

Reporte y trabajo de investigación
Proyecto Final
Asignatura: Cómputo Móvil
Profesor: Ing Marduk Pérez de Lara Domínguez
Semestre: 2025-2
Grupo: 03
Fecha: 23 de abril de 2025
Equipo: CompuMoviliOS
Caballero Hernandez Juan Daniel
Ramirez Gonzalez José Miguel

Introducción

En un entorno marcado por la hiperconectividad digital y el uso intensivo de dispositivos móviles, las personas enfrentan desafíos crecientes relacionados con su bienestar físico, mental y emocional. El uso excesivo de redes sociales, la exposición constante a estímulos digitales y la dificultad para establecer rutinas saludables han generado un contexto en el que es urgente ofrecer herramientas tecnológicas que contribuyan a recuperar el equilibrio y la autorregulación personal.

En este escenario surge **GreenGrocery!**, una aplicación móvil desarrollada con el propósito de convertirse en un acompañante digital enfocado en el bienestar del usuario. La propuesta se centra en integrar funcionalidades orientadas al control consciente del tiempo digital, la práctica de actividad física, la meditación guiada y la interacción dentro de una comunidad positiva. Todo esto se ofrece desde una experiencia de usuario unificada, minimalista y altamente intuitiva, diseñada para dispositivos iPhone dentro del ecosistema iOS.

Este documento técnico reúne los principales lineamientos para el desarrollo profesional de la aplicación, partiendo del trabajo conceptual y funcional realizado durante el Hackathon 2025. El contenido proporciona una visión estructurada de la arquitectura, diseño y planeación del producto, con el objetivo de establecer una base sólida que permita su evolución desde un prototipo académico hacia una aplicación escalable y publicable en tiendas digitales.

A lo largo del documento se abordan aspectos clave como la definición del producto mínimo viable (MVP), compatibilidad técnica con dispositivos y versiones de sistema operativo, selección de tecnologías, planeación ágil del desarrollo, lineamientos de seguridad de la información, así como la documentación detallada de cada pantalla de la aplicación. Todo ello

orientado a garantizar que GreenGrocery! no solo cumpla con los requerimientos funcionales definidos, sino que también logre posicionarse como una solución real y efectiva frente a las problemáticas digitales contemporáneas.

En conjunto, esta introducción sirve como punto de partida para comprender la visión integral de GreenGrocery! y el enfoque metodológico adoptado en su desarrollo, alineando los objetivos del proyecto con buenas prácticas de ingeniería de software y diseño centrado en el usuario.

Alcance y Producto Mínimo Viable (MVP)

Alcance general de la aplicación

La aplicación **GreenGrocery!** tiene como objetivo principal convertirse en una herramienta integral de acompañamiento digital para mejorar el bienestar físico, mental y digital de los usuarios. Está diseñada para jóvenes adultos (18 a 35 años) con alta exposición al entorno digital y un interés creciente en prácticas de autocuidado.

El alcance funcional de la aplicación abarca los siguientes dominios:

- **Actividad física:** Acceso a rutinas básicas de ejercicio guiado (fuerza, estiramiento, cardio) mediante contenido externo validado.
- **Meditación y mindfulness:** Inclusión de meditaciones programadas y meditaciones sorpresa para reducir el estrés y fomentar la atención plena.
- **Control de tiempo frente a pantallas:** Monitorización del tiempo en apps de entretenimiento (TikTok, YouTube, Instagram) y posibilidad de bloqueo temporal.
- **Comunidad y apoyo entre usuarios:** Implementación de un foro público donde los usuarios pueden compartir experiencias, consejos o preguntas relacionadas con el bienestar.
- **Seguimiento de hábitos:** Registro y control de hábitos saludables definidos por el usuario, enfocados en desconexión digital y bienestar mental.

El alcance técnico incluye:

- Desarrollo exclusivo para dispositivos iPhone (iOS 16 o superior).
- Uso del framework SwiftUI para diseño responsivo.
- Almacenamiento local mediante CoreData.
- Soporte para operación sin conexión en módulos esenciales.

Definición del Producto Mínimo Viable (MVP)

El MVP de GreenGrocery! representa la versión inicial funcional que valida la propuesta de valor con usuarios reales, ofreciendo las funciones esenciales con una experiencia de usuario fluida, sin incluir todas las características avanzadas previstas en versiones futuras.

El MVP incluye los siguientes módulos funcionales mínimos:

- **Inicio de sesión e identificación del usuario:** Acceso mediante credenciales básicas. No se requiere integración con Apple ID en esta etapa.
- **Dashboard diario:** Visualización compacta del estado diario (hábitos completados, minutos de meditación, tiempo de pantalla).
- **Registro de hábitos:** Posibilidad de marcar hábitos como completados (ej. caminar, desconectarse).
- **Rutinas de ejercicio:** Acceso a una selección reducida de rutinas por categoría, enlazadas a contenido externo.
- **Meditación sorpresa básica:** Generación de hasta 2 alertas diarias con sugerencia de meditación corta integrada.
- **Control digital simple:** Visualización del tiempo diario de uso de otras aplicaciones. El bloqueo estará disponible solo como una sugerencia no vinculante.
- **Foro comunitario básico:** Lectura y publicación de mensajes públicos, sin sistema de reacciones ni moderación avanzada.

Módulos excluidos del MVP

Los siguientes componentes quedan fuera del MVP y se consideran para fases posteriores del desarrollo:

- **Funciones premium o de suscripción.**
- **Bloqueo real de aplicaciones mediante permisos del sistema operativo.**
- **Meditaciones guiadas profundas y personalizadas.**
- **Análisis avanzado del comportamiento digital.**
- **Moderación automática y filtrado de contenido en el foro.**
- **Gamificación y sistema de recompensas por progreso.**
- **Soporte multiplataforma (Android, iPad).**

Objetivo del MVP

El objetivo del MVP es validar con usuarios reales que:

- La propuesta de bienestar integral es atractiva y comprensible.
- El modelo de interfaz y navegación es funcional e intuitivo.
- Existe interés sostenido por parte de los usuarios al usar la app durante al menos 5 días consecutivos.

