



UNIVERSIDAD
NACIONAL
DE COLOMBIA

PROYECTO FINAL

Ingeniería de Software I

Wulffredo Javier Barco Godoy - wbarco@unal.edu.co
Brahian Camilo Gómez Carvajal - bgomezca@unal.edu.co
Juan David Rivera Buitrago - jriverabu@unal.edu.co
William Darío Vanegas Marín - wvanegas@unal.edu.co

Profesor
Oscar Eduardo Alvarez Rodríguez

Universidad Nacional de Colombia
Facultad de Ingeniería
Ingeniería de Sistemas y Computación
Bogotá, Colombia
31 de Enero de 2025

1. Pre Requerimientos:

Están en el archivo README.md que está en el repositorio de nuestro proyecto, sin embargo aquí se resume lo más elemental de dicho archivo:

Nombre del Proyecto: ChazaUNApp

Descripción del Proyecto: ChazaUNApp es una plataforma pensada y diseñada para la gestión y promoción de pequeños negocios y ventas informales dentro de la Universidad Nacional de Colombia. Permite a los vendedores registrar sus negocios y a los compradores encontrar productos y servicios fácilmente.

Tecnologías utilizadas:

- **Lenguaje de Programación:** Dart.
- **Frameworks:** Flutter.
- **Servicios externos:** Firebase (Autenticación, Base de datos en tiempo real), Google Maps API.

Ícono o Logo del Proyecto:



Imagen 1. Logo del Proyecto. Imagen creada con OpenArt AI (<https://openart.ai/create>).

2. Levantamiento de Requerimientos

Origen de la Idea

La idea de desarrollar **ChazaUNApp** surgió de la observación y análisis de la dinámica de comercio informal dentro del campus de la sede Bogotá de la Universidad Nacional de Colombia. Muchos estudiantes y emprendedores venden productos y servicios en espacios informales o chazas sin contar con una plataforma que facilite su visibilidad y gestión. A través de entrevistas con vendedores y compradores, identificamos que existe una necesidad clara de digitalizar y organizar este tipo de comercio para hacerlo más accesible y eficiente.

La principal problemática que se busca resolver con ChazaUNApp es la falta de un canal de comunicación eficiente entre los vendedores de productos o servicios informales de las chazas y los estudiantes o personal universitario que desean acceder a ellos de manera rápida y confiable.

Proceso de Selección de la Idea

Para la elección del proyecto, el equipo realizó una sesión de brainstorming en la que se propusieron varias ideas. Posteriormente, se evaluaron en función de su viabilidad técnica, impacto en la comunidad y alineación con nuestras habilidades y expectativas de aprendizaje. A través de votaciones y debate entre todos los integrantes del grupo, se concluyó que **ChazaUNApp** era una solución innovadora y con alto potencial de utilidad dentro del entorno universitario. Se asignaron roles iniciales para la investigación y planificación del proyecto, buscando con ello garantizar un trabajo coordinado y eficiente.

Problemáticas que Busca Resolver

Las principales problemáticas identificadas en el comercio informal dentro de la universidad son:

- **Falta de visibilidad:** Los dueños y vendedores de las chazas no tienen una plataforma digital donde promocionar sus productos o servicios.
- **Dificultad en la localización:** Hasta el momento no existe un medio unificado para conocer la ubicación y disponibilidad de los puestos de venta.
- **Falta de confianza y seguridad:** Los compradores potenciales, es decir, los miembros de la comunidad universitaria no cuentan con referencias ni reseñas sobre los vendedores, lo que puede generar incertidumbre al momento de realizar compras.
- **Falta de organización:** Hasta ahora no existe un sistema centralizado que permita una mejor gestión de la oferta y demanda.

Expectativas de los Usuarios Potenciales

A través de una consulta de opinión sobre una propuesta de aplicación como la que se ha venido planteando en este documento a otros compañeros, compañeras y demás miembros de la comunidad universitaria, se destaca y se sintetiza que los usuarios potenciales de **ChazaUNApp** esperan:

- Encontrar fácilmente productos y servicios dentro de la universidad.
- Conocer la ubicación de los vendedores y horarios de atención.
- Acceder a reseñas y calificaciones de otros usuarios para tomar decisiones informadas.
- Tener una interfaz sencilla e intuitiva que facilite la navegación.
- Contar con notificaciones y actualizaciones sobre disponibilidad de productos.

Beneficios Esperados del Proyecto

Para todos los integrantes del equipo, el desarrollo de **ChazaUNApp** representa en conjunto una oportunidad de crecimiento en diversas áreas, tales como:

- Aplicación de conocimientos en desarrollo móvil y uso de **Flutter**.
- Experiencia en gestión de proyectos de software en un entorno real.
- Aprendizaje sobre integración de bases de datos en la nube mediante **Firebase**.
- Desarrollo de habilidades de UX/UI para la creación de una interfaz amigable.
- Impacto positivo en la comunidad universitaria al proporcionar una solución digital innovadora.

Funcionalidades Identificadas

Con base en las problemáticas mencionadas, se han identificado las siguientes funcionalidades clave para la aplicación:

- Registro y autenticación de usuarios.
- Creación y gestión de perfiles de vendedores.
- Geolocalización de puestos de venta en un mapa interactivo.
- Sistema de búsqueda y filtrado de productos/servicios.
- Calificación y reseñas de vendedores por parte de los compradores.
- Notificaciones sobre disponibilidad y promociones.

3. Análisis de Requerimientos

Funcionalidad	Prioridad (MoSCoW)	Esfuerzo (Fibonacci)	Justificación
Registro y autenticación de usuarios	Must	5	Es una funcionalidad crítica para el acceso al sistema ya que requiere integración con Firebase Authentication, configuración de seguridad y pruebas de autenticación.
Creación y gestión de perfiles de vendedores	Must	8	Implica la creación de perfiles con opciones de personalización, subida de imágenes y edición de datos, lo que aumenta su complejidad técnica.

Geolocalización de puestos de venta en un mapa interactivo	Should	13	La implementación de Google Maps API requiere permisos de ubicación, bases de datos para almacenar coordenadas y una UI dinámica, lo que implica una mayor complejidad.
Sistema de búsqueda y filtrado de productos/servicios	Must	8	La búsqueda eficiente es esencial. Su desarrollo implica bases de datos indexadas y optimización de consultas para mejorar la experiencia del usuario.
Calificación y reseñas de vendedores	Should	5	Aunque no es esencial para el funcionamiento inicial, mejora la confiabilidad de la plataforma. Requiere la implementación de una base de datos y lógica de manejo de reseñas.
Notificaciones sobre disponibilidad y promociones	Could	3	Una funcionalidad útil pero no esencial en la primera versión. Su implementación con Firebase Cloud Messaging es factible, pero podría añadirse en una fase posterior.

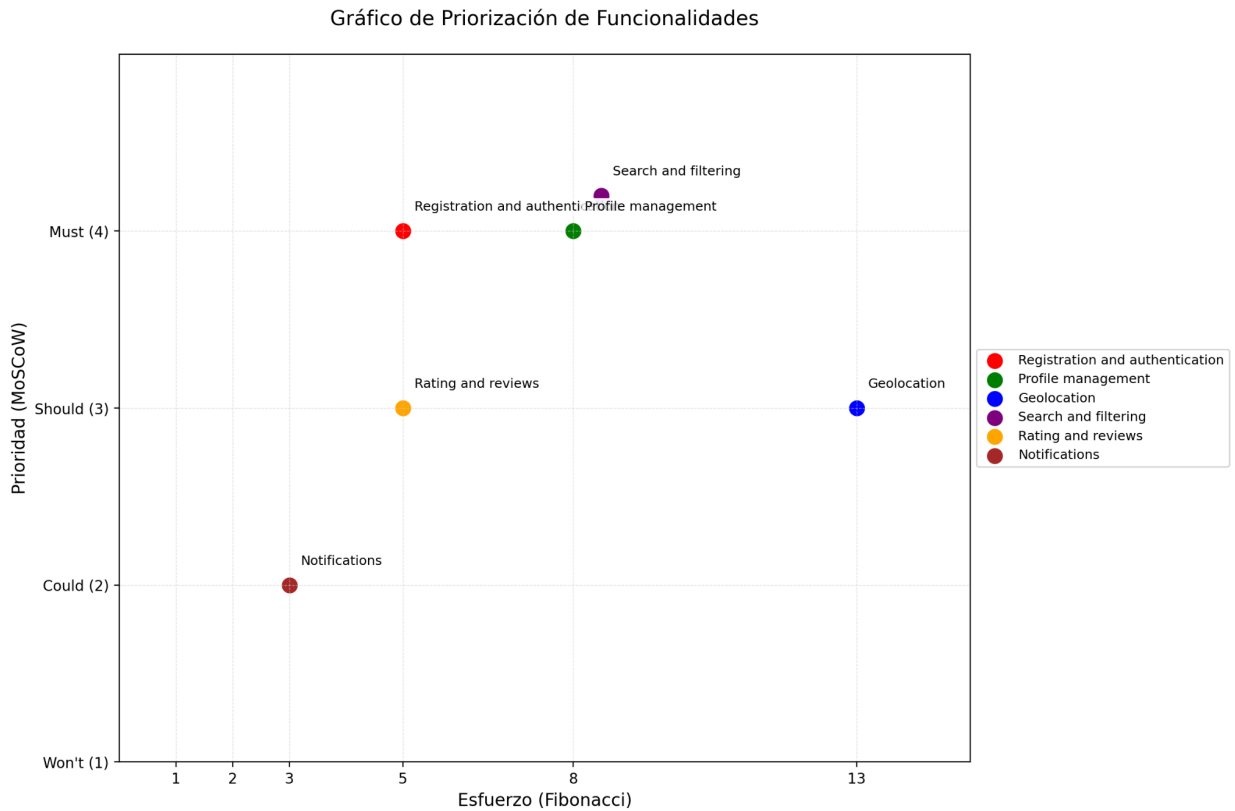


Imagen 2. Gráfica de Priorización de Funcionalidades. Esta gráfica está realizada con base en el análisis de requerimientos hecho previamente. El eje horizontal representa la medida del esfuerzo representada por los números de la serie de Fibonacci, y el eje vertical representa las categorías del método de Moscow.

4. Análisis de Gestión de Software

Tiempo

Se estima que el desarrollo del proyecto tome aproximadamente **4 meses**, los cuáles se distribuyen en las etapas que están descritas en la siguiente tabla:

Fase	Duración
Diseño y planificación	3 semanas
Desarrollo del backend	5 semanas
Desarrollo del frontend	6 semanas
Pruebas y corrección de errores	3 semanas
Implementación y despliegue	3 semanas

Costo

A continuación, se presentan los costos estimados del proyecto:

Concepto	Cantidad	Costo Unitario (USD)	Total (USD)
Desarrollador Junior	1	1,500 / mes	6,000
Desarrollador Senior	1	4,000 / mes	16,000
UX/UI Designer	1	1,500 / mes	6,000
Infraestructura (Servidor y BD)	-	500 / mes	2,000
Licencias y APIs	-	1,000 (único)	1,000
Total	-	-	31,000 USD

Alcance

Se espera que el MVP (Minimum Viable Product) incluya las siguientes funcionalidades:

- Registro y autenticación de usuarios.
- Creación de perfiles de vendedores.
- Geolocalización de los puestos de venta.
- Sistema de búsqueda y filtrado básico.
- Calificación de vendedores.

