

APTC106

Semana 9

Sumativa 3:

Propuesta y Visualización al Repositorio

Nombres integrantes:

- Alonso Cid Riveros
- Scarlett Espinoza Contreras
- Christian Mattioni Avila

ÍNDICE

ÍNDICE	2
1 Introducción	3
1.1 Objetivos del proyecto:	3
1.2 Objetivos de este informe:	3
2 Desarrollo del Prototipo de Alta Fidelidad	3
2.1 Decisión de elaborar un prototipo de alta fidelidad	3
2.2 Justificación global de las mejoras según los elementos entregados	4
2.2.1 Relación entre decisiones y elementos entregados	4
2.3 Diseño del sistema Service Desk	5
2.3.1 Pantalla de Login.....	5
2.3.2 Pantalla de Lista de Tickets.....	6
2.3.3 Pantalla de Detalle del Ticket	6
2.3.4 Pantalla de Creación de Nuevo Ticket.....	7
3 Archivos JSON para Simulación de Datos	7
3.1 Archivo JSON de Tickets.....	7
3.2 Archivo JSON de Usuarios	7
4 Selección de Tecnología: Flutter en Ambiente Local	8
5 Despliegue del aplicativo en GitHub	8
6 Funcionamiento mínimo viable	9
6.1 Pantalla de Login.....	9
6.2 Pantalla de Lista de Tickets.....	9
6.3 Pantalla de Detalle del Ticket	9
6.4 Pantalla de Creación de Ticket.....	9
7 Propuesta de integración móvil/web	9
7.1 Componentes principales de la integración:	10
8 Mockups del Prototipo	10
9 Conclusiones	10

1 INTRODUCCIÓN

El desarrollo de soluciones tecnológicas orientadas a la gestión organizacional requiere, en sus primeras etapas, la construcción de prototipos que permitan validar la propuesta y avanzar hacia un producto funcional. Este informe presenta el diseño, justificación y componentes principales del prototipo de alta fidelidad correspondiente al sistema de Service Desk, cuya finalidad es facilitar la administración de tickets de soporte en un entorno corporativo.

La motivación detrás de este proyecto surge de la necesidad de implementar una herramienta que permita registrar incidencias, gestionar solicitudes y apoyar la continuidad operativa de áreas técnicas. En esta etapa, el objetivo es construir la base visual y estructural del aplicativo, preparando el camino para futuras integraciones con servicios web reales.

1.1 Objetivos del proyecto:

- Implementar una solución tecnológica orientada a la gestión eficiente de tickets de soporte.
- Mejorar la trazabilidad, administración y resolución de incidencias dentro de un entorno organizacional.
- Establecer la base funcional de un sistema de Service Desk que permita evolucionar hacia una versión completa en etapas posteriores.

1.2 Objetivos de este informe:

- Describir el diseño del prototipo de alta fidelidad del aplicativo.
- Explicar las decisiones técnicas y de estructura tomadas en esta etapa.
- Presentar los elementos utilizados para simular datos mediante archivos JSON.
- Evidenciar el cumplimiento de los indicadores de logro establecidos en la rúbrica de evaluación.

La estructura del documento incluye el desarrollo del prototipo, la descripción de las vistas, la explicación del uso de archivos JSON, la fundamentación técnica de Flutter, los mockups asociados y las conclusiones que integran el trabajo realizado.

2 DESARROLLO DEL PROTOTIPO DE ALTA FIDELIDAD

2.1 Decisión de elaborar un prototipo de alta fidelidad

Se optó por diseñar un prototipo de alta fidelidad dado que este tipo de prototipos permite avanzar de manera directa hacia la implementación posterior. A diferencia de los prototipos de baja fidelidad, este enfoque ofrece:

- Representación precisa de la interfaz gráfica final.
- Validación temprana de la navegabilidad y flujo de usuario.
- Reducción de errores de interpretación en etapas avanzadas.
- Mayor eficiencia en la transición hacia la codificación en Flutter.

Esta decisión está alineada con el aprendizaje esperado del curso, el cual exige crear una versión mínimamente viable de un aplicativo distribuible.

