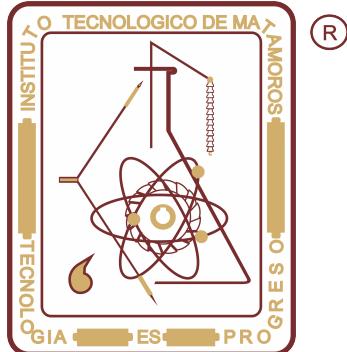

REPORTE DE DOCUMENTACIÓN DEL PROYECTO TRUEQUE VERDE

INSTITUTO TECNOLOGICO DE MATAMOROS

Ingenieria en Sistemas Computacionales



PROGRAMACIÓN FRONTEND

AUTORES

Marcos Iván Jiménez de la Cruz 21260170

Denes Ulises Mireles Rodríguez 21261105

Daniela Juan Hipólito 21260145

Esmeralda Teodoro Soreano 20260223

Juan Felipe Ávalos Pérez 21260130

Ernesto Vázquez Ramírez 21260181

DOCENTE

Anabel Pineda Briseño

Marzo 2025

Índice

1	Introducción	3
2	Implementación de la interfaz de registro	4
2.1	Login	4
2.2	Registro	5
2.3	Estructura Visual	6
2.4	Recuperación de Contraseña	6
2.5	Inicio de Sesión y Gestión de Usuario	6
3	Implementación del catálogo de productos de intercambio	8
3.1	Frontend	8
3.2	FormPublicar.js.	10
3.3	Actualización del frontend del formulario de productos	11
3.4	Producto.Js	12
3.5	Backend	12
3.6	PostController.	13
4	Implementación del sistema de intercambio	15
4.1	Funcionalidades Clave	15
4.2	Formulario y Chat	15
4.2.1	Funcionalidades Principales	15
4.2.2	Estructura	15
5	Implementacion de la Base de Datos	17
5.1	Sistema de Intercambio y Mensajes	17
5.2	Backend: Desarrollo de Tablas	19
5.3	Descripción del Modulo de Chat	19
5.3.1	Backend	19
5.3.2	Frontend	20
6	Implementacion de Mapa Interactivo	23
6.1	Pantalla de Mapa	23
6.1.1	Interacción y Funciones Dinámicas	23
6.1.2	Validaciones y Diseño	24
7	Implementación de la interfaz (Pagina Web)	25
7.1	Estructura del proyecto	25
7.1.1	Tecnologías Usadas	25
7.2	Funcionalidades	26
7.2.1	Sección de consejos	26
7.2.2	Sección de guías	26

7.2.3	Sección de beneficios	27
7.2.4	Sección de árboles frutales	27
7.2.5	Modal	28
7.2.6	Comunidad	29
7.2.7	Base de Datos Versión Web	30
7.2.8	Base de Datos Modal	30
8	Sistema de Reputación	31
8.1	Encuesta de Reputación	31
8.2	Funcionalidad de Foto de Perfil	31
8.3	Elementos del Perfil de Usuario	32
8.4	Integración del Sistema de Reputación	33
8.5	Comunicación Frontend–Backend	33
9	Implementación de la interfaz del panel de administración	34
9.1	Usuarios Problemáticos	34
9.2	Publicaciones	34
9.3	Tabla de Publicaciones	34
9.4	Ventanas Emergentes (Modales) con Detalles de la Publicación	35
9.5	Panel de Administracion	35
10	Equipos de Trabajos	37

Capítulo 1

Introducción

El proyecto Trueque Verde surge con la finalidad de fomentar el intercambio de bienes y servicios de manera sostenible, promoviendo una economía colaborativa que incentive el cuidado del medio ambiente. En un contexto donde la responsabilidad social y el compromiso ecológico cobran cada vez más relevancia, esta plataforma se plantea como una solución integral para conectar a usuarios interesados en el intercambio responsable, facilitando tanto la comunicación como la gestión de transacciones de forma segura y amigable.

El desarrollo del proyecto se ha dividido en diversas áreas de trabajo, en las que cada equipo se ha encargado de implementar componentes específicos. Entre ellos se destacan la creación de interfaces de registro, catálogos de productos, sistemas de mensajería y mapas interactivos, tanto para plataformas web como móviles. En este sentido, el equipo 8 ha asumido la responsabilidad de recopilar y documentar de manera uniforme todos los aportes referentes al FrontEnd, con el objetivo de garantizar coherencia, facilidad de uso y mantenimiento a futuro.

Este reporte de documentación tiene como propósito describir detalladamente la estructura, diseño y funcionalidades implementadas en el FrontEnd de Trueque Verde. A través de esta guía, se busca proporcionar una visión clara y precisa de los lineamientos adoptados, los retos enfrentados y las soluciones propuestas durante el desarrollo. Además, se pretende servir como herramienta de referencia para futuros procesos de evaluación y mejora del sistema, contribuyendo al fortalecimiento de la plataforma como un recurso innovador y escalable en el ámbito del intercambio sostenible.

Capítulo 2

Implementación de la interfaz de registro

En este capítulo, se presentan los principales lineamientos y estrategias empleadas en el desarrollo de la interfaz de usuario, abordando desde la estructura de las vistas hasta los principios de diseño que rigen la plataforma. Se detallan aspectos clave como la disposición de los elementos en la pantalla de inicio de sesión, la organización del formulario de registro y las técnicas utilizadas para lograr una experiencia de usuario óptima.

A través de esta documentación, se proporciona una visión clara del enfoque adoptado en la construcción del FrontEnd, con el objetivo de asegurar coherencia visual, facilidad de uso y escalabilidad futura.

2.1 Login

El componente Login maneja la autenticación de usuarios existentes. Proporciona un formulario con validación en tiempo real, permitiendo a los usuarios ingresar su correo electrónico y contraseña. También incluye una opción “Recuérdame” para mantener la sesión activa. Además, cuenta con enlaces para redirigir a la página de registro o a la recuperación de contraseña.

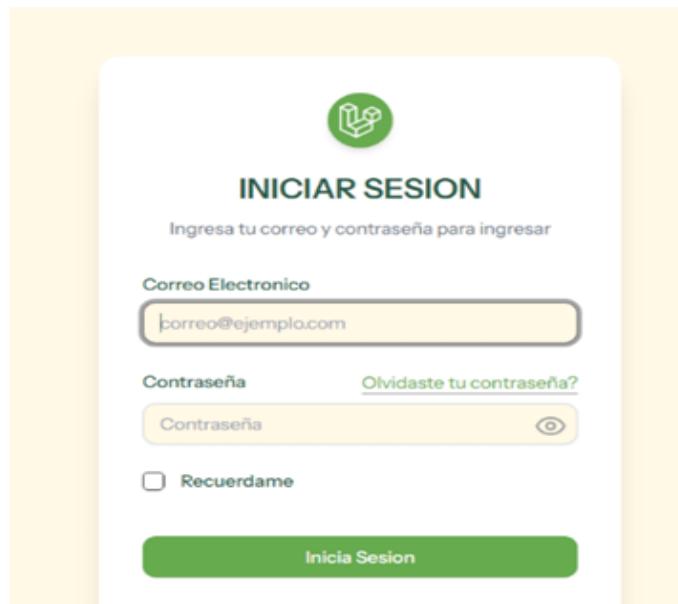


Figure 2.1: Inicio de sesión.

2.2 Registro

El componente Register gestiona el registro de nuevos usuarios. Utiliza un formulario dividido en dos secciones para facilitar la captura de información personal y de cuenta. Además, permite la carga de una foto de perfil y aplica validaciones completas para asegurar la correcta entrada de datos.

The figure consists of two side-by-side screenshots of a mobile application's registration screen. Both screens have a light yellow header and footer bar. The top header contains a green circular logo with a white stylized 'W' and 'U' shape. The main title 'CREA UNA CUENTA' is centered at the top of both sections. Below the title, a subtitle 'Ingresa tus datos para registrarte' is displayed.

Left Screen (User Information):

- Nombre:** A text input field with placeholder 'Nombre'.
- Apellido:** A text input field with placeholder 'Apellido'.
- Nombre de usuario:** A text input field with placeholder 'Nombre de usuario'.
- Correo Electrónico:** A text input field with placeholder 'correo@ejemplo.com'.
- Contraseña:** A text input field with placeholder 'Contraseña' and an eye icon for visibility.
- Confirmar Contraseña:** A text input field with placeholder 'Contraseña' and an eye icon for visibility.