- Las funciones básicas de autocuidado digital y mindfulness son efectivas como primer paso hacia una vida más equilibrada.

A partir de esta validación, se definirá un roadmap de evolución funcional que incluirá optimizaciones de experiencia, ampliación de módulos y monetización progresiva a través de contenido premium.

Requerimientos

Reglas de negocios

- **RN-01. Bloqueo de aplicaciones distractoras:** Si el usuario supera el tiempo límite configurado en una aplicación (por ejemplo, TikTok), esta será bloqueada automáticamente durante el tiempo establecido. No se podrá desbloquear manualmente hasta que el tiempo expire.
- **RN-02. Activación de meditaciones sorpresa:** El sistema debe activar entre 2 y 4 sesiones de meditación aleatorias por día. Estas no deben coincidir con sesiones activas de ejercicio ni horarios de alta actividad en pantalla.
- **RN-03. Recompensas por hábitos saludables:** Cada hábito marcado como completado se registra en el historial diario y se le asigna un punto para el seguimiento del progreso del usuario.
- **RN-04. Personalización de recomendaciones:** Las sugerencias de meditación, ejercicio y hábitos deben adaptarse al perfil e intereses del usuario, definidos al iniciar sesión.
- **RN-05. Restricción de foros a usuarios autenticados:** Solo usuarios autenticados pueden participar en los foros. Cada publicación debe asociarse al perfil del autor.
- **RN-06. Moderación de comunidad:** Todo contenido en el foro comunitario debe pasar por filtros automáticos para detectar lenguaje ofensivo o contenido inapropiado. Opcionalmente, pueden aplicarse mecanismos de denuncia.
- **RN-07. Registro de progreso diario:** El dashboard debe mostrar métricas acumuladas del día, incluso si la app ha sido cerrada temporalmente.
- **RN-08. Acceso limitado en versión gratuita:** En el modelo freemium, solo algunas funciones están habilitadas (por ejemplo, meditaciones básicas y rutinas generales). El acceso a funciones avanzadas requiere suscripción.
- **RN-09. Persistencia local obligatoria:** El progreso del usuario debe guardarse localmente para permitir el funcionamiento sin conexión (hábitos, rutinas marcadas, tiempo bloqueado, etc.).
- **RN-10. Control del uso digital compatible con políticas del sistema operativo:** La visualización del tiempo de uso de aplicaciones y su bloqueo debe apegarse a las políticas de privacidad y permisos del sistema operativo.

Requerimientos funcionales

ID	Nombre	Descripción Técnica
RF-01	Inicio de sesión	Permite la autenticación mediante email y contraseña o Apple ID.
RF-02	Dashboard diario	Muestra un resumen del día con estadísticas de actividad física, meditación y tiempo frente a pantalla.
RF-03	Actividad física	Permite seleccionar ejercicios (correr, estiramiento, fuerza) y seguir rutinas a través de videos externos.
RF-04	Meditaciones guiadas	Acceso a una galería de meditaciones y generación aleatoria de sesiones sorpresa durante el día.
RF-05	Control de uso digital	Muestra el tiempo de uso de aplicaciones y permite configurar bloqueos por intervalos definidos.
RF-06	Foro comunitario	Espacio de interacción para crear, comentar y reaccionar a publicaciones de otros usuarios.
RF-07	Gestión de hábitos	Configuración y seguimiento de hábitos saludables diarios (como caminar, meditar, desconectarse del teléfono, etc.).

Requerimientos no funcionales

ID	Nombre	Descripción Técnica
RNF-01	Plataforma inicial	La app se desarrollará exclusivamente para iOS utilizando SwiftUI.
RNF-02	Interfaz minimalista	Se utilizará un diseño centrado en el usuario, simple, intuitivo y coherente con el estilo de bienestar digital.
RNF-03	Tiempo de carga	El tiempo de inicio no debe superar los 2 segundos en dispositivos iOS modernos.
RNF-04	Operación sin conexión	Funciones como hábitos y meditaciones descargadas deben funcionar sin conexión a internet.

RNF-05	Privacidad y políticas	El acceso al tiempo de uso debe cumplir con las políticas de privacidad y permisos del sistema operativo.
RNF-06	Almacenamiento local	Se usará CoreData para persistencia local de datos con posibilidad de sincronización futura.

Compatibilidad de la aplicación GreenGrocery!

Dispositivos

La aplicación **GreenGrocery!** ha sido desarrollada exclusivamente para dispositivos móviles del ecosistema Apple, específicamente para **iPhone**. Esta decisión responde a razones tanto técnicas como estratégicas que favorecen una implementación más robusta, eficiente y mantenible:

- **Dispositivo objetivo:** iPhone (modelos con arquitectura ARM64), excluyendo iPad y iPod Touch. La aplicación no ha sido diseñada ni optimizada para ejecutarse en iPadOS ni macOS, lo que garantiza un enfoque preciso en la experiencia móvil vertical tradicional.
- **Tamaños de pantalla soportados:** Se ha asegurado compatibilidad con los tamaños de pantalla más comunes dentro del ecosistema iPhone:
 - 5.4 pulgadas (iPhone 12 Mini, 13 Mini)
 - 6.1 pulgadas (iPhone 11, 12, 13, 14, 15)
 - 6.5–6.7 pulgadas (iPhone 11 Pro Max, 12/13/14/15 Pro Max)

Estas resoluciones cubren una amplia gama de dispositivos actualmente activos en el mercado, garantizando una experiencia visual consistente.

- **Diseño adaptativo:** Se utilizó **SwiftUI** como framework de interfaz, permitiendo un diseño fluido y adaptativo a distintas resoluciones dentro de los límites definidos. El layout ha sido validado en simuladores y vistas previas dentro de Xcode, asegurando la correcta presentación de todos los componentes visuales en los modelos seleccionados.
- **Orientaciones soportadas:**
 - **Orientación vertical (portrait):** Es la única orientación soportada y ha sido considerada como la principal en el diseño y navegación. Toda la arquitectura visual de la aplicación está concebida para maximizar la usabilidad en este formato.
 - **Orientación horizontal (landscape):** No soportada. Esta decisión responde a un diseño centrado en la simplicidad de uso, accesibilidad con una sola mano y coherencia con los hábitos de interacción más comunes en el contexto de uso de aplicaciones enfocadas en bienestar y control digital.

