

****

**“ANIMATI APP”**

**Aplicación móvil de ventas, de productos personalizados.**

**Creadores:**

**Ferace, Luciana**

**Palacio, Lautaro**

**Arevalo, Luciano**

**Burrut, Yasmin**

**Palomeque, Dalila**

**De Nieto, Gabriel**

**Salinas, Juan**

**Samsam, Emir**

**Christiansen, Malvina**

**Rochetti, Uriel**

*Año 2024*

# ÍNDICE

[Ficha del documento](#_heading=h.k8e7iubkbjy1) 1

[Contenido](#_heading=h.8otus6er2h1o) 1

[Introducción](#_heading=h.b3jk81nflp95) 1

[Propósito](#_heading=h.kl4bdikzkj00) 1

[Alcance](#_heading=h.su9tbamj5lwf) 1

[Personal involucrado](#_heading=h.blq6ejvo12g5) 2

[Definiciones, acrónimos y abreviaturas](#_heading=h.15829rolcmgz) 3

[Referencias](#_heading=h.8ck493hg9tup) 4

[Resumen](#_heading=h.s5lt7zte993u) 4

[Descripción general](#_heading=h.l931jnv1yrsk) 4

[Perspectiva del producto](#_heading=h.kfno2w9j9oc3) 4

[Características de los usuarios](#_heading=h.1tejk5hcmlvm) 5

[Restricciones](#_heading=h.xik16bum05gd) 5

[Requisitos específicos](#_heading=h.zhfvsehf8znb) 5

[Product Backlog:](#_heading=h.3xii5mkoobft) 6

[US:](#_heading=h.dujknmh3ljr2) 6

[TK:](#_heading=h.m2xg83ekuaoc) 7

[Sprints](#_heading=h.wd96swq5agt1) 9

# 

# 

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Verificado dep. Calidad.** |
| --- | --- | --- | --- |
| 2024 | 1.0.0 | Team AnimatiWA | Animati Testing Team |

# Contenido

## Introducción

Somos un grupo de estudiantes del ISPC, cohorte 2023, y estamos desarrollando el proyecto **Tienda de Compra Animati App**. Esta aplicación móvil tiene como objetivo ofrecer una experiencia de compra sencilla y accesible para nuestros usuarios, permitiéndoles adquirir productos personalizados de alta calidad.

## Propósito

El propósito de **Animati App** es brindar a los usuarios una plataforma digital eficiente para la compra de productos personalizados. Queremos mejorar la experiencia de compra online ofreciendo una variedad de artículos creativos y únicos, que satisfagan las necesidades y gustos específicos de nuestros clientes.

## Alcance

El alcance del proyecto incluye el desarrollo de una aplicación móvil que permita a los usuarios:

* Registrarse e iniciar sesión.
* Navegar y buscar productos personalizados.
* Añadir productos a un carrito de compras.
* Realizar el proceso de pago de manera segura.
* Acceder a su historial de compras.
* Gestionar su perfil de usuario.

La aplicación estará disponible para dispositivos Android y contará con una interfaz intuitiva, diseñada para ofrecer una experiencia de usuario óptima.

## Personal involucrado

| **Nombre** | Luciana Ferace |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Facilitadora Team Develop y Team Testing |
| **Información de contacto** | feraceluciana@gmail.com |

| **Nombre** | Yasmin Burrut |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | nawsper@gmail.com |

| **Nombre** | Emir Samsam |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | ea.samsam@gmail.com |

| **Nombre** | Luciano Javier Arevalo |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | luciann2121@gmail.com |

| **Nombre** | Dalila Macarena Palomeque |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop y Team Testing |
| **Información de contacto** | dalilapalomeque@mi.unc.edu.ar |

| **Nombre** | Lautaro Palacio |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | pala797@gmail.com |

| **Nombre** | Malvina Christiansen |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | malvi\_989@hotmail.com |

| **Nombre** | Gabriel De Nieto |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Scrum Master, Team Develop y Team Testing |
| **Información de contacto** | gabrielnietodejesus@gmail.com |

| **Nombre** | Uriel Rochetti |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | urielrochetti@gmail.com |

| **Nombre** | Juan José Salinas |
| --- | --- |
| **Rol** | Developer |
| **Categoría Profesional** | Estudiante |
| **Responsabilidad** | Team Develop |
| **Información de contacto** | juansalinaa@gmail.com |