Right Screen (Account Information):

- Género:** A text input field with placeholder 'Selecciona tu género'.
- Fecha de nacimiento:** A text input field with placeholder '26/03/2025' and a calendar icon.
- Teléfono:** A text input field with placeholder 'Teléfono'.
- Dirección:** A text input field with placeholder 'Dirección'.

Buttons:

- Siguiente:** A green button located at the bottom left of the left section.
- Anterior:** A green button located at the bottom left of the right section.
- Crear Cuenta:** A green button located at the bottom right of the right section.

Footer:

At the bottom of both screens, there is a link 'Ya tienes una cuenta? [Inicia Sesión](#)'.

Figure 2.2: Formulario de registro.

2.3 Estructura Visual

AuthLayout proporciona una estructura visual consistente para las páginas de autenticación. Su diseño es responsivo, incluye un logo centralizado y un contenedor con sombras. Además, permite personalizar los títulos y descripciones de cada página.

2.4 Recuperación de Contraseña

El componente ForgotPassword permite a los usuarios recuperar el acceso a sus cuentas. Consiste en un formulario simple donde se solicita el correo electrónico registrado. Posteriormente, se envía un enlace de recuperación y se muestra un mensaje de estado al usuario.

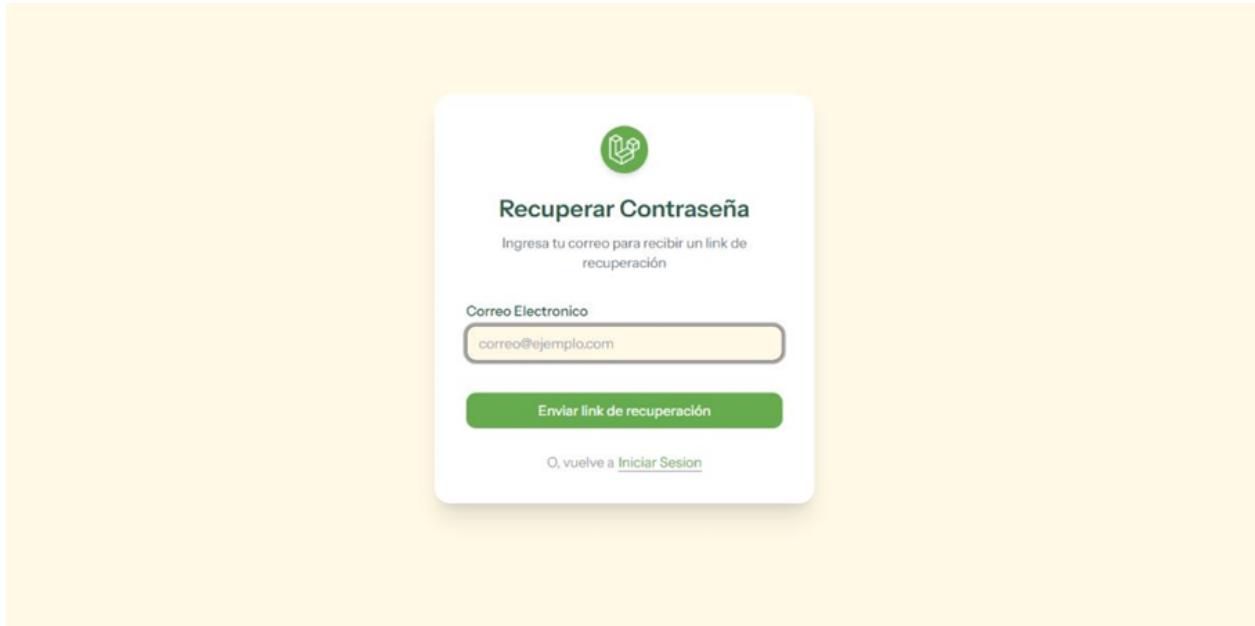


Figure 2.3: Recuperación de contraseña.

2.5 Inicio de Sesión y Gestión de Usuario

El sistema actual incorpora un módulo robusto de autenticación, navegación segura y persistencia de sesión, orientado a ofrecer una experiencia fluida, segura y escalable. A continuación se detallan sus características funcionales y técnicas.

Autenticación Segura

- Validación de credenciales tanto en frontend como en backend mediante JWT.
- Almacenamiento local seguro de tokens, con control de expiración y limpieza automática.
- Opción de persistencia condicional mediante el selector “Recuérdame”.
- Navegación basada en datos completos del usuario autenticado.

Experiencia de Usuario

- Interfaz accesible y responsiva, diseñada con componentes nativos.
- Indicadores visuales de carga e interacción inmediata.
- Estructura adaptada a pantallas de distintos tamaños.

Gestión de Estado y Navegación

- Manejo de usuario con `UserContext` y persistencia con `AsyncStorage`.
- Eliminación automática de datos obsoletos o sesiones inválidas.
- Navegación tipada con verificación estricta de tipos.

Compatibilidad y Patrones de Diseño

- Compatibilidad con:
 - React Native 0.70.0
 - React Navigation 6.0.0
 - React Native Paper 5.0.0
- Patrones utilizados:
 - Context Pattern
 - Repository Pattern
 - Observer Pattern
 - Factory Pattern

Rendimiento Estimado

- Tiempo promedio de login: \leq 2 segundos.
- Validación de token: \leq 500 ms.
- Carga inicial del módulo: \leq 1 segundo.

Capítulo 3

Implementación del catálogo de productos de intercambio

3.1 Frontend

Las nuevas actualizaciones en cuestión del frontend del catálogo son del rediseño en cuestión del menú en donde se muestra el catálogo, se agregó un pequeño menú en la parte inferior y también el diseño e implementación del formulario de publicar un producto.

En la parte del catálogo está conformado por tres partes principalmente, el cuales son formPublicar.js, producto.js y screenCatalogo.js:

ScreenCatalogo.js

El componente App representa la pantalla principal de productos dentro de una aplicación de intercambio. Permite a los usuarios buscar productos, filtrarlos por categoría, ver detalles mediante un modal, y navegar a otras pantallas. Los productos se obtienen desde un servidor externo y se actualizan periódicamente.

Funcionalidades Principales

- Búsqueda en tiempo real por nombre del producto.
- Filtro por categoría desde un menú desplegable.
- Visualización de productos en tarjetas dentro de un FlatList.
- Modal que muestra información detallada del producto mediante el componente Producto.
- Actualización automática de productos cada 5 segundos desde el backend.
- Carga de fuentes personalizadas (Poppins) usando expo-font.

Actualizaciones en el frontend del catalogo.

CHAPTER 3. IMPLEMENTACIÓN DEL CATÁLOGO DE PRODUCTOS DE INTERCAMBIO

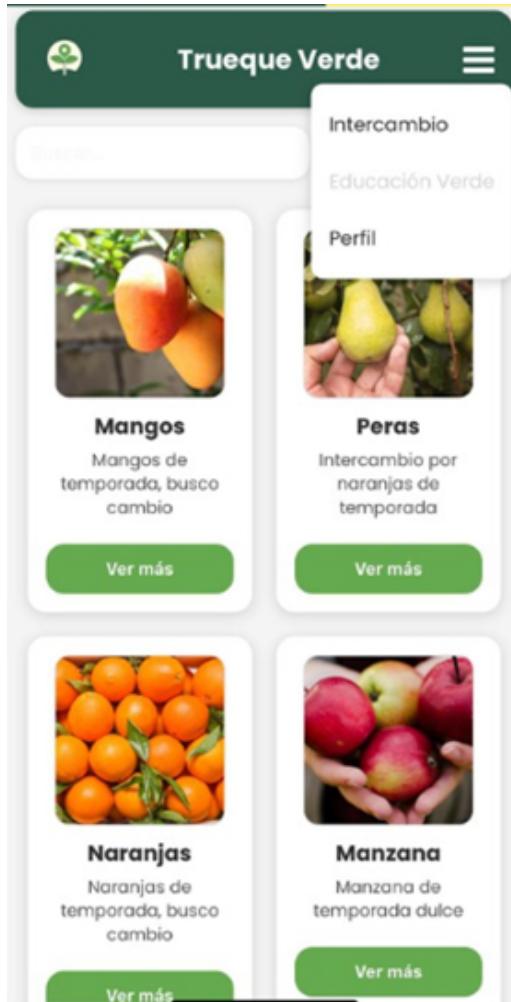


Figure 3.1: Menú antes de las modificaciones.



Figure 3.2: Diseño actualizado.

Como anteriormente se mencionó se cambió la parte donde se muestran los catálogos agregando un menú que permita al usuario moverse entre las diferentes vistas y poder publicar con esto se retiro el botón de publicar y el menú de la parte superior.

