

**Asignatura:**  
 **DESARROLLO DE SOFTWARE PARA ANDROID [DSM441]**

**Proyecto:**

**Mundo verde**

**Docente: Ing. Manuel Alexander Jiménez García.**

**Año: 2025, Ciclo II, Grupo 02T**

**Integrantes:**

**Manuel Ezequiel guerrero granados GG241501**

**Enlace:  
https://github.com/Manuelguerrero234/DSM441-G02T.git**

**Índice**

1. **Objetivos............................................................................................... 1**
2. **Introducción .............................................................................................................. 2**
3. **Diseño UX/UI (Mockups) ................................................................................................... 3**
4. **Lógica de Solución ............................................................................................................. 4**
5. **Herramientas de Desarrollo ............................................................................................... 5**
6. **Presupuesto .............................................................................................................. 6**

**1. Objetivos**

**Objetivo General:**Desarrollar una aplicación móvil nativa para Android llamada "Mundo Verde" que funcione como una herramienta educativa y de gestión para catalogar, aprender sobre y realizar seguimiento virtual del cuidado de diferentes especies de plantas, con un enfoque especial en aquellas que se encuentran en peligro de extinción o tienen propiedades terapéuticas.

Objetivos Específicos:

1. Implementar un sistema de autenticación de usuarios (registro e inicio de sesión) seguro.
2. Diseñar y desarrollar una interfaz de usuario (UI) intuitiva y una experiencia de usuario (UX) agradable, basada en los mockups propuestos.
3. Crear una base de datos local para almacenar la información del usuario y su inventario virtual de plantas.
4. Desarrollar la funcionalidad principal de la aplicación: un catálogo categorizado de plantas (Semillas, Extintas, En Peligro, Terapéuticas).
5. Implementar una característica de "alimentación" o recordatorio para simular el cuidado de una planta seleccionada por el usuario.
6. Realizar pruebas exhaustivas de funcionalidad y usabilidad para garantizar la calidad del producto final.

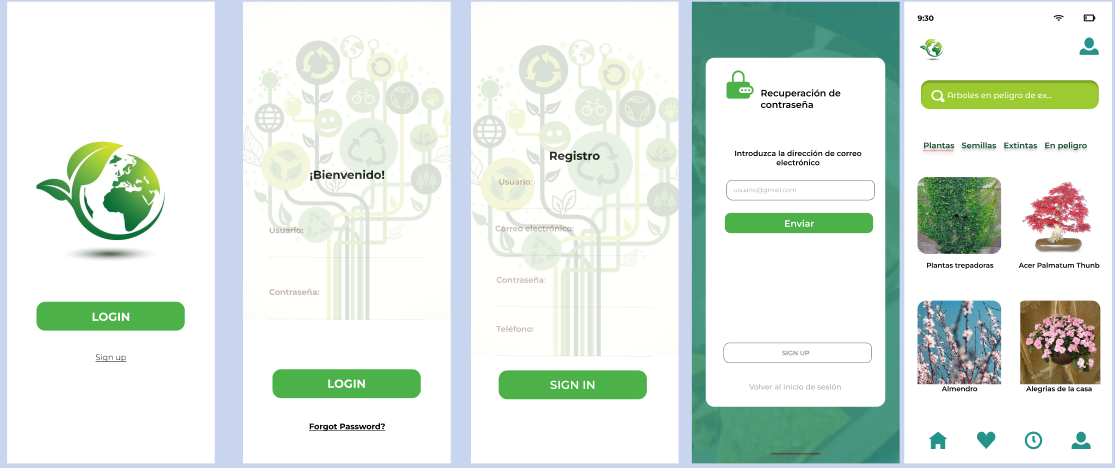
**2. Introducción**

En la era digital actual, la conservación ambiental y la educación ecológica encuentran en la tecnología una aliada poderosa. "Mundo Verde" nace como un proyecto que busca aprovechar el potencial de los dispositivos móviles para acercar a las personas al mundo de la botánica y la conservación de una manera interactiva y accesible.

Esta aplicación está dirigida a estudiantes, educadores, ambientalistas y al público en general interesado en aprender sobre diferentes especies de plantas, su estado de conservación y sus usos. La app simula la experiencia de cuidar una planta, enviando recordatorios al usuario, con el objetivo principal de generar conciencia y fomentar el interés por la preservación de la biodiversidad vegetal.

El presente documento detalla la planificación de la primera fase del desarrollo de "Mundo Verde", describiendo la estrategia de solución, el diseño visual propuesto a través de mockups, las herramientas tecnológicas seleccionadas y un presupuesto estimado para el proyecto.

**3. Diseño UX/UI (Mockups)**

****

**Descripción de las Pantallas:**

* Pantalla de Login (Inicio de Sesión): Diseño minimalista con campos para usuario y contraseña, y opciones para registrarse o recuperar contraseña.
* Pantalla de Registro (Sign Up): Formulario para que un nuevo usuario cree una cuenta con sus credenciales.
* Pantalla de Recuperación de Contraseña: Permite al usuario restablecer su contraseña mediante su correo electrónico.
* Pantalla Principal (Home/Planta): El corazón de la aplicación. Muestra las categorías de plantas (Semillas, Extintas, En Peligro, Terapéuticas), información de una planta en específico ("Acer Panestecin Thumb") y un botón para "Alimentar" o cuidar de ella.

Flujo de Usuario: Inicio → Login/Registro → Recuperación de Contraseña (si es necesario) → Home (Menú Principal).