## Definiciones, acrónimos y abreviaturas

| ***Nombre*** | ***Descripción*** |
| --- | --- |
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **SIS-I** | Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos |
| **ERS** | Especificación de Requisitos Software |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |
| **FTP** | Protocolo de Transferencia de Archivos |
| **Moodle** | Aula Virtual |
| **US** | Historias de usuario |
| **TK** | Tareas a realizar |

## Referencias

| **Título del Documento** | **Referencia** |
| --- | --- |
| Standard IEEE 830 - 1998 | IEEE |

## Resumen

El proyecto Tienda de Compra Animati App tiene como objetivo crear una aplicación móvil que facilite la compra de productos personalizados, brindando una experiencia de usuario intuitiva y segura. Con un enfoque en la personalización y la facilidad de uso, Animati App busca satisfacer las demandas del mercado de e-commerce ofreciendo una amplia gama de productos únicos. El proyecto está alineado con las tendencias actuales del comercio electrónico y se desarrollará cumpliendo con los requisitos establecidos y dentro del cronograma fijado. El equipo de desarrollo se compromete a entregar una solución completa y funcional que cumpla con las expectativas de los usuarios y las necesidades del mercado.

## Descripción general

**Animati App** es una tienda de compras móvil que ofrece una amplia gama de productos personalizados, incluyendo separadores, sets de stickers, cubecrafts, cajas personalizadas, colgantes, blocks de notas, llaveros, y más. Los usuarios pueden explorar el catálogo, aplicar filtros de búsqueda, y realizar compras directamente desde la aplicación.

## Perspectiva del producto

**Animati App** se enmarca dentro de la tendencia creciente del comercio electrónico y la personalización de productos. Su diseño y funcionalidad están orientados a facilitar la compra de artículos únicos y personalizados, brindando a los usuarios una plataforma moderna y confiable para realizar sus adquisiciones.

## 

## Características de los usuarios

| **Tipo de usuario** | Administrador |
| --- | --- |
| **Formación** | Manejo de herramientas informáticas |
| **Actividades** | Control y manejo del sistema en general |

| **Tipo de usuario** | Visitante |
| --- | --- |
| **Formación** | Manejo de entornos App |
| **Actividades** | Observa e indaga información sobre productos y servicios disponibles y compras que puede realizar en la tienda virtual de papelería. |

*.*

## Restricciones

- La aplicación debe ser compatible con la versión estable actual de Android.

- La seguridad de los datos de los usuarios es prioritaria, por lo que se implementarán medidas de encriptación para proteger la información personal y las transacciones.

- El proyecto deberá completarse dentro del cronograma establecido por el ISPC.

## Requisitos específicos

- **Registro e inicio de sesión:** El usuario debe poder registrarse con un correo electrónico y contraseña, y poder iniciar sesión de manera segura.

- **Navegación de productos:** La aplicación debe permitir la visualización y búsqueda de productos con opciones de filtrado.

- **Carrito de compras:** El usuario debe poder añadir productos a su carrito, revisar los artículos seleccionados y proceder al pago.

- **Proceso de pago:** Se debe integrar una pasarela de pago segura para procesar las compras.

- **Historial de compras:** Los usuarios deben poder acceder a un historial de todas sus compras realizadas.

- **Gestión de perfil:** Los usuarios deben poder editar su perfil para mantener su información actualizada.

## Product Backlog:

### US:

**#US1:** Registro de Usuarios: Como usuario, quiero poder registrarme en la aplicación para poder acceder a las funciones de compra.

**#US2:** Inicio de Sesión: Como usuario registrado, quiero poder iniciar sesión en la aplicación para acceder a mi cuenta.

**#US3:** Recuperación de contraseña: Como usuario, quiero poder recuperar mi contraseña en caso de olvidarla para seguir usando mi cuenta.

