Universidad Nacional Autónoma De México
Facultad De Ingeniería
Cómputo Móvil
Semestre 2026-1
Proyecto Final

Equipo “Code4Change”

Integrantes:

- Argueta Bravo Angel Jacob
- Gaytán Herrera Bélen
- Solís Espinosa Andrea Vianney
- Soria Aguilar Angel

Profesor: Ing.Marduk Pérez de Lara Domínguez

Grupo 03

Fecha de entrega: 28.11.2025

Proyecto Final

Índice

Proyecto Final

Introducción

La tecnología móvil se ha convertido en una herramienta fundamental para acompañar actividades esenciales de la vida diaria, entre ellas la alimentación. En un entorno donde el ritmo de vida es acelerado y la información nutricional suele ser extensa o confusa, muchas personas buscan soluciones prácticas que les ayuden a tomar decisiones saludables sin complicar su rutina. Bajo esta idea surge Meal Plan, una aplicación diseñada para ofrecer recomendaciones simples, rápidas y personalizadas que apoyen a los usuarios en la elección de snacks y comidas ligeras.

El objetivo principal de Meal Plan es facilitar la planificación alimenticia mediante una experiencia intuitiva que permita filtrar alimentos según preferencias, objetivos físicos o restricciones particulares. La aplicación integra elementos visuales claros, datos relevantes y un flujo de navegación accesible, buscando que cualquier persona pueda encontrar opciones saludables en cuestión de segundos, sin necesidad de conocimientos avanzados en nutrición.

Durante el desarrollo del proyecto se analizaron distintos escenarios de uso, así como los factores que influyen en los hábitos alimenticios cotidianos. Esta reflexión permitió establecer una propuesta equilibrada entre funcionalidad, diseño y facilidad de uso. El resultado es una herramienta adaptable, capaz de funcionar incluso sin conexión a internet, y preparada para evolucionar a futuro mediante la integración de nuevas tecnologías.

A lo largo de este documento se presenta la estructura técnica y conceptual que da forma a Meal Plan. Se detallan los requerimientos de la aplicación, la justificación de su viabilidad técnica, el alcance del producto, el diseño del MVP, los wireframes, la documentación de cada pantalla, el flujo de navegación, el análisis de datos, los dispositivos compatibles, así como los aspectos relacionados con sensores, permisos y publicación. También se incluye la descripción del equipo de trabajo, las tecnologías utilizadas y una estimación general de tiempos y costos.

Meal Plan refleja la intención de aprovechar la tecnología como un aliado práctico y accesible para mejorar la relación cotidiana con los alimentos, ofreciendo una experiencia personalizada que busca fomentar hábitos saludables sin sacrificar la simplicidad ni la comodidad.

Requerimientos de la aplicación

El diseño de Meal Plan requirió establecer una base sólida que guiara su funcionamiento y garantizara que cada recomendación fuera coherente con el perfil del usuario. Para ello, fue necesario definir un conjunto de reglas, acciones y condiciones que determinaran el comportamiento de la aplicación, tanto a nivel de lógica interna como de experiencia de uso. Este apartado presenta de manera detallada las reglas de

negocio, los requerimientos funcionales y los requerimientos no funcionales que estructuran el proyecto y marcan el alcance técnico de la herramienta.

Reglas de negocio

Las reglas de negocio representan los lineamientos que determinan cómo debe operar la aplicación y de qué manera se procesará la información del usuario para generar recomendaciones confiables y personalizadas.

- Para generar sugerencias, el usuario debe registrar información básica como edad, estatura, peso, sexo y objetivo físico.
- La aplicación debe excluir automáticamente cualquier alimento marcado como alergia o restricción alimentaria.
- Cada receta debe incluir información completa: lista de ingredientes, pasos de preparación, tiempo estimado y valor nutricional aproximado.
- Las recomendaciones deben adaptarse al objetivo físico del usuario (bajar de peso, mantenerlo o aumentar masa muscular).
- Las recetas marcadas como favoritas deben conservarse incluso sin conexión a internet.
- Las preferencias alimentarias tienen prioridad sobre el catálogo general y modifican de manera inmediata las recomendaciones.
- La aplicación debe permitir su funcionamiento básico en modo sin conexión, utilizando únicamente la información almacenada localmente.
- Cualquier modificación en el perfil del usuario o en sus preferencias debe reflejarse de forma automática en el catálogo.
- Meal Plan no generará dietas médicas ni menús clínicos; su función es orientar mediante opciones prácticas y saludables.
- El uso de datos biométricos, como los que proporciona HealthKit, solo podrá realizarse con autorización explícita del usuario y con fines de personalización.

Requerimientos funcionales

Los requerimientos funcionales describen las acciones que la aplicación debe permitir al usuario y las funcionalidades principales que forman parte de su operación.

Registro del usuario

- Permite ingresar edad, sexo, peso, estatura y objetivo físico.
- Guarda la información de forma local para personalizar futuras recomendaciones.