3.2 FormPublicar.js.

El componente FormPublicar es un formulario que permite a los usuarios autenticados publicar productos para intercambio. Los usuarios deben proporcionar información como el nombre del producto, una descripción, fotos, lo que desean recibir a cambio, una categoría y su ubicación.

Funcionalidades Principales

- Captura de datos del producto: nombre, descripción, categoría y elemento deseado en intercambio.
- Selección de múltiples imágenes desde la galería del dispositivo utilizando expo-image-picker.
- Obtención de ubicación del usuario (opcional).
- Envío de los datos y archivos al backend mediante axios.
- Validación básica de campos y del estado de autenticación del usuario.
- Feedback visual durante el envío de datos.

Renderizado del Formulario

Incluye:

- Campo de texto para nombre del producto.
- Campo de texto para descripción (multilínea).
- Botón para subir imágenes + vista previa de las imágenes seleccionadas.
- Campo de texto para "cambio por".
- Selector de categoría (Picker).
- Texto de ubicación basada en userLocation o coordenadas por defecto.
- Botón de envío que se desactiva mientras se procesa.

3.3 Actualización del frontend del formulario de productos

Anteriormente no se contaba con el diseño ni con la funcionalidad del formulario destinado a la creación de publicaciones. Sin embargo, esta sección ya se encuentra completamente implementada. Ahora, los usuarios tienen la posibilidad de crear publicaciones en las que especifican los productos de su cosecha que desean intercambiar, así como aquellos que están buscando obtener a cambio. Este formulario incluye campos como el título, la descripción, imágenes, la categoría del producto, la ubicación y otros datos relevantes. Con esta implementación, se facilita la interacción entre los usuarios, promoviendo el intercambio directo de productos agrícolas y fomentando la colaboración dentro de la comunidad.

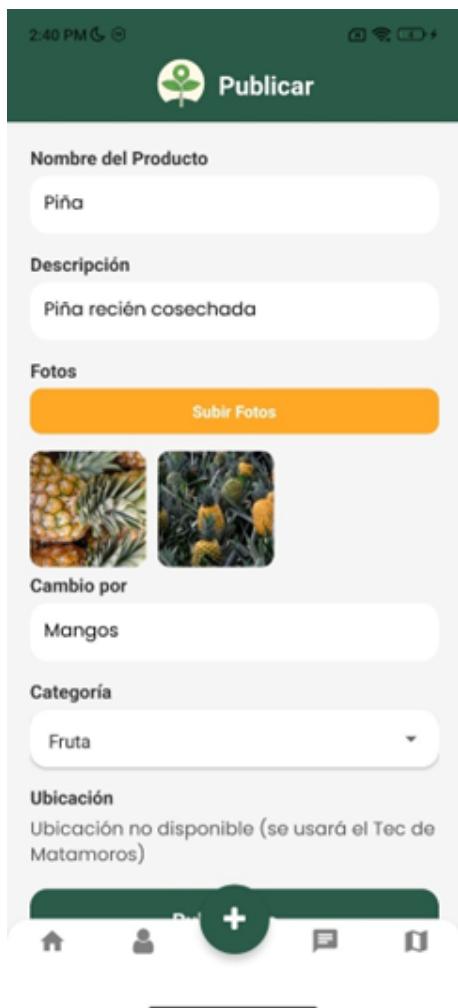


Figure 3.3: Imagen del acceso a Git

3.4 Producto.Js

Este componente muestra los detalles de un producto en un modal: un carrusel de imágenes, información del intercambio, descripción, datos del dueño, ubicación, y un campo para enviar un mensaje al propietario para iniciar un chat.

Funciones principales:

- Visualización de imágenes: carrusel horizontal de imágenes del producto, ya sea desde URLs remotas o imágenes locales por defecto.
- Envío de mensaje: permite al usuario interesado escribir y enviar un mensaje para iniciar una conversación con el propietario del producto.
- Inicio de conversación: al presionar "Enviar", se crea (o reutiliza) una conversación y se redirige al usuario al chat correspondiente.
- Visualización de detalles: se muestran la descripción del producto, información del propietario y ubicación.
- Modal de cierre: puede cerrarse tocando el botón de cierre superior.

3.5 Backend

Para que funcione todo lo que se encuentra en las vistas del menú principal, el filtrado, la visualización del de producto y su publicación contamos con tres controladores los cuales son CategoryController, ItemController y PostController, junto a sus modelos Category, Item y Post. Algo que tomar en cuenta es que en todos los controladores manejan método index, store, show, update y destroy, pero en cada uno de los controladores funcionara de manera distinta.

Controladores.

CategoryController.

El controlador CategoryController tiene como propósito central gestionar todas las operaciones relacionadas con el modelo Category.

Metodos.

Index: El primer método del controlador es index(), que tiene como función devolver una lista completa de todas las categorías almacenadas en la base de datos.

Store: El siguiente método es store, que se encarga de crear una nueva categoría. Este método primero realiza una validación de los datos recibidos. Se asegura de que el campo name esté presente, sea una cadena y tenga una longitud máxima de 255 caracteres. Si los datos no cumplen con estas reglas, se devolverá automáticamente un error de validación.

Show: Después, se encuentra el método show, que recibe directamente una instancia del modelo Category gracias a la funcionalidad de route model binding de Laravel.

Update: El método update permite actualizar una categoría existente.

Destroy: El último método es destroy, que elimina la categoría proporcionada.

ItemController. El controlador ItemController es responsable de gestionar todas las operaciones relacionadas con los items dentro del sistema.

3.6 PostController.

El controlador PostController, se encarga de gestionar las operaciones relacionadas con las publicaciones (posts) dentro de la aplicación.

Métodos.

Index: Este método se encarga de recuperar todos los posts existentes en la base de datos.

Store: Este es el método más complejo del controlador y tiene como objetivo crear una nueva publicación.

Show: Este método permite visualizar los detalles de un post en particular. Recibe un objeto Post gracias a la inyección automática de modelos de Laravel, y luego carga sus relaciones user e item para incluir la información completa relacionada.

Update: Este método se encarga de actualizar un post existente. Recibe los datos del request y los valida, permitiendo que muchos de los campos sean opcionales. Es decir, solo se actualizarán aquellos campos que estén presentes en el request y sean válidos.

Destroy: El propósito de este método es eliminar un post de la base de datos.

Modelos

Los controladores de esta aplicación trabajan directamente con los modelos para gestionar la información de la base de datos. Cada modelo representa una entidad con sus relaciones, y los controladores definen las operaciones que se pueden realizar sobre ellas.

Item

La función de category define una relación donde cada ítem pertenece a una sola categoría.

Por último, la función de posts indica que un ítem puede estar asociado a varios posts, es decir, un ítem puede aparecer en muchas publicaciones.

Category

En el arreglo protected fillable muestra que solo se tiene un campo assignable: el nombre de la categoría (por ejemplo: “Fruta”, “Planta”, etc.).

Mientras la función define que una categoría puede tener muchos ítems asociados.

Post

El arreglo permite asignar los siguientes datos al crear o actualizar un post:

- title: Título de la publicación.
- content: Contenido o descripción.
- image: Ruta(s) de la(s) imagen(es) asociada(s).
- userid: ID del usuario que publicó.
- itemid: ID del ítem relacionado.
- statusid: Estado del post (pendiente, aprobado, etc.).
- latitude, longitude: Coordenadas de ubicación.
- greenpointd: Punto verde asociado (si existe).

La función de user relaciona el post con el usuario que lo creó.

La ultima función de Item relaciona el post con el ítem asociado (por ejemplo, una planta, fruta, etc.).

Capítulo 4

Implementación del sistema de intercambio

4.1 Funcionalidades Clave

Solicitud de trueque:

1. Botón para solicitar un intercambio, con un formulario donde se especifique qué se ofrece y qué se busca.
2. Validación automática para evitar duplicados o datos incompletos.
3. Mensajería interna
4. Sistema de chat para que los usuarios negocien detalles del trueque (cantidad, punto de encuentro, etc.).

4.2 Formulario y Chat

4.2.1 Funcionalidades Principales

- Chat integrado: Permite a los usuarios ponerse de acuerdo fácilmente.
- Formulario de intercambio: Ayuda a registrar qué se ofrece, qué se busca a cambio, la ubicación y hasta agregar fotos.
- Diseño sencillo e intuitivo: La aplicación está pensada para que cualquiera pueda usarla sin problemas.