**#US4:** Navegación por Productos: Como usuario, quiero poder navegar y buscar productos en la tienda para encontrar lo que deseo comprar.

**#US5:** Detalle de Producto: Como usuario, quiero poder ver los detalles de un producto para tomar una decisión de compra informada.

**#US6:** Añadir Producto al Carrito: Como usuario, quiero poder añadir productos a un carrito de compras para comprarlos más tarde.

**#US7:** Visualizar Carrito de Compras: Como usuario, quiero poder ver los productos en mi carrito de compras para revisarlos antes de hacer una compra.

**#US8:** Proceso de Pago: Como usuario, quiero poder realizar el pago de los productos en mi carrito para completar mi compra.

**#US9:** Historial de Compras: Como usuario, quiero poder ver mi historial de compras para revisar las transacciones anteriores.

**#US10:** Gestión de Perfil de Usuario: Como usuario, quiero poder editar mi perfil para mantener mi información actualizada.

**#US11:** Como desarrollador, quiero contar con que el back-end previamente utilizado en el proyecto web se pueda utilizar en la aplicación mobile.

### 

### TK:

**#TK1:** Registro de Usuario

**-TK1.1:** Diseñar la interfaz de registro de usuario.

**-TK1.2:** Implementar la funcionalidad de registro con validación de datos.

**-TK1.3:** Conectar la pantalla de registro con la base de datos.

**-TK1.4:** Implementar notificaciones de error en caso de fallos de registro.

**#TK2:** Inicio de Sesión

**-TK2.1:** Diseñar la interfaz de inicio de sesión.

**-TK2.2**: Implementar la funcionalidad de inicio de sesión.

**-TK2.3:** Integrar el inicio de sesión con la base de datos.

**-TK2.4:** Implementar el manejo de sesiones y cookies.

**#TK3:** Recuperación de Contraseña

**-TK3.1:** Diseñar la interfaz de recuperación de contraseña.

**-TK3.2:** Implementar la funcionalidad de recuperación de contraseña.

**-TK3.3:** Configurar el envío de correos electrónicos de recuperación.

**-TK3.4:** Implementar la validación del token de recuperación.

**#TK4:** Navegación por Productos

**-TK4.1:** Diseñar la interfaz de listado de productos.

**-TK4.2:** Implementar la funcionalidad de búsqueda y filtrado.

**-TK4.3:** Integrar la pantalla de productos con la base de datos.

**-TK4.4:** Implementar la paginación para la lista de productos.

**#TK5:** Detalle de Producto

**-TK5.1:** Diseñar la interfaz de detalle de producto.

**-TK5.2:** Implementar la funcionalidad de visualización de detalles.

**-TK5.3:** Conectar la pantalla de detalles con la base de datos.

**-TK5.4:** Implementar la sección de opiniones de usuarios.

**#TK6:** Añadir Producto al Carrito

**-TK6.1:** Diseñar la interfaz de carrito de compras.

**-TK6.2:** Implementar la funcionalidad de añadir productos al carrito.

**-TK6.3:** Conectar el carrito con la base de datos para almacenar los productos añadidos.

**-TK6.4:** Implementar notificaciones de confirmación de adición al carrito.

**#TK7:** Visualizar Carrito de Compras

**-TK7.1:** Diseñar la interfaz de visualización del carrito.

**-TK7.2:** Implementar la funcionalidad de modificación y eliminación de productos en el carrito.

**-TK7.3:** Integrar la pantalla del carrito con la base de datos.

**-TK7.4:** Implementar el cálculo automático del total de la compra.

**#TK8:** Proceso de Pago

**-TK8.1:** Diseñar la interfaz del proceso de pago.

**-TK8.2:** Implementar la funcionalidad de selección de método de pago.

**-TK8.3:** Integrar el proceso de pago con la API de un proveedor de pagos.

**-TK8.4:** Implementar la confirmación y notificación de compra exitosa.

**#TK9:** Historial de Compras

**-TK9.1:** Diseñar la interfaz del historial de compras.

**-TK9.2:** Implementar la funcionalidad de visualización del historial.

**-TK9.3:** Conectar el historial con la base de datos para mostrar las compras anteriores.

**-TK9.4:** Implementar filtros de búsqueda en el historial.

**#TK10:** Gestión de Perfil de Usuario

**-TK10.1:** Diseñar la interfaz de edición de perfil.

**-TK10.2:** Implementar la funcionalidad de actualización de datos del perfil.

**-TK10.3:** Integrar la edición de perfil con la base de datos.

**-TK10.4:** Implementar la validación de datos y notificaciones de éxito o error.

**#TK11:** Identificar y documentar los roles de Scrum Master y del equipo de desarrollo en el Wiki del proyecto.

**#TK12:** Configurar el proyecto Kanban en GitHub.

**#TK13:** Crear y agregar Issues en GitHub para cada Historia de Usuario.

**#TK14:** Documentar el proyecto según el estándar IEEE 830, incluyendo requisitos adicionales.

**#TK15:** Revisar y alinear el contenido y la disposición de la tienda con el nuevo desarrollo de e-commerce.

**#TK16:** Establecer la estructura del Wiki en GitHub y registrar las ceremonias Scrum.

**#TK17:** Crear Milestones en GitHub para los Sprints con fechas de inicio y fin.

**#TK18:** Implementar y mantener el tablero Kanban con las columnas: To Do, In Progress, testing test, testing Passed, Done.

**#TK19:** Revision del back antiguo

**-TK19.1:** Configurar el back-end antiguo para que se ejecute correctamente. Incluida la creación de un entorno virtual para DJango.

**-TK19.2:** Analizar los los modelos, vistas y serializers preexistentes para definir que uso tiene cada endpoint del back-end.

**-TK19.3:** Hacer las correcciones necesarias para que cada endpoint funcione correctamente y mantengan un formato estandarizado.

**-TK19.4:** Testear los endpoints para comprobar su funcionamiento, y corroborar el formato de la peticion que reciben y la información que devuelven.