Preferencias y restricciones alimentarias

- Permite seleccionar alimentos a evitar, ya sea por alergias, gustos o restricciones personales.
- Filtra automáticamente el catálogo para excluir opciones incompatibles.

Proyecto Final

Configuración calórica

- Calcula un rango calórico recomendado en función del perfil del usuario.
- Permite realizar ajustes manuales.

Catálogo de snacks personalizados

- Muestra una lista dinámica de opciones saludables según el perfil y las preferencias.
- Actualiza el catálogo en tiempo real cuando el usuario cambia su información.

Vista detallada de recetas

- Presenta ingredientes, pasos, tiempo estimado de preparación y valor nutricional aproximado.
- Muestra una pequeña nota sobre los beneficios del alimento.

Gestión de favoritos

- Permite guardar y eliminar recetas preferidas.
- Almacena esta información localmente para asegurar acceso sin conexión.

Modo sin conexión

- Permite visualizar el perfil, los favoritos y el catálogo precargado sin internet.
- Navegación intuitiva
- Permite desplazarse entre pantallas mediante toques, deslizamientos y botones claros.

Integración opcional con datos biométricos

- Si el usuario lo permite, utiliza información de actividad física para ajustar recomendaciones calóricas.

Requerimientos no funcionales

- Los requerimientos no funcionales definen la calidad, comportamiento y restricciones técnicas de la aplicación, más allá de las funciones específicas que implementa.

Usabilidad

- La interfaz debe ser limpia, clara y fácil de entender.
- Los elementos visuales deben transmitir simplicidad, bienestar y organización.

Rendimiento

- La aplicación debe ofrecer tiempos de respuesta rápidos al generar catálogos o cargar pantallas.
- Debe funcionar adecuadamente en dispositivos de gama media.

Seguridad

- Los datos del usuario se almacenan únicamente de forma local.
- El acceso a datos biométricos requiere autorización explícita.

Compatibilidad

- Optimizada para dispositivos iOS en orientación vertical.
- Debe ajustarse correctamente según el tamaño de pantalla del dispositivo.

Confiabilidad

- La información almacenada (perfil, favoritos y preferencias) debe persistir aunque la aplicación se cierre.

Mantenibilidad

- El proyecto debe contar con una estructura clara que facilite futuras actualizaciones, como ampliar el catálogo o integrar nuevas funciones.

Escalabilidad

- La arquitectura debe permitir incorporar tecnologías futuras, como inteligencia artificial generativa o la expansión a otros sistemas operativos.

Viabilidad técnica

Durante el diseño de Meal Plan surgió la posibilidad de incorporar una funcionalidad avanzada basada en inteligencia artificial generativa, cuyo objetivo sería crear recetas nuevas y completamente personalizadas a partir del perfil, gustos y hábitos del usuario. Esta característica tendría la capacidad de combinar ingredientes disponibles, adaptar porciones y sugerir preparaciones originales que no formen parte del catálogo predefinido. Sin embargo, al evaluar su inclusión en la versión actual del proyecto, fue necesario analizar su viabilidad técnica y las implicaciones que conlleva su implementación.

Desde un punto de vista conceptual, la integración de un modelo generativo es posible, ya que existen APIs y servicios que permiten procesar texto y generar contenido de manera dinámica. Este tipo de tecnología podría enriquecer la experiencia del usuario, ofreciendo recomendaciones únicas y adaptadas a cada caso específico. Sin embargo, su incorporación en Meal Plan presenta varios retos técnicos que deben considerarse con cuidado.

El primer desafío está relacionado con el costo y la infraestructura necesaria. Para que la generación de recetas sea precisa, segura y relevante, es indispensable contar con un modelo entrenado en datos nutricionales confiables y con un servidor capaz de procesar solicitudes en tiempo real. Esto implica costos de mantenimiento,

procesamiento y almacenamiento que superan el alcance de una aplicación ligera pensada para operar también sin conexión.

Además, la validación nutricional es un aspecto crítico. Aunque un modelo generativo puede crear textos de recetas de forma eficiente, no garantiza que los valores calóricos, porciones o combinaciones propuestas sean adecuadas o seguras para el usuario. Para asegurar la calidad de la información generada sería necesario desarrollar un sistema adicional de validación o filtros que incrementan la complejidad técnica del proyecto.

Por otro lado, la integración de esta funcionalidad requiere conexión permanente a internet, lo cual contrasta con uno de los principios centrales de Meal Plan: funcionar incluso en modo sin conexión mediante almacenamiento local. Depender de un servicio externo rompe parcialmente esa experiencia y limita el uso de la aplicación en contextos donde la conectividad es inestable.