2.2 Justificación global de las mejoras según los elementos entregados

La propuesta desarrollada integra todos los elementos solicitados en las instrucciones de esta evaluación y en el trabajo de la semana 8. Cada decisión tomada se fundamenta según los requerimientos, asegurando coherencia con los objetivos del curso y con la mejora planteada previamente.

2.2.1 Relación entre decisiones y elementos entregados

- **Prototipo de alta fidelidad:** se eligió este enfoque porque permite representar con precisión las vistas finales, facilitando la validación temprana del sistema Service Desk. Esto responde directamente al requerimiento de diseñar una versión mínimamente viable del aplicativo y avanzar hacia su implementación real.
- **Estructura de pantallas (Login, Lista, Detalle y Creación de Ticket):** estas vistas abordan los problemas detectados en la propuesta de mejora de la semana 8, especialmente la necesidad de contar con un flujo claro para gestionar incidencias, asegurar trazabilidad y permitir el cierre controlado de tickets.
- **Uso de archivos JSON locales:** esta decisión responde al elemento entregado que solicita simular datos antes de implementar una API. Permite validar navegación, gestión básica de información y pruebas locales sin depender de un backend, cumpliendo con el alcance de la etapa actual del proyecto.
- **Uso de Flutter en ambiente local:** responde al requisito de crear un aplicativo distribuible fuera de una tienda y al indicador de logro que busca generar una estructura funcional del sistema. La elección se fundamenta en su capacidad de prototipado rápido, compatibilidad multiplataforma y manejo eficiente de datos simulados.

Con estas decisiones, el diseño final integra completamente los elementos entregados en las instrucciones, resuelve los problemas identificados previamente y establece una base sólida para el desarrollo posterior.

2.3 Diseño del sistema Service Desk

El sistema diseñado responde a la necesidad de administrar tickets de soporte de forma ordenada y clara. Las pantallas desarrolladas permiten cubrir el flujo básico de operación: acceso → visualización → detalle → creación → cierre.

2.3.1 Pantalla de Login

La pantalla de inicio de sesión permite validar usuarios antes de acceder a la plataforma. Esta medida garantiza control de acceso y simula un proceso real de autenticación que será reemplazado por un servicio API en futuras etapas.

La pantalla contiene (ver Ilustración 1):

- Campos de ingreso de usuario y contraseña.
- Validación local contra el JSON de usuarios.
- Redirección a la pantalla principal de tickets.

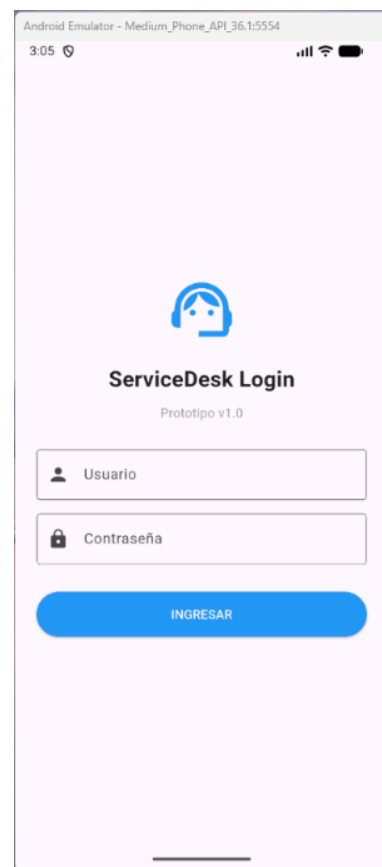


Ilustración 1 - Pantalla de Login

2.3.2 Pantalla de Lista de Tickets

Esta vista central agrupa todos los tickets disponibles (abiertos y cerrados), mostrando para cada uno (ver Ilustración 2):

- Título
- Estado (abierto/cerrado)
- Fecha de creación
- Identificador del ticket

Desde esta pantalla el usuario puede:

- Seleccionar un ticket para ver sus detalles.
- Crear un nuevo ticket.

Su diseño está orientado a la claridad visual y a la eficiencia en la gestión del soporte.