En consecuencia, las funcionalidades que quedarían fuera del MVP son las siguientes:

- Chat en tiempo real.
- Integración de pagos.
- Notificaciones avanzadas.

5. Casos de Uso del Sistema:

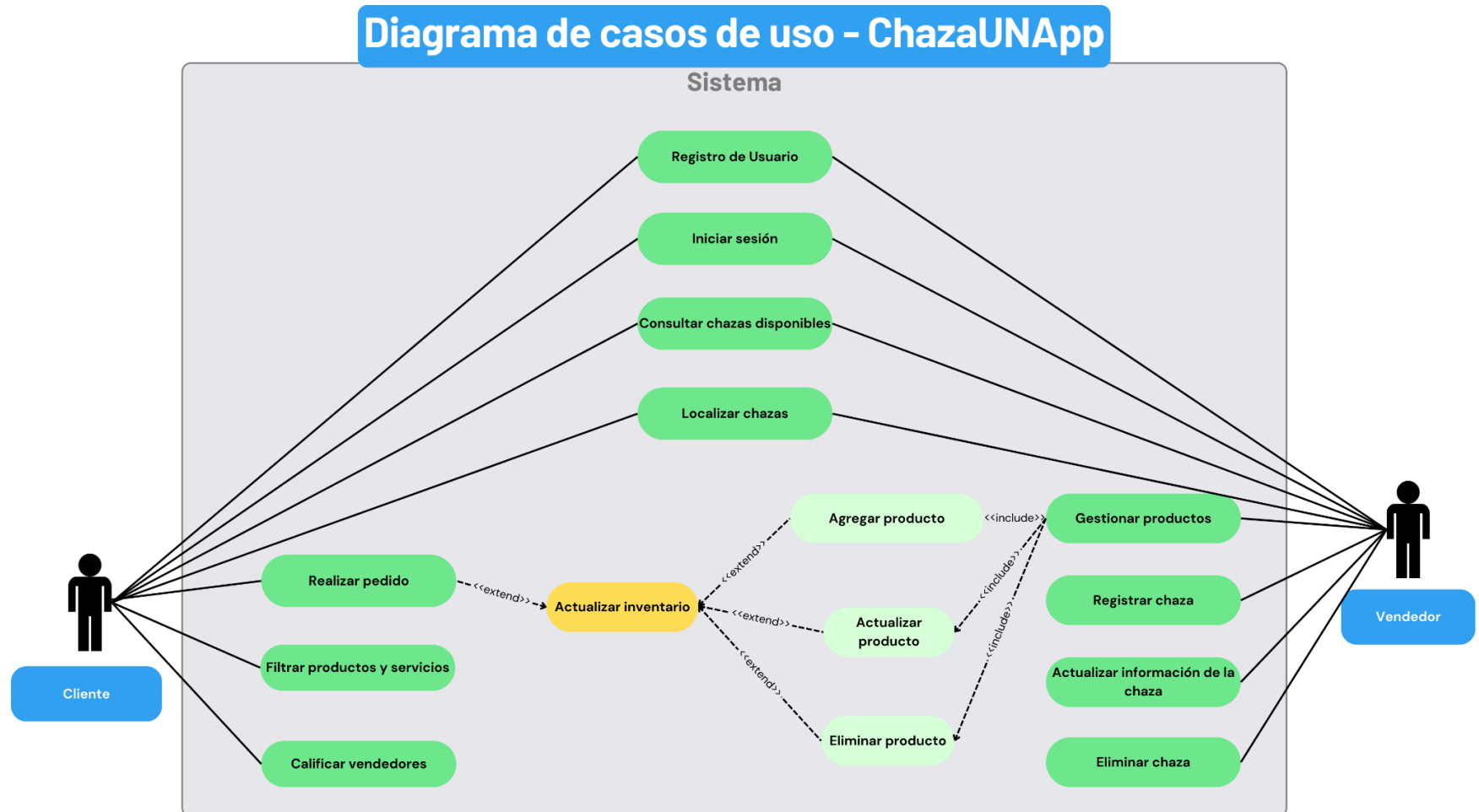


Imagen 3. Diagrama de casos de uso para el proyecto ChazaUNApp

Registro de Usuario
Actor Cliente, Vendedor
Descripción Permite a un usuario registrarse en la aplicación proporcionando sus datos personales.
Precondiciones <ul style="list-style-type: none"> El usuario no debe estar registrado previamente.
Flujo de eventos <ol style="list-style-type: none"> El usuario ingresa a la opción de registro. El usuario ingresa su información personal, que consiste de los siguientes datos: (nombre, correo, contraseña, etc.) <ul style="list-style-type: none"> Nombre. Apellido. Correo. El usuario confirma los datos y envía la solicitud de registro. <ol style="list-style-type: none"> Si el correo ya está registrado, se muestra un mensaje de error (Volver al paso 2). El sistema valida los datos y registra al usuario. El usuario recibe un mensaje de confirmación del registro de sus datos.
Postcondiciones <ul style="list-style-type: none"> El usuario queda registrado en el sistema para que posteriormente pueda iniciar sesión.

Iniciar sesión
Actor Cliente, Vendedor
Descripción Permite a un usuario acceder a su cuenta en la aplicación.
Precondiciones <ul style="list-style-type: none"> El usuario previamente se ha registrado en la aplicación.
Flujo de eventos <ol style="list-style-type: none"> El usuario ingresa a la aplicación ingresando los siguientes datos: <ul style="list-style-type: none"> Correo. Contraseña. El sistema realiza la validación de las credenciales. Si la verificación es correcta, el usuario ingresa a la pantalla principal.

a) Si el proceso es incorrecto, el sistema arroja un mensaje de error (Volver al paso 1).
Postcondiciones <ul style="list-style-type: none"> El usuario ha accedido con los datos de su cuenta a la aplicación.

Registrar chaza
Actor Vendedor
Descripción Permite que un dueño o vendedor pueda registrar su puesto de venta en la aplicación.
Precondiciones <ul style="list-style-type: none"> El vendedor debe tener una cuenta activa en la aplicación. El vendedor ha ingresado con los datos de su cuenta a la aplicación.
Flujo de eventos <ol style="list-style-type: none"> El vendedor accede a la opción "Registrar chaza" que está ubicada en el menú principal. El vendedor ingresa la información del puesto que consta de los siguientes campos: <ul style="list-style-type: none"> Nombre. Ubicación. Categoría. Nombre del dueño El vendedor confirma los datos y envía la solicitud de registro. <ol style="list-style-type: none"> Si falta información en alguno de los campos, o si los datos ingresados son inválidos, se muestra un mensaje de error (Volver al paso 2). El sistema almacena la información registrada.
Postcondiciones <ul style="list-style-type: none"> El puesto de venta o chaza queda registrado en la aplicación. La información de la chaza es accesible a todos los usuarios de la aplicación.

Actualizar información de la chaza
Actor Vendedor
Descripción Permite a un vendedor hacer la modificación de la información sobre una chaza registrada y seleccionada.
Precondiciones <ul style="list-style-type: none"> El vendedor ha ingresado con los datos de su cuenta a la aplicación. El vendedor ha registrado previamente la información de la chaza.

Flujo de eventos

1. El vendedor accede a la sección con nombre tentativo "Mis puestos" ubicada en el menú principal.
2. El vendedor selecciona la chaza.
3. El vendedor actualiza la información, que puede ser de:
 - Nombre.
 - Ubicación.
 - Categoría.
 - Nombre del dueño.
4. El vendedor confirma los cambios realizados.
 - a) Si falta información en alguno de los campos, se muestra un mensaje de error (Volver al paso 3).
5. El sistema almacena la información.

Postcondiciones

- La información de la chaza se actualiza correctamente.

Eliminar chaza**Actor**

Vendedor

Descripción

Autoriza al vendedor o dueño a eliminar la información sobre una chaza registrada y seleccionada en la aplicación.

Precondiciones

- La chaza junto con su información están registradas en la aplicación.
- El vendedor ha ingresado con los datos de su cuenta a la aplicación.

Flujo de eventos

1. El vendedor accede a la sección con nombre tentativo "Mis puestos" ubicada en el menú principal.
2. El vendedor selecciona la chaza.
3. El vendedor selecciona la opción de eliminar chaza.
4. El vendedor confirma la eliminación realizada.
5. El sistema elimina el registro de la chaza.

Postcondiciones

- La información de la chaza desaparece de la aplicación.

Consultar chazas disponibles**Actor**

Cliente
Descripción Permite a los usuarios hacer la exploración y la búsqueda de chazas registradas en la aplicación.
Precondiciones <ul style="list-style-type: none"> • Las chazas deben estar registradas en el sistema. • El usuario debe tener una cuenta activa en la aplicación. • El usuario ha ingresado con los datos de su cuenta a la aplicación.
Flujo de eventos <ol style="list-style-type: none"> 1. El usuario accede a la sección con nombre tentativo “Puestos disponibles” ubicada en el menú principal. 2. El usuario puede ver la lista de chazas disponibles. 3. El usuario selecciona alguna de las chazas para consultar más información.
Postcondiciones <ul style="list-style-type: none"> • El usuario obtiene el acceso a la información disponible de las chazas existentes.

Agregar productos
Actor Vendedor
Descripción Permite al vendedor de una chaza poder agregar la información de productos vendidos en la chaza seleccionada.
Precondiciones <ul style="list-style-type: none"> • El vendedor debe tener una cuenta activa en la aplicación. • El vendedor ha ingresado con los datos de su cuenta a la aplicación. • El vendedor debe haber registrado una chaza previamente.
Flujo de eventos <ol style="list-style-type: none"> 1. El vendedor accede a la sección con nombre tentativo “Mis puestos” ubicada en el menú principal. 2. El vendedor selecciona la chaza. 3. El vendedor accede a la sección con nombre tentativo “Gestión de Productos” 4. El vendedor hace el ingreso de la información del producto con base en los siguientes campos: <ul style="list-style-type: none"> • Nombre. • Descripción. • Precio. • Cantidad disponible. • Imágenes del producto. 5. El vendedor confirma el registro de la nueva información.

<p>a) Si falta información en alguno de los campos, o si los datos son inválidos se muestra un mensaje de error (Volver al paso 4).</p> <p>6. El sistema almacena la información del producto con respecto a la chaza seleccionada previamente.</p>
<p>Postcondiciones</p> <ul style="list-style-type: none"> El producto queda publicado en el puesto.