Tomando en cuenta estos factores, la implementación de inteligencia artificial generativa se considera parcialmente viable. Es una función que podría integrarse en una etapa futura del proyecto, siempre y cuando se cuente con la infraestructura, recursos económicos y mecanismos de validación necesarios para garantizar la calidad y seguridad de las recomendaciones. Por ahora, su desarrollo queda fuera del alcance inmediato de Meal Plan, pero se mantiene como una oportunidad de crecimiento para una versión avanzada de la aplicación.

Alcance y MVP de la aplicación

El desarrollo de Meal Plan se planteó con un enfoque realista que permitiera construir una herramienta funcional, práctica y accesible para los usuarios, evitando complicar la primera versión del proyecto. Con esto en mente, fue necesario definir con claridad el alcance de la aplicación y determinar cuáles serían las funciones esenciales que debían incluirse en el producto mínimo viable (MVP) para garantizar una experiencia coherente desde el inicio.

Alcance de la aplicación

El alcance de Meal Plan contempla todas las funciones que forman parte de la versión base de la app. Este conjunto de características permite que el usuario registre sus datos, obtenga recomendaciones personalizadas y navegue de forma sencilla por las diferentes opciones alimenticias. La aplicación está pensada para operar tanto en línea como fuera de ella y para ofrecer una interfaz clara e intuitiva.

El alcance actual de la aplicación incluye:

- Registro de datos básicos del usuario (edad, estatura, peso, sexo y objetivo físico).

- Selección de preferencias y restricciones alimentarias.
- Configuración calórica con cálculo automático y opción de ajuste manual.
- Generación de un catálogo de snacks y comidas ligeras basadas en el perfil del usuario.
- Vista detallada de cada receta con ingredientes, pasos, tiempo estimado y valor nutricional aproximado.
- Sistema de favoritos completamente funcional y almacenado de forma local.
- Navegación fluida y diseño visual basado en claridad y simplicidad.
- Operación en modo sin conexión mediante almacenamiento local con Core Data.
- Arquitectura lista para futuras ampliaciones, como el uso de sensores, IA o integración con servicios externos.

Este alcance permite que Meal Plan sea una herramienta estable y útil desde la primera versión, sin depender de funciones complejas o costosas para cumplir su propósito principal: simplificar la elección de alimentos saludables.

Producto Mínimo Viable (MVP)

El MVP representa la versión más simple que puede lanzarse al público manteniendo la funcionalidad esencial del proyecto. Su objetivo es asegurar que la aplicación sea capaz de resolver el problema principal para el que fue diseñada, evitando incluir características que requieran tiempos de desarrollo extensos o costos elevados.

Para Meal Plan, el MVP está compuesto por los siguientes elementos:

Funciones esenciales del MVP

- Registro del usuario: permite ingresar datos básicos necesarios para personalizar las recomendaciones.
- Preferencias alimentarias: selección de gustos, restricciones y alergias.
- Catálogo de snacks precargado: base de recetas simples y saludables.
- Generación de recomendaciones personalizadas: usando datos del usuario y sus preferencias.
- Vista detallada de recetas: con información nutricional, pasos y tiempo de preparación.
- Favoritos: sistema básico para guardar y eliminar recetas preferidas.
- Modo sin conexión: acceso al catálogo, favoritos y perfil sin necesidad de internet.
- Diseño visual básico, responsivo y fácil de navegar.

Objetivo del MVP

El MVP busca validar que Meal Plan realmente facilita la elección de snacks saludables y que la experiencia resulta clara, rápida y accesible para el usuario. Esta versión permite comprobar si la lógica de recomendación funciona correctamente y si las

personas encuentran valor en la aplicación sin depender de tecnologías adicionales como IA, sensores o servicios en la nube.