4.2.2 Estructura

Pantallas Principales:

- Pantalla de inicio para explorar opciones de trueque
- Pantalla del chat para negociar intercambios

Se utilizó React Native junto con Expo para el desarrollo y prueba de la aplicación, y NativeWind para estilizar las pantallas con clases de Tailwind CSS, lo que nos permitió mantener un diseño uniforme y adaptable. Adaptando esta configuración a nuestras pantallas, se visualizan de la siguiente forma.

- Formulario de Intercambios

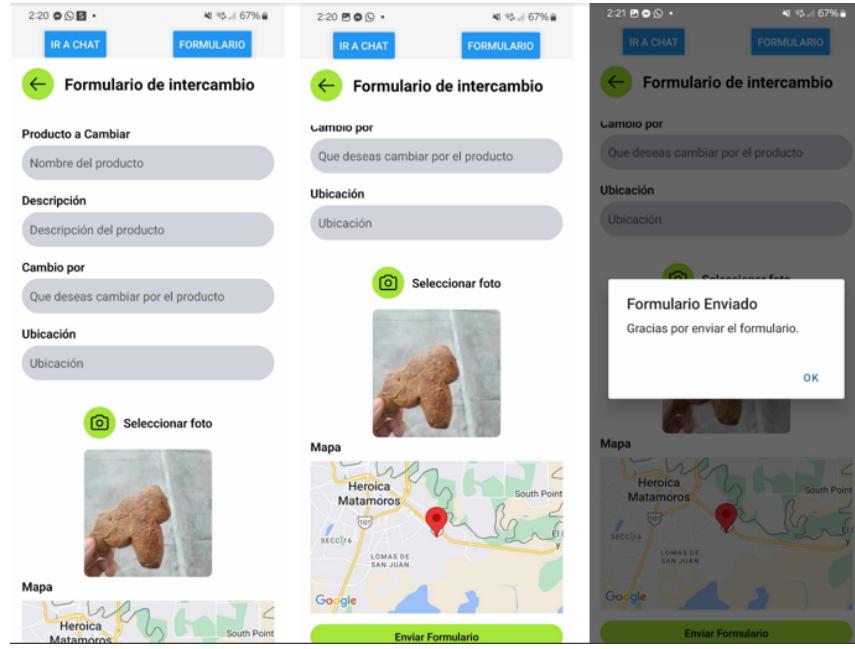


Figure 4.1: Formulario

Capítulo 5

Implementacion de la Base de Datos

5.1 Sistema de Intercambio y Mensajes

Esta base de datos consta de varias tablas interrelacionadas, cuya función es gestionar y almacenar las solicitudes de trueques, las interacciones entre los usuarios, y el contenido de las conversaciones. A continuación se describen las principales tablas y sus campos:

Tabla de Trueques: Esta tabla está diseñada para gestionar las solicitudes de trueque y almacenar la información relacionada con las transacciones entre usuarios. Los campos que contiene son los siguientes:

1. – Id Trueque: Un identificador único para cada solicitud de trueque.
- Id publicacion: Relaciona la solicitud con la publicación en la que se basa el trueque.
- Id usuario ofertante: Relaciona la solicitud con el usuario que realiza la oferta, referenciando a la tabla de usuarios para obtener su información.
- Descripción: Permite agregar una descripción detallada de la solicitud de trueque.
- Id status: Indica el estado del trueque, como activo, cerrado o rechazado. Este campo se relaciona con la tabla de estados (status).
- Latitud y Longitud: Guardan las coordenadas geográficas de la ubicación del trueque.
- Fecha: Indica la fecha y hora en que se creó la solicitud de trueque.

2. Tabla de Conversaciones:

Esta tabla es fundamental para almacenar las interacciones entre los usuarios que participan en una publicación, facilitando la comunicación. Los campos que componen esta tabla son:

- Id conversacion: Un identificador único para cada conversación.
- Id publicacion: Relaciona la conversación con una publicación específica
- Id usuario creador: Hace referencia al usuario que creó la publicación, obteniendo su información desde la tabla de usuarios.
- Id usuario ofertante: Relaciona la conversación con el usuario interesado en la publicación.
- Id status: Indica si la conversación está activa o cerrada, relacionándose con la tabla de estados (status).

- Fecha: Registra la fecha y hora de creación de la conversación.

3. Tabla de Mensajes:

Esta tabla se utiliza para almacenar los mensajes enviados dentro de las conversaciones. Los campos que incluye son:

- Id mensajes: Un identificador único para cada mensaje.
- Id conversacion: Relaciona el mensaje con una conversación específica.
- Idusuario: Hace referencia al usuario que envió el mensaje, obteniendo su información desde la tabla de usuarios.
- Contenido: Almacena el texto del mensaje
- Leido: Indica si el mensaje ha sido leído por el receptor
- Fecha: Registra la fecha y hora en que se envió el mensaje.

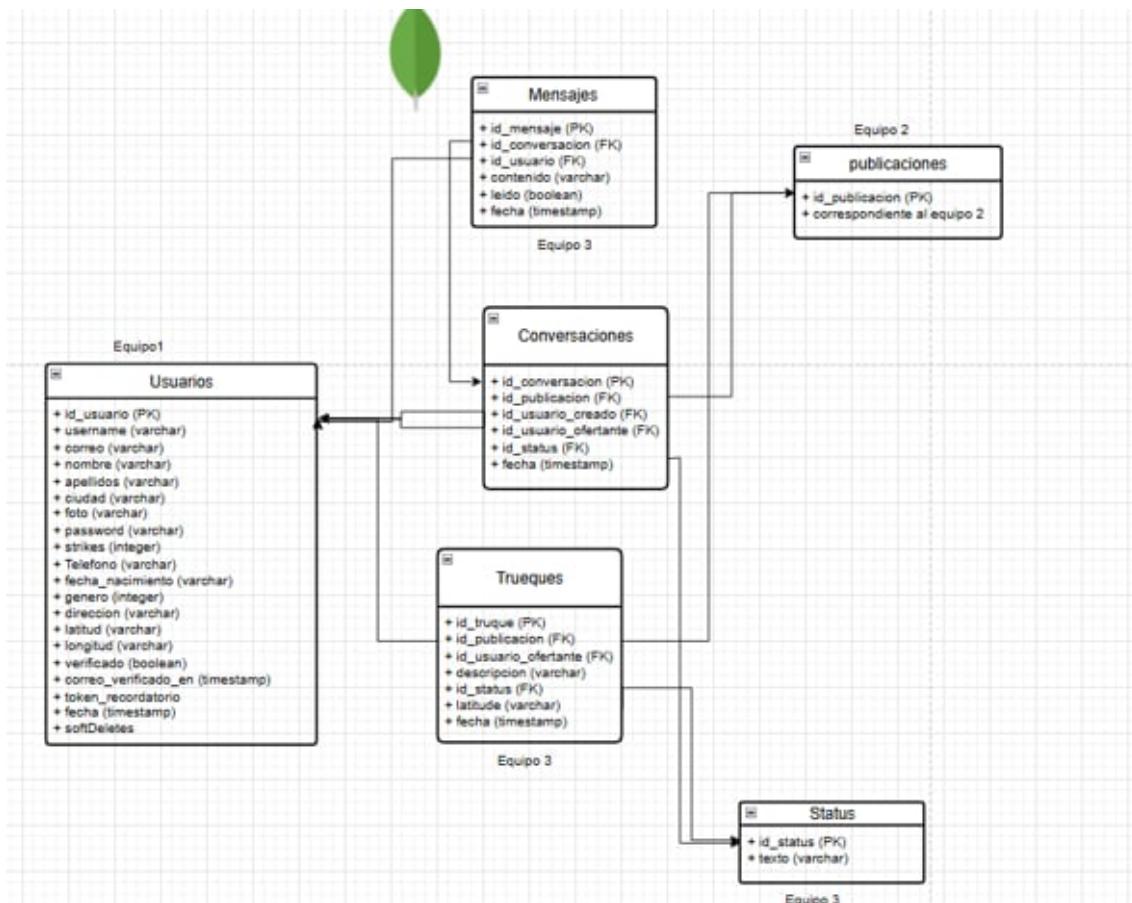


Figure 5.1: Diagrama

5.2 Backend: Desarrollo de Tablas

los fragmentos de código muestran el desarrollo de backend en Laravel que definen la estructura de las tablas "barters", "conversations" y "posts" dentro de una base de datos.

- "barters" gestiona los intercambios entre usuarios, almacenando información sobre la oferta, el estado y la ubicación del trueque.
- "conversations" administra las interacciones entre usuarios en relación con una publicación, vinculando a los participantes y su estado.
- "posts" representa las publicaciones dentro de la aplicación, incluyendo detalles como título, contenido, imagen, ubicación y estado.