Actualizar productos
<p>Actor</p> <p>Vendedor</p>
<p>Descripción</p> <p>Permite al vendedor de una chaza poder modificar la información de alguno de sus productos vendidos en la chaza seleccionada.</p>
<p>Precondiciones</p> <ul style="list-style-type: none"> El vendedor debe tener una cuenta activa en la aplicación. El vendedor ha ingresado con los datos de su cuenta a la aplicación. El vendedor debe haber registrado una chaza previamente. Hay productos previamente registrados para la chaza seleccionada.
<p>Flujo de eventos</p> <ol style="list-style-type: none"> El vendedor accede a la sección con nombre tentativo “Mis puestos” ubicada en el menú principal. El vendedor selecciona la chaza. El vendedor accede a la sección con nombre tentativo “Gestión de Productos”. El vendedor selecciona el producto a actualizar. El vendedor hace la modificación de la información del producto en alguno o en todos los siguientes campos: <ul style="list-style-type: none"> Nombre. Descripción. Precio. Cantidad disponible. Imágenes del producto. El vendedor confirma la modificación en el sistema. <ol style="list-style-type: none"> Si falta información en alguno de los campos, o si los datos ingresados son inválidos, se muestra un mensaje de error (Volver al paso 5). El sistema almacena la información del producto con respecto a la chaza seleccionada previamente.
<p>Postcondiciones</p> <ul style="list-style-type: none"> El producto queda actualizado con la información ingresada.

Eliminar productos
Actor Vendedor
Descripción Permite al vendedor de una chaza poder eliminar alguno de los productos vendidos en la chaza seleccionada.
Precondiciones <ul style="list-style-type: none"> • El vendedor debe tener una cuenta activa en la aplicación. • El vendedor ha ingresado con los datos de su cuenta a la aplicación. • El vendedor debe haber registrado una chaza previamente. • Hay productos previamente registrados para la chaza seleccionada.
Flujo de eventos <ol style="list-style-type: none"> 1. El vendedor accede a la sección con nombre tentativo “Mis puestos” ubicada en el menú principal. 2. El vendedor selecciona la chaza. 3. El vendedor accede a la sección con nombre tentativo “Gestión de Productos”. 4. El vendedor selecciona el producto a eliminar. 5. El vendedor confirma la eliminación en el sistema. 6. El sistema elimina el registro del producto para la chaza seleccionada.
Postcondiciones <ul style="list-style-type: none"> • La información del producto queda eliminada.

Realizar pedido
Actor Cliente
Descripción Permite a un cliente poder realizar una compra de un producto determinado de una chaza seleccionada.
Precondiciones <ul style="list-style-type: none"> • Debe haber productos registrados y disponibles. • Las chazas deben estar registradas en el sistema. • El usuario debe tener una cuenta activa en la aplicación. • El usuario ha ingresado con los datos de su cuenta a la aplicación.
Flujo de eventos <ol style="list-style-type: none"> 1. El cliente accede a la sección con nombre tentativo “Puestos disponibles” ubicada en el menú principal. 2. El cliente puede ver la lista de chazas disponibles.

3. El cliente selecciona alguna de las chazas disponibles.
4. El cliente accede a la lista de productos disponibles.
5. El cliente selecciona los productos que se requieran y especifica las cantidades.
 - a) Si no hay cantidad suficiente de productos disponibles:
 - I. El sistema notifica al cliente que no hay suficiente disponibilidad del o de los productos requeridos.
 - II. El cliente ajusta la cantidad requerida o puede eliminar productos del pedido. (Volver al paso 5).
6. El cliente revisa el pedido.
 - a) Si el cliente lo requiere, puede ajustar el pedido (Volver al paso 5).
7. El cliente confirma el pedido y escoge el método de pago.
3. El sistema almacena el pedido realizado.

Postcondiciones

- El pedido queda registrado en el sistema.
- El inventario se actualiza de manera automática una vez se realiza el pedido.
- Se envía una notificación al cliente y al vendedor sobre el nuevo pedido.

Localizar chazas

Actor

Vendedor, Cliente

Descripción

- Este caso de uso permite a los usuarios visualizar en un mapa interactivo la ubicación de los diferentes puestos de venta dentro de la Universidad Nacional de Colombia. A través de la API de Google Maps, los vendedores pueden registrar la ubicación de sus chazas y los compradores pueden acceder a esta información en tiempo real para encontrar productos y servicios de manera rápida y eficiente.

Precondiciones

- El comprador debe estar registrado y autenticado en la aplicación.
- El dispositivo debe tener conexión a Internet para acceder a la API de Google Maps.
- Los vendedores deben haber registrado previamente sus puestos de venta en la base de datos de Firebase.

Flujo de eventos

1. El comprador accede a la aplicación ChazaUNApp.
2. El comprador selecciona la opción "Ver mapa de chazas" en la interfaz.
3. El sistema carga el mapa interactivo utilizando la API de Google Maps.
4. El sistema recupera la información de ubicación de los puestos de venta registrados.
5. El sistema muestra en el mapa los puestos de venta con marcadores interactivos.
6. El comprador puede hacer zoom y desplazarse en el mapa para explorar los diferentes puestos de venta.
7. El comprador puede hacer clic en un marcador para ver detalles del puesto de venta

<p>(nombre, productos/servicios ofrecidos, horario de atención, calificaciones y comentarios).</p> <p>8. El comprador puede utilizar el sistema de búsqueda para filtrar los puestos de venta según productos, servicios o categorías.</p>
<p>Postcondiciones</p> <ul style="list-style-type: none"> • El comprador podrá visualizar los puestos de venta disponibles en el mapa. • El comprador podrá acceder a información detallada sobre los puestos de venta. • El sistema actualizará dinámicamente la ubicación de los puestos de venta en tiempo real.

Filtrar productos/servicios
<p>Actor</p> <p>Cliente</p>
<p>Descripción</p> <p>Este caso de uso permite a los compradores buscar productos y servicios dentro de la Universidad Nacional de Colombia a través de un sistema de búsqueda y filtros avanzados. Los resultados se mostrarán de acuerdo con los criterios ingresados por el usuario, permitiendo una experiencia más eficiente para encontrar puestos de venta relevantes.</p>
<p>Precondiciones</p> <ul style="list-style-type: none"> • El comprador debe estar registrado y autenticado en la aplicación. • Debe haber productos y servicios registrados en la base de datos.
<p>Flujo de eventos</p> <ol style="list-style-type: none"> 1. El comprador accede a la aplicación y selecciona la opción "Buscar productos/servicios". 2. El sistema muestra un campo de búsqueda y opciones de filtrado. 3. El comprador ingresa un término en la barra de búsqueda (ejemplo: "Arepas", "Café", "Ropa", etc.). 4. El comprador puede aplicar filtros adicionales, como: <ul style="list-style-type: none"> • Categoría (comida, ropa, tecnología, etc.). • Precio mínimo y máximo. • Horario de disponibilidad. • Calificación del vendedor. 5. El sistema consulta y obtiene los puestos de venta que coinciden con los criterios de búsqueda. 6. El sistema muestra los resultados en forma de lista y/o mapa interactivo. 7. El comprador puede seleccionar un resultado para ver información detallada sobre el puesto de venta, productos, horarios y ubicación. 8. El comprador puede contactar al vendedor o marcarlo como favorito.

Postcondiciones

- El comprador podrá visualizar productos y servicios según los criterios de búsqueda.
- El comprador podrá acceder a información detallada de los puestos de venta que ofrecen los productos buscados.
- El sistema actualizará dinámicamente los resultados según los filtros aplicados.

Calificar vendedores

Actor

Comprador
Vendedor

Descripción

Este caso de uso permite a los compradores calificar y dejar reseñas sobre los vendedores en la plataforma, ayudando a otros usuarios a tomar decisiones informadas sobre dónde comprar productos y servicios dentro de la Universidad Nacional de Colombia. Los vendedores podrán visualizar los comentarios y mejorar su servicio en función del feedback recibido.

Precondiciones

- El comprador debe estar registrado y autenticado en la aplicación.
- El comprador debe haber realizado una compra con el vendedor para poder calificarlo.
- Debe existir una base de datos con vendedores registrados.

Flujo de eventos

1. El comprador accede a la aplicación y busca un puesto de venta.
2. El comprador selecciona un vendedor para ver su perfil.
3. El sistema muestra la información del vendedor, incluyendo calificaciones y reseñas previas.
4. El comprador selecciona la opción "Calificar y dejar reseña".
5. El sistema muestra un formulario donde el comprador puede:
 - Asignar una calificación en estrellas (1 a 5).
 - Escribir un comentario sobre su experiencia con el vendedor.
6. El comprador completa la reseña y presiona "Enviar".
7. El sistema almacena la calificación y la reseña en Firebase.
8. El sistema actualiza la calificación promedio del vendedor en su perfil.
9. El vendedor puede visualizar las nuevas calificaciones y reseñas en su perfil.

Postcondiciones

- La calificación promedio del vendedor se actualiza en su perfil.
- La reseña queda visible para otros compradores.
- El vendedor puede visualizar los comentarios recibidos.

6. Historias de Usuario

Historia de Usuario #1: Registro y Autenticación

Anexo de Documentos Relacionados:

- Sección 2 de este documento: Gestión de Perfiles

Descripción conceptual

Módulo	Autenticación y Registro
Descripción de la(s) funcionalidad(es) requerida(s):	Permite a los usuarios registrarse o iniciar sesión en la aplicación para acceder a sus funcionalidades. Se validará la autenticación mediante correo y contraseña, asegurando que los datos sean correctos antes de permitir el acceso.

Descripción técnica

Backend

1. Registro de Usuario

URL	Método	Código html
localhost:8080/auth	POST	200, 422
Caso de uso técnico Cuando un vendedor intenta crear su perfil, si los datos son correctos, se retorna un código 200. Si los datos son incorrectos o faltan campos obligatorios, se retorna un código 422.		
Datos de entrada 200: { "nombre": "Juan Pérez", "email": "usuario@ejemplo.com", "password": "password123" }	Datos de salida 200: { "status": "success", "data": { "usuario_id": 1, "nombre": "Juan Pérez" } }	
422: { "email": "usuario@ejemplo.com" }	422: { "status": "error", "message": "La contraseña es obligatoria" }	

--	--

2. Inicio de Sesión

URL	Método	Código html
localhost:8080/auth/login	POST	200, 401, 422
Caso de uso técnico Cuando un usuario intenta iniciar sesión, si las credenciales son correctas, se retorna un código 200 con los datos del usuario autenticado. Si las credenciales son incorrectas, se retorna un código 401. Si faltan campos obligatorios, se retorna un código 422.		
Datos de entrada 200: <pre>{ "email": "usuario@ejemplo.com", "password": "password123" }</pre>		Datos de salida 200: <pre>{ "status": "success", "token": "abc123", "usuario": { "usuario_id": 1, "nombre": "Juan Pérez" } }</pre>
401: <pre>{ "perfil_id": "" }</pre>		401: <pre>{ "status": "error", "message": "El ID del perfil es obligatorio" }</pre>
422: <pre>{ "email": "usuario@ejemplo.com" }</pre>		422: <pre>{ "status": "error", "message": "La contraseña es obligatoria" }</pre>

3. Recuperación de Contraseña

URL	Método	Código html
localhost:8080/auth/recover	POST	200, 422
Caso de uso técnico		

Cuando un usuario solicita recuperar su contraseña, si el correo está registrado en el sistema, se retorna un código 200 con un mensaje de confirmación. Si el correo no está registrado o no se ingresa, se retorna un código 422.