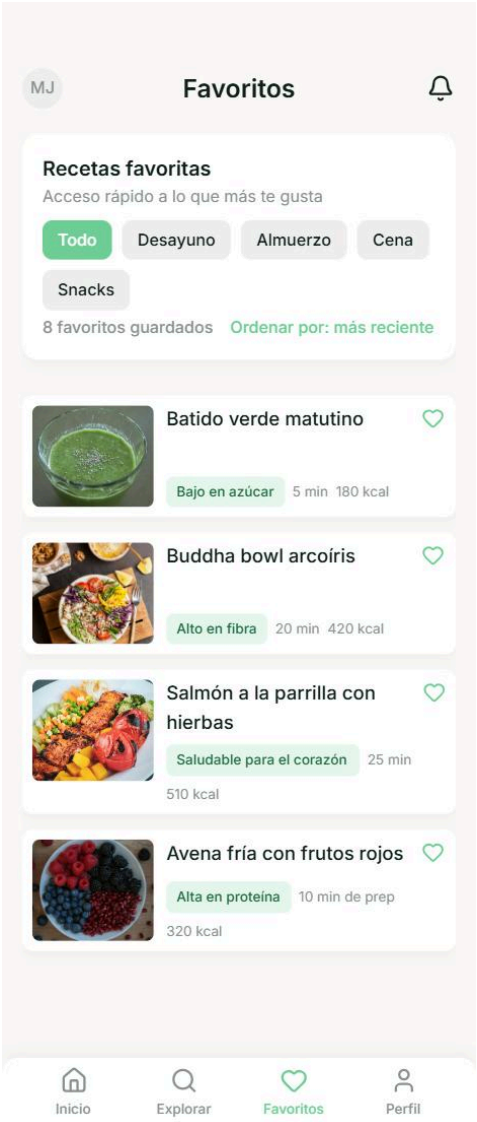
Wireframes de la aplicación

Pantalla 1. Compartir recetas con otros usuarios

Proyecto Final

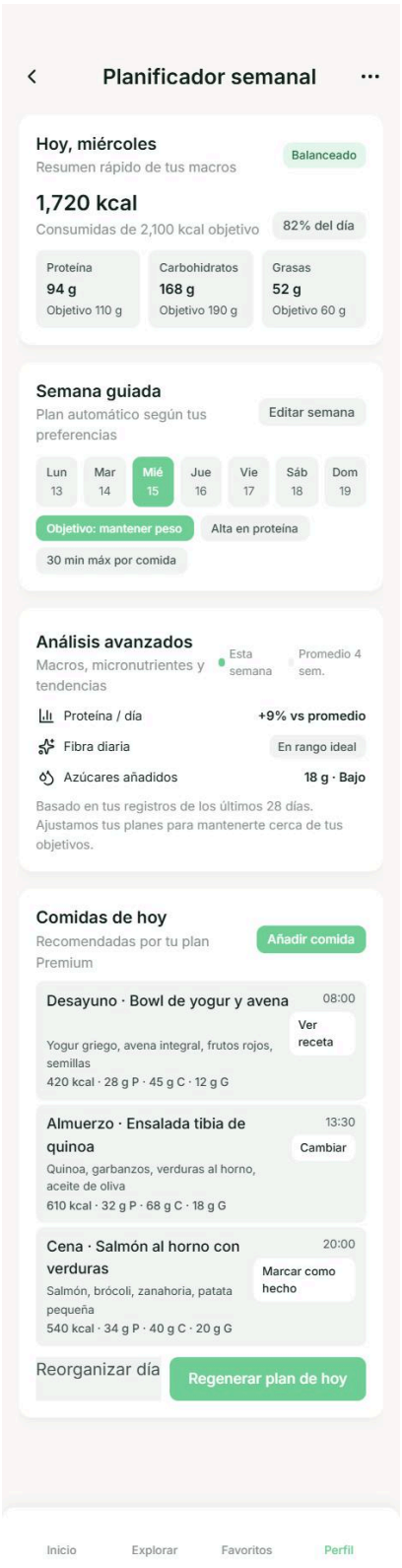
Pantalla 2. Explorar nuevas recetas en base a gustos del usuario

Proyecto Final



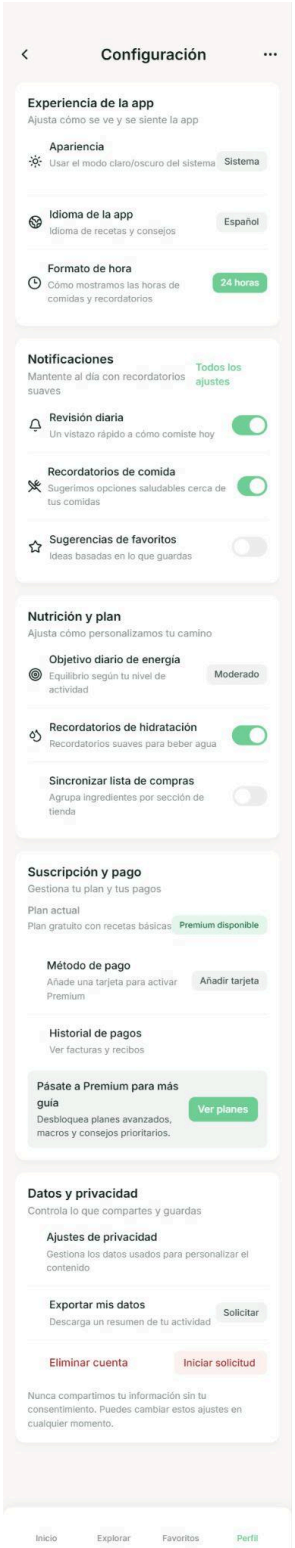
Pantalla 3. Recetas favoritas del usuario

