



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**EMPAT-E: APLICACIÓN MOVIL DEDICADA A LA CREACIÓN Y
GESTIÓN DE MAPAS DE EMPATIA**

**Proyecto de titulación previo a la obtención del título de: Tecnólogo Superior en
Desarrollo de Software**

Autor: Alan David Pérez Tipantuña

Tutor: Pablo Andrés Pérez Martínez

Quito, Ecuador

2024

DECLARACIÓN y AUTORIZACIÓN

Yo, **Alan David Pérez Tipantuña** con C.I. 1750789750 autor(a) del trabajo de titulación intitulado: **“Empat-e: APLICACIÓN MOVIL DEDICADA A LA CREACIÓN Y GESTIÓN DE MAPAS DE EMPATIA”**, previa a la obtención del título de **TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, Octubre de 2024

Alan David Pérez Tipantuña

C.I. 1750789750

Tabla de contenido

Objetivos.....	5
Objetivo General.....	5
Objetivos Específicos	6
Capítulo I. Introducción a Mobile-D y Fase Exploratoria.....	7
Explore.....	7
Product Inception.....	8
Why are we here?	9
Elevator Pitch.....	10
Product Box	11
Show your Solution	14
Size it Up	15
Distribución en Releases	16
What is Going to Give?	17
What is Going to Take?	18
Historias de Usuario	19
ÉPICA 1: Gestión de Usuarios	19
Épica 2: Gestión de Mapas de Empatía	24
Capítulo II Initialize.....	32
Project Setup	32

Frontend	32
Backend.....	33
Herramientas Transversales	34
Planning the day in zero iteration	35
Working Day in Zero Iteration.....	35
Release Day in Zero Iteration	39
Capítulo III. Productionize, Stabilize, System Test & Fix.....	39
Conclusiones.....	45
Recomendaciones	47
Referencias	48

Introducción

En el mundo de los negocios y la educación, entender bien a tu público es clave para que tus proyectos y productos triunfen. Ahí es donde entran los mapas de empatía, que son una súper herramienta para descubrir qué piensan, cómo actúan y qué necesitan las personas a las que quieres llegar. Pero de aquí proviene el problema, que normalmente estos mapas se hacen en papel, lo cual puede llegar a ser un inconveniente porque se desorganizan fácil y actualizarlos es un dolor de cabeza. Por ello se construyó la idea de "Empat-e", una app móvil pensada para estudiantes, emprendedores y gente de marketing que quiera crear y manejar mapas de empatía de manera digital. No solo haciendo todo más fácil y organizado, sino que también es más ecológico porque no gastas papel ayudado así a disminuir la huella de carbono en nuestro planeta. Además, te ayuda a tomar decisiones más rápido porque toda la información está bien ordenada y accesible.

Y para lograr este propósito de crear "Empat-e" se usará la metodología Mobile D, que es súper ágil y flexible. Esto significa que la app se va mejorando constantemente gracias a la organización y diseño que permite esta metodología. O sea, no es una app cualquiera, sino una que evoluciona y se adapta a lo que la gente necesita. Para convertir una idea en la herramienta que crece contigo y te ayuda a conquistar el mercado.

Objetivos

Objetivo General

Se busca crear una solución digital, que permita a los usuarios diseñar y gestionar mapas de empatía de manera efectiva, rápida, utilizando un canvas digital, para que haga más sencillo el

estudio de mercado y el análisis de audiencias. Trayendo la herramienta que ayude a entender mejor a las personas y a tomar decisiones más informadas, todo desde el celular.

Objetivos Específicos

- Diseñar una interfaz que sea fácil de usar, que cualquier persona, pueda crear y personalizar sus mapas de empatía sin complicaciones.
- Eliminar la necesidad de usar materiales físicos, como papel o post-its, ofreciendo una alternativa digital que no solo es más práctica, sino también más sostenible, porque cuidar el medio ambiente que también es importante.
- Una app que crezca con el usuario, diseñada con una arquitectura sólida y flexible, para permitir añadir nuevas funciones en el futuro sin problemas, buscando que esta herramienta evolucione junto a las necesidades.
- Utilizar la metodología Mobile-D para desarrollar la aplicación de manera ágil, incorporando el feedback de los usuarios en cada etapa, lo que permitirá ajustar y mejorar la app constantemente, asegurando que sea de alta calidad y que realmente se vuelva una solución digital.

Capítulo I. Introducción a Mobile-D y Fase Exploratoria

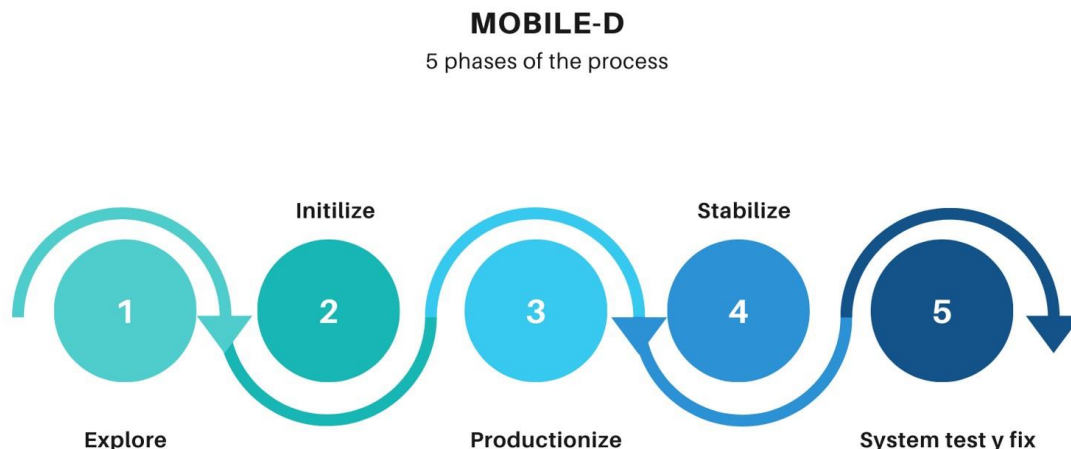


Ilustración 1. Fases Mobile-D

El desarrollo de "Empat-e" se realizará utilizando la metodología Mobile-D, que cuenta con un enfoque ágil ya que está específicamente diseñado para crear aplicaciones móviles. Esta metodología será importante para el éxito del proyecto porque nos permite avanzar paso a paso, de manera flexible y adaptativa, es decir, si surgen cambios o nuevas necesidades durante el proceso, podremos ajustarnos sin problemas, asegurando que el producto final sea justo lo que el usuario vaya a usar realmente.

Explore

En esta primera etapa, se hace un análisis profundo para entender bien las necesidades de los usuarios. Esta investigación es súper importante porque nos ayudará a definir exactamente cómo debe ser la aplicación, para asegurarnos de que no nos desviemos del camino y de que cada funcionalidad que desarrollemos tenga un propósito claro y útil.

Durante esta fase, se sentó las bases del proyecto, definiremos el alcance, las funcionalidades principales, quiénes son los actores clave (stakeholders) y a quién va dirigida la aplicación.

- **Stakeholders:**
 - Desarrollador FullStack: Responsable del desarrollo y el mantenimiento de la aplicación a nivel de programación.
 - Product Owner: Encargado de establecer la visión general del producto y de priorizar las funcionalidades en función de las necesidades del usuario.
- **Target (público objetivo):** ○ Estudiantes, emprendedores y profesionales del marketing
- **Alcance del proyecto:**
 - Sincronización y creación de mapas de empatía: Permitir a los usuarios crear, sincronizar y modificar mapas de empatía de manera virtual.
 - Generación de documentos exportables: Brindar la capacidad de generar documentos descargables para compartir los resultados del análisis.
- **Tiempo de ejecución:** ○ El desarrollo completo se llevará a cabo en un periodo de 4 meses.

Product Inception

En esta fase del proyecto, me centre en los detalles de lo que ya he definido, utilizando el Inception Deck, que esta es una actividad colaborativa en la que participan todos los miembros clave del equipo y los stakeholders, con el objetivo de crear una visión clara y compartida del producto, asegurando que todos estemos alineados en cuanto a su propósito, alcance y requisitos técnicos, esto convirtió al Inception Deck es una gran oportunidad para discutir aspectos importantes del producto, como la arquitectura técnica, la estructura del equipo y una estimación inicial de los costos de desarrollo. Además ayudó a acercar al equipo de desarrollo con el cliente, facilitando un acuerdo sobre plazos de tiempo, recursos y expectativas.

Durante esta fase, realicé un análisis profundo de la factibilidad técnica y económica del producto, esto permitió tomar decisiones informadas sobre su viabilidad y aseguro de que estoy construyendo algo que realmente funcione. También se identificó los principales riesgos y desafíos que podrían surgir durante el desarrollo, y preparar estrategias para enfrentarlos de manera proactiva.

Why are we here?

Estoy aquí para revolucionar la manera en que emprendedores y estudiantes crean y organizan sus ideas a través de mapas de empatía, ya que hoy en día, este proceso se hace principalmente en papel, lo que trae varios problemas como: falta de portabilidad, dificultad para organizar y compartir información, y un impacto ambiental negativo debido al uso excesivo de recursos físicos. Razón por lo que propongo Empat-e, proyecto con el propósito de digitalizar este proceso, una app móvil específicamente diseñada para crear y gestionar mapas de empatía de manera intuitiva, accesible y sostenible, al ofrecer una herramienta que no solo facilite el trabajo, sino que también al ahorrar recursos físicos contribuya a un futuro más ecológico y eficiente. Dando un salto hacia la innovación y ayudando a dejar atrás el papel.