Sistema Operativo

GreenGrocery! ha sido diseñada para ejecutarse exclusivamente en **iOS 16.0 o superior**. Esta elección obedece a criterios técnicos que aseguran el uso de tecnologías modernas y una experiencia de usuario sólida:

- **Compatibilidad mínima:** iOS 16.0 ha sido establecido como la versión base debido a que:
 - Es compatible con iPhone 8 en adelante, lo que abarca un porcentaje considerable de dispositivos activos.
 - Representa una cuota significativa del mercado según datos de adopción oficial de Apple para 2024.
 - Permite aprovechar **SwiftUI 4.0**, cuya evolución facilita interfaces reactivas, animaciones fluidas y mayor modularidad.
- **Compatibilidad con APIs clave:** Las funcionalidades principales —como el control de uso de aplicaciones, navegación mediante tabs y stacks, y animaciones en tiempo real— dependen de APIs modernas disponibles únicamente a partir de iOS 16.0.
- **Razonamiento de exclusión de versiones anteriores:**
 - Versiones como iOS 15 no ofrecen soporte completo para las últimas herramientas de SwiftUI sin depender de soluciones intermedias que aumentarían la complejidad técnica.
 - Apple ya no mantiene soporte de seguridad activo para versiones anteriores a iOS 16, lo cual comprometería la integridad del producto.
- **Enfoque estratégico en iOS:** El equipo cuenta con dominio técnico sobre el entorno de desarrollo nativo de Apple, lo cual facilita una implementación optimizada, disminuye los errores en tiempo de ejecución y mejora la integración con servicios del sistema operativo.
- **Compatibilidad futura:** GreenGrocery! ha sido verificada para funcionar correctamente en dispositivos con iOS 17, sin requerir modificaciones sustanciales. Se planea mantener la compatibilidad con versiones futuras mediante pruebas automatizadas y actualizaciones continuas.

En resumen, GreenGrocery! está cuidadosamente diseñada para ejecutarse en dispositivos iPhone con iOS 16.0 o superior, exclusivamente en orientación vertical, y con soporte garantizado para las resoluciones más comunes. Este enfoque limitado pero optimizado permite ofrecer una experiencia de usuario coherente, eficiente y alineada con las capacidades modernas del ecosistema Apple, reduciendo la carga técnica y asegurando un producto estable y escalable.

Lenguajes de programación y herramientas utilizadas

La aplicación **GreenGrocery!** fue desarrollada específicamente para el sistema operativo **iOS**, utilizando el lenguaje de programación **Swift** y el framework **SwiftUI**. Esta elección responde tanto a criterios de eficiencia como de modernidad y alineación con las tendencias actuales de desarrollo de aplicaciones móviles en entornos Apple.

Swift es un lenguaje moderno, seguro y de alto rendimiento, creado por Apple, que ofrece una sintaxis clara y concisa, facilitando el desarrollo de aplicaciones robustas y mantenibles. Su integración nativa con el ecosistema de Apple permite un aprovechamiento completo de las capacidades de hardware y software de los dispositivos iPhone y iPad.

SwiftUI es el framework declarativo de interfaces gráficas de usuario introducido por Apple. Fue seleccionado por sus ventajas en rapidez de prototipado, reutilización de componentes, compatibilidad con animaciones fluidas, adaptabilidad entre dispositivos y su estrecha integración con herramientas como Xcode y Preview.

Herramientas utilizadas:

- **Xcode 15 (IDE oficial de Apple):** Utilizado para todo el proceso de desarrollo, diseño de interfaces, pruebas locales y emulación de dispositivos.
- **Swift 5.9:** Versión moderna y estable del lenguaje utilizada en el proyecto, con soporte completo para estructuras reactivas y asincrónica.
- **iOS Simulator:** Para pruebas funcionales en distintos modelos de iPhone (ej. iPhone 14, iPhone SE).
- **Miro:** Herramientas complementarias para diseñar los wireframes, mockups y la estructura visual previa a la implementación.

Esta combinación de tecnologías permite construir una aplicación con excelente rendimiento, visualmente atractiva, y preparada para futuras ampliaciones o integración con servicios en la nube como Firebase, HealthKit o CoreData, si se decidiera escalar la app.

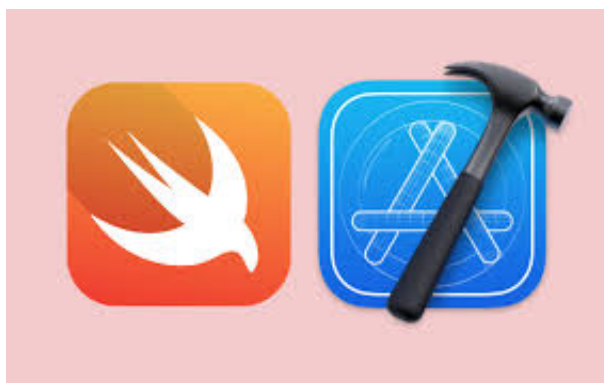


Figura 1: Swift & Xcode

Permisos y adecuaciones para su publicación en App Store

Para que la aplicación **GreenGrocery!** pueda ser publicada oficialmente en la **App Store** de Apple, es necesario cumplir con una serie de requisitos técnicos, legales y administrativos establecidos por la **Apple Developer Program**.