Proyecto Final

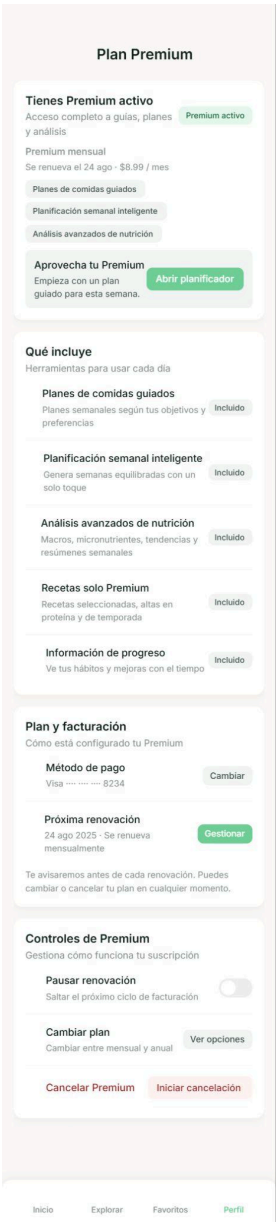


Pantalla 4. Planificación de dieta semanal

Proyecto Final



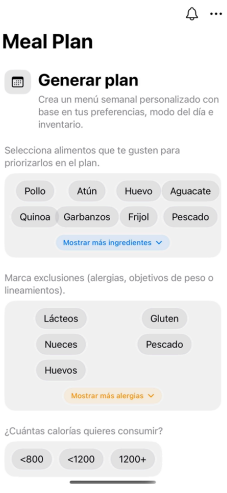
Pantalla 5. Configuración de la aplicación



Pantalla 6. Subscripción mensual a la app

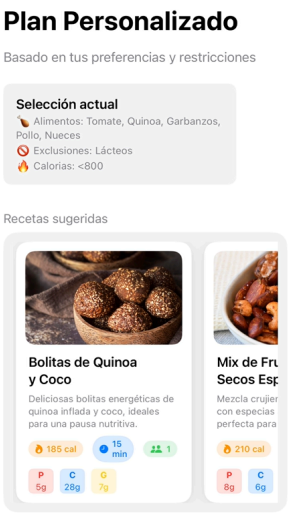
Proyecto Final

Pantalla 7. Configuración del perfil del usuario



Pantalla 8. Generación de recetas

Proyecto Final



Pantalla 9. Recetas sugeridas

Pantalla 10. Detalles de las recetas

Documentación por pantalla

Pantalla 1. Compartir receta

Esta pantalla permite al usuario difundir una receta específica con otras personas a través de diferentes canales

- Funcionalidad: Previsualización y configuración de metadatos de una receta para ser enviada mediante UIActivityViewController o enlaces profundos.
- Información: Título de la receta, imagen, calorías, etiquetas dietéticas, opciones de envío (enlace, mensaje, redes) y toggles de privacidad.
- Tipo de datos: String (título, enlace), Image (binario), Boolean (interruptores de "Mostrar nombre", "Info nutricional").

- Vigencia del dato: Transitoria. Los estados de los interruptores solo viven durante la sesión de compartir; no se guardan en base de datos.
- Origen: Mixto. Los datos de la receta provienen de la Base de Datos Local (Core Data); la configuración de envío es input del Usuario en tiempo real.
- Operaciones:
    - Read: Lectura de los detalles de la receta seleccionada.
    - Process: Generación de cadena de texto o deeplink con los parámetros seleccionados.

Pantalla 2. Explorar

Interfaz de búsqueda y descubrimiento sobre el catálogo general

- Funcionalidad: Filtrado dinámico de recetas y presentación de categorías agrupadas.
- Información: Barras de búsqueda, filtros rápidos (tags), recetas destacadas ("Destacado para ti"), categorías y tendencias.
- Tipo de datos: String (query de búsqueda), Array<Recipe> (listas de resultados), Enum (Categorías: Desayuno, Batch cooking).
- Vigencia del dato: Persistente (Catálogo) / Transitoria (Búsqueda). El catálogo es estático/actualizable; los resultados de búsqueda son efímeros.
- Origen: Base de Datos Local. El catálogo está precargado (Json/Core Data) para funcionar offline.
- Operaciones:
    - Read: Consulta masiva de recetas con predicados de filtro (ej. "Alta en proteína").
    - Filter: Algoritmo de coincidencia de texto para la barra de búsqueda

Pantalla 3. Favoritos

Gestión de la colección personal de recetas guardadas por el usuario.

- Funcionalidad: Listado y filtrado de recetas marcadas previamente como favoritas para acceso rápido
- Información: Lista de tarjetas de recetas, filtros por tiempo de comida (Todo, Desayuno, Cena).
- Tipo de datos: Array <Recipe> (Lista filtrada), Boolean (Estado *isFavorite*)
- Vigencia del dato: Persistente. La relación de favorito se almacena permanentemente hasta que el usuario decida eliminarla.
- Origen: Usuario. Es una sub-selección de la base de datos definida por la interacción previa del usuario.
    - Operaciones:
    - Read: Consulta de recetas donde isFavorite == true.

    - Update: Al pulsar el corazón, se cambia el estado a false y se remueve de la vista.

Pantalla 4. Planificador semanal

Dashboard de seguimiento nutricional y calendario.