Cada tabla establece claves foráneas para mantener la integridad referencial en la base de datos, asegurando la correcta relación entre los distintos elementos del sistema.

5.3 Descripción del Módulo de Chat

Dentro de la sección se muestra el backend donde se explica cómo funciona, los controladores que se manejan son: ConversationController, MessageController y BarterController

5.3.1 Backend

Controladores implementados:

1. ConversationController:

Maneja las conversaciones entre usuarios. Incluye Funcionalidades para:

- Listar todas las conversaciones en las que el usuario participa.
- Crear una nueva conversación entre dos usuarios para un post específico.
- Mostrar detalles de una conversación.
- Actualizar el estado de la conversación.
- Eliminar una conversación.

Este controlador usa autenticación para garantizar que solo los participantes puedan acceder a las conversaciones.

2. MessageController:

Administra los mensajes enviados dentro de las conversaciones. Permite:

- Consultar los mensajes de una conversación específica.
- Enviar un nuevo mensaje.
- Marcar un mensaje como leído.
- Eliminar un mensaje. También incluye una función privada que valida si el usuario tiene acceso a la conversación.

3. BarterController:

Controla las solicitudes de intercambio. Su funcionalidad principal es:

- Registrar un nuevo trueque, validando los datos recibidos y actualizando el estado del producto relacionado para reflejar que está en proceso de intercambio
- También permite ver, actualizar o eliminar intercambios específicos del usuario autenticado.

5.3.2 Frontend

Pantalla de Producto

En la pantalla de producto se habilitó la opción para que el usuario interesado en un artículo pueda escribir un mensaje al propietario. Una vez redactado y enviado, el sistema crea una conversación vinculada al producto y guarda el mensaje.



Figure 5.2: modal protocolo

Pantalla de Chat

La pantalla de chat fue diseñada para mostrar todos los mensajes de la conversación activa, distinguiendo visualmente cuáles fueron enviados por el usuario y cuáles por el interlocutor.

Se agregó un botón con un ícono de "+" que despliega un menú animado. Este menú ofrece funciones adicionales como acceder a un mapa o abrir el formulario de intercambio, en caso de que el usuario sea el ofertante del producto.

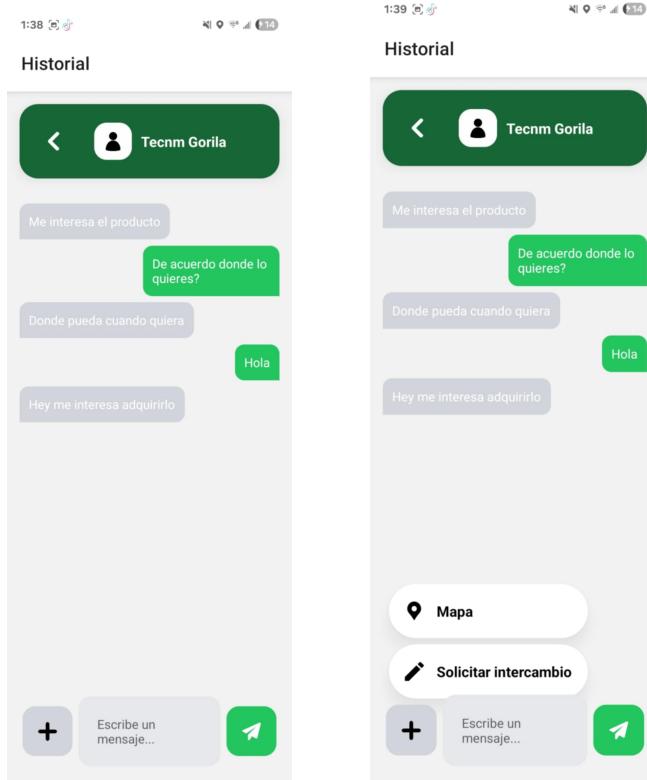


Figure 5.3: Estilo de chat

Formulario de Trueque

En cuanto al formulario de trueque, se implementó una pantalla que permite al usuario proponer un intercambio. Esta pantalla contiene campos para ingresar el nombre del producto que ofrece, una descripción, el artículo que desea recibir a cambio y su ubicación. Se utiliza un mapa interactivo para facilitar la selección del lugar del intercambio.

Figure 5.4: Diseño Formulario

Pantalla de lista de conversaciones

Por último, la pantalla de lista de conversaciones permite al usuario ver todas las conversaciones activas en las que participa. En cada elemento de la lista se muestra el nombre del interlocutor, el último mensaje enviado, el título del producto asociado y la hora de la última actividad. Si el usuario selecciona una conversación, es redirigido automáticamente a la pantalla de chat correspondiente.

44

Figure 5.5: modal Historial

El proceso inicia cuando un usuario contacta al propietario de un producto desde su pantalla. Esto crea o reutiliza una conversación, permitiendo el intercambio de mensajes en un chat dinámico. Si el ofertante desea un trueque, puede enviar la solicitud desde el chat, la cual se valida y almacena en el backend, actualizando el estado del producto.

Capítulo 6

Implementacion de Mapa Interactivo

En esta sección se describe la funcionalidad de la pantalla de mapa dentro de la aplicación, la cual cumple un papel fundamental al permitir a los usuarios ubicar, visualizar y filtrar diferentes tipos de puntos relacionados con la oferta e intercambio de productos

6.1 Pantalla de Mapa

La pantalla de mapa tiene como objetivo principal facilitar la conexión entre usuarios mediante un sistema de geolocalización. Al ingresar, la aplicación solicita el permiso para acceder a la ubicación actual del usuario. Si se concede, el mapa se centra en su posición, y a partir de allí se muestran diferentes marcadores correspondientes a:

- **Puntos verdes:** Lugares comunitarios o designados donde se pueden realizar intercambios de forma segura.
- **Publicaciones:** Productos específicos que otros usuarios han puesto a disposición, como frutas, semillas u otros alimentos.

Cada uno de estos elementos está representado con un ícono visual personalizado, lo que permite identificar fácilmente la categoría del producto.

6.1.1 Interacción y Funciones Dinámicas

El diseño del mapa no solo muestra ubicaciones, sino que ofrece una experiencia dinámica mediante un sistema de filtros inteligentes que permiten al usuario personalizar lo que desea ver:

- **Filtro por tipo de marcador:** El usuario puede elegir entre ver todos los elementos, solo publicaciones o únicamente puntos verdes.
- **Filtro por categoría:** Se puede seleccionar una categoría específica (por ejemplo, semillas), lo que ajusta automáticamente los marcadores visibles a esa selección.
- **Filtro por distancia:** Un control deslizante permite definir un rango máximo en kilómetros, mostrando únicamente aquellos elementos que se encuentren dentro de ese radio.

Estos filtros están disponibles desde un menú desplegable accesible mediante un botón flotante en la pantalla.

Además, se incluye la posibilidad de actualizar manualmente la ubicación del usuario, en caso de que se haya desplazado o necesite refrescar los datos del mapa.

6.1.2 Validaciones y Diseño

Para garantizar una experiencia de usuario fluida y confiable, la pantalla de mapa incorpora diversas validaciones y consideraciones de diseño que contribuyen a su funcionalidad y atractivo visual.

Validaciones y manejo de errores:

- Si la aplicación no obtiene permiso para acceder a la ubicación, muestra un mensaje claro acompañado de un botón que permite al usuario intentar la acción.
- En caso de problemas de conexión con el servidor o errores al obtener los datos de publicaciones y puntos verdes, se informa al usuario con un mensaje visible y se le ofrece la opción de intentar nuevamente.

Diseño y experiencia visual: El mapa está diseñado con una estética coherente y neutral, utilizando colores que evitan saturar la vista y facilitan la interpretación. Los marcadores cuentan con íconos personalizados que representan claramente la categoría o tipo de punto, mejorando la identificación rápida.

Al pulsar cualquier marcador, se despliega un *callout* que muestra información relevante, como el nombre del producto o detalles del punto verde, sin necesidad de salir de la pantalla principal. Esto facilita la consulta rápida y mejora la interacción.