Datos de entrada	Datos de salida
200: { "email": "usuario@ejemplo.com" }	200: { "status": "success", "message": "Se ha enviado un correo con instrucciones para recuperar la contraseña" }
422: { "email": "" }	422: { "status": "error", "message": "Debe ingresar un correo válido" }

Frontend

Interacción esperada:

- El usuario ingresa a la pantalla de inicio de sesión mediante un botón en la aplicación.
- El usuario puede registrarse completando un formulario con su información personal.
- En caso de olvidar su contraseña, puede solicitar un enlace de recuperación.

Mockups/Prototipos:

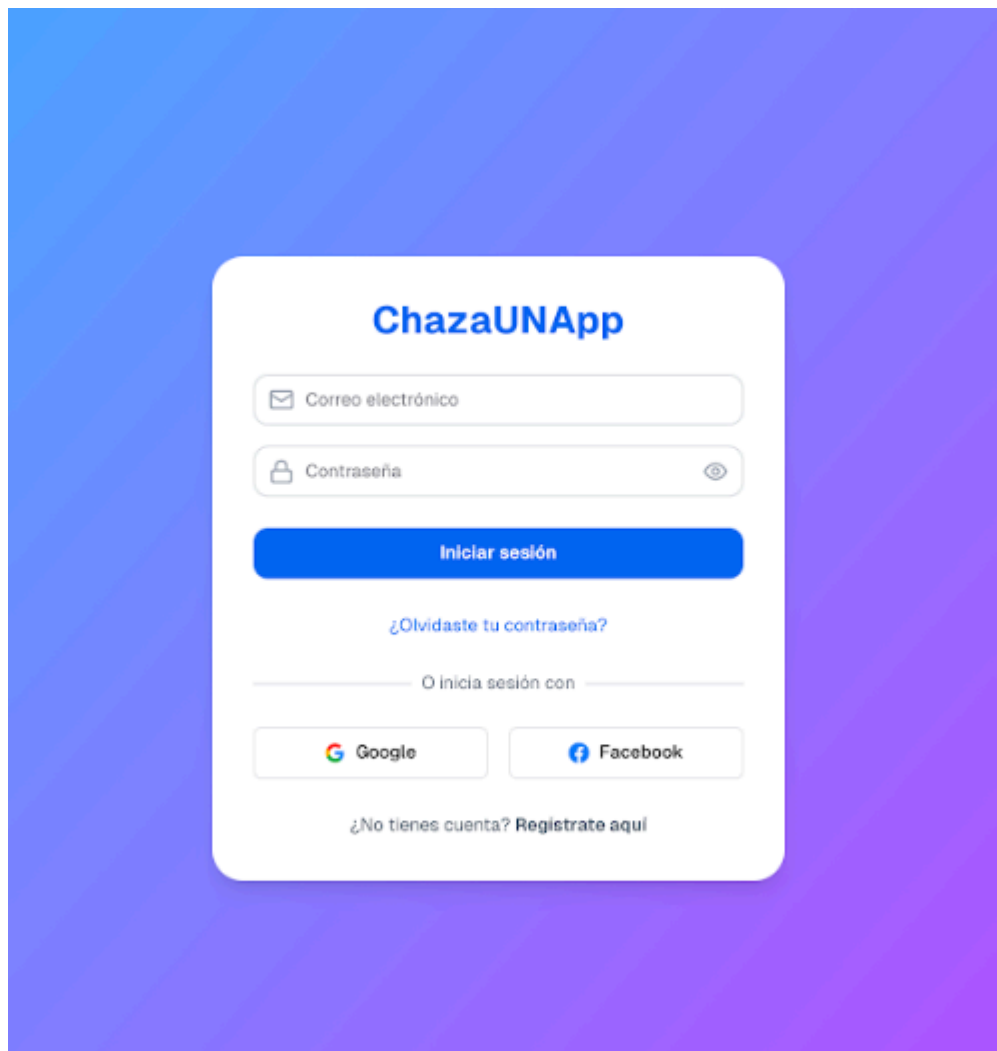
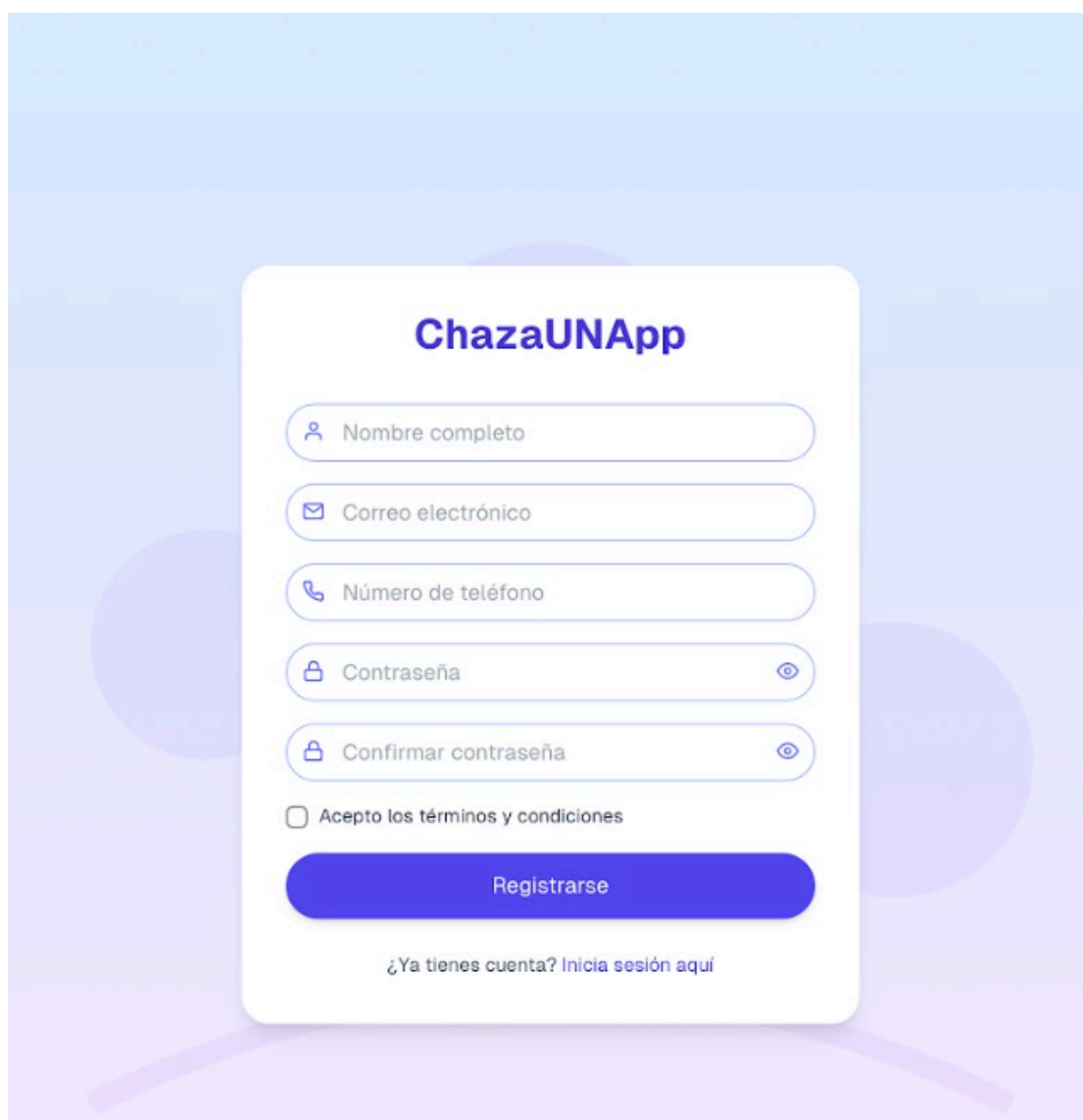


Imagen 4. Prototipo de la ventana de inicio de sesión.



The image shows a registration form for 'ChazaUNApp' centered on a light blue background with abstract shapes. The form is a white rounded rectangle containing the following elements: a title 'ChazaUNApp' in bold blue text; five input fields with blue borders and icons (person, envelope, phone, and two padlocks) and placeholder text; a checkbox for terms and conditions; a blue 'Registrarse' button; and a link 'Inicia sesión aquí'.

ChazaUNApp

Nombre completo

Correo electrónico

Número de teléfono

Contraseña

Confirmar contraseña

☐ Acepto los términos y condiciones

Registrarse

¿Ya tienes cuenta? [Inicia sesión aquí](#)

Imagen 5. Prototipo del formulario de registro.

Recupera tu contraseña

Ingresa tu correo electrónico y te enviaremos un enlace para restablecer tu contraseña.

usuario@correo.com

Enviar enlace

[← Volver al inicio de sesión](#)

Imagen 6. Prototipo del formulario de recuperación de contraseña.

Flujo visual y eventos:

- El usuario selecciona una opción entre iniciar sesión, registrarse o recuperar contraseña.
- Aparece el formulario correspondiente de manera emergente.
- Se procesa la información ingresada y se muestra un mensaje de éxito o error según el caso.