- Funcionalidad: Cálculo de ingesta calórica diaria, visualización de macros y gestión de comidas por horario.
- Información: Calorías consumidas vs. objetivo, gramos de Proteína/Carbos/Grasa, calendario semanal, lista de comidas del día (Desayuno, Almuerzo, Cena).
- Tipo de datos: Integer (Calorías), Float (Macros), Date (Fechas calendario), Struct (Plan de comida).
- Vigencia del dato: Persistente. El historial de consumo y la planificación futura se guardan en la base de datos.
- Origen: Calculado / Base de Datos. Los macros totales son la suma de las recetas registradas en el día; los objetivos vienen del Perfil.
- Operaciones:
    - Read: Obtener comidas asociadas a la fecha seleccionada.
    - Calculate: Sumatoria en tiempo real de calorías y macros de las comidas registradas.
    - Update: Marcar comida como "Hecho".

Pantalla 5. Configuración

Panel de administración de preferencias de la aplicación.

- Funcionalidad: Modificación de variables globales de la app y gestión de cuenta.
- Información: Ajustes de apariencia, idioma, notificaciones (push), estado de suscripción y controles de privacidad.
- Tipo de datos: Boolean (Toggles de notificaciones), Enum (Idioma, Tema), String (ID de usuario).
- Vigencia del dato: Persistente. Se almacenan en UserDefaults o entidad de Configuración en Core Data.
- Origen: Usuario.
- Operaciones:
    - Update: Guardar cambios inmediatamente.
    - Delete: "Eliminar cuenta" borra todos los registros locales asociados al usuario.

Pantalla 6. Suscripción (Plan Premium)

Gestión del estado de facturación y acceso a funciones bloqueadas.

- Funcionalidad: Visualización del nivel de servicio y gestión de pasarela de pago (simulada o conectada a StoreKit).
- Información: Estado (Activo), fecha de renovación, método de pago enmascarado, lista de beneficios.
- Tipo de datos: Boolean (isPremium), Date (Fecha vencimiento), String (Masked Credit Card).
- Vigencia del dato: Persistente.
- Origen: Sistema / Externo. Datos validados contra el recibo de compra (Apple Store) o bandera local.
- Operaciones:
    - Read: Verificar estado de suscripción al inicio de la app.
    - Update: Cambiar plan o cancelar suscripción.

Pantalla 7. Perfil

Visualización de datos biométricos y preferencias dietéticas.

- Funcionalidad: Resumen del usuario ("User Persona") y CRUD de restricciones alimentarias.
- Información: Nombre, racha de días, etiquetas de preferencias (ej. "Basada en plantas"), alergias seleccionadas.
- Tipo de datos: String (Nombre), Integer (Racha, contadores), Array<String> (Tags de alergias/gustos).
- Vigencia del dato: Persistente. Datos almacenados en Core Data (Entidad User).
- Origen: Usuario. Ingresados durante el onboarding o edición.
- Operaciones:
    - Read: Cargar perfil del usuario.
    - Update: Modificar alergias o cambiar foto de perfil.

Pantalla 8. Generación de recetas

Formulario de entrada para el motor de recomendación.

- Funcionalidad: Recolección de parámetros para filtrar el catálogo y generar un plan a medida.
- Información: Selección múltiple de ingredientes (ej. Pollo, Atún), exclusiones (ej. Gluten), rango calórico.
- Tipo de datos: Set<Ingredient> (Inclusiones), Set<Allergen> (Exclusiones), Enum/Int (Rango calórico).
- Vigencia del dato: Transitoria. Los filtros se reinician o mantienen solo en memoria durante la sesión de generación.
- Origen: Usuario.
- Operaciones:
    - Input: Selección y validación de criterios incompatibles.

    - Process: Envío de parámetros al algoritmo de filtrado al presionar "Generar".

Pantalla 9. Recetas sugeridas

Presentación de resultados del algoritmo de generación.

- Funcionalidad: Visualización de recetas que cumplen estrictamente con los criterios de la Pantalla 8.
- Información: Resumen de filtros aplicados y lista de recetas resultantes con sus macros resumidos.
- Tipo de datos: Array<Recipe> (Resultados).
- Vigencia del dato: Transitoria. Los resultados se muestran al momento; si se sale de la pantalla, se requiere regenerar.
- Origen: Sistema (Algoritmo). Procesamiento local sobre la base de datos de recetas.
- Operaciones:
    - Read: Mostrar detalles preliminares en tarjetas.

Pantalla 10. Detalles de las recetas

- Vista de lectura detallada (Solo lectura).
- Funcionalidad: Guía paso a paso e información nutricional profunda de un ítem específico.
- Información: Título, tiempo, porciones, desglose de macros (gr), lista de ingredientes con gramaje.
- Tipo de datos: Struct/Object (Modelo completo de Receta).
- Vigencia del dato: Persistente (Datos de la receta) / Solo Lectura (Vista).
- Origen: Base de Datos Local.
- Operaciones:
    - Read: Deserialización completa del objeto receta.
    - Action: Disparadores para "Añadir a favoritos" o "Compartir" (que llevan a otras pantallas).