**#TK 20:** Diseño del Proyecto en Figma

**-TK20.1:** Crear el diseño inicial de la aplicación en Figma.

**-TK20.2:** Definir el estilo visual general (colores, tipografías, iconos).

**-TK20.3:** Diseñar las pantallas principales de la aplicación (Inicio, Registro, Contacto, etc.).

**-TK20.4:** Compartir los enlaces y especificaciones del diseño con el equipo de desarrollo.

## Sprints

| **N° de sprint** | **00** |
| --- | --- |
| **Sprint Backlog** | 1-Identificar y documentar los roles de Scrum Master y del equipo de desarrollo en el Wiki del proyecto.  2-Configurar el proyecto Kanban en GitHub.  3-Crear y agregar Issues en GitHub para cada Historia de Usuario.  4-Documentar el proyecto según el estándar IEEE 830, incluyendo requisitos adicionales.  5-Revisar y alinear el contenido y la disposición de la tienda con el nuevo desarrollo de e-commerce.  6-Establecer la estructura del Wiki en GitHub y registrar las ceremonias Scrum.  7-Crear Milestones en GitHub para los Sprints con fechas de inicio y fin.  8-Implementar y mantener el tablero Kanban con las columnas:To Do, In Progress, testing test, testing Passed, Done. |
| **Responsabilidades** | **Descripción del Trabajo en Equipo:**  El equipo llevó a cabo las tareas del proyecto de manera colaborativa y eficiente. A continuación, se detalla el proceso seguido:   1. **Identificación de Roles y Documentación:**    * El equipo definió y documentó los roles del Scrum Master y los miembros del equipo en el Wiki del proyecto. Esta tarea se realizó en conjunto para asegurar que cada miembro entendiera sus responsabilidades y contribuyera a la documentación de manera precisa. 2. **Configuración del Proyecto Kanban:**    * Se configuró el proyecto Kanban en GitHub para gestionar el flujo de trabajo. Todos los miembros participaron en la configuración inicial, estableciendo las columnas y asegurando que el tablero reflejara las necesidades del proyecto. 3. **Creación y Gestión de Issues:**    * Los Issues se crearon y añadieron a GitHub para cada Historia de Usuario. El equipo colaboró en la redacción y categorización de los Issues para garantizar que se alinearan con los requisitos del proyecto. 4. **Documentación IEEE 830:**    * La documentación del proyecto se realizó siguiendo el estándar IEEE 830, con aportes de todos los miembros para asegurar que se cubrieran todos los aspectos necesarios y se incluyeran requisitos adicionales. 5. **Revisión del Contenido de la Tienda:**    * Se revisó el contenido y la disposición de la tienda para alinearlo con el nuevo desarrollo de e-commerce. El equipo trabajó en conjunto para evaluar y ajustar el diseño y las funcionalidades de la tienda. 6. **Estructuración del Wiki:**    * Se estableció la estructura del Wiki en GitHub, incluyendo el registro de ceremonias Scrum. Cada miembro del equipo contribuyó a la organización del contenido y al registro de las ceremonias, asegurando que toda la información relevante estuviera documentada. 7. **Creación de Milestones:**    * Se crearon Milestones en GitHub para los Sprints, con fechas de inicio y fin. El equipo colaboró en la planificación de los Milestones para asegurar que el cronograma del proyecto se mantuviera en línea con los objetivos establecidos. 8. **Implementación del Tablero Kanban:**    * El equipo implementó y mantuvo el tablero Kanban con las columnas necesarias: Product Backlog, Por hacer, En proceso, Finalizado. Todos los miembros participaron en la actualización del tablero para reflejar el estado actual del proyecto.   A través de una comunicación constante y una colaboración efectiva, el equipo logró completar estas tareas de manera organizada y alineada con los objetivos del proyecto. |
| **Calendario** | Start Date = 15/08/24 - End Date = 29/08/24 |
| **Inconvenientes:** | |