Historia de Usuario #2: Creación y Gestión de Perfiles de Vendedores

Anexo de Documentos Relacionados:

- Sección 2 de este documento: Creación y Gestión de Perfiles de Vendedores

Descripción conceptual

Módulo	<i>Perfiles de Vendedores</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Permite a los vendedores crear y administrar su perfil dentro de la aplicación. Se validará que los datos sean correctos antes de permitir la creación y modificación del perfil, asegurando la integridad de la información.</i>

Descripción técnica

Backend

1. Creación de Perfil de Vendedor

URL	Método	Código html
localhost:8080/perfil	POST	200, 422
Caso de uso técnico Cuando un vendedor intenta crear su perfil, si los datos son correctos, se retorna un código 200. Si los datos son incorrectos o faltan campos obligatorios, se retorna un código 422.		
Datos de entrada 200: <pre>{ "nombre": "Tienda Juan", "descripcion": "Venta de snacks y bebidas", "ubicacion": "Plaza Central, UNAL" }</pre>	Datos de salida 200: <pre>{ "status": "success", "data": { "perfil_id": 45, "nombre": "Tienda Juan" } }</pre>	
422: <pre>{ "nombre": "" }</pre>	422: <pre>{ "status": "error", "message": "El nombre es obligatorio" }</pre>	

2. Actualización de Perfil de Vendedor

URL	Método	Código html
localhost:8080/perfil	PUT	200, 422
Caso de uso técnico Cuando un vendedor actualiza su perfil, si los datos son correctos, se retorna un código 200. Si hay datos incorrectos o falta información requerida, se retorna un código 422.		
Datos de entrada 200: <pre>{ "perfil_id": 45, "descripcion": "Venta de comidas rápidas", "ubicacion": "Edificio Central, UNAL" }</pre>	Datos de salida 200: <pre>{ "status": "success", "message": "Perfil actualizado correctamente" }</pre>	
422: <pre>{ "perfil_id": "" }</pre>	422: <pre>{ "status": "error", "message": "El ID del perfil es obligatorio" }</pre>	

3. Eliminación de Perfil de Vendedor

URL	Método	Código html
localhost:8080/perfil	DELETE	200, 404
Caso de uso técnico Cuando un vendedor desea eliminar su perfil, si la solicitud es válida, se retorna un código 200 . Si el perfil no existe, se retorna un código 404 .		
Datos de entrada 200: <pre>{ "perfil_id": 45 }</pre>	Datos de salida 200: <pre>{ "status": "success", "message": "Perfil eliminado correctamente" }</pre>	
404: <pre>{</pre>	4: <pre>{</pre>	

<pre>"perfil_id": 9999 }</pre>	<pre>"status": "error", "message": "Perfil no encontrado" }</pre>
--------------------------------	---


Frontend


Interacción esperada:


- El vendedor ingresa a la sección de perfil desde el menú principal.
- Puede completar un formulario para crear o actualizar su perfil.
- En caso de eliminación, se mostrará un mensaje de confirmación antes de proceder.


Mockups/Prototipos:


Crear perfil de vendedor




Nombre del negocio

Descripción del negocio

Ubicación

Número de contacto

Subir imagen de perfil

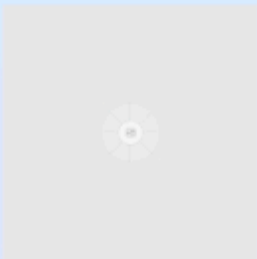
Guardar perfil


Cancelar

Imagen 7. Pantalla de Creación de Perfil de Vendedor.

✕

Editar perfil de vendedor





Mi Chaza UN

Vendemos deliciosos snacks y bebidas para estudiantes universitarios.

Edificio de Ingeniería, Universidad Nacional

300 123 4567

Guardar cambios

Cancelar

Imagen 8. Prototipo de la pantalla de edición del perfil.

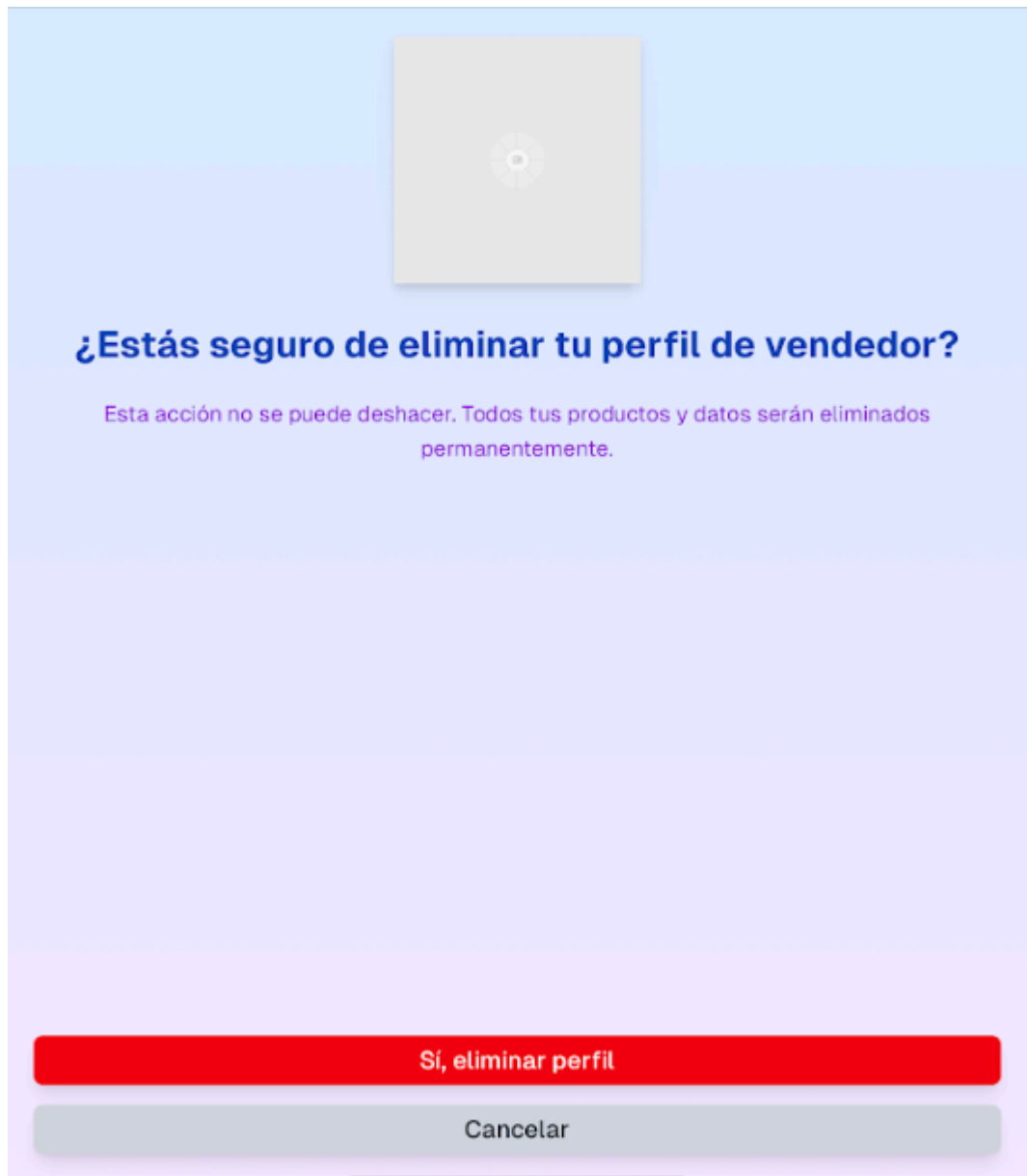


Imagen 9. Prototipo de la confirmación de eliminación.

Flujo visual y eventos:

- El vendedor accede a la sección de perfil.
- Completa o modifica los campos del formulario.
- Guarda los cambios y recibe un mensaje de éxito o error.
- En caso de eliminar el perfil, se muestra una advertencia antes de proceder.

Historia de Usuario #3: Búsqueda y Filtrado de Productos

Anexo de Documentos Relacionados:

- Sección 2 de este documento: Gestión de Búsqueda y Filtrado.

Descripción conceptual

Módulo	<i>Búsqueda de Productos</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Permite a los usuarios buscar y filtrar productos disponibles en la plataforma según diferentes criterios como categoría, ubicación y precio. Se validará la entrada del usuario para optimizar la búsqueda y mejorar la experiencia de usuario.</i>

Descripción técnica

Backend

1. Búsqueda de Productos por Categoría y Ubicación

URL	Método	Código html
localhost:8080/productos	GET	200, 422
Caso de uso técnico Cuando un usuario realiza una búsqueda de productos, si hay resultados, se retorna un código 200 con los datos. Si no hay coincidencias, se retorna un código 422.		
Datos de entrada 200: <pre>{ "categoria": "Bebidas", "ubicacion": "Plaza Central, UNAL" }</pre>	Datos de salida 200: <pre>{ "status": "success", "data": [{ "producto_id": 10, "nombre": "Café", "precio": 2000, "vendedor": "Tienda Juan" }, { "producto_id": 12, "nombre": "Jugo de Naranja", "precio": 3000, "vendedor": "Tienda Juana" }] }</pre>	

	<pre> }] } </pre>
422: <pre> { "categoria": "" } </pre>	422: <pre> { "status": "error", "message": "Debe ingresar una categoría" } </pre>

2. Filtrado de Productos por Rango de Precio

URL	Método	Código html
localhost:8080/productos	GET	200, 422
Caso de uso técnico Cuando un usuario filtra productos por un rango de precio, si hay resultados, se retorna un código 200. Si no hay coincidencias, se retorna un código 422.		
Datos de entrada 200: <pre> { "precio_min": 1000, "precio_max": 5000 } </pre>	Datos de salida 200: <pre> "status": "success", "data": [{ "producto_id": 15, "nombre": "Empanada", "precio": 2500, "vendedor": "Tienda Carlos" }] } </pre>	
422: <pre> { "precio_min": "" } </pre>	422: <pre> { "status": "error", "message": "Debe especificar un rango de precio" } </pre>	

3. Sugerencias de Búsqueda Basadas en Historial del Usuario

URL	Método	Código html
-----	--------	-------------

localhost:8080/sugerencias	GET	200, 422
<p>Cuando un usuario accede a la sección de búsqueda, se le muestran sugerencias basadas en su historial de búsqueda previo. Si hay sugerencias, se retorna un código 200. Si no hay datos disponibles, se retorna un código 422.</p>		
<p>Datos de entrada</p> <p>200:</p> <pre>{ "usuario_id": 5 }</pre>	<p>Datos de salida</p> <p>200:</p> <pre>{ "status": "success", "data": [{ "producto_id": 8, "nombre": "Chocolate", "precio": 4000, "vendedor": "Tienda María" }] }</pre>	
<p>422:</p> <pre>{ "usuario_id": "" }</pre>	<p>422:</p> <pre>{ "status": "error", "message": "No hay historial disponible" }</pre>	

Frontend

Interacción esperada:

- El usuario ingresa a la barra de búsqueda desde la pantalla principal.
- Puede escribir palabras clave, seleccionar categorías y definir un rango de precio.
- Si existen coincidencias, se muestran los productos relacionados.
- En caso de no encontrar resultados, se sugiere modificar los filtros de búsqueda.