Flujo de navegación de la app

La demo de Meal Plan representa la primera versión visual y funcional del proyecto, basada en los wireframes previamente diseñados y en el alcance definido para el MVP. Esta maqueta final permite observar cómo se integran las pantallas, la navegación y los componentes principales de la aplicación, así como la manera en que el usuario interactúa con el sistema.

La maqueta incluye:

Pantalla de inicio y registro del usuario

Se presenta un formulario claro para capturar datos como edad, peso, estatura, sexo y objetivo físico. La interfaz utiliza elementos visuales simples que facilitan la lectura y aceleran el proceso de registro.

Pantalla de preferencias y restricciones alimentarias

La demo muestra una lista organizada de alimentos comunes que pueden activarse o desactivarse según alergias, gustos o restricciones. Los elementos se muestran como interruptores o tarjetas seleccionables para mantener la experiencia intuitiva.

Pantalla principal del catálogo de snacks

La maqueta incluye un catálogo precargado donde cada snack aparece como una tarjeta con imagen, nombre y breve descripción. Estas tarjetas permiten desplazamiento vertical y se adaptan al perfil configurado por el usuario.

Vista detallada de receta

Cada tarjeta del catálogo lleva a una pantalla detallada con información completa: ingredientes, pasos de preparación, tiempo aproximado y valor nutricional estimado. También se muestra un botón para agregar o quitar la receta de favoritos.

Sección de favoritos

La demo incorpora una sección accesible desde el menú inferior donde se pueden visualizar todas las recetas guardadas. La interfaz está optimizada para funcionar incluso en modo sin conexión.

Menú de navegación inferior

La maqueta incluye una barra inferior con accesos directos a las secciones principales: Catálogo, Favoritos y Perfil, permitiendo un flujo claro y consistente.

Pantalla de perfil del usuario

En esta vista se puede revisar y modificar la información personal y las preferencias alimentarias. Cualquier cambio se refleja automáticamente en el catálogo.

Compatibilidad: dispositivos, tamaños y orientaciones

Meal Plan fue diseñada para adaptarse de forma natural al ecosistema de dispositivos móviles actuales, priorizando la claridad visual, la legibilidad y la experiencia del usuario. La aplicación está optimizada para funcionar en teléfonos inteligentes debido a que representan el dispositivo más utilizado en situaciones cotidianas, como el trabajo, la escuela o el hogar, donde las recomendaciones rápidas son más necesarias.

La compatibilidad actual considera los siguientes aspectos:

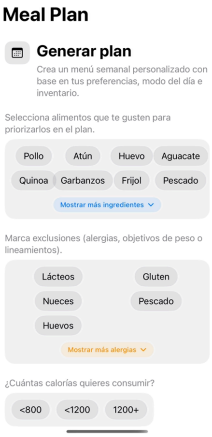
Proyecto Final

- **Dispositivos**: Meal Plan está pensada para funcionar en iPhone, desde modelos recientes hasta generaciones previas compatibles con iOS moderno.
- **Sistema operativo**: iOS 15 o superior, garantizando estabilidad, seguridad y compatibilidad con SwiftUI.
- **Tamaños de pantalla**: La interfaz se adapta a diferentes resoluciones, permitiendo una experiencia óptima tanto en pantallas compactas como en pantallas más amplias de generaciones recientes.
- **Orientación**: La orientación principal es vertical, ya que facilita una navegación lineal y natural para el usuario. Aunque la aplicación puede ajustarse visualmente al modo horizontal, este no forma parte del flujo principal de interacción.
- **Tabletas**: Meal Plan puede visualizarse en iPad, aunque por el tipo de contenido no se considera una prioridad dentro de esta versión del proyecto.

Con estas especificaciones, la aplicación garantiza estabilidad y un diseño consistente en todos los dispositivos contemplados.

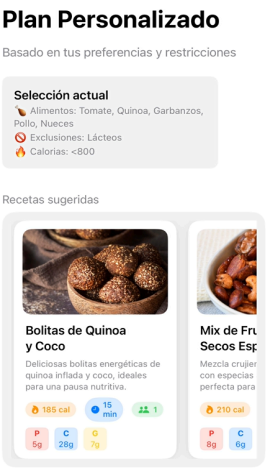
Demo o maqueta final

Al abrir la aplicación el usuario tendrá esta pantalla,deberá seleccionar los datos que se indican para comenzar a generar un plan



Una vez seleccionados se genera su plan personalizado en el que se muestran con mayor detalle las recetas dandole una amplia variedad de alternativas en base a las selecciones de la pantalla anterior

Proyecto Final



Al seleccionar una receta en específico se mostrará información con mayor detalle nutricional de la comida elegida

Lenguajes, herramientas y tecnologías utilizadas

El desarrollo de Meal Plan se llevó a cabo utilizando un conjunto de herramientas modernas que facilitan la creación de interfaces limpias, funcionales y altamente eficientes. La selección tecnológica se basó en la compatibilidad con el entorno iOS y en la necesidad de mantener un código escalable y fácil de mantener.