- **Funcionalidades previstas:**

- **Creación interactiva de mapas de empatía:** Plantillas dinámicas que permiten agregar, mover y organizar contenido en tiempo real.
- **Sincronización en la nube:** Mantén tu trabajo seguro y accesible desde cualquier lugar.
- **Exportación flexible:** Genera documentos en formatos PDF e imagen para compartir y presentar tu trabajo.
- **Interfaz móvil amigable:** Adaptada para un uso ágil y sin complicaciones.

- **Versiones del producto:**

- **Versión inicial:** Funcionalidades básicas de creación y edición de mapas.
- **Actualizaciones futuras:** Colaboración en tiempo real, analíticas sobre el contenido de los mapas y mayor personalización.

- **Principales riesgos:**

- Competencia con herramientas genéricas como Miro y Canva.
- Resistencia inicial del público a abandonar métodos tradicionales.
- Problemas técnicos en la sincronización y rendimiento de la app.

- **Insumos requeridos:**

- Equipo de desarrollo (desarrollador FullStack).
- Software de diseño y herramientas de gestión de proyectos.
- Recursos para pruebas y validación con usuarios finales.

Elevator Pitch

¿A quién va dirigida la aplicación?	Estudiantes, emprendedores y profesionales del marketing que buscan herramientas especializadas para comprender y gestionar las necesidades de sus clientes o mercados.
Necesidad	Digitalizar y simplificar el proceso de creación de mapas de empatía, permitiendo una mayor movilidad, organización y distribución de ideas.
Nombre del producto	Empat-e
Tipo de solución	Aplicación móvil intuitiva y especializada para la creación, gestión y exportación de mapas de empatía.

Misión del producto	Revolucionar la forma en que los usuarios conceptualizan y estructuran información sobre sus clientes o mercados, proporcionando una herramienta eficiente, accesible y sostenible.
Competidores	• Herramientas genéricas como Miro, Canva y Lucidchart. • Métodos tradicionales basados en papel.
Principales características	• Plantillas prediseñadas adaptadas al modelo de mapas de empatía. • Sincronización en la nube para trabajo remoto y seguro. • Exportación de mapas en formatos PDF e imagen. • Diseño optimizado para dispositivos móviles. • Facilidad para gestionar y modificar mapas de empatía en cualquier momento.

Product Box



Ilustración 2. Icono App

Imagina ver una caja en un estante de una tienda. La caja de Empat-e tendría lo siguiente:

Nombre del producto: Empat-e **Eslogan:**

¡Transforma ideas en acción!

Beneficios destacados:

- **Crea mapas de empatía en minutos:** Accede a plantillas prediseñadas para simplificar el proceso.
- **Sincronización en la nube:** Trabaja desde cualquier lugar y mantén tus datos seguros.
- **Exportación y compatibilidad:** Comparte tus mapas en formato PDF o imagen con tu equipo o clientes.
- **Diseño móvil intuitivo:** Adaptado para una experiencia cómoda y ágil en dispositivos móviles.

Frase atractiva:

La herramienta definitiva para emprendedores que buscan entender y conectar con su mercado.

Not List

A continuación, se utiliza el proceso de “not list” para definir claramente lo que no está dentro del alcance del proyecto.

In Scope:

- Diseño y creación de mapas de empatía a través de plantillas interactivas.
- Sincronización de datos en tiempo real con almacenamiento en la nube.
- Exportación en formatos PDF e imagen.
- Funcionalidad para editar y personalizar mapas existentes.
- Registro y autenticación de usuarios, incluyendo recuperación de contraseñas.
- Acceso multiplataforma (Android e iOS).

Out of Scope:

- Edición colaborativa en tiempo real por múltiples usuarios.
- Integración con herramientas externas.
- Funciones de inteligencia artificial para analizar datos de los mapas.
- Soporte para modo offline.

Unresolved:

- Nivel de personalización permitido en las plantillas (colores, fuentes, etc.).
- Estructura y costos de las suscripciones premium.
- Colaboración en tiempo real.
- Posible implementación de analíticas básicas para los mapas de empatía.

Meet your Neighbors

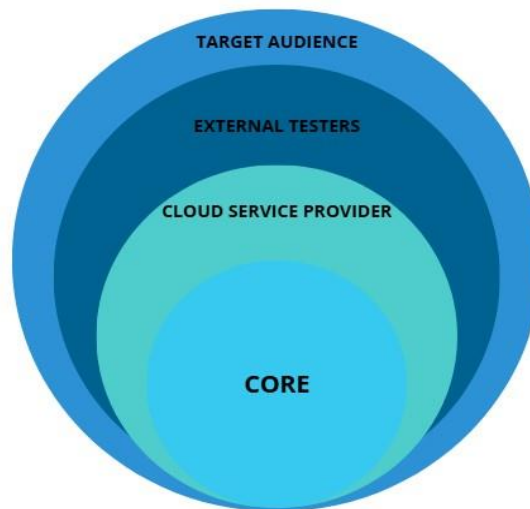


Ilustración 3. Meet your Neighbors

Show your Solution

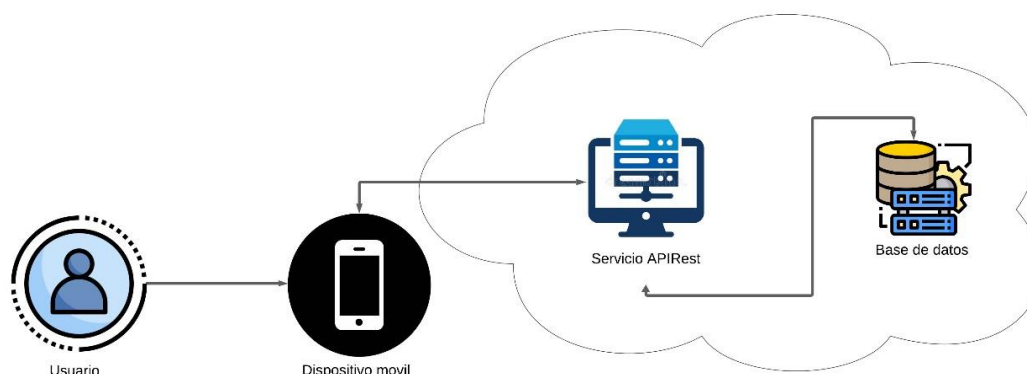


Ilustración 4. Arquitectura inicial

Empat-e es una aplicación móvil diseñada para crear, gestionar y compartir mapas de empatía, facilitando la comprensión y organización de ideas sobre clientes y mercados, optimizando el tiempo y recursos de los usuarios. Como solución para llevar a cabo el desarrollo y garantizar el buen funcionamiento de la aplicación, se optó por diseñar una arquitectura básica que detalle cómo interactuarán sus distintos componentes, ya que el proyecto requerirá principalmente el uso de servicios en la nube, incluyendo un servidor web que permita realizar llamadas mediante API REST, porque permite llamadas que facilitarán la comunicación entre la aplicación móvil y un servidor de base de datos, asegurando un acceso rápido y seguro a la información almacenada.

Características clave:

- Plantillas prediseñadas adaptadas al formato estándar del mapa de empatía.
- Un editor visual que permite agregar, modificar y mover post-its dentro de cada sección.
- Sincronización en tiempo real para mantener la información actualizada.
- Exportación de mapas en formatos PDF e imagen para facilitar el uso compartido.
- Una interfaz intuitiva diseñada específicamente para dispositivos móviles.

Size it Up

Con el Size it Up se evaluó cuánto esfuerzo requiere cada funcionalidad planeada para el proyecto y qué tan importante es. Por eso, organizamos las funcionalidades en dos categorías:

- Feature Points (FP): Representan la dificultad técnica, es decir, qué tan complejo es desarrollar cada función.
- Business Units (BU): Reflejan el valor estratégico o la importancia que tiene esa funcionalidad para los usuarios.

Esta evaluación es clave porque nos permite priorizar las tareas en cada release (versión del proyecto). Saber qué funciones son más importantes para los usuarios y cuáles requieren más esfuerzo, esto nos ayuda a tomar decisiones inteligentes, asegurando que entreguemos valor desde el principio y no atascarse en tareas demasiado complicadas al inicio.