Mockups/Prototipos:

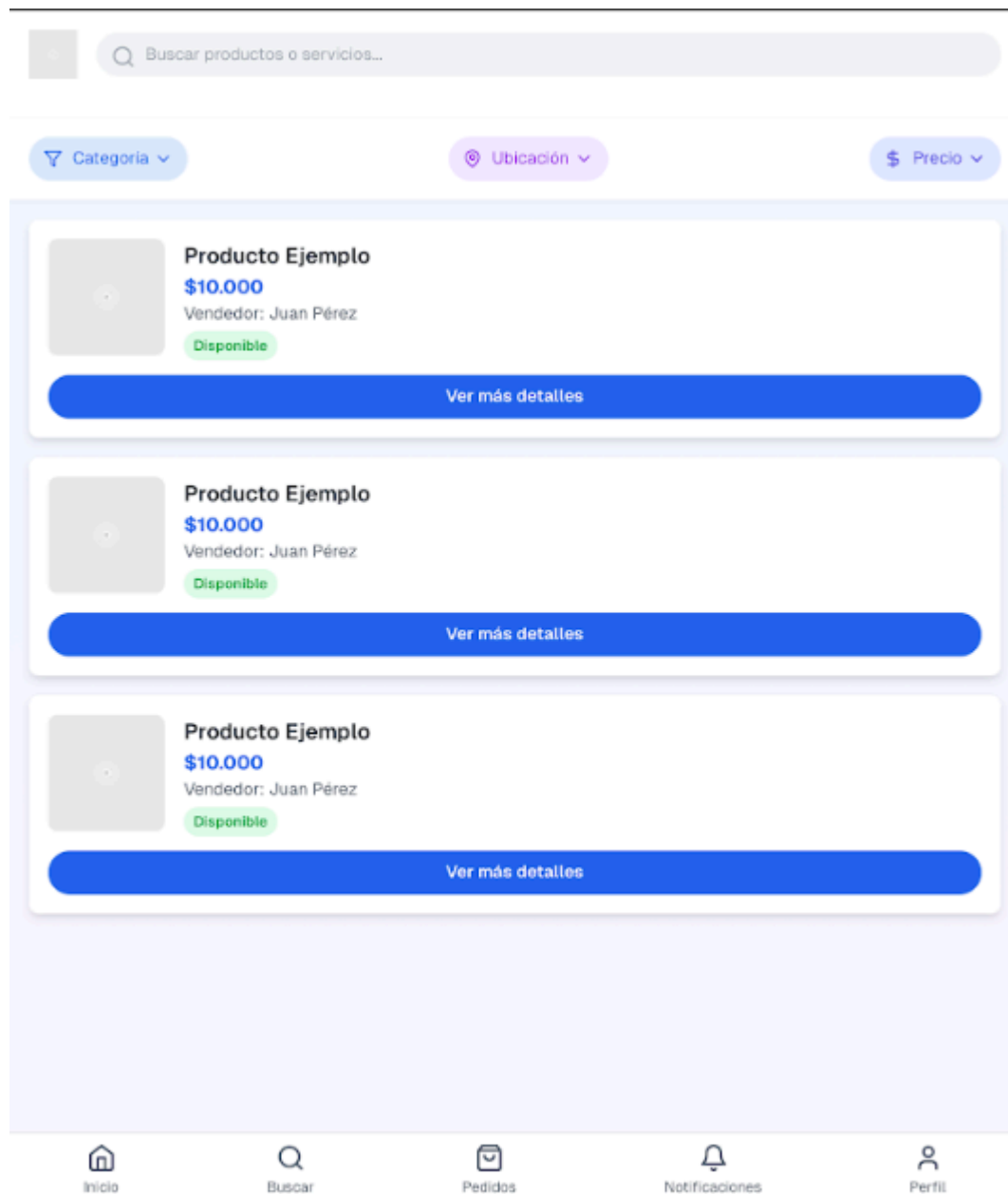


Imagen 10. Prototipo de la barra de búsqueda.

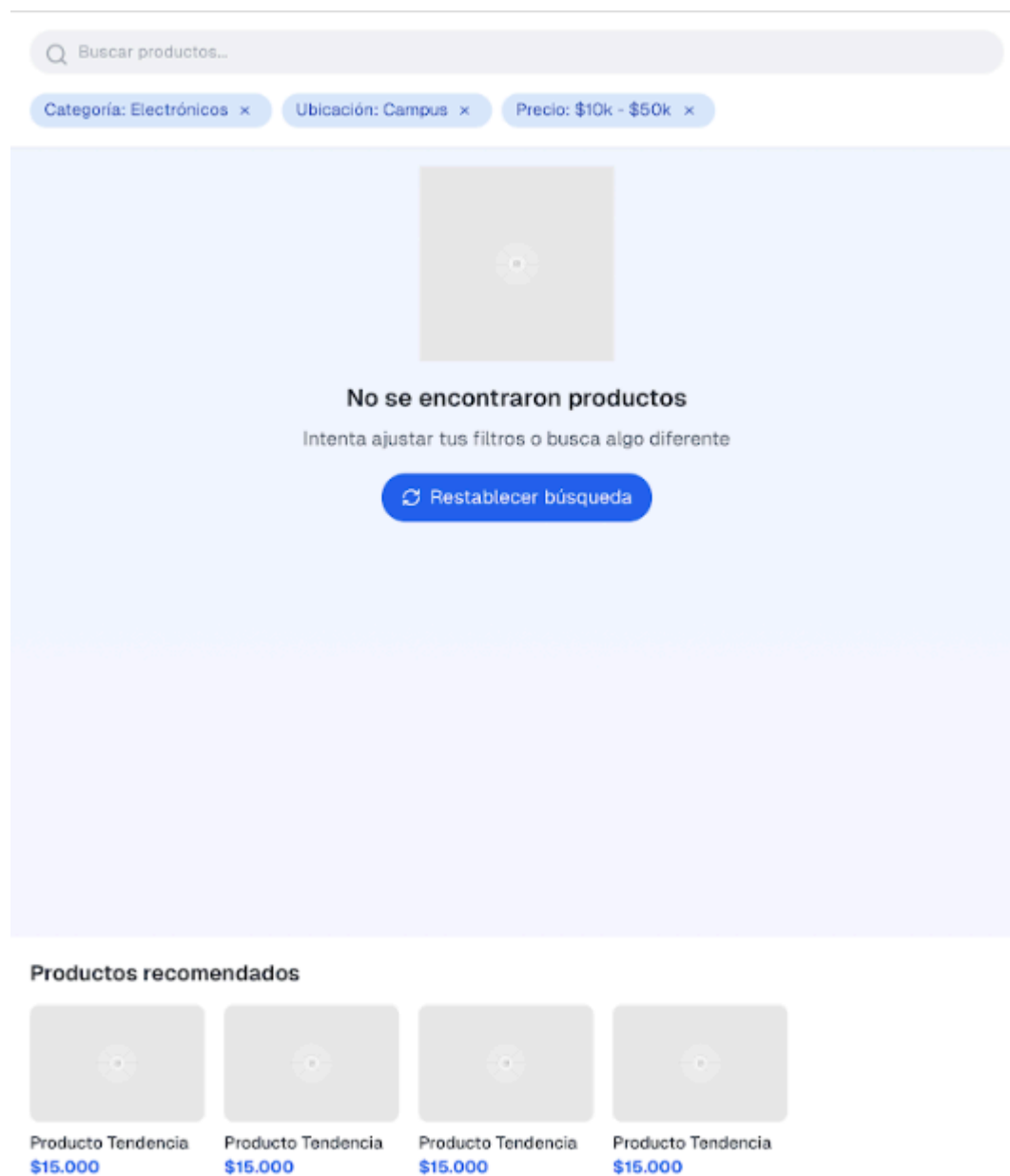


Imagen 11. Prototipo de los resultados de búsqueda.

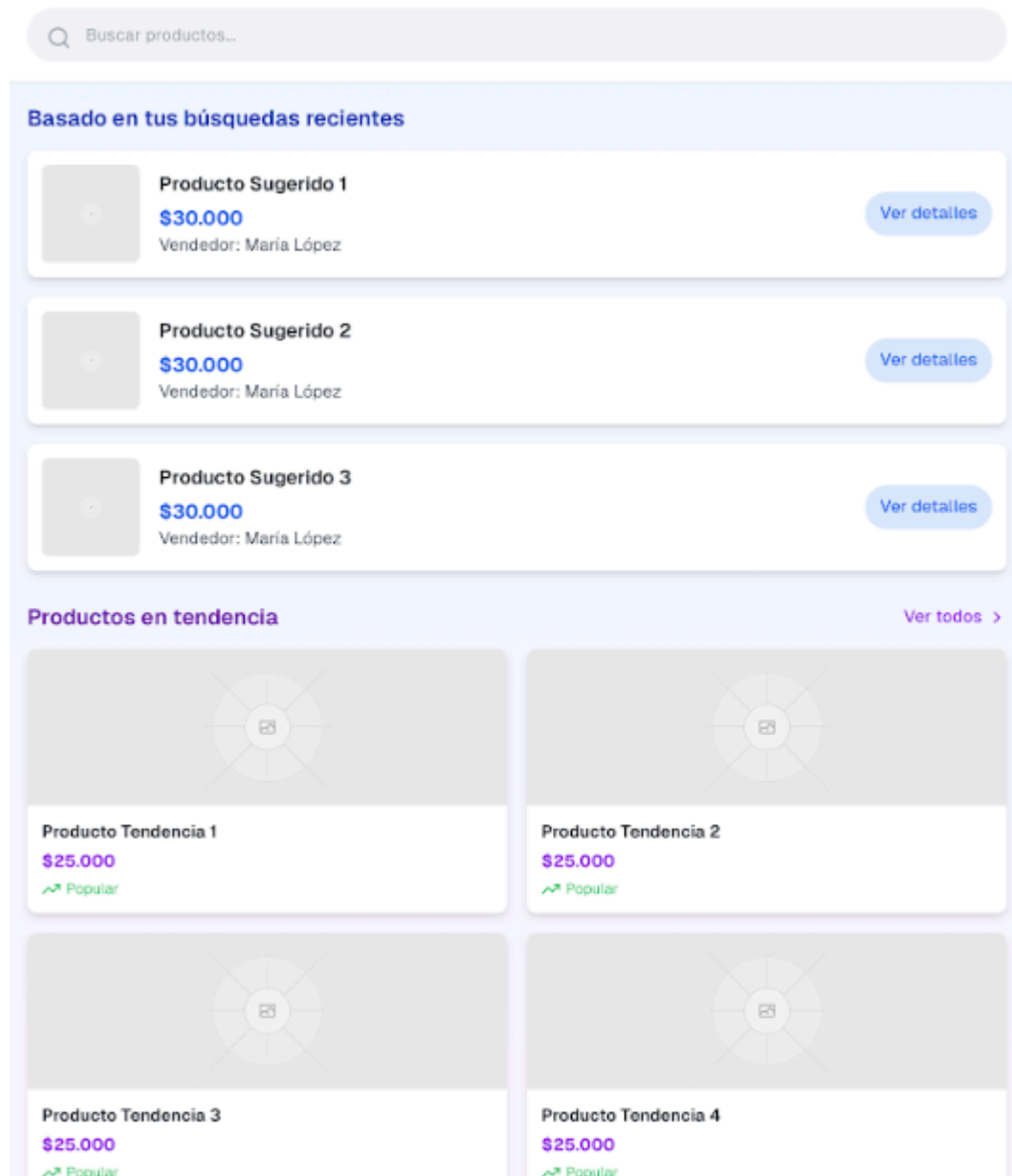


Imagen 12. Prototipo de las sugerencias de búsqueda.