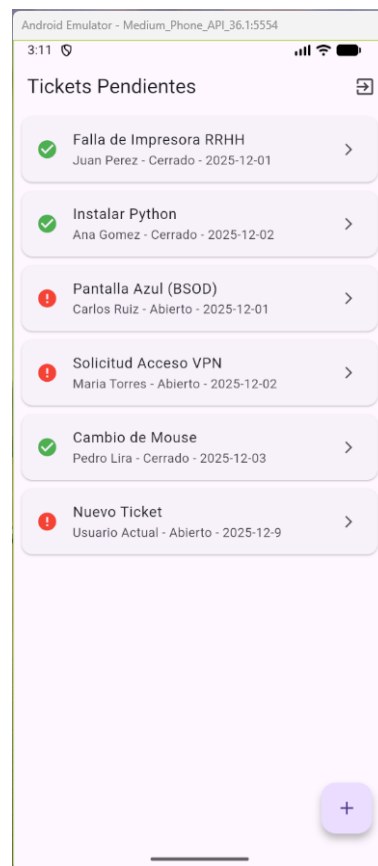


Ilustración 2 - Listado de Tickets

2.3.3 Pantalla de Detalle del Ticket

Esta pantalla permite revisar la información completa de un ticket seleccionado, incluyendo descripción detallada, responsable e historial básico.

Además, incluye un botón para cerrar el ticket cuando este está en estado “abierto”. Esta funcionalidad permite completar el flujo mínimo viable esperado para esta etapa (ver Ilustración 3).

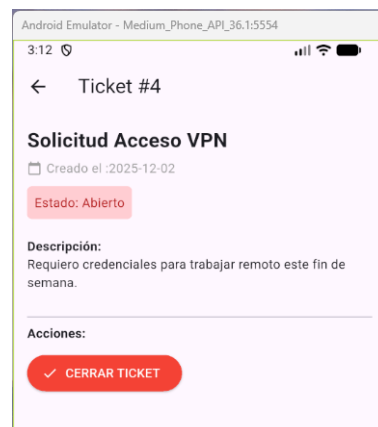


Ilustración 3 - Detalle del ticket

2.3.4 Pantalla de Creación de Nuevo Ticket

Permite registrar un nuevo ticket ingresando (ver Ilustración 4):

- Título
- Categoría
- Descripción

La aplicación toma automáticamente la fecha actual como fecha de creación, lo cual asegura consistencia en los datos simulados.

Este diseño responde a objetivos de uso simples y prácticos, propios de un sistema de soporte operativo.

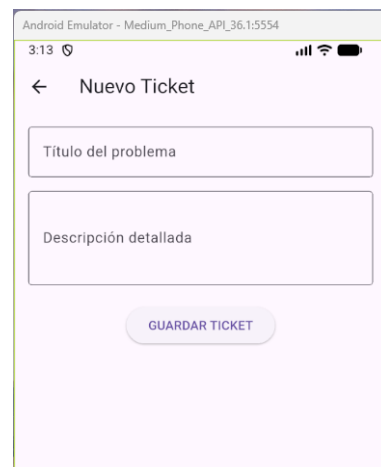


Ilustración 4 - Nuevo Ticket

3 ARCHIVOS JSON PARA SIMULACIÓN DE DATOS

Para simular la futura interacción con una API REST se desarrollaron dos archivos JSON. Esta decisión permite avanzar sin depender de un backend real, facilitando pruebas locales y demostraciones funcionales.

3.1 Archivo JSON de Tickets

El archivo se encuentra al interior de `../lib/data/` y el nombre del archivo es `mock_data.dart`

Incluye los registros estructurados de tickets con campos como:

- Id
- Titulo
- Descripción
- Estado
- fecha_creacion

Este archivo representa la respuesta que entregaría un endpoint real en la próxima etapa de desarrollo.

3.2 Archivo JSON de Usuarios

El archivo se encuentra al interior de `../lib/data/` y el nombre del archivo es `mock_users.dart`

Simula una pequeña base de datos de credenciales, permitiendo validar el acceso desde la pantalla de Login, incluye campos como:

- Username
- Password

- rol

Estos datos son esenciales para probar el flujo completo sin requerir conexión a un servicio externo.