**4. Explicación Detallada de la Lógica de Solución**

**La lógica para resolver el problema se dividirá en los siguientes módulos:**

1**. Módulo de Autenticación (Login, Registro, Recuperación):**

* Lógica: Se utilizará Firebase Authentication como servicio backend para gestionar de forma segura el registro (creando un usuario con email y contraseña), el inicio de sesión (validando las credenciales contra Firebase) y el restablecimiento de contraseña (enviando un correo electrónico con un enlace seguro a través de los servicios de Firebase).
* Flujo: La app primero checkeará si hay un usuario ya logueado (sesión persistente). Si no lo hay, mostrará la pantalla de Login. Desde ahí, el usuario puede ir a Registro o Recuperación. Tras un login exitoso, se redirigirá a la pantalla principal (Home).

**2. Módulo de Base de Datos:**

* Lógica: Se empleará Room Persistence Library, que es parte de Android Jetpack, para crear y gestionar una base de datos SQLite local en el dispositivo.
* Estructura: Se crearán tablas para almacenar la información del usuario (preferencias) y el inventario de plantas del usuario (qué plantas ha "adquirido" o está "cuidando").
* Ventaja: Room facilita el acceso a la base de datos, reduce código boilerplate y permite una integración fluida con los componentes de Android.

**3. Módulo de la Interfaz Principal (Home) y Catálogo:**

* Lógica: La pantalla principal se construirá usando Fragments y un ViewPager2 o un BottomNavigationView para navegar entre las diferentes categorías de plantas (cada categoría podría ser un Fragment).
* Datos: La información de las plantas (nombre, descripción, imagen, categoría, etc.) se almacenará inicialmente en un archivo JSON local dentro de la carpeta res/raw. Esto se cargará la primera vez que la app se ejecute y se insertará en la base de datos Room.
* Navegación: Se utilizará el componente Navigation de Android Jetpack para gestionar la navegación entre pantallas (Login -> Home) y entre fragments (categorías).

**4. Módulo de "Alimentación" / Recordatorios:**

* Lógica: Se implementará usando la clase WorkManager de Android Jetpack. WorkManager programará una tarea periódica (por ejemplo, cada 24 horas) que:
  1. Verificará si el usuario ha "alimentado" su planta hoy.
  2. Si no lo ha hecho, mostrará una Notification (notificación del sistema) recordándoselo.
* Persistencia: El estado de la última "alimentación" se guardará en la base de datos Room.

**5. Detalle de Herramientas a Utilizar**

| **Categoría** | **Herramienta/Lenguaje/Tecnología** | **Propósito** |
| --- | --- | --- |
| **Entorno de Desarrollo** | **Android Studio Giraffe/Hedgehog** | **IDE principal para el desarrollo, debugging y empaquetado de la aplicación.** |
| **Lenguaje de Programación** | **Kotlin** | **Lenguaje moderno, conciso y seguro oficialmente soportado por Google para el desarrollo Android.** |
| **Arquitectura** | **MVVM (Model-View-ViewModel)** | **Patrón de arquitectura para separar la lógica de negocio (ViewModel) de la interfaz de usuario (View), haciendo el código más testeable y mantenible.** |
| **Componentes de Android Jetpack** | **- ViewModel - LiveData - Room Persistence Library - Navigation Component - WorkManager - DataBinding** | **Conjunto de librerías que ayudan a seguir las mejores prácticas, reducir código boilerplate y crear apps robustas.** |
| **Base de Datos** | **SQLite (a través de Room)** | **Base de datos local para almacenar la información de la aplicación de manera persistente.** |
| **Autenticación** | **Firebase Authentication** | **Servicio de Google para gestionar la autenticación de usuarios de forma segura y sin necesidad de un backend propio.** |
| **Diseño y Mockups** | **Figma / Adobe XD** | **Herramientas de diseño para crear los mockups y prototipos de la interfaz de usuario.** |
| **Control de Versiones** | **Git / GitHub** | **Sistema para controlar los cambios en el código fuente y facilitar el trabajo en equipo.** |

**6. Presupuesto Estimado del Proyecto**

**El presupuesto se calcula considerando el valor hora de un desarrollador profesional y el tiempo estimado de desarrollo.**

| **Concepto** | **Detalle** | **Costo Estimado (USD)** |
| --- | --- | --- |
| **Mano de Obra (Desarrollo)** | **1 desarrollador senior ($50/hora) x 120 horas estimadas de desarrollo (3 semanas a 40 h/sem)** | **$6,000** |
| **Mano de Obra (Diseño UX/UI)** | **1 diseñador UX/UI ($35/hora) x 40 horas estimadas** | **$1,400** |
| **Licencias y Servicios** | **Cuenta Google Cloud Platform/Firebase (plan Blaze, costos por uso, estimación baja para tráfico inicial)** | **$50 mensuales** |
| **Contingencia (10%)** | **Imprevistos y gastos adicionales** | **$745** |
|  | **Costo Total de Desarrollo (Estimado)** | **$8,195** |

**Nota sobre el presupuesto:**Este es un presupuesto estimado para el desarrollo de la aplicación desde cero por parte de profesionales. Para el contexto académico del proyecto, los costos reales en los que incurrirá el equipo serán principalmente de tiempo y posiblemente de una cuenta gratuita de Firebase, que ofrece un tier gratuito muy generoso para este tipo de proyectos de aprendizaje.