Flujo visual y eventos:

- El usuario accede a la sección de búsqueda.
- Introduce criterios de búsqueda y aplica filtros.
- Se muestran los resultados y sugerencias en tiempo real.
- El usuario puede refinar su búsqueda para obtener mejores coincidencias.

Historia de Usuario #4: Calificación de Vendedores

Anexo de Documentos Relacionados:

- Sección 2 de este documento: Gestión de Calificaciones y Reseñas

Descripción conceptual

Módulo	Evaluaciones y Calificaciones
Descripción de la(s) funcionalidad(es) requerida(s):	Permite a los compradores calificar a los vendedores y dejar reseñas sobre su experiencia de compra. La calificación debe ser en un rango de 1 a 5 estrellas y puede incluir un comentario opcional. Se validará la entrada del usuario para asegurar que las reseñas sean relevantes y evitar spam o contenido inapropiado.

Descripción técnica

Backend

1. Envío de Calificación y Reseña

URL	Método	Código html
localhost:8080/calificaciones	POST	200, 422
Caso de uso técnico Cuando un usuario califica a un vendedor, si los datos son correctos, se retorna un código 200. Si los datos son incorrectos o faltan campos obligatorios, se retorna un código 422.		
Datos de entrada 200: { "vendedor_id": 45, "calificacion": 5, "comentario": "Excelente servicio y atención" }	Datos de salida 200: { "status": "success", "message": "Calificación registrada correctamente" }	
422: { "vendedor_id": "", "calificacion": 6 }	422: { "status": "error", "message": "La calificación debe estar en un rango de 1 a 5" }	

	}
--	---

2. Obtención de calificaciones de un vendedor


URL	Método	Código html
localhost:8080/calificaciones/ /{vendedor_id}	GET	200, 404
Caso de uso técnico Cuando un usuario consulta las calificaciones de un vendedor, si existen, se retorna un código 200 con los datos. Si no hay calificaciones registradas, se retorna un código 404.		
Datos de entrada 200: <pre>{ "vendedor_id": 45 }</pre>		Datos de salida 200: <pre>{ "status": "success", "data": [{ "usuario": "Juan Pérez", "calificacion": 5, "comentario": "Excelente servicio y atención" }, { "usuario": "María Gómez", "calificacion": 4, "comentario": "Buen producto, pero la entrega fue lenta" }] }</pre>
422: <pre>{ "vendedor_id": 9999 }</pre>		422: <pre>{ "status": "error", "message": "No hay calificaciones registradas para este vendedor" }</pre>

Frontend

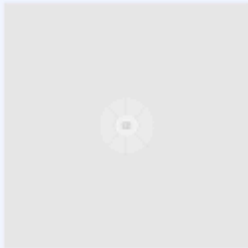
Interacción esperada:


- El usuario accede a la página del vendedor y visualiza la sección de calificaciones.
- Puede enviar una calificación seleccionando entre 1 y 5 estrellas y escribiendo un comentario opcional.
- Puede ver las calificaciones y comentarios previos dejados por otros compradores.

Mockups/Prototipos:



Calificar vendedor





Juan Pérez
📍 Campus Central
★ 4.8 (120 reseñas)

Tu calificación

★ ★ ★ ★ ★

Comentario (opcional)

Escribe tu experiencia con este vendedor...

Enviar calificación

Cancelar

Imagen 13. Prototipo del formulario para enviar una calificación.

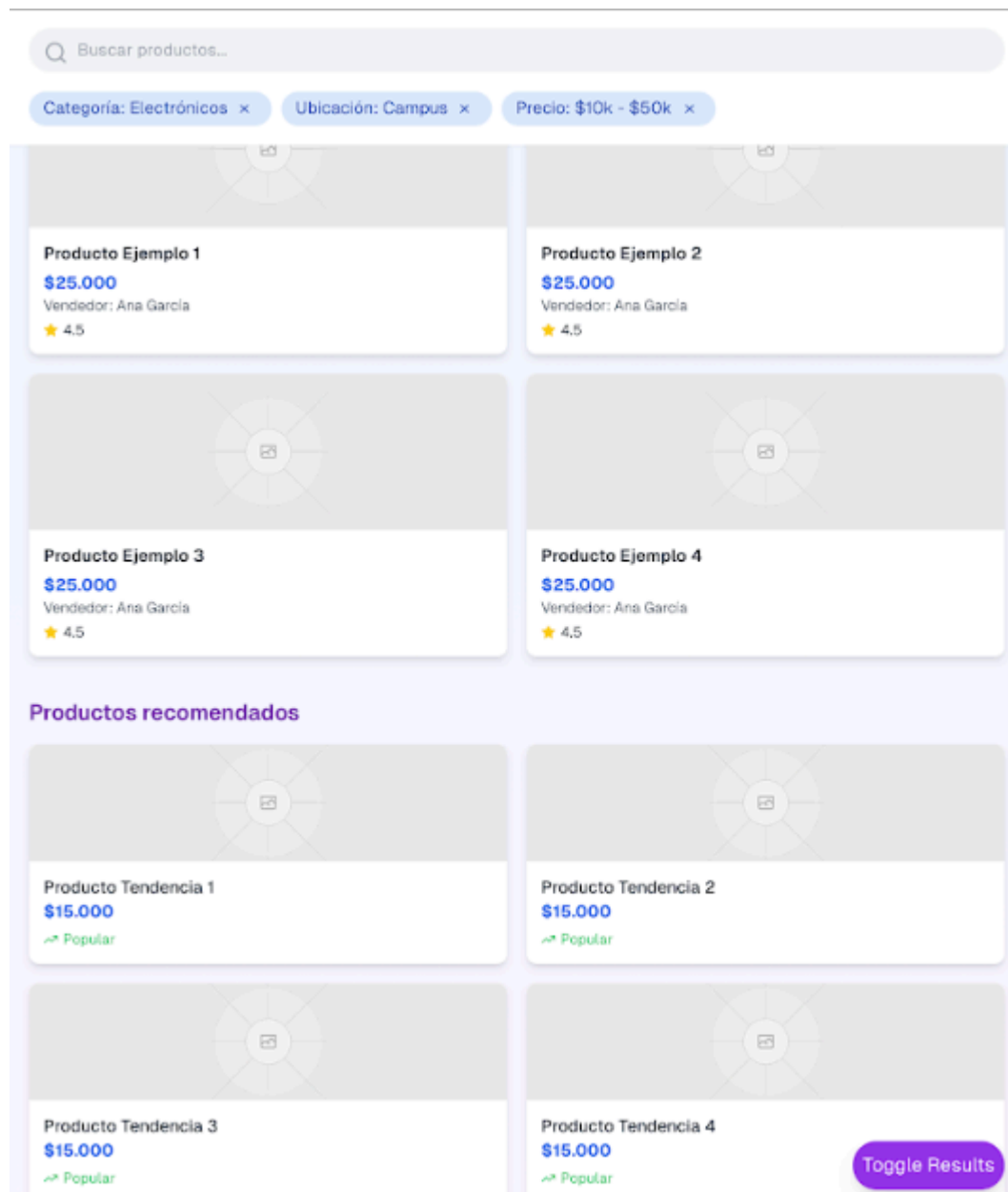


Imagen 14. Prototipo de la sección de calificaciones en el perfil del vendedor.

Flujo visual y eventos:

- El usuario accede al perfil del vendedor.
- Puede visualizar el historial de calificaciones.
- Ingresa una nueva calificación y la envía.
- El sistema valida los datos y muestra un mensaje de éxito o error.

Historia de Usuario #5: Notificaciones sobre disponibilidad

Anexo de Documentos Relacionados:

- Sección 2 de este documento: Gestión de Notificaciones

Descripción conceptual

Módulo	Evaluaciones y Calificaciones
Descripción de la(s) funcionalidad(es) requerida(s):	Permite a los usuarios recibir alertas sobre la disponibilidad de productos. Los usuarios pueden suscribirse a notificaciones de productos específicos y recibir alertas cuando haya cambios en stock o nuevas ofertas. La funcionalidad debe garantizar que las notificaciones sean enviadas en tiempo real.

Descripción técnica

Backend

1. Envío de Calificación y Reseña

URL	Método	Código html
localhost:8080/notificaciones	POST	200, 422
Caso de uso técnico Cuando un usuario se suscribe a notificaciones de productos, si la solicitud es válida, se retorna un código 200. Si faltan datos o la solicitud es incorrecta, se retorna un código 422.		
Datos de entrada 200: { "producto_id": 10, "usuario_id": 5, "notificacion": "Activada" }	Datos de salida 200: { "status": "success", "message": "Notificación activada correctamente" }	
422: { "producto_id": "" }	422: { "status": "error", "message": "Debe proporcionar un producto válido" }	

	}
--	---

2. Envío de Notificación de Disponibilidad

URL	Método	Código html
localhost:8080/notificaciones/envio	POST	200, 422
Caso de uso técnico Cuando un producto cambia su estado de disponibilidad y hay usuarios suscritos, el sistema envía notificaciones automáticas. Si la operación es válida, se retorna un código 200. Si hay un error en la configuración, se retorna un código 422.		
Datos de entrada 200: <pre>{ "producto_id": 10, "estado": "Disponible" }</pre>		Datos de salida 200: <pre>{ "status": "success", "message": "Notificaciones enviadas a los usuarios suscritos" }</pre>
422: <pre>{ "producto_id": 9999 }</pre>		422: <pre>{ "status": "error", "message": "No se encontraron usuarios suscritos" }</pre>

Frontend

Interacción esperada:

- El usuario selecciona productos y activa la opción de recibir notificaciones.
- Cuando el producto cambia su disponibilidad, el usuario recibe una alerta en la aplicación.
- Puede gestionar sus preferencias de notificación desde el perfil.
- **Mockups/Prototipos:**