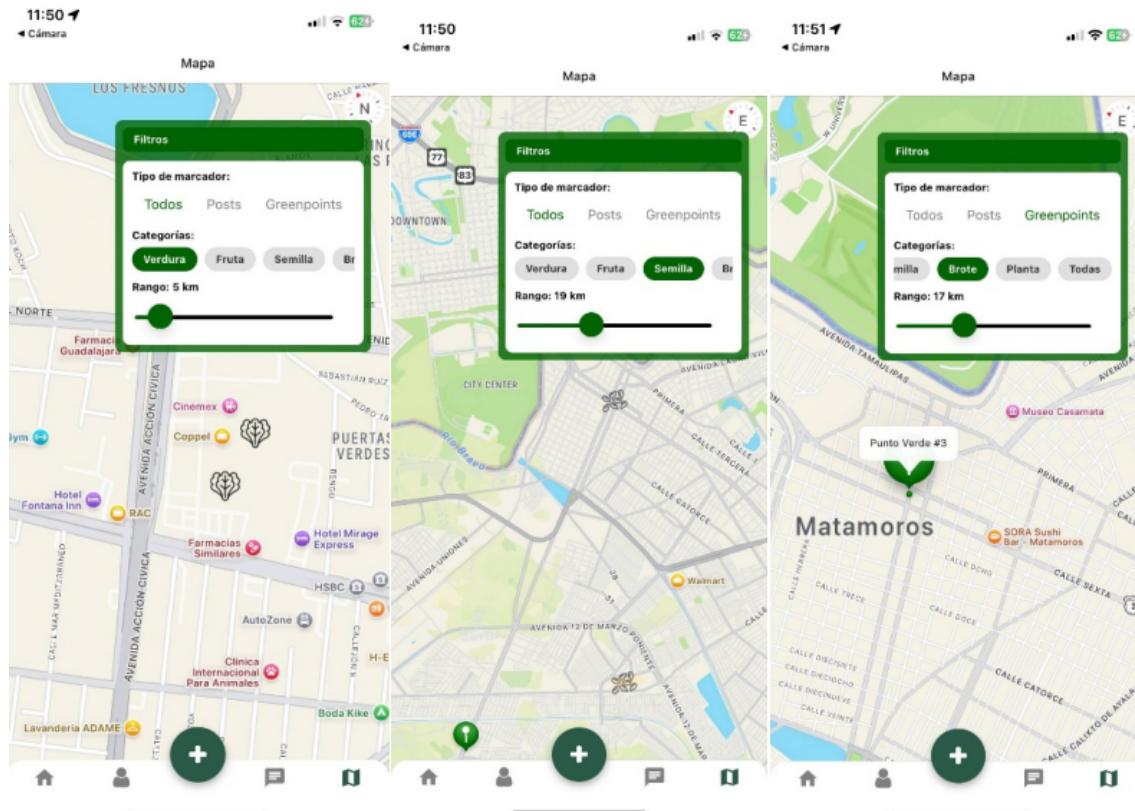


Figure 6.1: Localizaciones

Capítulo 7

Implementación de la interfaz (Pagina Web)

7.1 Estructura del proyecto

El proyecto ”**Educación Verde**” es una página web diseñada para educar a los usuarios sobre el cuidado de árboles frutales. La página incluye secciones con consejos, guías, beneficios y características de diferentes árboles frutales. Además, utiliza Bootstrap para el diseño responsive y JavaScript para generar contenido dinámico el proyecto está compuesto por los siguientes archivos:

7.1.1 Tecnologías Usadas

Archivo principal que contiene la estructura HTML de la página.

1 Encabezado (`<header>`):

Contiene el logo y el título de la página.

Diseño responsivo con Bootstrap.

HTML5: Para la estructura de la página.

CSS3: Para los estilos personalizados.

Tailwind css : Clases utilitarias predefinidas

JavaScript: Para generar contenido dinámico y manejar interacciones.

PostgreSQL: Sistema Gestor de Base de Datos para almacenar el contenido de la aplicación.

R2: Será usado para el almacenamiento de imágenes.

Imágenes: Utilizadas para ilustrar consejos, guías, beneficios y árboles frutales.

7.2 Funcionalidades

7.2.1 Sección de consejos

Muestra una lista de consejos para cuidar árboles frutales. Cada consejo incluye una imagen, un título, una descripción breve y un botón "Leer más" que abre un modal con detalles adicionales.



Figure 7.1: Imagen de Consejos

7.2.2 Sección de guías

Proporciona guías prácticas para plantar y mantener árboles jóvenes. Similar a la sección de consejos, incluye imágenes, títulos, descripciones y un botón para ver más detalles.



Figure 7.2: Referencia del Radio de comunicación

7.2.3 Sección de beneficios

Enumera los beneficios de los árboles frutales para el medio ambiente. Cada beneficio tiene una imagen y un botón para ver más información en el modal.

Educación Verde

Beneficios de los árboles frutales para el medio ambiente

Los árboles frutales no solo embellecen el entorno, sino que también desempeñan un papel fundamental en la sostenibilidad ambiental, la biodiversidad y el bienestar humano.

- 1. Mejora de la calidad del aire y reducción de la contaminación
- 2. Conservación de la biodiversidad y protección de especies
- 3. Enriquecimiento y protección del suelo

[Ler más](#)

Figure 7.3: Referencia del Radio de comunicación

7.2.4 Sección de árboles frutales

Presenta una lista de árboles frutales con sus características, cuidados y recomendaciones de plantación.

Educación Verde

Principales árboles frutales, características y cuidados

Manzano (Malus domestica) Características Cuidados Plantado	Peral (Pyrus) Características Cuidados Plantado	Cítricos (Citrus) Características Cuidados Plantado
--	--	--

Figure 7.4: Referencia del Radio de comunicación

Cada árbol tiene tres botones para ver detalles sobre sus características, cuidados y cómo plantarlo. Beneficios Enumera los beneficios de los árboles frutales para el medio ambiente. Cada beneficio tiene una imagen y un botón para ver más información en el modal



Figure 7.5: Referencia del modelo de beneficios

7.2.5 Modal

Un modal reutilizable que muestra información detallada dependiendo del botón que se haya hecho clic. El contenido del modal se genera dinámicamente con JavaScript.

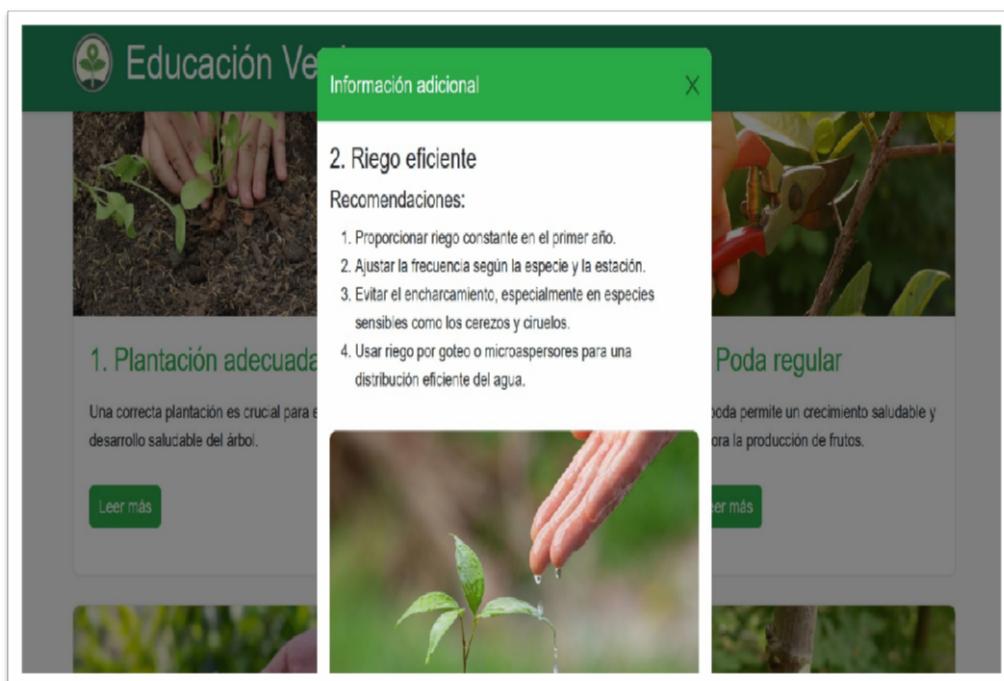


Figure 7.6: Referencia del Radio de comunicación

Los modales mejoran la experiencia de usuario al mostrar información detallada sin abandonar la página actual, manteniendo el contexto y la fluidez de navegación.