4 SELECCIÓN DE TECNOLOGÍA: FLUTTER EN AMBIENTE LOCAL

Para el desarrollo del prototipo se decidió utilizar **Flutter**, ejecutado de forma local en un entorno Windows. Esta elección se fundamenta en que Flutter permite construir interfaces móviles modernas y consistentes utilizando un único código base, lo que facilita tanto el prototipado como el desarrollo posterior.

Entre sus principales ventajas destacan:

- Prototipado rápido gracias a Hot Reload, lo que agiliza la iteración de las pantallas.
- Interfaz responsiva y de alta fidelidad, ideal para simular el producto final.
- Facilidad para trabajar con datos simulados mediante archivos JSON locales, lo que permite avanzar sin depender aún de una API real.
- Ejecución y distribución local, cumpliendo con el requisito del curso de generar un aplicativo que pueda ser probado directamente fuera de una tienda de aplicaciones.

Esta tecnología permite diseñar la estructura del aplicativo móvil, generar una primera versión distribuible y construir una versión mínimamente viable basada en navegación real y datos simulados.

5 DESPLIEGUE DEL APLICATIVO EN GITHUB

Para cumplir con el requisito de despliegue en GitHub, se creó un repositorio público que contiene el código fuente del prototipo de Service Desk desarrollado en Flutter. El repositorio se encuentra disponible en la siguiente dirección:

- https://github.com/MaidoniaN/flutter_sumativa3

En este repositorio se incluyen:

- El proyecto Flutter con su estructura completa (carpetas lib/, assets/, etc.).
- Los archivos JSON utilizados para la simulación de datos de tickets y usuarios.
- El archivo pubspec.yaml con las dependencias necesarias.
- Un archivo README.md con instrucciones básicas para clonar y ejecutar el proyecto en ambiente local.

Con este despliegue se cumple el requerimiento de versionar el aplicativo en GitHub, asegurando trazabilidad de los cambios y facilitando la colaboración entre los integrantes del grupo.

6 FUNCIONAMIENTO MÍNIMO VIABLE

El prototipo desarrollado cumple con el funcionamiento mínimo viable solicitado para esta etapa. Las vistas implementadas en Flutter están completamente operativas, permitiendo recorrer el flujo principal del sistema Service Desk. A continuación, se detalla el funcionamiento validado:

6.1 Pantalla de Login

- Permite ingresar usuario y contraseña.
- La validación se realiza contra el archivo `mock_users.dart`.
- En caso de credenciales correctas, navega a la pantalla de lista de tickets.

6.2 Pantalla de Lista de Tickets

- Carga los registros almacenados en `mock_data.dart`.
- Muestra título, estado e identificador de cada ticket.
- Permite seleccionar un ticket para ver su detalle.
- Incluye botón para crear un nuevo ticket.

6.3 Pantalla de Detalle del Ticket

- Muestra la información completa del ticket seleccionado.
- Si el ticket está en estado “abierto”, permite cerrarlo.
- La acción actualiza el estado dentro del prototipo (simulado localmente).

6.4 Pantalla de Creación de Ticket

- Permite ingresar título, categoría y descripción.
- El prototipo agrega automáticamente la fecha de creación.
- El registro se simula de manera local.

Estas funcionalidades fueron probadas en entorno local, verificando que la navegación entre pantallas funciona correctamente, que los datos JSON se cargan y muestran sin errores y que el flujo completo del sistema puede ejecutarse sin interrupciones. Con ello se garantiza que el prototipo cumple con el funcionamiento mínimo viable solicitado en esta evaluación.

7 PROPUESTA DE INTEGRACIÓN MÓVIL/WEB

La integración entre la aplicación móvil desarrollada en Flutter y una futura aplicación web se basa en una arquitectura común soportada por servicios REST. Esta propuesta permite que ambos

canales compartan la misma lógica de negocio, los mismos datos y un flujo coherente para los usuarios, asegurando consistencia en la gestión de tickets de soporte.

