad del sistema con diferentes niveles de permisos de usuario	Ejecutivo/Tecnico de Area	El sistema debe funcionar de manera adecuada en todos los niveles de permisos especificados	Éxito
6	Comprobar la compatibilidad del sistema con diferentes versiones de las bibliotecas y dependencias utilizadas	Versión 1.0, versión 2.0	El sistema debe funcionar de manera adecuada con todas las versiones especificadas de las bibliotecas y dependencias utilizadas	Éxito

## **Plan de mantención.**

Referente al plan de mantención, se debe mencionar que este ser planificado y adaptable al crecimiento que el sitio tenga, esto también pensando en las posibles solicitudes que el cliente tenga para modificar la página, ya sea un cambio significativo o menor, es por esto que el cliente en caso de que quiera lo ya mencionado, debe ser solicitado con tiempo y de manera formal y registrada, para que así se pueda tomar acción, reunir requerimientos, y desarrollar una carta Gantt con la planificación y las peticiones que se tiene para con el cliente(formato, imágenes, etc.), esto con motivo de dejar en claro la planificación del desarrollo de los cambios.

Lo necesario para proponer un protocolo de mantención, son muchos puntos para abarcar, iniciando con:

- La creación de una copia de seguridad periódica, esto con motivo de en caso de ser víctima de un ataque (cualquiera sea su naturaleza), se pueda tener acceso a información valiosa relacionada a los registros de actividad para así determinar la falla y realizar mejoras.
- Realizar monitoreo constante del servicio ofrecido, lo mencionado consta en realizar vigilancia del sitio, particularmente en su correcto funcionamiento, dando como ejemplo la continuidad operacional, revisión de que sus funciones se desempeñen correctamente, verificar que el sitio no tenga errores visuales, y que este se mantenga con conectividad estable.
- Implementar pruebas de velocidad, también destacar en este punto que está sujeto a actualizaciones, dependiendo de las tecnologías implementada en el desarrollo de la página. Retomando, este es un factor para considerar, puesto que ciertos navegadores utilizan este factor al momento de mostrar resultados de búsqueda, este protocolo se puede llevar a cabo con la utilización de herramientas tales como Google PageSpeed o GTMetrix, esto dará como resultado, más tráfico en la página web.
- El Analizar la arquitectura del sitio también es un factor para considerar, puesto que la pagina es propensa a cambios, es importante el realizar de manera correcta y planificada los ya mencionados, esto para cuidar la estética y usabilidad de la página en sí y además no afectar la orientación del usuario al momento en el que esté navegando por ella.
- Realizar búsqueda de enlaces rotos, como su nombre lo indica, consta en la mantención de los enlaces tanto internos o externos dependiendo de las funcionalidades de la página, y la dificultad de su realización es directamente proporcional con el tamaño de esta. Herramientas las cuales se utilizan para esto son Google Search Console



- Actualizar diseño, contenido y funcionalidades. Al momento de la manteniendo la página en el tiempo, esto resulta crítico, puesto que los usuarios puede que no queden conforme con el diseño de la página, es por esto que se debe de realizar estas actualizaciones con el fin de que esta sea más agradable para con el usuario tomando en cuenta una ambientación, y las funcionalidades que esta deba tener para satisfacer al ya mencionado.
- Revisar el trabajo del SEO, debido a que los navegadores están en constante actualización, es altamente recomendable el hacer revisión de lo ya mencionado, lo cual cumple con la función de optimizar las palabras clave y otros aspectos similares.
- Renovación de certificados de seguridad SSL, es esencial para demostrar la seguridad de la página el renovar los protocolos HTTP a HTTPS, esto la protege de lecturas, modificaciones y robo de la información.
- Analizar las vulnerabilidades, como lo indica, consta en realizar revisión de las vulnerabilidades que la página tiene y puede llegar a tener, en esto también se debe considerar que se requiere de personal especializado.

## Conclusión

Para dar cierre, se puede destacar que se adquiere una mejor perspectiva del desarrollo de un proyecto, esto debido a que se toma el punto de vista inicial de una necesidad de implementar un sistema informático el cual cubra las necesidades presentes, por otro lado, no es un proceso simple y corto de realizar, puesto que se tiene que utilizar múltiples herramientas para identificar dichas necesidades, de los diferentes usuarios. Esto luego nos lleva a la planificación del proyecto con sus respectivos sprint, lo cual también requiere de atención, puesto que se tiene que utilizar el tiempo, personal y recursos de manera óptima. Por otra parte, también dar buena creación de los diagramas UML considerando la aplicación del paradigma 4+1, con el fin de plasmar las funcionalidades e interacciones usuario-sistema.

## Referencias Bibliográficas.

Atlassian. (s. f.). *Jira | Software de seguimiento de proyectos e incidencias*. Recuperado 14 de septiembre de 2022, de <https://www.atlassian.com/es/software/jira>

Debrauwer, L., & Van der Heyde, F. (2016). *UML 2.5: iniciación, ejemplos y ejercicios corregidos*. Ediciones ENI.

*¿Qué son las pruebas de software y cómo funcionan? | IBM*. (s. f.). Recuperado 10 de octubre de 2022, de <https://www.ibm.com/cl-es/topics/software-testing>

Estrada, G. H. (2022, 28 febrero). *Las mejores prácticas al desarrollar software – FORTE INNOVATION*. <https://forteinnovation.mx/revista/las-mejores-practicas-al-desarrollar-software/>

*¿Qué es el almacenamiento en la nube?* (s. f.). R 2022, de <https://www.redhat.com/es/topics/data-storage/what-is-cloud-storage>