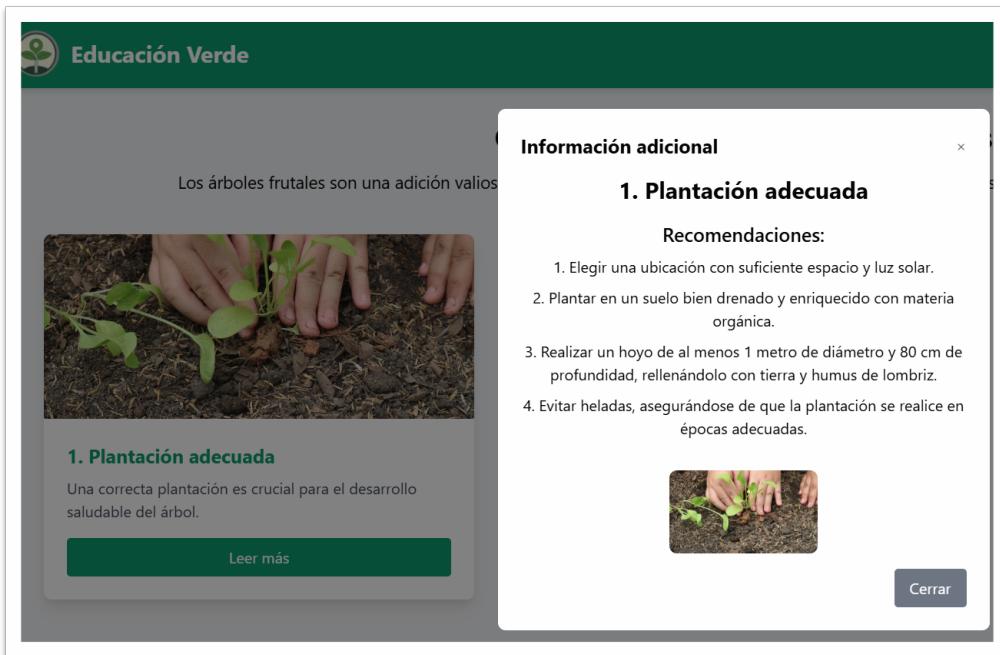


Figure 7.7: Informacion adicional

7.2.6 Comunidad

Presenta publicaciones relacionadas a la temática. Dichas publicaciones son hechas por parte de los usuarios, donde cada publicación deberá contener: título, contenido, imagen y tema.

The screenshot shows the "Comunidad" section. On the left, there's a form titled "Crear Nueva Publicación" with fields for "Título", "Contenido", "Imagen (opcional)" (with a "Choose File" button), and "Tema" (with a dropdown menu). A "Publicar" button is at the bottom. On the right, there's a published post titled "Consejo para plantar guayabas" with the content "En esta publicación les recomiendo plantar guayabas en una estación más calida" and the tag "Tema: conservacion". A "Crear Publicación" button is at the top right of the community area.

Figure 7.8: Visualización del apartado de Comunidad

7.2.7 Base de Datos Versión Web

La tabla de secciones será utilizada para filtrar las cartas o tarjetas dependiendo de su contenido. Las secciones serán especificadas más adelante en el documento. Esta tabla posee los siguientes atributos:

- **id_seccion (PK)**: Clave primaria, utilizada para identificar cada sección agregada.
- **nombre (varchar)**: Nombre de la sección, algunos ejemplos: “consejos”, “guías”, “beneficios”, etc.
- **descripción**: Breve descripción del contenido de cada sección.

7.2.8 Base de Datos Modal

La tabla de cartas será utilizada para almacenar la información de las tarjetas utilizadas para la visualización modal. La tabla posee los siguientes atributos:

- **id_carta (PK)**: Clave primaria que identifica cada tarjeta disponible en la aplicación.
- **id_seccion (FK)**: Clave foránea que referencia la tabla **secciones**, utilizada para clasificar las tarjetas en sus respectivas secciones.
- **id_usuario (FK)**: Clave foránea que referencia la tabla **usuarios**, permitiendo relacionar la tarjeta con el usuario creador.
- **titulo (varchar)**: Contiene el título de la tarjeta.
- **descripción (varchar)**: Contiene la información dentro de la tarjeta.
- **fecha (date)**: Almacena la fecha en la que se generó la tarjeta.

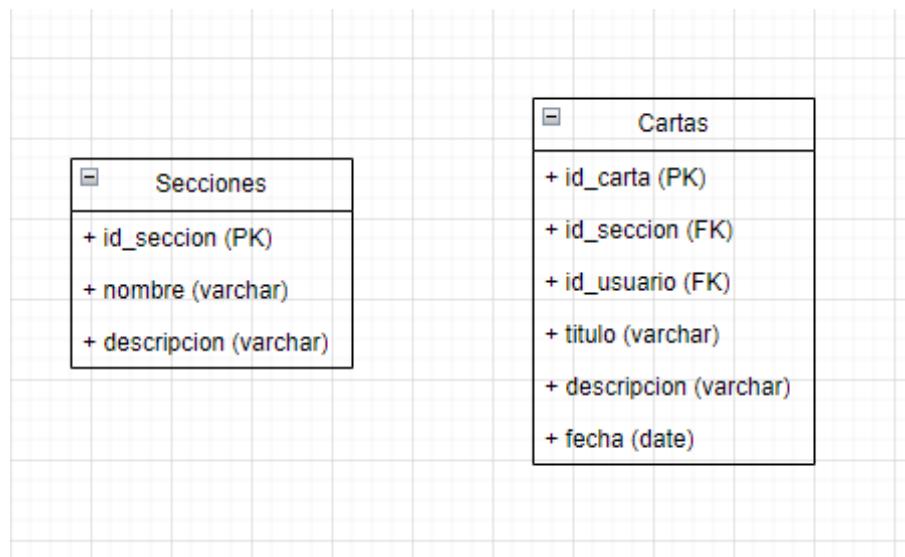


Figure 7.9: Visualización del apartado de Comunidad

Capítulo 8

Sistema de Reputación

8.1 Encuesta de Reputación

Se diseñó la interfaz de la encuesta de reputación con la inclusión de imágenes gráficas representativas del nivel de satisfacción del usuario. Estas imágenes fueron integradas a la pantalla de evaluación para mejorar la experiencia visual y facilitar la comprensión del sistema de calificación.



Figure 8.1: Perfil de usuario.



Figure 8.2: Diseño de medallas.

8.2 Funcionalidad de Foto de Perfil

Se habilitó una funcionalidad provisional para el cambio de foto de perfil. Al presionar la imagen actual, el sistema abre una ventana con los archivos locales del dispositivo, permitiendo al usuario seleccionar y recortar una nueva imagen. Esta funcionalidad se mantiene activa únicamente durante la sesión actual de ejecución.

8.3 Elementos del Perfil de Usuario

Se implementaron las siguientes características en la vista de perfil:

- **Medallas:** Se muestran correctamente según el nivel de reputación del usuario. Al seleccionar una medalla, se despliega una descripción detallada del logro correspondiente.
- **Menú lateral:** Ubicado en la parte superior izquierda, permite acceso a funciones adicionales, como la activación del modo oscuro. El menú se despliega en la misma posición al ser presionado.
- **Nivel de reputación:** Visualizado mediante diseños representativos en forma de plantas, que indican el progreso del usuario.
- **Banner superior:** Se añadió un encabezado con el nombre del proyecto “Trueque Verde” para reforzar la identidad visual.

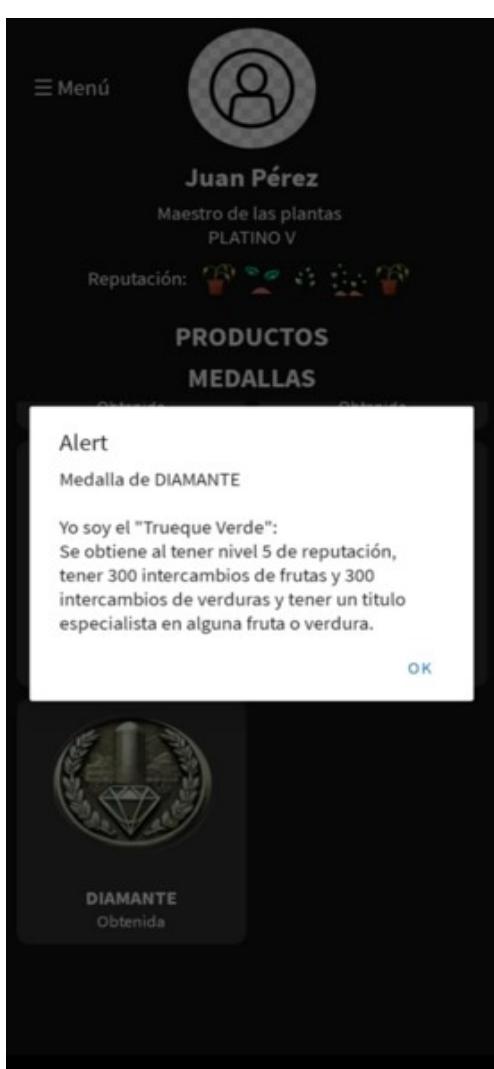


Figure 8.3: alerta.