Registro de desarrollador: El primer paso es inscribirse en el Apple Developer Program, lo cual implica el pago de una tarifa anual (actualmente de \$99 USD). Este registro permite acceder a herramientas avanzadas de desarrollo, distribución, pruebas internas (TestFlight) y análisis de rendimiento. [5]

Permisos y políticas a considerar:

- **Política de privacidad:** La aplicación debe incluir una política de privacidad clara y accesible, donde se especifique el tipo de datos recopilados, su uso y tratamiento, de acuerdo con el *App Store Review Guidelines*.
- **Gestión de datos personales:** Si se almacena o procesa información como correo electrónico, hábitos de uso o tiempos de pantalla, se deberá cumplir con normativas como el **Reglamento General de Protección de Datos (GDPR)** en Europa o la **Ley de Privacidad del Consumidor de California (CCPA)** en EE.UU.
- **Permisos de uso del sistema:** Si en versiones futuras la app utiliza sensores como acelerómetro, HealthKit, notificaciones o localización, será obligatorio solicitar el consentimiento del usuario y declarar su propósito en el archivo **Info.plist**.
- **Accesibilidad:** Apple recomienda cumplir con las directrices de accesibilidad de iOS, asegurando que la aplicación sea utilizable por personas con discapacidades (uso de VoiceOver, tamaño de texto, contraste, etc.).

Pruebas y revisión: Antes de ser aprobada, la aplicación debe pasar por un proceso de revisión por parte del equipo de la App Store. Se evaluarán aspectos como estabilidad, experiencia de usuario, cumplimiento con las políticas de contenido, rendimiento y seguridad.

Equipo de trabajo y roles que intervienen

El desarrollo de la aplicación **GreenGrocery!** ha requerido conformar un equipo multidisciplinario, aplicando principios de desarrollo ágil para estructurar, diseñar e implementar una solución digital centrada en el bienestar físico y mental del usuario. La composición del equipo ha permitido cubrir todas las etapas necesarias para llevar a cabo un proyecto de desarrollo móvil, desde su concepción hasta su ejecución funcional.

Estructura del equipo y funciones:

- **Actuar como Project Manager / Scrum Master:** Organizar y facilitar el proceso de desarrollo siguiendo la metodología ágil Scrum. Definir los sprints, planificar reuniones de seguimiento (daily stand-ups), coordinar la comunicación entre los integrantes

del equipo y eliminar obstáculos que impidan el avance. Documentar decisiones, avances y cambios durante todo el proceso.

- **Diseñar como UI/UX Designer:** Elaborar experiencias de usuario funcionales, accesibles y coherentes. Prototipar wireframes interactivos con herramientas como Figma o Adobe XD. Establecer una guía visual que contemple tipografía, paleta de colores, iconografía y espaciado. Validar que las interfaces sean intuitivas y consistentes con los principios de diseño centrado en el usuario.
- **Desarrollar como iOS Frontend Developer:** Convertir los prototipos visuales en vistas funcionales mediante el uso de **Swift** y **SwiftUI**. Implementar navegación, diseño responsivo, campos de entrada, componentes reutilizables y lógica básica de interacción. Adaptar la aplicación para múltiples tamaños de pantalla dentro del ecosistema iOS.
- **Estructurar como Desarrollador de lógica / Backend local:** Diseñar modelos de datos, estructuras de control y lógica de negocio. Preparar la base para una futura integración con almacenamiento local (ej. CoreData) o remoto (ej. Firebase). Anticipar posibles rutas de expansión para nuevas funcionalidades sin comprometer el rendimiento ni la arquitectura base.
- **Verificar como Tester / QA:** Probar manualmente las distintas funcionalidades de la app en dispositivos físicos o simuladores. Detectar errores de navegación, visualización o experiencia de usuario. Documentar fallos y sugerencias de mejora. Validar que los flujos funcionen correctamente bajo distintos escenarios de uso.

Colaborar de forma interdisciplinaria: A pesar de haber definido roles específicos, asumir funciones cruzadas en función de las necesidades de cada etapa. Apoyar tareas de otros integrantes del equipo y trabajar de manera flexible para mejorar la eficiencia y la cohesión del desarrollo.

Adoptar un enfoque profesional: Emular la estructura organizativa de equipos reales de desarrollo de software. Desarrollar habilidades de especialización, comunicación efectiva y planificación iterativa. Valorar el trabajo en equipo como elemento clave en la creación de productos digitales funcionales, escalables y orientados a mejorar la vida de las personas.

Plan de trabajo y enfoque ágil de desarrollo

Para desarrollar la aplicación **GreenGrocery!**, se adoptó un enfoque de trabajo ágil basado en la metodología **Scrum**. Esta elección permitió organizar las tareas en sprints cortos, iterativos e incrementales, donde en cada ciclo se generó un entregable funcional de la aplicación. El desarrollo se dividió en **épicas** (objetivos generales) y **user stories** (funcionalidades concretas), priorizando aquellas que aportaran mayor valor al usuario desde las primeras versiones.

El equipo trabajó con **sprints de una semana**, con reuniones de planificación, daily stand-ups, revisión y retrospectiva al cierre de cada ciclo. Este enfoque permitió detectar problemas tempranos, ajustar prioridades y mantener una visión clara de avance continuo.

A continuación se muestra un esquema del cronograma de trabajo:

Tabla 1. Cronograma de actividades (modelo Gantt simplificado)

Actividad / Épica	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Planeación del proyecto y backlog	✓					
Diseño de wireframes y prototipos	✓	✓				
Desarrollo de pantalla de inicio/login		✓				
Implementación del dashboard		✓	✓			
Control de uso de aplicaciones			✓			
Sección de meditación			✓	✓		
Sección de actividad física				✓		
Foro y comunidad				✓	✓	
Testing general y correcciones					✓	
Documentación final y entrega						✓

Notas:

- Cada entrega funcional fue validada en un entorno de pruebas, integrando retroalimentación inmediata del equipo.
- Se priorizó el desarrollo del MVP (Producto Mínimo Viable) en las primeras semanas, y las funciones complementarias en etapas posteriores.

Análisis de seguridad de la información

El desarrollo de la aplicación **GreenGrocery!** contempla principios fundamentales de protección de datos personales y buenas prácticas en seguridad informática. Aunque en su versión actual no incluye una base de datos en la nube ni almacenamiento masivo de información sensible, se proyecta una evolución del sistema hacia funcionalidades que involucrarán la gestión de datos del usuario. Por ello, resulta fundamental identificar qué información manejará la app y cómo se protegerá.