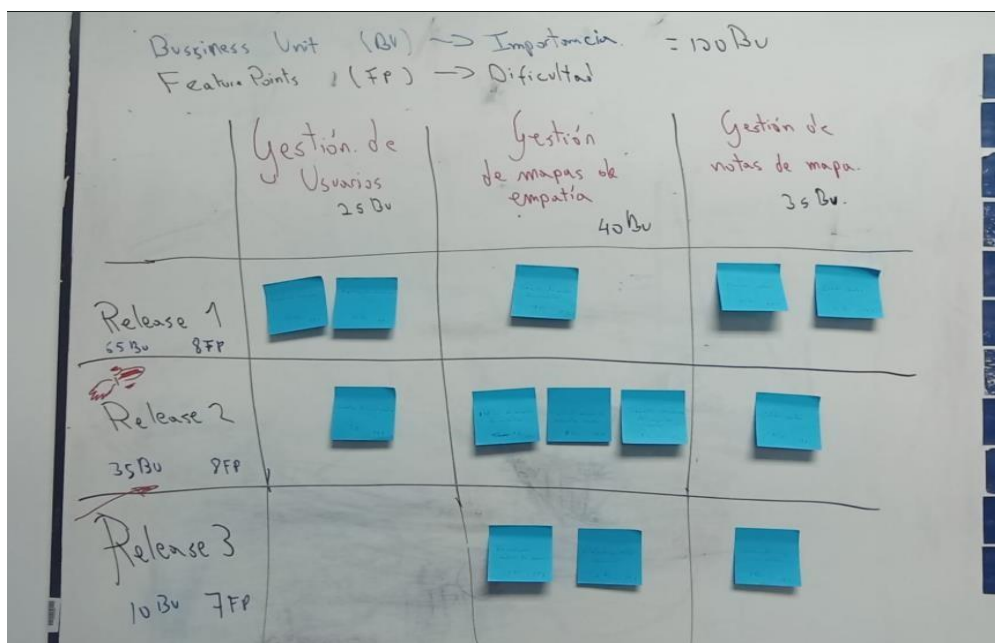


Ilustración 5. Inicio Roadmap

Distribución en Releases

Para optimizar el desarrollo, las funcionalidades se dividirán en tres releases de acuerdo con su prioridad:

- **Release 1:**

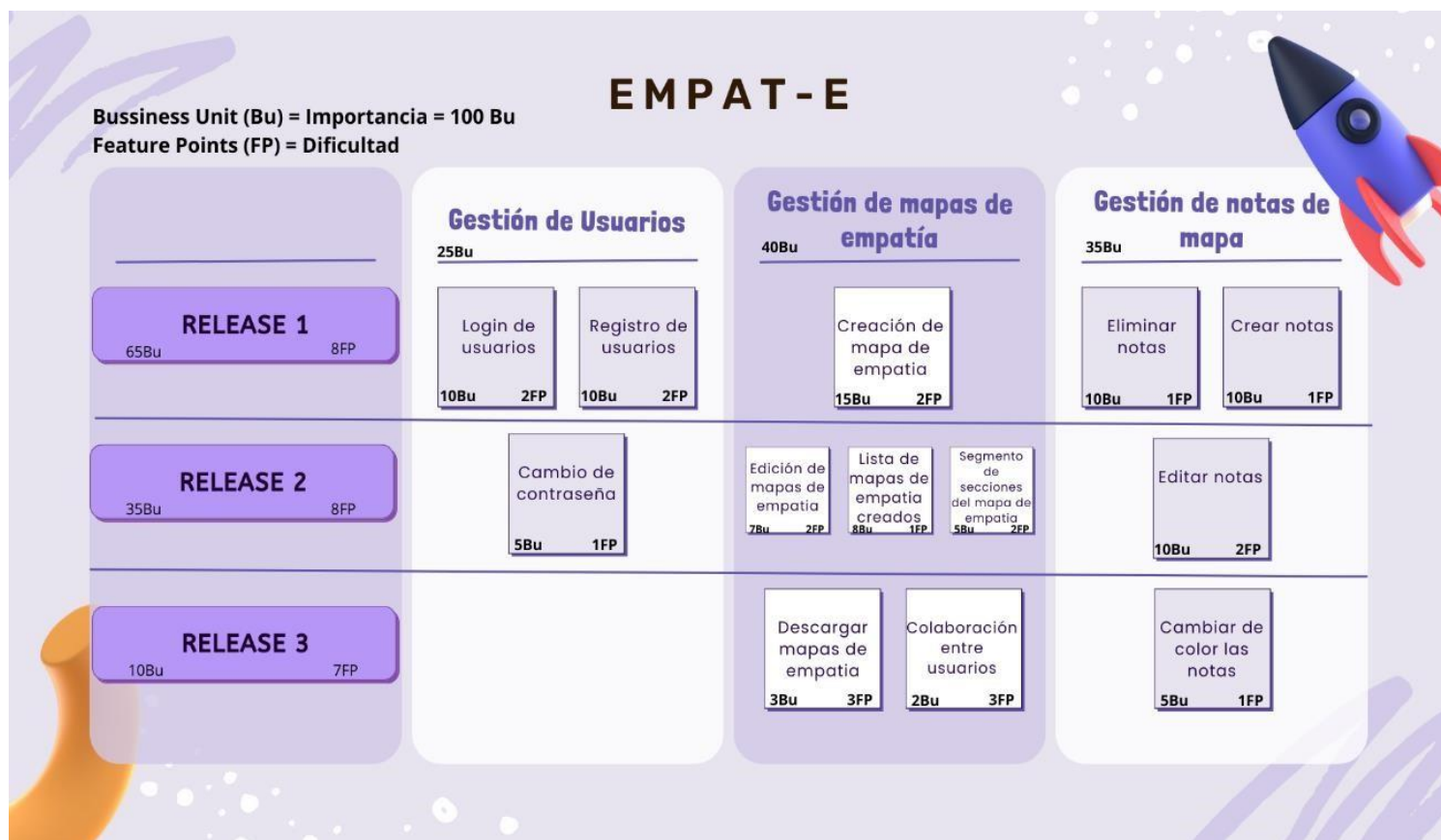
- Login de usuarios (10 Bu, 2 FP). ○ Registro de usuarios (10 Bu, 2 FP).
- Creación de mapa de empatía (15 Bu, 2 FP).
- Crear notas (10 Bu, 1 FP).
- Eliminar notas (10 Bu, 1 FP).

- **Release 2:**

- Cambio de contraseña (5 Bu, 1 FP). ○ Edición de mapas de empatía (7 Bu, 2 FP).
- Editar notas (10 Bu, 2 FP). ○ Lista de mapas de empatía creados (8 Bu, 1 FP).
- Segmento de secciones del mapa (5 Bu, 2 FP).

- **Release 3:**

- Descargar mapas de empatía (3 Bu, 3 FP). ○ Cambiar de color las notas (2 Bu, 3 FP). ○ Colaboración entre usuarios (5 Bu, 1 FP).



El proyecto Empat-e incluye funcionalidades críticas de alta importancia (Bu) en las primeras releases para garantizar el impacto en el usuario final. Las funcionalidades de menor prioridad, aunque de mayor complejidad técnica, se reservan para el último release para optimizar los esfuerzos.

Este enfoque asegura un desarrollo incremental, permitiendo obtener un producto funcional desde el primer release mientras se añaden mejoras progresivamente.

What is Going to Give?

En el desarrollo de Empat-e, se evaluaron varios factores clave para determinar qué elementos serán priorizados durante el proyecto. Para ello, se utilizó una escala de 0 a 100, donde 100 representa la máxima prioridad. El análisis dio los siguientes resultados:

Disponibilidad	30	Se prioriza garantizar que la aplicación esté accesible en todo momento para los usuarios, con sincronización en la nube.
Calidad	10	La estabilidad y confiabilidad de la app son fundamentales para ofrecer una experiencia sólida y libre de errores técnicos que puedan afectar el uso.
Usabilidad y Experiencia de Usuario (UX)	20	Se busca que la interfaz sea intuitiva y fácil de usar, con un diseño que simplifique la creación y gestión de mapas de empatía.
Tiempo	40	Se busca cumplir con un cronograma realista de 4 meses, equilibrando la rapidez de desarrollo con la calidad del producto final.
Costo	0	Se procurará optimizar los costos, utilizando recursos disponibles de manera eficiente y sin comprometer la calidad.

Este análisis asegura que Empat-e esté alineada con las necesidades de los usuarios y las prioridades del proyecto.

What is Going to Take?

Para el desarrollo de Empat-e, se definieron los roles clave y los recursos necesarios para llevar a cabo el proyecto con éxito. A continuación, se presenta un desglose del equipo y los costos mensuales estimados:

Cantidad	Rol	Costo por Persona (\$)/Mes	Costo Total (\$)/Mes
1	Desarrollador de Apps Móviles	1200	1200
1	Desarrollador de Backend	1300	1300
1	Diseñador UX/UI	1200	1200
1	Product Owner	2000	2000

Total Mensual: 5,700 \$

Duración Estimada: 4 meses

Costo Total del Proyecto: 22,800 \$

Historias de Usuario

ÉPICA 1: Gestión de Usuarios

US1. Inicio de Sesión de Usuarios

Nombre: Autenticación segura para usuarios registrados.

Descripción:

El sistema debe permitir a los usuarios registrados iniciar sesión de forma segura proporcionando su correo y contraseña, y acceder al menú principal si las credenciales son válidas.

Conversación:

Pantalla de inicio de sesión con campos para correo y contraseña.

Botón de "Iniciar sesión".

Mensajes de error claros como "Usuario o contraseña incorrectos".

Implementación de autenticación con tokens JWT y encriptación de contraseñas para proteger los datos del usuario.

Tareas:

Diseñar una pantalla de inicio de sesión con campos para correo electrónico y contraseña.

Validar las credenciales ingresadas contra la base de datos y generar un token JWT si son válidas.

Configurar el almacenamiento seguro del token en el dispositivo del usuario.

Implementar manejo de errores con mensajes descriptivos.

Pruebas:

Dado que soy un usuario registrado, **cuando** ingreso mi correo y contraseña correctos, **entonces** debo ser redirigido al menú principal.

Dado que ingreso un correo o contraseña incorrectos, **cuando** intento iniciar sesión, **entonces** debo recibir un mensaje de error: "Usuario o contraseña incorrectos".



Ilustración 7. Inicio App

US2. Registro de Nuevos Usuarios

Nombre: Creación de cuentas para nuevos usuarios.

Descripción:

El sistema debe permitir a los visitantes registrarse proporcionando un correo único, nombre y contraseña, y almacenar de manera segura los datos en la base de datos.

Conversación:

Pantalla con formulario de registro que incluya validaciones para correo, nombre y contraseña.

Verificación en la base de datos para que el correo sea único.

Almacenamiento seguro de contraseñas mediante encriptación.

Inicio de sesión automático tras un registro exitoso.

Tareas:

Crear un formulario de registro con validación en tiempo real para correo, nombre y contraseña.