Figure 8.4: menu lateral.

8.4 Integración del Sistema de Reputación

El sistema de reputación fue integrado a la navegación principal del proyecto, permitiendo su acceso desde el resto de las pantallas. Se realizaron pruebas con diferentes usuarios, verificando lo siguiente:

- Visualización correcta del nivel de reputación.
- Asignación automática de medallas (Bronce, Plata, Platino, Diamante) según el puntaje alcanzado.
- Presentación del título más alto logrado por el usuario.
- Actualización dinámica de la calificación global.



Figure 8.5: alerta.



Figure 8.6: menu lateral.

8.5 Comunicación Frontend–Backend

Se resolvieron problemas de conexión entre la interfaz y el backend, permitiendo la extracción y visualización en tiempo real de los datos del usuario. Esto garantiza una sincronización efectiva del estado de reputación y los logros obtenidos.

Capítulo 9

Implementación de la interfaz del panel de administración

El sistema de moderación de usuarios y publicaciones está diseñado para garantizar un entorno seguro dentro de la plataforma. Para ello, se han implementado funcionalidades que permiten la gestión de reportes, la revisión de publicaciones y la toma de decisiones sobre su estado.

Para optimizar el proceso de moderación, se han incorporado ventanas emergentes (modales) que muestran información detallada de cada reporte, incluyendo la fecha, el motivo y el usuario denunciante, además de opciones para tomar medidas directamente desde la interfaz.

9.1 Usuarios Problemáticos

Listado de Características y Funcionalidades del código, búsqueda de reportes tabla de usuarios responsables.

- Usuario reportado.
- Razón del reporte (ejemplo: "Contenido ofensivo").
- Cantidad de reportes recibidos.
- Estado del usuario.

9.2 Publicaciones

Este código implementa una página de administración para la moderación de publicaciones reportadas dentro de una plataforma. Su propósito es permitir que los administradores revisen, aprueben o eliminen publicaciones que han sido reportadas por los usuarios.

1-Filtros para Buscar Publicaciones Reportadas

- Pendiente de revisión" → No ha sido moderada aún.
- "Eliminada" → Se consideró inadecuada y se borró.
- "Aprobada" → Se validó y es visible para los usuarios.
- Botón "Filtrar" para aplicar los criterios de búsqueda.

9.3 Tabla de Publicaciones

- ID único de la publicación.
 - Usuario que realizó la publicación.
 - Razón del reporte (ejemplo: "Contenido ofensivo").

Estado actual de la publicación:

- Pendiente de revisión → Necesita ser moderada.
- Eliminada → Se eliminó por incumplimiento de normas.
- Aprobada con advertencia → Permitida, pero el usuario fue advertido.

Acciones disponibles para cada publicación:

- "Ver Detalles" → Muestra información completa de la publicación en un modal.
- "Advertir" → Envía una advertencia al usuario sobre su contenido.
- "Eliminar" → Elimina la publicación por incumplir normas.
- "Aprobar" → Permite que la publicación se mantenga visible.

9.4 Ventanas Emergentes (Modales) con Detalles de la Publicación

Usuario 1 tiene un modal con información detallada:

- Fecha del reporte: 21-02-2025.
- Razón: Contenido ofensivo.
- Reportado por: @usuarioReportadorX.

Contenido de la publicación:

- Imagen publicada (placeholder de imagen).
- Texto de la publicación, visible en un bloque de cita.

Acciones disponibles dentro del modal:

- Advertir al usuario sobre el contenido.
- Eliminar la publicación si infringe normas.
- Aprobar la publicación si no hay problemas.

9.5 Panel de Administracion

El panel de la interfaz de usuario del panel de administracion se esta realizado en Tailwind y contiene lo siguiente:

* Estadísticas de intercambios

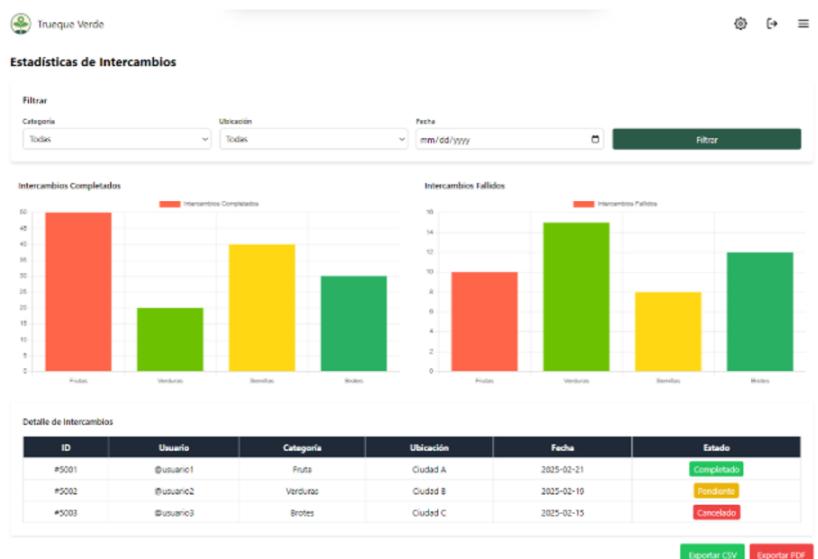


Figure 9.1: Visualización de Administracion

* Moderación de publicaciones

ID	Usuario	Razón	Estado	Acciones
#1234	@usuario1	Contenido ofensivo	Pendiente de revisión	Ver Detalles Advertir Eliminar Aprobar
#1235	@usuario2	Spam	Eliminado	Ver Detalles
#1236	@usuario3	Publicidad no permitida	Aprobada con advertencia	Ver Detalles

Figure 9.2: Visualización de la Moderacion

* Moderación de publicaciones nuevas

ID	Usuario	Contenido	Acciones
#2001	@usuario1	Ver Contenido	Aprobar Rechazar
#2002	@usuario2	Ver Contenido	Aprobar Rechazar
#2003	@usuario3	Ver Contenido	Aprobar Rechazar

Figure 9.3: Visualización de moderación de entrada

* Administración de usuarios

Usuario	Razón	Reportes	Estado	Acciones
@usuario1	Contenido ofensivo	3	Pendiente de revisión	Ver Detalles Advertir Suspender
@usuario2	Spam	1	Advertido	Ver Detalles Re-advertir Suspender
@usuario3	Estafa reportada	5	Cuenta suspendida	Ver Detalles
@usuario4	Lenguaje inapropiado	2	Advertido	Ver Detalles Re-advertir Suspender

Figure 9.4: Visualización de la administrador de usuarios

Capítulo 10

Equipos de Trabajos

Equipo 1

- Isaac Segovia Cazares
- Johan Adrian Gonzalez Loza
- Felipe Jesus Hernandez Guzman
- Mauricio Eliseo Rosales Weigend
- Kevin del Angel Colunga Vazquez
- Santiago Emmanuel Garcia Villanueva

Equipo 2

- Camarillo Estrada Alonso Emmanuel
- González Rodríguez Kevin Eduardo
- Jiménez Reséndiz Karla Rebeca
- Muñoz Ordoñez José Daniel

Equipo 3

- Alcocer Rangel Fernando
- Del Angel Díaz Danniel Ehud
- López Nava Alejandro
- Lozano Velazquez Brandon Armando
- Torres Martínez José Francisco
- Ramos González Edgar Karol

Equipo 4

- Jan Karlo Armendariz Gonzalez
- Leslie Yiharem Aguirre Isassi
- Andrea Sordel Guzmán
- Jessica Rubi Gallegos Rodriguez
- Joel Sánchez Gutiérrez
- Fabiola de Jesus Cuellar Hiyan

Equipo 5

- Garibaldi Zúñiga, Daniel
- González Pérez, Erick Eduardo
- Herrera Galaviz, Leonardo
- Márquez Ruiz, José de Jesús
- Rivera Jaramillo, Jesús Naum

Equipo 6

- Padron Zaleta Jared Alfredo
- Patiño Herrera Edgar Ivan
- Perez Calderon Gerardo Adrian
- Lopez Cantu Andres Manuel
- Zaleta Diaz Dalia Nohemi
- Gutierrez Mendoza Osbel Emmanuel

Equipo 7

- Abellaneda Mejia Grecia Guadalupe
- Borjas Vega Gustavo Adrián
- Flores Torres Jose Gerardo
- Juárez Clemente Julio Antonio
- Paz Miguel Angel
- Ramirez Reyes Eduardo