Tipos de información que puede manejar la aplicación:

- **Correo electrónico:** utilizado para autenticación del usuario y recuperación de contraseña.
- **Datos de uso:** hábitos de ejercicio, meditación, uso de apps y seguimiento de metas diarias.
- **Preferencias personales:** configuraciones del usuario, hábitos alimenticios o intereses en tipos de contenido.

Riesgos identificados:

- **Filtración de datos personales:** acceso no autorizado a cuentas o a la información almacenada localmente.

- **Pérdida de información:** por fallos en el dispositivo o errores en la sincronización futura con servidores externos.
- **Falsificación de identidad:** en caso de no implementar correctamente validaciones y control de sesiones.

Medidas de protección propuestas:

- **Autenticación segura:** implementar validaciones robustas en el login (validación de email, contraseña cifrada).
- **Almacenamiento local protegido:** utilizar mecanismos como **Keychain** en iOS para proteger credenciales y datos sensibles del usuario.
- **Política de privacidad clara:** informar al usuario sobre qué datos se recopilan, cómo se utilizan y con qué propósito.
- **Cifrado de información:** aplicar cifrado en tránsito (TLS) en futuras integraciones con servidores o servicios en la nube.
- **Control de sesiones y caducidad:** validar sesiones activas y establecer tiempos límite de autenticación para mayor seguridad.

Consideraciones futuras: En versiones avanzadas, se contempla la integración con servicios como **Firebase Authentication**, bases de datos en la nube (**Firestore**) o **HealthKit**, lo que implicará cumplir con normativas internacionales de protección de datos como el **GDPR** (Europa) y la **CCPA** (California). Asimismo, para la publicación en la App Store, se requerirá declarar explícitamente qué datos se recopilan y para qué se utilizan, conforme a la política de privacidad de Apple.

Estimaciones de tiempo de desarrollo y costos

La planificación del desarrollo de la aplicación **GreenGrocery!** se ha realizado considerando una aproximación realista basada en fases, tareas y roles definidos. A continuación se presentan las estimaciones generales de tiempo y los costos aproximados asociados a cada etapa del ciclo de vida de la aplicación, teniendo en cuenta tanto la fase de desarrollo inicial como los requerimientos de mantenimiento a mediano plazo.

Tiempo estimado de desarrollo:

- **Fase de planeación y diseño de UX/UI:** 2 semanas
- **Desarrollo del MVP (versión funcional mínima):** 4 semanas
- **Testing, revisión y corrección de errores:** 1 semana
- **Documentación y preparación para entrega/publicación:** 1 semana

Total estimado: 8 semanas de trabajo, considerando un equipo de 3 a 5 personas con dedicación parcial.

Costos estimados (en USD):

Concepto	Costo estimado (USD)
Diseño UI/UX (freelancer o interno)	\$500 – \$800
Desarrollo iOS (Swift + SwiftUI)	\$2,000 – \$3,500
Pruebas y QA	\$400 – \$600
Registro en Apple Developer Program (anual)	\$99
Mantenimiento mensual estimado (actualizaciones, bugs)	\$100 – \$200
Costos opcionales (dominio, servidor backend, base de datos)	\$250 – \$500
Total aproximado (sin backend)	\$3,000 – \$5,000

Consideraciones adicionales:

- Estos costos corresponden a estimaciones para un proyecto académico o de bajo alcance comercial. En contextos empresariales, el presupuesto podría duplicarse o triplicarse.
- El mantenimiento se considera a razón de pequeñas mejoras, correcciones o actualizaciones visuales.
- No se incluye marketing, legalidad ni costos de infraestructura en la nube (si se quisiera escalar).

Con una buena organización del equipo y herramientas gratuitas de desarrollo (Xcode, Miro), el desarrollo de **GreenGrocery!** resulta viable y sostenible, tanto a nivel técnico como económico.

Wireframes

Para una mejor visualización de los wireframes, estos fueron añadidos de forma anexa en la ruta **wireframes/miro** en formato PDF.

1. Splash Screen / Carga

Funcionalidad: Mostrar logotipo e identidad visual de la app mientras se inicializan recursos locales.

Datos: Ninguno procesado o ingresado por el usuario.

Tipo de datos: No aplica.

Vigencia: No aplica.

Origen: Interno (imagen local de recursos de la app).

Operaciones: Inicialización visual.

Servicios externos: Ninguno.

Autenticación/API: No requerida.

Almacenamiento: No aplica.



Figura 2: Splash screen de GreenGrocery!

2. Inicio de Sesión

Funcionalidad: Permite al usuario autenticarse con email y contraseña.

Datos:

- Email (**string**, requerido, validado con expresión regular).
- Contraseña (**string**, encriptado).

Vigencia: Solo durante la sesión.

Origen: Ingresado manualmente por el usuario.

Operaciones: Autenticación (consulta contra backend).

Servicios externos:

- API REST de autenticación (POST /auth/login).
- Medio: HTTPS con JSON, respuesta con token JWT.

Almacenamiento:

- Local: token JWT (para mantener sesión activa).
- Remoto: datos del usuario en base de datos (Firestore o PostgreSQL).

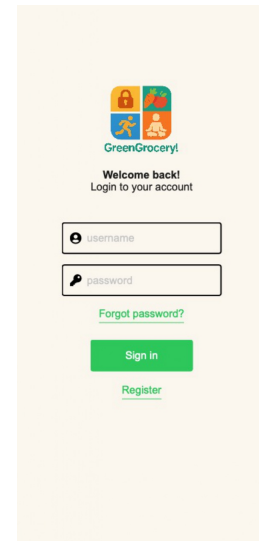


Figura 3: Splash screen de GreenGrocery!

3. Recuperar contraseña

Funcionalidad: Enviar enlace de recuperación al correo registrado.

Datos: Email (**string**).

Vigencia: Temporal (solo para el proceso de recuperación).

Origen: Usuario.

Operaciones: Envío de enlace por correo.