Verificar en la base de datos que el correo ingresado no esté registrado.

Encriptar la contraseña antes de guardar los datos.

Permitir al usuario iniciar sesión automáticamente después del registro exitoso.

Pruebas:

Dado que soy un visitante, cuando ingreso un correo y contraseña válidos, entonces debo recibir una confirmación de que mi cuenta fue creada exitosamente.

Dado que uso un correo ya registrado, cuando intento crear una cuenta, entonces debo recibir un mensaje: "Correo ya registrado".



The image shows a registration form with the title "Registrarse". It features four text input fields stacked vertically, labeled "Nombre", "Correo Electrónico", "Contraseña", and "Confirmar Contraseña". Below these fields are two buttons: a blue button labeled "Registrarse" and an orange button labeled "Cancelar". The form is presented on a white background with rounded corners, set against a dark, patterned border.

Ilustración 8. Registro Usuario

US3. Recuperación de Contraseña

Nombre: Recuperación de acceso mediante restablecimiento de contraseña.

Descripción:

El sistema debe permitir a los usuarios recuperar su acceso a la aplicación enviando una contraseña temporal al correo registrado, para restablecer la contraseña.

Conversación:

Enlace "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión.

Pantalla para ingresar el correo registrado.

Mensaje de error si el correo no está registrado.

Contraseña temporal enviada por correo para restablecer la contraseña.

Pantalla para ingresar y confirmar una nueva contraseña.

Tareas:

Agregar un enlace "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión.

Diseñar un formulario para que el usuario ingrese su correo electrónico.

Validar que el correo esté registrado y enviar contraseña temporal para restablecer la contraseña.

Implementar una pantalla para que el usuario ingrese una nueva contraseña al obtener la temporal.

Verificar la actualización de la contraseña en la base de datos.

Pruebas:

Dado que ingreso un correo registrado, cuando solicito restablecer la contraseña, entonces debo recibir un correo con una contraseña temporal para restablecer mi contraseña.

Dado que ingreso un correo no registrado, cuando intento restablecer la contraseña, entonces debo recibir un mensaje: "El correo ingresado no está registrado".

Dado que sigo recibo la contraseña temporal para reestablecer mi contraseña, cuando ingreso una nueva contraseña válida, entonces debo poder iniciar sesión con mi nueva contraseña.

Ilustración 9. Cambiar contraseña

Épica 2: Gestión de Mapas de Empatía

US4. Creación de Mapas de Empatía

Nombre: Generación de un nuevo mapa de empatía.

Descripción:

El sistema debe permitir al usuario crear un nuevo mapa de empatía proporcionando un nombre y una descripción, que se mostrará en la lista de mapas y estará disponible para futuras ediciones.

Conversación:

Un formulario que capture el nombre y la descripción del mapa.

Un botón para guardar el mapa creado.

Mensajes de error si el nombre está vacío o duplicado.

Tareas:

Diseñar un formulario que permita capturar el nombre y la descripción del mapa.

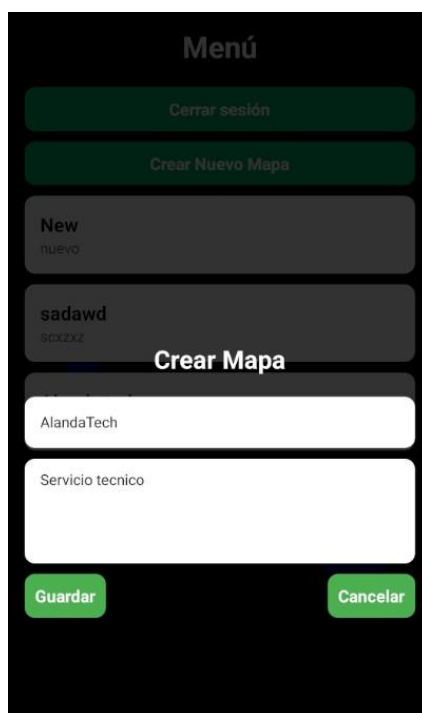
Validar que el nombre sea obligatorio y único.

Guardar los datos en la base de datos.

Redirigir al usuario a la pantalla de edición del mapa tras la creación.

Pruebas:

Dado que soy un usuario registrado, cuando creo un mapa con un nombre y descripción válidos, entonces este debe ser visible en mi lista de mapas.



The image shows a mobile application interface. At the top, there is a dark header with the word "Menú" in white. Below the header, there are two dark green buttons with white text: "Cerrar sesión" and "Crear Nuevo Mapa". Underneath these buttons is a list of items, each with a dark background and white text. The first item is labeled "New" with "nuevo" below it. The second item is labeled "sadaud" with "sadaud" below it. Below the list, there is a section titled "Crear Mapa" in white text. This section contains two white input fields on a dark background. The first input field contains the text "AlandaTech". The second input field contains the text "Servicio tecnico". At the bottom of the "Crear Mapa" section, there are two green buttons with white text: "Guardar" and "Cancelar".

Ilustración 10. Menú

US5. Crear Notas en el Mapa

Nombre: Adición de notas en las secciones del mapa.

Descripción:

El sistema debe permitir al usuario agregar notas en cualquier sección del mapa, vinculándolas correctamente a la sección seleccionada.

Conversación:

Un botón para añadir una nueva nota.

Un cuadro de texto donde se ingrese el contenido de la nota.

Las notas deben aparecer automáticamente en la sección seleccionada tras guardarlas.

Tareas:

Habilitar funcionalidad para crear nuevas notas en la interfaz del mapa.

Guardar las notas asociándolas a la sección seleccionada en la base de datos.

Pruebas:

Dado que estoy trabajando en un mapa, cuando agrego una nota en una sección, entonces esta debe ser visible en esa sección.

Dado que intento guardar una nota vacía, cuando presiono el botón de guardar, entonces debo recibir un mensaje indicando "La nota no puede estar vacía".

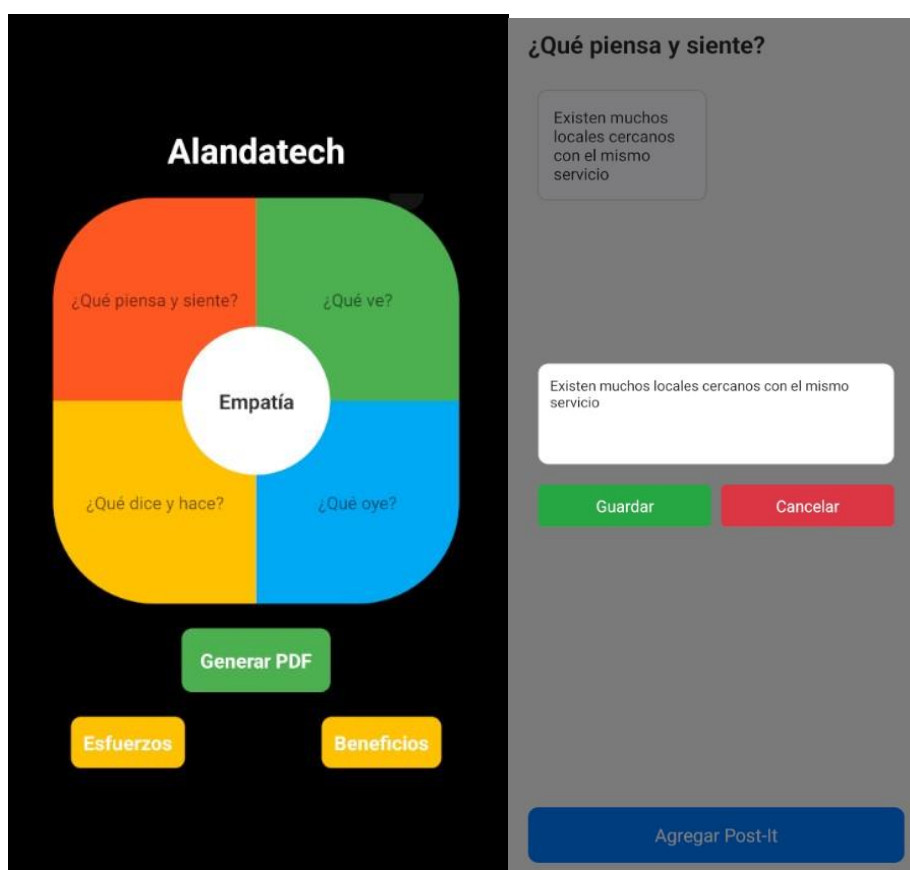


Ilustración 11. Empat-e

US6. Eliminar Notas de una Sección Nombre:

Borrado de notas innecesarias.

Descripción:

El sistema debe permitir al usuario seleccionar notas específicas para eliminarlas, solicitando confirmación antes de proceder y reflejando el cambio en la interfaz.

Conversación:

Un botón en cada nota para eliminarla.

Un cuadro de diálogo para confirmar la eliminación.

Las notas eliminadas deben desaparecer de la sección.

Tareas:

Implementar funcionalidad para seleccionar y eliminar notas de una sección.

Solicitar confirmación al usuario antes de borrar una nota.

Reflejar la eliminación de la nota en la interfaz.

Pruebas:

Dado que elimino una nota, cuando confirmo la acción, entonces esta debe desaparecer de la sección del mapa.

Dado que cancelo la acción de eliminación, cuando reviso el mapa, entonces la nota debe seguir visible.