7.1 Componentes principales de la integración:

- **Backend unificado:** La aplicación móvil y la versión web consumirán el mismo servicio REST, encargado de administrar tickets, usuarios, autenticación y actualización de estados. Esto permitirá centralizar la información y evitar duplicidad de datos.
- **Autenticación y manejo de sesiones:** Ambos canales utilizarán un mismo servicio de login, el cual gestionará tokens de autenticación. La aplicación móvil utilizará almacenamiento seguro local, mientras que la versión web empleará sesiones o cookies.
- **Sincronización de datos:** Los cambios realizados en la aplicación móvil (crear ticket, cerrar ticket, editar descripción) se reflejarán automáticamente en la plataforma web mediante las solicitudes API. Esto permite que técnicos y usuarios administrativos trabajen de manera simultánea.
- **Diseño responsivo y consistente:** Las interfaces web seguirán la misma estructura visual del prototipo móvil, adaptando su disposición a un entorno de escritorio. Esto facilita que los usuarios tengan una experiencia homogénea en ambos canales.
- **Escenarios de uso integrados:**
 - El usuario crea un ticket desde el móvil.
 - El área de soporte lo gestiona desde la web.
 - El usuario ve el cierre o actualización desde su app móvil.

Esta integración permitirá extender el prototipo actual hacia un ecosistema completo, conectando dispositivos móviles con una plataforma web robusta y permitiendo colaboración en tiempo real entre usuarios y técnicos.

8 MOCKUPS DEL PROTOTIPO

El detalle del funcionamiento de la aplicación lo pueden revisar en detalle en el link <https://www.youtube.com/watch?v=QmwmZs9ijC8>, el cual esta indicado en el README.md del repositorio (https://github.com/MaidoniaN/flutter_sumativa3).

9 CONCLUSIONES

Integración con la propuesta de mejora de la semana 8 y evaluación de objetivos

La presente propuesta se alinea directamente con la mejora planteada en la semana 8, ya que las funcionalidades diseñadas abordan las principales problemáticas identificadas anteriormente: la necesidad de una gestión centralizada de incidencias, la falta de trazabilidad en los tickets y la ausencia de un flujo uniforme de registro y cierre. El prototipo de alta fidelidad desarrollado refleja

de manera concreta estas mejoras, integrando un flujo coherente que responde a dichas necesidades.

En relación con los objetivos planteados tanto para el proyecto como para este informe, se concluye que estos fueron alcanzados satisfactoriamente. Se logró diseñar la estructura visual del sistema, justificar las decisiones técnicas adoptadas, simular datos mediante archivos JSON y documentar el proceso de acuerdo con los indicadores establecidos en la rúbrica. Con ello, se evidencia un nivel de logro pleno respecto de las metas propuestas para esta etapa del desarrollo.

El desarrollo del prototipo permitió establecer una base sólida para la futura implementación completa del sistema de Service Desk. Se lograron construir las vistas principales, decidir un flujo claro de interacción y simular datos mediante archivos JSON, permitiendo así un funcionamiento coherente con un entorno real.

Entre los resultados obtenidos destacan:

- La correcta estructuración de un prototipo de alta fidelidad reutilizable.
- La implementación de cuatro pantallas clave que cubren el flujo mínimo viable del sistema.
- La simulación satisfactoria de datos mediante JSON.
- La configuración adecuada del proyecto en Flutter, preparando su integración con servicios REST.

Desde una perspectiva crítica, se identifica como desafío futuro la necesidad de integrar un backend real para manejar autenticación, persistencia y actualización de datos en tiempo real. Asimismo, será necesario optimizar la experiencia de usuario al incorporar validaciones avanzadas y roles diferenciados.

Como marco de acción futura, se espera:

- Conectar el prototipo con una API REST real.
- Implementar control de sesiones y seguridad.
- Integrar almacenamiento en la nube para manejar tickets de manera persistente.
- Ampliar las funciones del Service Desk incorporando asignación automática, prioridades y notificaciones.

Estas acciones permitirán transformar el prototipo actual en un producto robusto y profesional, listo para etapa de pruebas y despliegue.