Servicios externos:

- API de recuperación (POST /auth/forgot-password).
- Servicio de correo (Firestore o SendGrid).
- Autenticación no requerida, solo validación del email.

Almacenamiento: No se almacena información localmente.

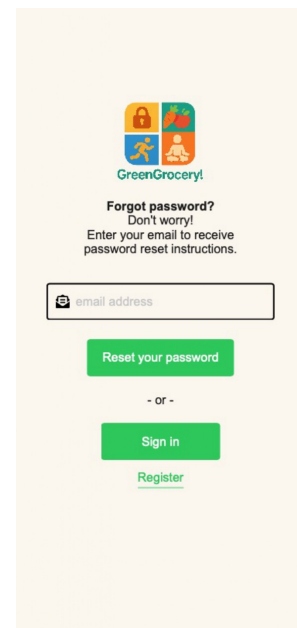


Figura 4: Splash screen de GreenGrocery!

4. Registro de Usuario

Funcionalidad: Permite crear una cuenta en la plataforma.

Datos:

- Email (**string**)
- Contraseña (**string**)
- Confirmación de contraseña

Vigencia: Permanente hasta que el usuario elimine su cuenta.

Origen: Formulario de usuario.

Operaciones: Registro (alta en base de datos).

Servicios externos:

- API de usuarios (POST /auth/register).
- Autenticación: JSON Web Token generado tras registro.

Almacenamiento:

- Remoto: datos en base de datos segura.
- Local: token y nombre del usuario.

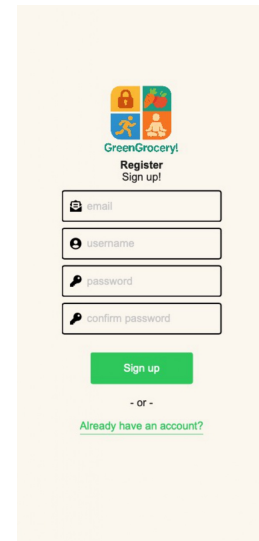


Figura 5: Splash screen de GreenGrocery!

5. Dashboard Principal

Funcionalidad: Vista general con resumen de uso digital, meditaciones y recomendaciones.

Datos:

- Tiempo de uso por app (**int**)
- Nombre de usuario (**string**)
- Recomendación diaria (**string**)
- Estado de hábitos (**boolean[]**)

Vigencia: Datos diarios (reseteo al siguiente día).

Origen: Screen Time API (iOS) + base de datos local para hábitos.

Operaciones: Consulta.

Servicios externos:

- API de iOS para uso de apps.
- Backend propio para recomendaciones.

Almacenamiento: Local, con sincronización opcional al backend.

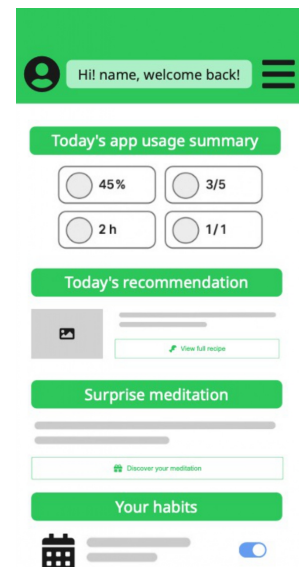


Figura 6: Splash screen de GreenGrocery!

6. Estadísticas de Tiempo Digital

Funcionalidad: Muestra detalle de apps usadas y su tiempo de uso.

Datos:

- Lista de apps (`string[]`)
- Tiempos individuales (`int`)

Vigencia: Diaria.

Origen: API Screen Time.

Operaciones: Consulta.

Servicios:

- Permisos requeridos de iOS.
- No hay costo si se usa API del sistema.

Almacenamiento: Local con agregados temporales. Exportable al backend si se desea guardar historial.

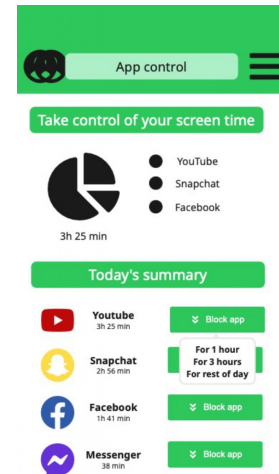


Figura 7: Splash screen de GreenGrocery!

7. Foro - Lista de publicaciones

Funcionalidad: Lista los posts creados por usuarios en la comunidad.

Datos:

- ID de post (`int`)
- Título (`string`)
- Fecha de creación (`datetime`)
- Autor (`string`)

Vigencia: Permanente.

Origen: Backend.

Operaciones: Consulta.

Servicios:

- API (`GET /posts`)
- Formato: JSON

Almacenamiento: No aplica en local.



Figura 8: Splash screen de GreenGrocery!

8. Foro - Detalle de publicación

Funcionalidad: Ver contenido completo de una publicación y sus comentarios.

Datos:

- Contenido (**string**)
- Imagen adjunta (**media**)
- Comentarios (**string[]**)

Vigencia: Permanente mientras no se elimine.

Origen: Backend.

Operaciones: Consulta.

Servicios:

- API (GET /posts/:id)
- Formato: JSON

Almacenamiento: No aplica en local.

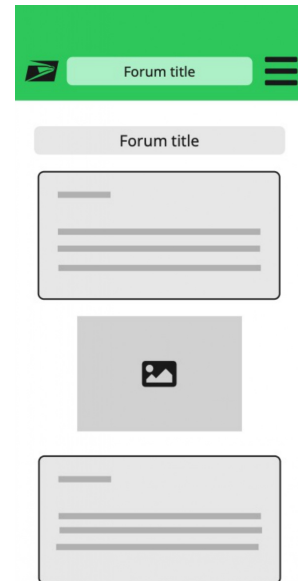


Figura 9: Splash screen de GreenGrocery!

9. Creación nuevo foro

Funcionalidad: Permitir crear una nueva publicación en el foro, agregando un título y distintos tipos de contenido como texto, imágenes, enlaces o casillas de verificación.

Datos: Título del foro, bloques de contenido.

Tipo de datos: Texto, imagen, URL, booleano (checkbox).