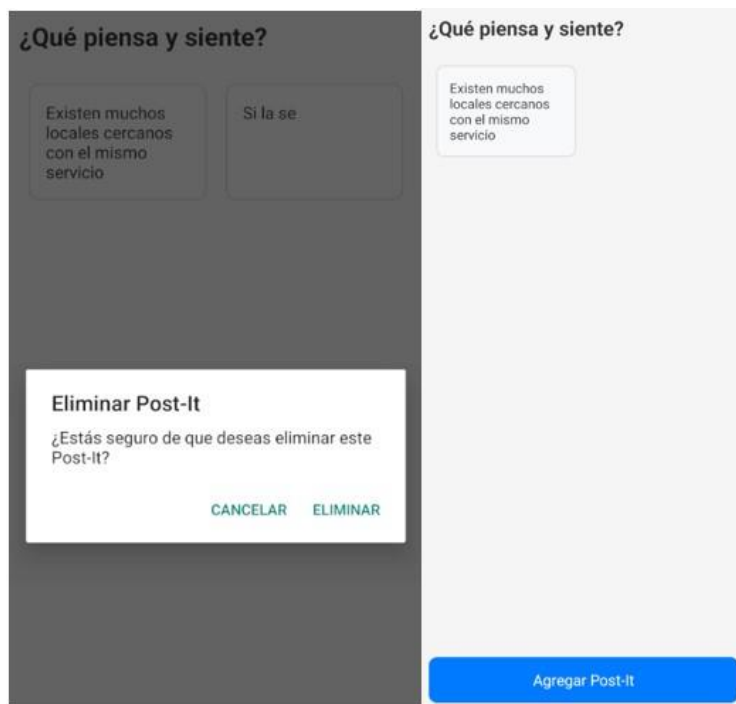


Ilustración 12. Acciones de notas

US7. Edición de Mapas de Empatía

Nombre: Actualización de nombre y descripción de mapas de empatía.

Descripción:

El sistema debe permitir al usuario editar el nombre y la descripción de un mapa existente para reflejar cambios en su análisis o ideas.

Conversación:

Un botón de edición en cada mapa de la lista.

Un formulario modal para capturar el nuevo nombre y descripción.

Mensajes de error si el nombre es duplicado o vacío.

Tareas:

Habilitar un botón de edición en cada mapa de la lista.

Diseñar un formulario modal para capturar el nuevo nombre y descripción.

Validar que el nuevo nombre sea único.

Guardar los cambios en la base de datos y actualizar la lista de mapas en la interfaz.

Pruebas:

Dado que selecciono un mapa existente, cuando ingreso un nuevo nombre y descripción válidos, entonces estos deben reflejarse en la lista de mapas.

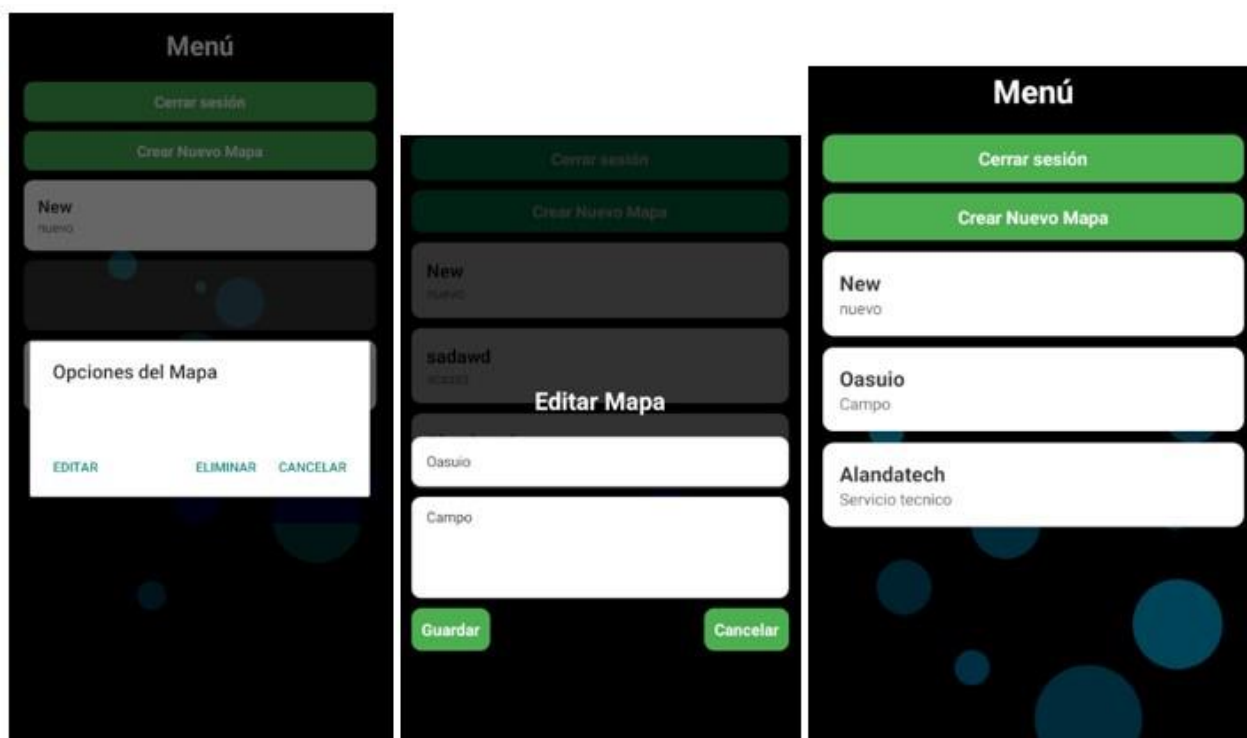


Ilustración 13. Acciones de Mapa

US8. Edición de Notas en una Sección del Mapa Nombre:

Modificación del contenido de notas existentes.

Descripción:

El sistema debe permitir al usuario editar el contenido de una nota en una sección del mapa para actualizar o corregir la información almacenada.

Conversación:

Un botón para editar cada nota.

Un cuadro de texto editable que permita realizar cambios en el contenido de la nota.

Un botón para guardar los cambios.

Tareas:

Habilitar un botón de edición para cada nota dentro de una sección.

Implementar un cuadro de texto editable que permita realizar cambios en el contenido.

Guardar los cambios automáticamente o mediante un botón de confirmación.

Pruebas:

Dado que selecciono una nota, cuando edito el texto y guardo los cambios, entonces la nota debe mostrar el nuevo contenido en la sección correspondiente.

Dado que edito una nota y cierro la ventana sin guardar, cuando reviso el contenido de la nota, entonces esta debe conservar su texto original.

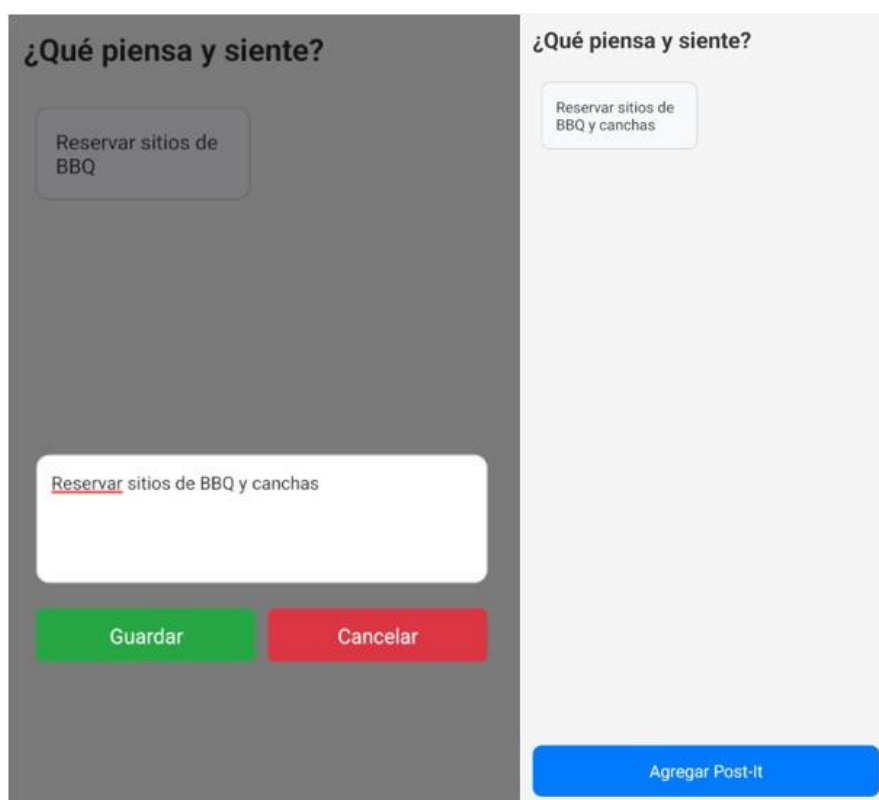


Ilustración 14

Capítulo II Initialize

En esta fase inicial, sentamos las bases del proyecto Empat-e, definiendo los objetivos principales y estableciendo un marco de trabajo tanto técnico como organizacional, por lo que realizamos un análisis detallado de los requerimientos funcionales y no funcionales, asegurándonos de que las metas del proyecto estén claras y alineadas con las necesidades de los usuarios finales.