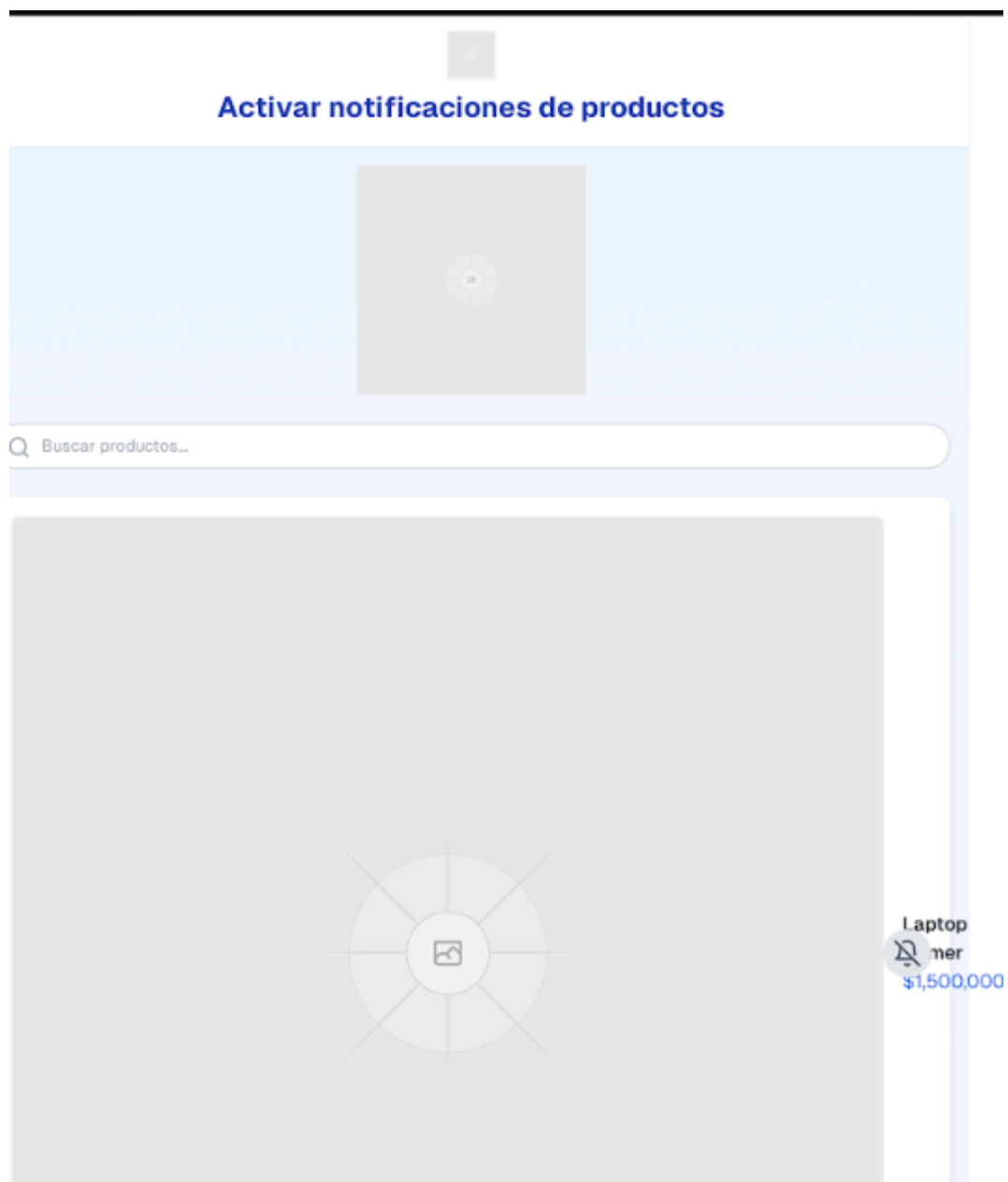


Imagen 16. Prototipo de la pantalla de suscripción a notificaciones.

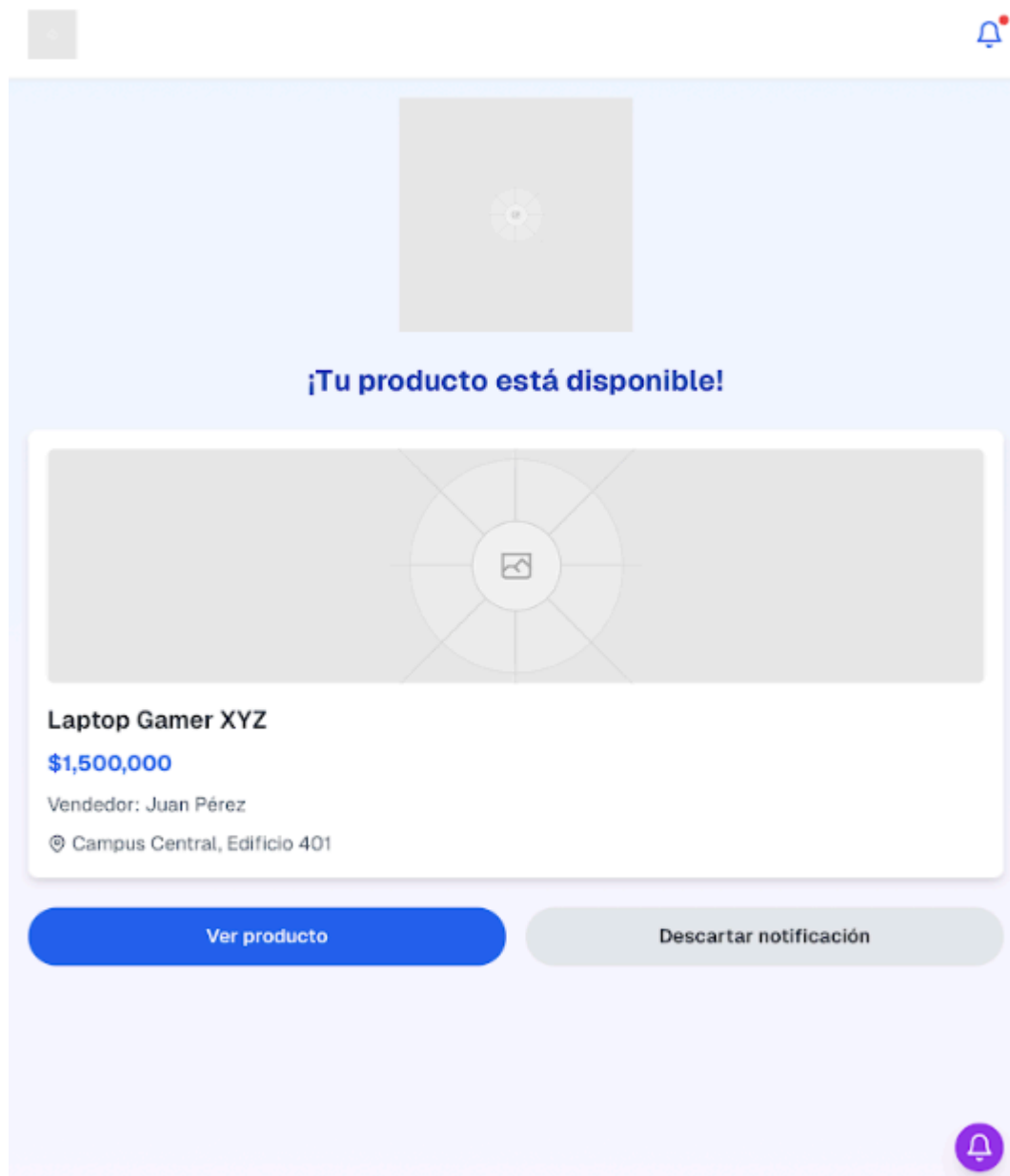


Imagen 17. Prototipo de la alerta de notificación en la aplicación.

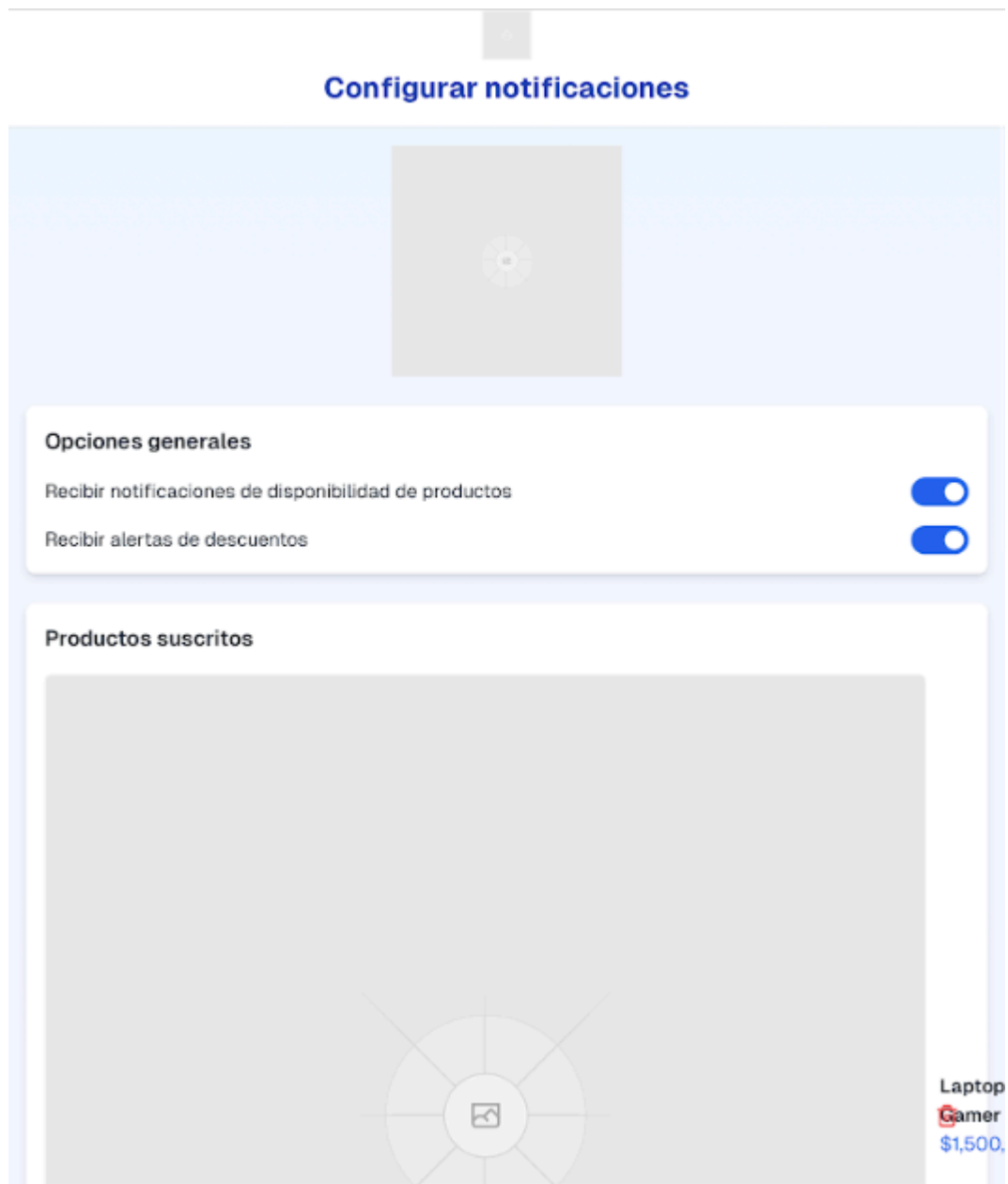


Imagen 18. Prototipo de la gestión de notificaciones en el perfil del usuario.

Flujo visual y eventos:

- El usuario accede a la sección de productos y activa la opción de notificaciones.
- Cuando el estado de un producto cambia, se genera una notificación automática.
- El usuario recibe la alerta y puede tomar acción según su interés.
- Puede gestionar o desactivar notificaciones desde su perfil.