Vigencia: Hasta que el usuario elimine o edite su publicación.

Origen: Ingresado manualmente por el usuario desde el dispositivo.

Operaciones: Crear título, añadir secciones, editar o eliminar contenido.

Servicios externos:

- **Almacenamiento en la nube (ej. Firebase Storage):** Para guardar imágenes que se integren en la publicación del foro.

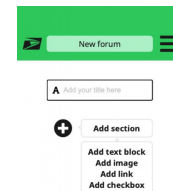


Figura 10: New Forum

10. Workout

Funcionalidad: Mostrar al usuario las categorías de ejercicio disponibles y permitir la navegación hacia rutinas o recursos relacionados.

Datos: Tipo de ejercicio seleccionado (ej. Running, Stretching, Strength).

Tipo de datos: Texto (etiquetas), selección (toques en botones).

Vigencia: Mientras esté activa la sesión o hasta que se reinicie la app.

Origen: Precargado en la aplicación.

Operaciones: Seleccionar categoría, acceder a contenidos relacionados.

Servicios externos:

- **YouTube:** Redireccionar al usuario a videos externos sobre ejercicios según la categoría elegida.

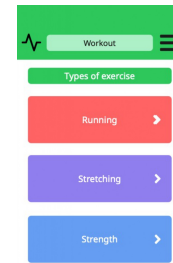


Figura 11: Workout

11. Running

Funcionalidad: Mostrar ejercicios relacionados con la categoría seleccionada (Running), permitiendo al usuario iniciar rutinas con control de tiempo.

Datos: Nombre del ejercicio, duración del temporizador, estado de inicio/pausa.

Tipo de datos: Texto, número (minutos), interacción (play/pausa).

Vigencia: Temporal, solo durante la sesión o la duración del ejercicio.

Origen: Precargado localmente o definido por el usuario si se habilita ajuste de tiempo.

Operaciones: Iniciar ejercicio, ajustar duración, detener o pausar.

Servicios externos:

- **Temporizador del sistema:** Utilizar el temporizador del dispositivo para controlar la sesión de ejercicio.

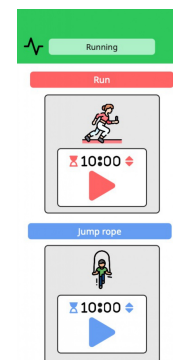


Figura 12: Running

12. Stretching

Funcionalidad: Mostrar una rutina de estiramiento con ejercicios guiados, control de tiempo y opciones para desplegar instrucciones adicionales de cada postura.

Datos: Nombre del ejercicio, duración del temporizador, estado del ejercicio (iniciado/pausado), instrucciones opcionales.

Tipo de datos: Texto, número, estado booleano (desplegado o no).

Vigencia: Temporal durante la sesión activa o rutina actual.

Origen: Precargado localmente en la app.

Operaciones: Iniciar rutina, ajustar duración, desplegar información adicional por ejercicio.

Servicios externos:

- **Temporizador del sistema:** Usado para contar el tiempo de cada estiramiento durante la rutina.

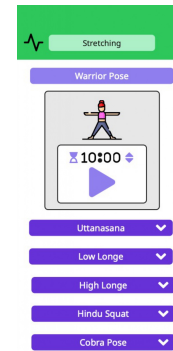


Figura 13: Stretching

13. Strength

Funcionalidad: Presentar ejercicios de fuerza con temporizador y opciones para desplegar instrucciones por cada actividad física.

Datos: Nombre del ejercicio, duración del tiempo, estado de ejecución (play/pausa), detalles de cada movimiento.

Tipo de datos: Texto, número, booleano (desplegado).

Vigencia: Solo mientras la rutina esté activa.

Origen: Contenido precargado en la app.

Operaciones: Iniciar ejercicio, modificar duración, consultar instrucciones específicas.

Servicios externos:

- **Temporizador del sistema:** Controla el tiempo de ejecución de cada ejercicio de fuerza seleccionado.

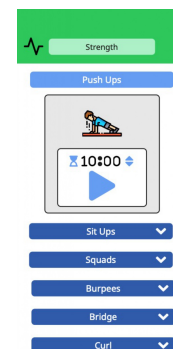


Figura 14: Strength

14. Meditation

Funcionalidad: Mostrar una meditación aleatoria acompañada de imagen, texto explicativo y temporizador para guiar la sesión.

Datos: Nombre de la meditación, descripción, duración, estado (iniciado/pausado).

Tipo de datos: Texto, imagen, número (minutos), booleano.

Vigencia: Durante la sesión activa o hasta que el usuario reinicie.

Origen: Selección aleatoria de una lista predefinida en la aplicación.

Operaciones: Reproducir meditación, visualizar imagen y descripción, controlar duración.

Servicios externos:

- **Temporizador del sistema:** Utilizado para guiar la duración de la sesión de meditación.

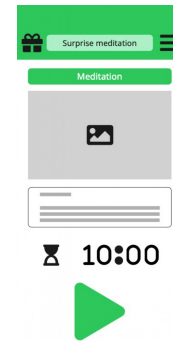


Figura 15: Meditation

15. List meditation

Funcionalidad: Mostrar una lista de meditaciones disponibles con nombre, imagen ilustrativa y descripción breve.

Datos: Nombre de la meditación, imagen asociada, texto explicativo.

Tipo de datos: Texto, imagen.

Vigencia: Permanente mientras esté disponible en el catálogo de la app.

Origen: Precargado desde la base de datos local o configurable desde el backend.

Operaciones: Visualizar lista, seleccionar una meditación para iniciar.

Servicios externos:

- **Repositorio de contenido (opcional):** Para almacenar las imágenes e información descriptiva de las meditaciones si se integra un backend futuro.

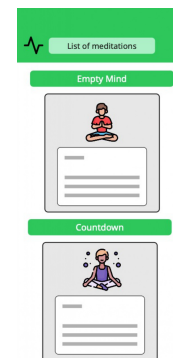


Figura 16: List meditation

Simulación de aplicación en Xcode

Para una mejor visualización de los wireframes, estos fueron añadidos en la carpeta anexa **wireframes/xcode** en formato PNG, con su nombre correspondiente al de este apartado.