Project Setup

En esta etapa, definimos la estructura inicial del sistema, seleccionando las tecnologías clave que nos permitirían desarrollar una aplicación robusta, segura y escalable, llevando también a cabo un análisis preliminar de los requisitos del proyecto, estableciendo un marco de trabajo que serviría como guía para las iteraciones futuras. Por lo que se presenta las herramientas que use en el desarrollo:

Frontend

- JavaScript debido a que es la opción principal para el desarrollo web y se puede usar para el desarrollo móvil, por eso lo decidí como el motor de mi frontend. Desde su creación en 1995 por Brendan Eich, este lenguaje ha conquistado la industria gracias a su versatilidad y capacidad para crear interfaces interactivas y dinámicas.
- Para construir una interfaz moderna y eficiente, hice uso de React, la biblioteca estrella de Facebook lanzada en 2013, ya que React me permite desarrollar componentes reutilizables y para mejorar el rendimiento de la aplicación, y lo que gracias a su enorme comunidad y su facilidad de integración, me ayuda a ofrecer un frontend atractivo, interactivo y súper optimizado.

El diseño si importa, y para asegurarme de que mi interfaz sea elegante y funcional, utilice Material-UI. Basado en las pautas de Google Material Design, esta biblioteca de componentes, lanzada en 2014, permite crear una UI consistente, moderna y altamente intuitiva.

Backend

- En el backend, me decidí por Java por su increíble robustez, seguridad y escalabilidad. Desde su lanzamiento en 1995 por Sun Microsystems (y ahora bajo el ala de Oracle), Java ha demostrado ser un lenguaje confiable y potente en el mundo empresarial. Su famosa filosofía de "escribe una vez, ejecuta en cualquier lugar" (WORA) lo convierte en la elección perfecta para mi sistema, garantizando que funcione sin problemas en cualquier entorno, así aseguramos estabilidad y rendimiento.
- Para hacer que mi desarrollo sea ágil y eficiente, hice uso de Spring Boot, un framework de código abierto que ha revolucionado la forma en que se crean aplicaciones en Java. Lanzado en 2014 por Pivotal, Spring Boot permitió desarrollar APIs REST potente y escalable sin perdernos en configuraciones complicadas. Su capacidad para gestionar dependencias de forma automática y su integración con múltiples herramientas lo convierten en un gran aliado, gracias a eso Spring Boot, hace del backend rápido, seguro y preparado para crecer.
- La seguridad es clave, y por eso el uso de JWT (JSON Web Tokens) para la autenticación de los usuarios. Este estándar, introducido por el grupo IETF en 2010, me permite generar tokens seguros y livianos que garantizan una comunicación protegida entre el frontend y el backend, ya que me ofrece una solución moderna, eficiente y segura.

Para almacenar y gestionar los datos de manera eficiente pensé en hacer uso de un BD SQL, por lo que implemente MySQL, una de las bases de datos más populares del mundo. Creada en 1995 por MySQL AB y actualmente propiedad de Oracle, MySQL se destaca por su velocidad, confiabilidad y facilidad de uso, ya que nos permite manejar grandes volúmenes de información sin complicaciones, asegurando que la información siempre esté disponible.

Herramientas Transversales

- Para llevar un control de versiones, mi conocimiento y familiaridad con Git, lo convirtió en mi favorita para usar en este proyecto. Creado por Linus Torvalds en 2005, Git revolucionó la forma en que los desarrolladores manejan el código, ya que permite controlar cada cambio hecho, así como las múltiples versiones de un proyecto y colaborar sin miedo a perder avances. En la app, Git nos ayuda con un desarrollo fluido y ordenado, donde cada cambio está controlado.
- Mi elección de GitHub para la organización de tareas y revisión de código, fue porque me es familiar y conocido. Desde su adquisición por Microsoft en 2018, esta herramienta se ha convertido en la plataforma favorita de los equipos de desarrolladores a nivel mundial. Funciones como pull requests, issues y revisiones de código, nos permite mantener la calidad y eficiencia del proyecto, mientras trabajamos.
- Para garantizar que mi app funcioné sin problemas en cualquier entorno, hice uso de Docker, la plataforma de contenedores que ha revolucionado el desarrollo y despliegue de software desde su lanzamiento en 2013 y nos permite encapsular el backend y la base de datos en contenedores ligeros y portátiles, asegurando que todo corra de manera por igual sin importar dónde se implemente.

Visual Studio Code desarrollado por Microsoft en 2015, es más que un simple editor de código, es una herramienta poderosa que mejora el flujo de trabajo de cualquier desarrollador, gracias a su ligereza, flexibilidad y compatibilidad con múltiples tecnologías. Con características como autocompletado, depuración y muchas extensiones, VS Code ayudó a escribir código limpio y eficiente tanto en el frontend como en el backend.

Planning the day in zero iteration

En esta fase, elegí una de las historias de usuario más importantes para probar que la estructura y las herramientas elegidas funcionaran correctamente, con el fin de verificar que las interacciones en la interfaz móvil se sincronizaran de manera adecuada con los datos manejados en la base de datos MySQL, a través del backend desarrollado. En concreto, me enfoqué en la funcionalidad de crear un nuevo mapa de empatía, asegurándome de que la información dada por el usuario en la app se sincronizara de manera eficiente y estuviera disponible de inmediato; Ya que esta funcionalidad es vital porque impacta directamente en la experiencia del usuario (UI/UX) y es fundamental para garantizar una comunicación eficiente entre el frontend y el backend.

Por lo que su correcta implementación me dio una base sólida para avanzar correctamente hacia las siguientes fases del proyecto.

Working Day in Zero Iteration

Una vez definida la tarea principal, me centre en configurar la sincronización entre los datos ingresados en la interfaz móvil y la base de datos, para ello utilizamos el backend desarrollado con Spring Boot, y que usa de base de datos a MySQL.



Ilustración 15. Dispositivo Objetivo

Por lo que el primer paso fue investigar y tratar de establecer cómo gestionar la comunicación entre el frontend, que hace uso de React Native, por lo que librerías como axios me ayudó para mi objetivo, y se conectó con el backend. Se logró al configurar las rutas de la API que permitirían la creación y actualización de los mapas de empatía en tiempo real. Esto al establecer un servicio REST en Spring Boot que recibiera las solicitudes desde la app móvil y las almacenara en la base de datos MySQL.

Configuramos las primeras rutas de la API que permitirían la creación y actualización de los mapas de empatía en tiempo real, esto al crear un servicio REST en Spring Boot que recibiera las solicitudes desde la app móvil y las almacenara en la base de datos MySQL. También definimos los modelos de datos para los mapas de empatía y sus secciones. Creamos una tabla en la base de datos para almacenar detalles como el nombre y la descripción de cada mapa, junto con la estructura necesaria para guardar la información de las secciones y los post-its que los usuarios crearían en la aplicación.

Para la capa de almacenamiento, configuramos MySQL y definimos las relaciones entre las tablas utilizando Spring Data JPA, lo que facilitó las operaciones CRUD (Crear, Leer, Actualizar,

Eliminar). Esto aseguró que los datos estuvieran disponibles para la app móvil y se sincronizaran de manera eficiente.

Por otro lado, en el frontend, configuramos la persistencia de datos en el dispositivo móvil utilizando AsyncStorage. Esto permitió que los tokens de autenticación y los datos temporales se mantuvieran entre sesiones sin necesidad de hacer solicitudes constantes al backend. También gracias a que la arquitectura inicial se estructuró para funcionar en un entorno Dockerizado. Usamos Docker-Compose para orquestar los contenedores del backend (que aloja el servicio en Spring Boot y MySQL). Esta configuración nos permitió establecer un flujo de trabajo coherente y estable.

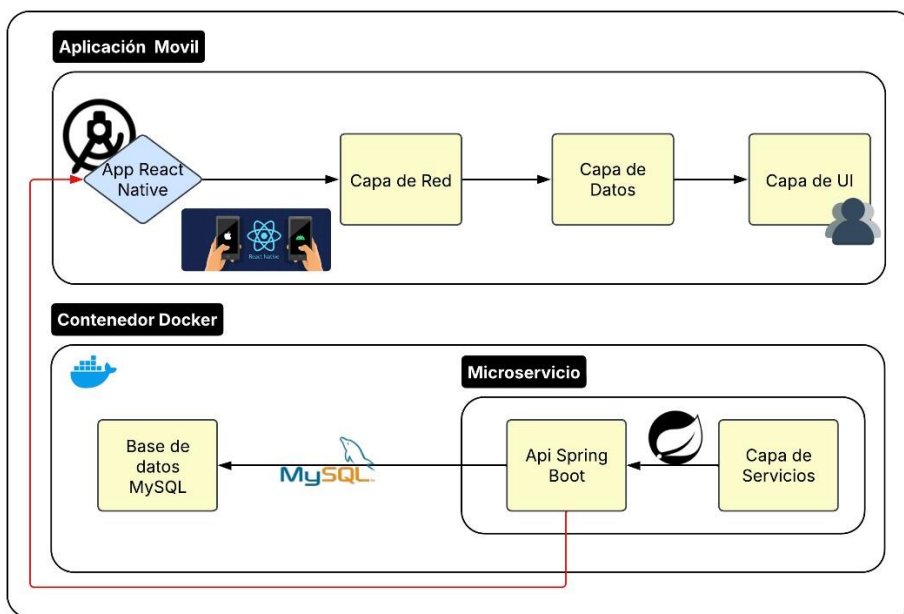


Ilustración 16. Arquitectura

Al final del día valide la comunicación entre la app móvil y la base de datos, verificando que las acciones realizadas por los usuarios en la interfaz móvil se reflejaran correctamente en la

base de datos. Esto al probar incluir la creación de un nuevo mapa de empatía y su correcta visualización en el frontend, para continuar con las próximas iteraciones.