## Sprints

| **N° de sprint** | **01** |
| --- | --- |
| **Sprint Backlog** | 1. \*\*Crear el Product Backlog\*\* detallando las nuevas características y requerimientos para la aplicación.  2. \*\*Generación de layouts para las Activity\*\*:  - 2.1. Crear diseños preliminares de las interfaces.  - 2.2. Implementar los diseños en Android Studio.  3. \*\*Revisión de Backend\*\*:  - 3.1. Revisar las APIs ya existentes y actualizar las necesarias.  - 3.2. Integrar los nuevos endpoints requeridos.  4. \*\*Generación de Testing Design\*\*:  - 4.1. Diseñar pruebas unitarias y de integración para las funcionalidades del backend.  5. \*\*Creación de un plan de seguridad\*\*:  - 5.1. Detallar las políticas de seguridad a implementar basadas en buenas prácticas.  - 5.2. Especificar las vulnerabilidades que las políticas van a prevenir (ejemplo: ataques de inyección SQL, CSRF).  6. \*\*Diseñar la autenticación con Auth0 o JWT\*\*:  - 6.1. Fundamentar la elección entre Auth0 o JWT basándote en los requisitos del proyecto (seguridad, facilidad de implementación, escalabilidad).  7. \*\*Revisión del modelo relacional\*\*:  - 7.1. Actualizar el modelo de base de datos según los nuevos requerimientos.  - 7.2. Realizar ajustes en el diseño de las tablas si es necesario.  8. \*\*Actualización de las APIs de Django\*\*:  - 8.1. Implementar nuevas APIs para soportar las funcionalidades del frontend.  - 8.2. Revisar la documentación de las APIs para asegurar que esté actualizada. |
| **Responsabilidades** | **Descripción del Trabajo en Equipo:**  El equipo llevó a cabo las tareas asignadas para el Sprint 01 de manera colaborativa y eficiente. A continuación, se detalla el proceso seguido: 1. Identificación y Documentación del Product Backlog: El equipo trabajó en conjunto para definir y documentar el **Product Backlog**, incluyendo nuevas características y requerimientos para la aplicación. Esta tarea fue crucial para asegurar que el backlog reflejara las necesidades actuales del proyecto. 2. Creación de Layouts para las Activity: El equipo de diseño colaboró en la creación de los layouts preliminares para las distintas **Activity** de la aplicación. Posteriormente, los desarrolladores implementaron los diseños en **Android Studio**, asegurando que se ajustaran a los requisitos establecidos en el backlog. 3. Revisión y Actualización del Backend: Los miembros responsables del backend revisaron las APIs existentes y realizaron las actualizaciones necesarias. Además, se integraron los nuevos endpoints requeridos para cumplir con las funcionalidades previstas en este sprint. 4. Diseño de Pruebas (Testing Design): El equipo de testers diseñó las pruebas unitarias y de integración necesarias para verificar el correcto funcionamiento de las nuevas funcionalidades del backend. Este proceso se realizó de manera coordinada con los desarrolladores. 5. Plan de Seguridad: Se diseñó un plan de seguridad detallado basado en buenas prácticas. El equipo discutió y acordó las políticas de seguridad que se implementarán para prevenir vulnerabilidades comunes, como ataques de inyección SQL y Cross-Site Request Forgery (CSRF). 