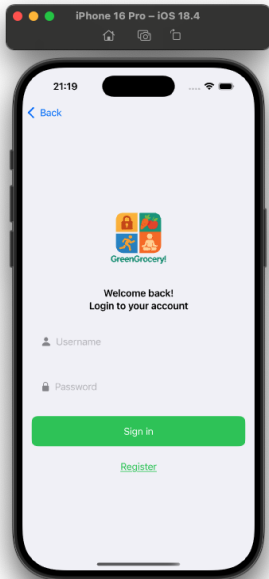


Figura 17: Login

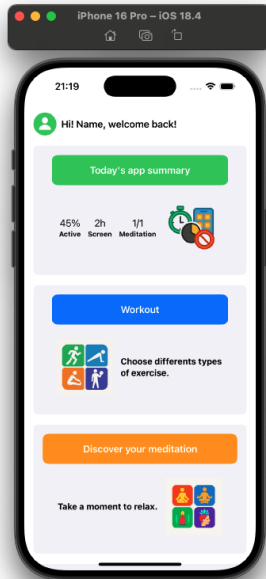


Figura 18: Dashboard

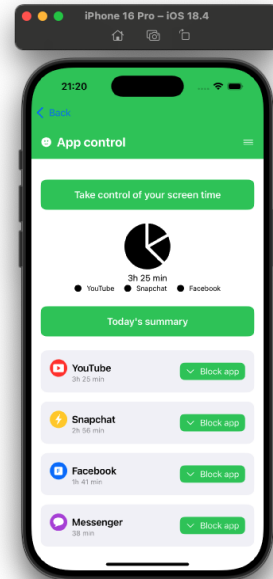


Figura 19: App Control

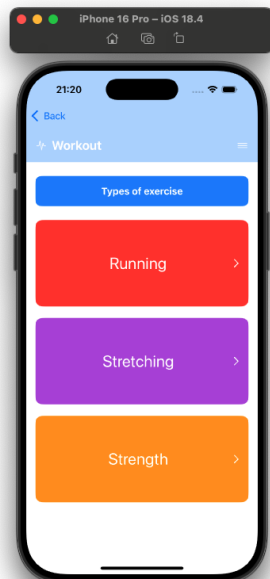


Figura 20: Workout

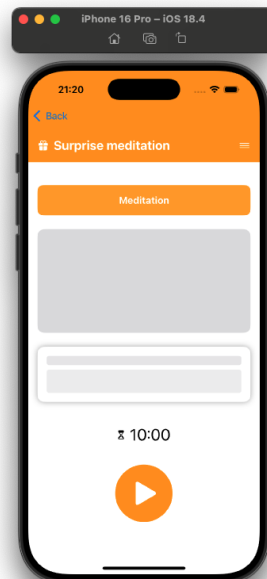


Figura 21: Meditation

Conclusión

El desarrollo de la aplicación **GreenGrocery!** ha representado una experiencia formativa, técnica y socialmente significativa. A lo largo del proyecto, se abordaron de manera integral problemáticas actuales como el uso excesivo de redes sociales, la falta de hábitos saludables y la desconexión con el bienestar físico y mental en un mundo altamente digitalizado.

Desde su concepción, GreenGrocery! fue pensada como una solución innovadora y accesible que integrara recomendaciones alimenticias, rutinas de ejercicio, sesiones de meditación y control de tiempo en pantalla, todo en una sola plataforma. La aplicación no solo responde a una necesidad real en la sociedad, sino que también pone en práctica principios de diseño centrado en el usuario, accesibilidad y desarrollo ágil.

Durante el proceso de trabajo se aplicaron herramientas modernas como **Swift**, **SwiftUI**, Xcode y metodologías ágiles, lo cual permitió construir un producto funcional, visualmente consistente y con potencial de escalabilidad. La estructura del equipo, organizada por roles específicos, permitió una distribución efectiva de tareas y fomentó la colaboración interdisciplinaria.

Finalmente, este proyecto no solo cumplió con los objetivos propuestos, sino que también reforzó la importancia de crear tecnología con propósito. GreenGrocery! es una prueba de cómo una app bien diseñada puede convertirse en un aliado en la vida cotidiana de las personas, ayudándolas a recuperar el equilibrio entre salud, mente y entorno digital.

Referencias

- [1] A Alba. *Descubriendo Swift: Características, sintaxis y usos principales*. 2023. URL: <https://profile.es/blog/swift/>.
- [2] Requeridos Blogs. *Requerimientos Funcionales y No Funcionales, ejemplos y tips*. 2018. URL: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>.
- [3] PubMed Central. *Effects of a Mindfulness Meditation App on Subjective Well-Being: Active Randomized Controlled Trial an Experience Sampling Study*. 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6329416/>.
- [4] PubMed Central. *Exercise or lie down? The impact of fitness app use on users' well-being*. 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10806118/>.
- [5] Apple Inc. *Program enrollment*. 2025. URL: <https://developer.apple.com/help/account/membership/program-enrollment/>.
- [6] A Manchanda. *What Is an MVP App? A Guide to Building Successful Products*. 2024. URL: <https://www.netsolutions.com/hub/minimum-viable-product/app/>.
- [7] OPSWAT. *13 Application Security Best Practice*. 2023. URL: <https://www.opswat.com/blog/application-security-best-practices>.
- [8] RankMyApp. *Qué es aplicación de bloqueo y cómo afecta a la aplicación*. 2021. URL: <https://rankmyapp.com/es/que-es-aplicacion-de-bloqueo-y-como-afecta-a-la-aplicacion/>.
- [9] Annual Reviews. *A Review of the Quality and Impact of Mobile Health Apps*. 2022. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev-publhealth-052020-103738>.
- [10] R Schmidt. *Marco ágil para el desarrollo de aplicaciones móviles*. 2023. URL: <https://appmaster.io/es/blog/marco-agil-para-el-desarrollo-de-aplicaciones-moviles>.