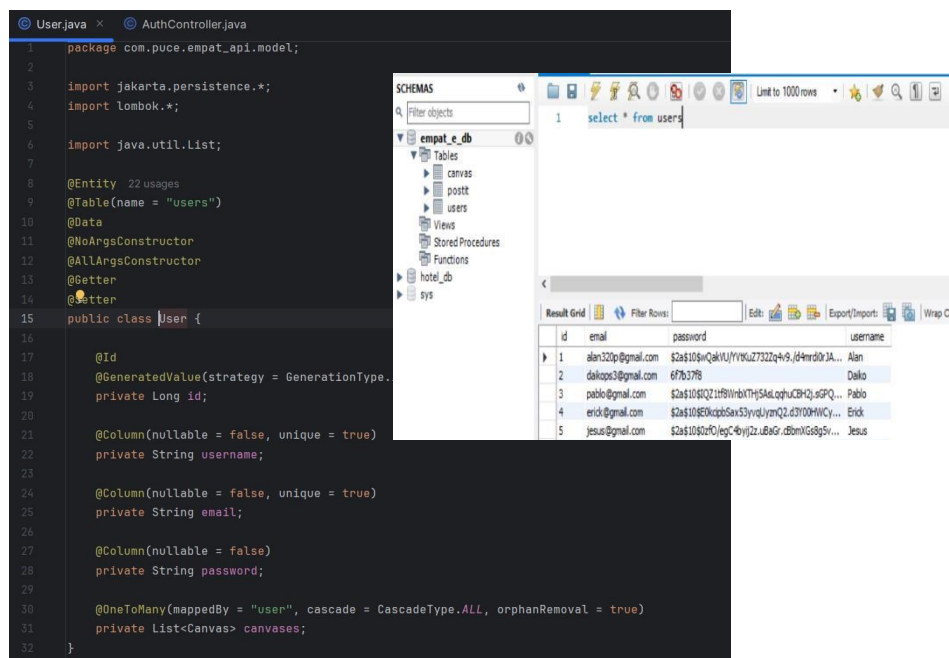


Ilustración 17. Gestion Usuario-Mapa

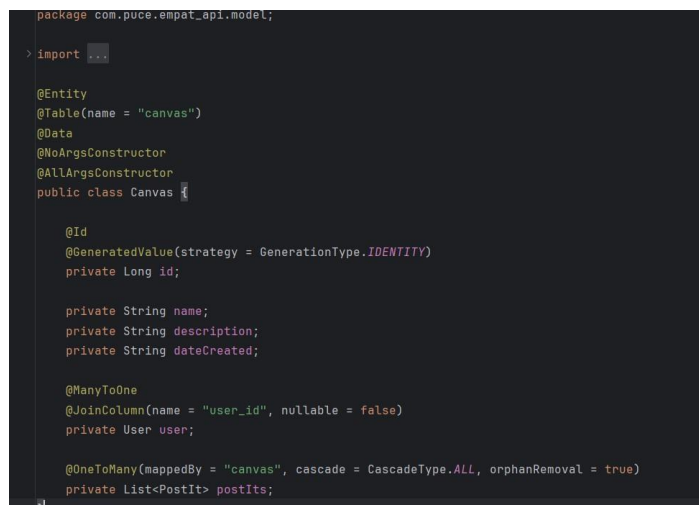


Ilustración 18

Release Day in Zero Iteration

En esta fase, compruebe que la funcionalidad principal de la aplicación, que vendría siendo la creación y sincronización de mapas de empatía entre el frontend y la base de datos, estaba correctamente implementada. Al confirmar que la información ingresada por el usuario en la interfaz móvil se sincronizaba en tiempo real con la base de datos MySQL a través del backend, fue exitosa, avanzamos con el desarrollo del proyecto. Para organizar el trabajo y las tareas futuras, hice uso de GitHub como plataforma central de gestión de los cambios tanto para el frontend como para el backend, y cree y gestione las historias de usuario directamente, en plantillas. Esto me permitió un seguimiento más ágil del progreso del proyecto.

Con todas las pruebas superadas y las funcionalidades correctamente implementadas, la Zero Iteration fue oficialmente culminada, y está lista para su implementación en el entorno de producción. Esta liberación marcó el comienzo del ciclo de retroalimentación para mí, y ajustes que guiaría el desarrollo de las siguientes iteraciones del proyecto.

Capítulo III. Productionize, Stabilize, System Test & Fix

Una vez definí los requisitos del proyecto, las tareas específicas y la arquitectura del software, actualice mi tablero Kanban en GitHub donde organizo y superviso las actividades necesarias para completar el Producto Mínimo Viable (MVP) de la aplicación, por lo que el tablero esta estructurado en tres categorías principales: “Pendiente”, “En curso” y “Completado”.

- “En curso” fue la categoría central, ya que incluía las fases de desarrollo, pruebas e implementación, lo que me ayudó a priorizar las tareas relacionadas con el diseño y la validación del sistema.
- “Pendiente” agrupaba las tareas en fase de planificación o desarrollo inicial.

- “Completado” me permitió realizar un seguimiento efectivo de los avances, ya que incluía las tareas validadas y terminadas.

Este enfoque me mantuvo organizado y enfocado en lo que realmente importaba que es construir el núcleo del MVP, el corazón del que le da vida siendo “El Mapa de Empatía”.

El foco principal durante esta etapa fue la creación del mapa de empatía, la herramienta esencial de la aplicación para gestionar y presentar información relevante a los usuarios. Los objetivos clave para esta funcionalidad fueron:

- Crear una interfaz gráfica intuitiva y fácil de usar.
- Integrar los datos del backend y visualizarlos de manera adecuada.
- Gestionar el acceso de los usuarios de forma eficiente.

El proceso comenzó con un análisis detallado del flujo de trabajo del usuario, asegurando una experiencia fluida y coherente, combinando esta planificación con el tablero Kanban, priorizamos las tareas relacionadas con la interfaz de usuario y la integración del backend. Dividimos el desarrollo en pantallas específicas, cada una enfocada en una funcionalidad clave de la aplicación.

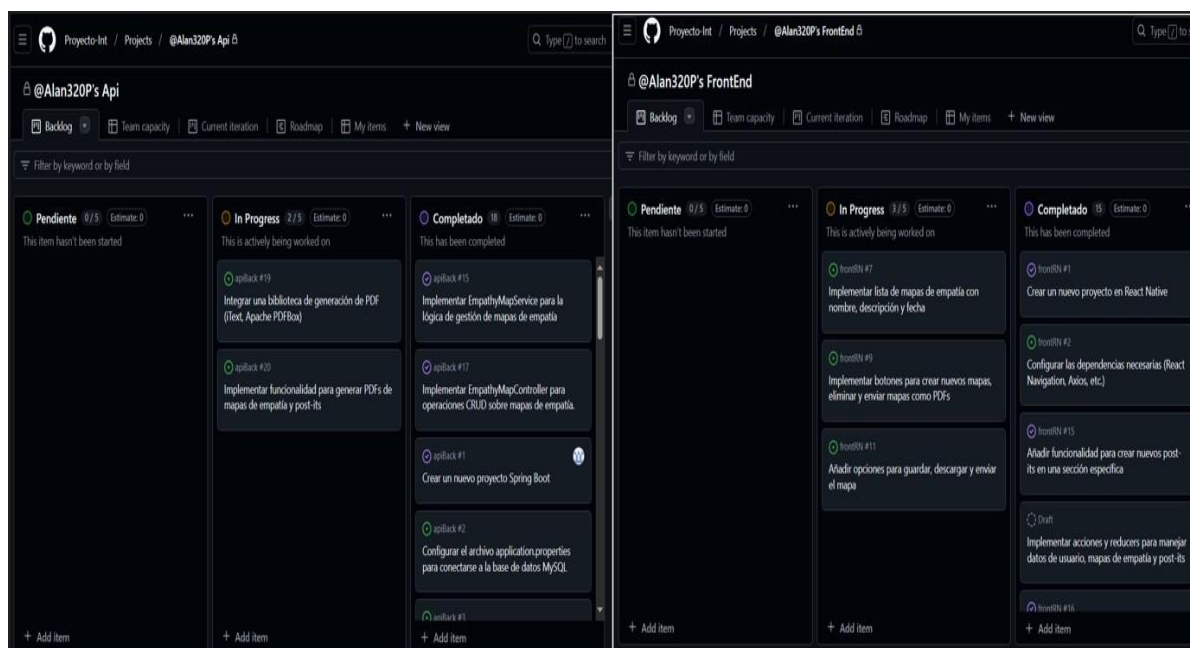


Ilustración 19. Kanban

Por lo que para gestionar las solicitudes y almacenar la información de los usuarios, despliegue el servicio REST utilizando Docker, ya que este servicio actuó como intermediario entre la interfaz de usuario y la base de datos, facilitando peticiones. El configurar la conexión con la IP del contenedor Docker, permitió acceder a los datos y gestionar las solicitudes de los usuarios de manera adecuada.