6. Implementación de la Autenticación: Después de una revisión de las opciones de autenticación, el equipo decidió implementar **JWT** como mecanismo de autenticación por su facilidad de integración y escalabilidad. Se documentaron las razones detrás de esta elección en el Wiki del proyecto. 7. Revisión del Modelo Relacional: Se realizó una actualización del modelo relacional para reflejar los nuevos requerimientos del proyecto. El equipo de backend, en colaboración con los diseñadores de bases de datos, hizo ajustes en el diseño de las tablas cuando fue necesario. 8. Actualización de APIs en Django: El equipo también trabajó en la actualización de las **APIs de Django**, implementando nuevas funcionalidades para soportar el frontend. Además, se revisó y actualizó la documentación de las APIs para asegurar que estuviera alineada con el desarrollo actual. 9. Herramientas de Seguimiento y Gestión: El equipo configuró el tablero **Kanban** en GitHub, añadiendo todas las tareas en las columnas **To Do**, **In Progress**, **Testing** y **Done**. También se crearon los Issues correspondientes a cada tarea e historia de usuario, permitiendo una gestión más clara del flujo de trabajo. 10. Creación de Milestones: Para asegurar un buen seguimiento del cronograma del proyecto, el equipo creó y ajustó los **Milestones** del Sprint 01. |
| **Calendario** | Start Date = 30/08/24 - End Date = 18/09/24 |
| **Inconvenientes:**  Durante el desarrollo del Sprint 01, el equipo se encontró con algunos desafíos que fueron resueltos gracias a la colaboración y el enfoque en la mejora continua:   1. **Desajustes de Comunicación Inicial**: En las primeras etapas del sprint, hubo pequeñas dificultades para coordinar las tareas entre los miembros del equipo, lo que ocasionó algunos retrasos en la asignación de responsabilidades. Para solucionarlo, se reforzaron las reuniones diarias y el uso de herramientas como el tablero Kanban en GitHub, mejorando significativamente la sincronización entre los miembros. 2. **Compatibilidad entre Versiones de Herramientas**: Al implementar los layouts en Android Studio, algunos miembros del equipo encontraron problemas relacionados con la compatibilidad de versiones. Se optó por estandarizar las versiones de las herramientas utilizadas, lo que permitió avanzar sin inconvenientes mayores. 3. **Integración de APIs**: Durante la actualización de las APIs de Django, surgieron problemas con la integración de algunos endpoints debido a diferencias en los modelos de datos. Estos inconvenientes se resolvieron mediante reuniones adicionales entre el equipo de backend y frontend, asegurando una mejor comprensión de los requisitos técnicos. 4. **Tiempo Limitado para las Pruebas**: La generación de pruebas unitarias e integrales en el backend tomó más tiempo del previsto debido a la complejidad de algunos escenarios. Para futuras iteraciones, el equipo planea ajustar el cronograma y destinar más tiempo para pruebas, minimizando cualquier posible impacto en el desarrollo. | |