Gracias a que se logró la conexión exitosa al servicio web, nos permitió avanzar con confianza en el desarrollo del sistema.

```

apijsx X
src > services > apijsx > ...
1 import axios from 'axios';
2 import AsyncStorage from '@react-native-async-storage/async-storage';
3
4 const apiUrl = 'http://192.168.0.146:8080/api';
5
6 const api = axios.create({
7   baseURL: apiUrl,
8   timeout: 10000,
9 });
10
11 api.interceptors.request.use(
12   async config => {
13     try {
14       const token = await AsyncStorage.getItem('token');
15       if (token) {
16         config.headers.Authorization = `Bearer ${token}`;
17       }
18     } catch (error) {
19       console.error('Error getting token from AsyncStorage:', error);
20     }
21     return config;
22   },
23   error => Promise.reject(error)
24 );
25
26 export default api;

```

Ilustración 20. Conexión Api

```

src > services > canvasService.jsx > createCanvas > headers
1 import api from './api';
2 import AsyncStorage from '@react-native-async-storage/async-storage';
3
4 export const getAllCanvases = async () => {
5   try {
6     const response = await api.get('/canvas/all');
7     return response.data;
8   } catch (error) {
9     console.error('Error fetching canvases:', error);
10    throw error;
11  }
12 };
13
14 export const createCanvas = async (canvas) => {
15   try {
16     const token = await AsyncStorage.getItem('token');
17
18     const headers = {
19       Authorization: `Bearer ${token}`,
20       'Content-Type': 'application/json'
21     };
22
23     const response = await api.post('/canvas/create', canvas, { headers });
24
25     return response.data;
26   } catch (error) {
27     console.error('Error creating canvas:', error);
28     throw error;
29   }
30 };
31
32 export const getCanvasById = async (id) => {
33   try {
34     const response = await api.get(`/canvas/${id}`);
35     return response.data;
36   } catch (error) {

```

Ilustración 21. Consumo Api

Al momento de que logre establecer la conexión con el backend, hice uso del desarrollo ágil lo que permitió modularizar el desarrollo. Lo que me dio el enfoque que favoreció la reutilización de código, mejoró la mantenibilidad y escalabilidad del sistema, por lo que dividimos el frontend en módulos específicos para mejorar la organización del trabajo y optimizar la

experiencia del usuario. Las secciones que se tomaron para trabajar de forma independiente fueron:

- Pantalla de inicio
- Menú principal
- Mapa de empatía
- Mesa de trabajo

Este enfoque modular facilitó el control y la personalización de cada componente, permitiendo que cada sección del sistema pudiera gestionarse y actualizarse independientemente sin afectar otras partes. La organización de los componentes del frontend (como se observa en la figura 22) se alineó con el propósito de dar el funcionamiento que se espera de la app, proporcionando una experiencia fluida y funcional para los usuarios.

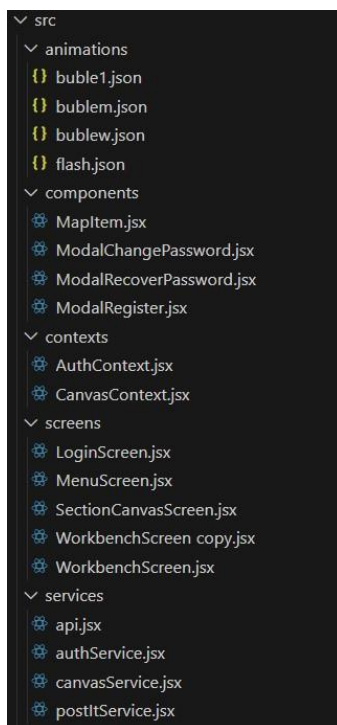


Ilustración 22. Estructura

Una vez desarrollados los componentes visuales, lo integre con el back mediante los endpoints definidos para cada acción, siendo este un paso esencial para procesar y presentar los datos almacenados en la base de datos, de manera estructurada y precisa en la interfaz de usuario.

Para que los usuarios visualicen los datos correspondientes, de manera clara y comprensible, completé y probé cada sección del mapa de empatía.

La integración no solo validó la correcta carga de los datos, sino que también garantizó una experiencia de usuario fluida, sin interrupciones ni fallos. Este proceso fue crucial para asegurar que el MVP del sistema estuviera listo para su implementación final. Con el MVP y la interfaz de usuario completados, realizamos pruebas exhaustivas para garantizar la calidad y funcionalidad del sistema:

- Pruebas de integración: Confirmamos que todos los componentes del frontend, backend y base de datos operaran de manera coherente y eficiente.
- Pruebas de usabilidad: Evaluamos que la experiencia de usuario fuera intuitiva y fácil de usar para hacer ajustes finales.
- Pruebas de seguridad: Verificamos que los datos de los usuarios estuvieran protegidos, cumpliendo con las mejores prácticas de seguridad, como la autenticación, encriptación y políticas de acceso.

```

public class WebSecurityConfig {
    public PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http.cors().and().csrf().disable()
            .exceptionHandling(exceptionHandlingConfigurer->exceptionHandling ->
                exceptionHandling.authenticationEntryPoint(unauthorizedHandler))
            .sessionManagement(sessionManagementConfigurer->sessionManagement ->
                sessionManagement.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
            .authorizeHttpRequests(authorizationManagerRequestMatchers ->
                authorizeRequests
                    .requestMatchers(patterns: "/api/auth/**").permitAll()
                    .requestMatchers(patterns: "/api/**").authenticated()
                    .requestMatchers(patterns: "/api/**").permitAll()
                    .requestMatchers(patterns: "/v3/**").permitAll()
                    .requestMatchers(patterns: "/swagger-ui/**").permitAll()
                    .anyRequest().authenticated()
                )
            .addFilterBefore(authenticationJwtTokenFilter(), UsernamePasswordAuthenticationFilter.class);

        return http.build();
    }

    @Bean
    public AuthenticationManager authenticationManager(AuthenticationConfiguration authenticationConfiguration) throws
        return authenticationConfiguration.getAuthenticationManager();
    }
}

```

Ilustración 23. seguridad web

Así al concluir las pruebas y resolver los errores detectados, estabilizamos el sistema para garantizar su robustez y disponibilidad con:

- Errores de visualización: Me asegure mediante pruebas, que los datos se mostraran correctamente en todas las pantallas, ya que los datos pertenecen únicamente a la base de datos si ella no funcionaría la aplicación.
- Optimización de rendimiento: optimice los controladores para las consultas al backend para reducir los tiempos de espera, al ajustar sus peticiones de entradas y salidas.
- Manejo de errores: en el desarrollo maneje mejor los mensajes de error para que sean más claros y útiles para los usuarios.

Conclusiones

Después de todo el largo proceso de desarrollo logre culminar el objetivo. La aplicación móvil está lista y funcionando con éxito. Y gracias a este tiempo he aprendido mucho, alcanzando

metas importantes en busca de cumplir con el objetivo planteado, lo que me permite hablar de los logros alcanzados:

- Haber creado una herramienta digital robusta que permite a los usuarios diseñar y gestionar mapas de empatía de manera efectiva. Gracias a la implementación del modelo de canvas digital, ahora para las personas que hagan uso de mi aplicación sera más fácil analizar audiencias y estudiar mercados de una forma visual y organizada.
- Desde el principio, el querer que la aplicación fuera fácil de usar para todos, desde principiantes hasta expertos, logro que mi resultado fuera una interfaz amigable y fluida, que permite trabajar sin complicaciones.
- Con esta digitalización, busco ayudar a reducir significativamente el uso de materiales físicos, porque la app facilita el acceso a la herramienta desde cualquier lugar, y también representa un impacto positivo en el medio ambiente.
- Al diseñar la arquitectura del software con un enfoque modular y escalable, significa que la aplicación está lista para seguir creciendo, por lo que puedo agregar nuevas funciones sin afectar su rendimiento y estabilidad, asegurando que siga siendo útil y eficiente a lo en el futuro con nuevas características.
- Aunque trabaje en idear una idea para que la aplicación fuera inclusiva, las limitaciones de mi conocimiento en la implementación de ciertos estándares de accesibilidad, no se pudo implementar por el momento. Esto me deja una gran lección y una meta clara para el futuro, mejorar aún más la experiencia para todos los usuarios.
- Al momento que hice uso y aplique la metodología Mobile-D pude asegurarme de que el proceso fuera dinámico y adaptativo. Gracias al enfoque que presenta esta metodología,

pude ajustar detalles y mejorar el rendimiento, logrando un producto final que realmente responde a las necesidades del público que tenía en mente.

Recomendaciones

1. Es clave seguir escuchando a los usuarios, actualizando la aplicación y agregando funciones que hagan la experiencia aún mejor, por lo que el trabajo no termina aquí.
2. Mejorar las herramientas de accesibilidad será una prioridad en futuras versiones, asegurando que la app sea inclusiva para todos, porque no queremos que nadie se quede fuera.
3. La digitalización es un gran avance, pero podemos hacer más, por lo que evaluar la huella ecológica de la aplicación nos permitirá encontrar nuevas formas de reducir su impacto y hacerla aún más sostenible.

Referencias

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley.

Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.

React Native Documentation. (2025). *Getting Started*.
<https://reactnative.dev/docs/getting-started>

Spring Boot Documentation. (2025). *Spring Boot*. <https://spring.io/projects/spring-boot>

Docker Documentation. (2025). *Docker Docs*. <https://docs.docker.com/>

MySQL Documentation. (2025). *MySQL Docs*. <https://dev.mysql.com/doc/>

GitHub Guides. (2025). *GitHub Guides*. <https://guides.github.com/>

Agile Alliance. (2025). *Agile Alliance*. <https://www.agilealliance.org/>

JSON Web Tokens (JWT) Introduction. (2025). *JWT Introduction*.
<https://jwt.io/introduction/>

Axios Documentation. (2025). *Axios Docs*. <https://axios-http.com/docs/intro>

Visual Studio Code Documentation. (2025). *Visual Studio Code Docs*.
<https://code.visualstudio.com/docs>

Mobile-D Methodology Overview. (2025). *Mobile-D*. <https://www.mobiled.org/>

Baeldung (2025). *How to Build a RESTful API with Spring Boot*.
<https://www.baeldung.com/rest-with-spring-series>

Auth0(2025).*Introduction to JWT (JSON Web Tokens)*.<https://auth0.com/docs/secure/tokens/json-web-tokens>

Atlassian(2025).*Agile Project Management with Kanban*.
<https://www.atlassian.com/agile/kanban>