Las tecnologías principales incluyen:

- **Lenguaje de programación: Swift 5**
  Un lenguaje moderno, seguro y orientado a estructuras claras, ideal para aplicaciones móviles en iOS.
- **Framework de interfaz: SwiftUI**
  Tecnología declarativa que permite diseñar interfaces de manera intuitiva y mantener una correspondencia directa entre la vista y el estado de la aplicación.
- **Base de datos local: Core Data**

Utilizada para almacenar el perfil del usuario, preferencias alimentarias, restricciones y recetas favoritas sin depender de internet.

- **Entorno de desarrollo: Xcode 15**
  Herramienta oficial de Apple para la construcción, simulación y depuración de aplicaciones móviles.
- **Control de versiones: GitHub**
  Plataforma utilizada para mantener un registro de cambios, respaldos y trabajo colaborativo.
- **Herramientas de apoyo:**
  - Figma o Canva para el diseño de wireframes y maquetas.
  - ChatGPT y Copilot como apoyo para documentación y generación de ideas.

La combinación de estas tecnologías permitió crear una aplicación moderna, ligera y preparada para escalar en versiones futuras.

Permisos necesarios para publicación

Aunque Meal Plan es una aplicación que opera principalmente con datos ingresados por el usuario y no utiliza sensores ni servicios avanzados del dispositivo, existen ciertos lineamientos que deben cumplirse para su publicación en la App Store.

Los permisos y adecuaciones necesarios incluyen:

- **Acceso al almacenamiento local**:Utilizado para guardar preferencias, recetas y datos básicos del usuario. Este permiso está integrado de manera nativa en iOS y no requiere solicitudes especiales al usuario.
- **Política de privacidad**:La aplicación debe incluir una política clara que explique que los datos permanecen en el dispositivo y no se comparten con terceros.
- **Cumplimiento de App Store Review Guidelines**: Es necesario verificar que la aplicación respete las normativas de Apple respecto a usabilidad, seguridad y transparencia.
- **Transparencia en datos del usuario**: Dado que Meal Plan no recopila información sensible ni utiliza sensores, no requiere permisos adicionales como tracking, HealthKit o GPS.

Gracias a su enfoque discreto y local, la aplicación presenta un proceso de publicación relativamente sencillo.

Equipo de trabajo y roles

El desarrollo de Meal Plan se realizó mediante la colaboración organizada del equipo **Code4Change**, conformado por cuatro integrantes que asumieron roles complementarios para asegurar un flujo de trabajo eficiente. Cada miembro participó en

diferentes etapas del proyecto, desde el análisis inicial hasta la creación de la demo final, distribuyendo responsabilidades de acuerdo con sus habilidades y conocimientos.

Solís Espinosa Andrea Vianney — Desarrollador principal / Lógica y arquitectura
Responsable de implementar la estructura técnica de la aplicación, gestionar la lógica del sistema y coordinar el funcionamiento entre pantallas y componentes. Participó en la integración del almacenamiento local (Core Data) y en la definición de la arquitectura general del proyecto.

Gaytán Herrera Bélen— Diseño UI/UX / Prototipado
Encargada del diseño visual de la aplicación, la creación de los wireframes y la elaboración de la maqueta final. Aseguró que la interfaz fuera clara, accesible y coherente con el objetivo de la app. Participó en decisiones sobre usabilidad y estilo.

Argueta Bravo Angel Jacob— Documentación / Análisis funcional
Responsable de estructurar el contenido del documento técnico, redactar la descripción del proyecto, definir reglas de negocio y requerimientos, así como asegurar que la documentación fuera clara y completa. También apoyó en validación de flujos y casos de uso.

Soria Aguilar Angel — Tester / Integración y soporte visual
Encargado de probar la aplicación en sus diferentes etapas, detectar inconsistencias y validar la navegación. También colaboró en tareas de diseño ligero, organización de contenidos y aseguramiento de la calidad en la demo final.

Estimaciones de tiempo y costos de desarrollo y mantenimiento

La estimación de tiempos y costos se realizó considerando las tareas necesarias para completar el MVP, perfeccionar la interfaz y garantizar un funcionamiento estable. Esta proyección solo incluye la parte técnica del desarrollo y no contempla gastos de marketing o infraestructura externa.

Estimación de tiempos

- Análisis y diseño del proyecto: 1–2 semanas
- Diseño de wireframes y prototipos: 1 semana
- Implementación de pantallas y navegación: 3–4 semanas
- Integración de lógica y almacenamiento local (Core Data): 2 semanas
- Pruebas, correcciones y optimización: 1–2 semanas
- Documentación y preparación para publicación: 1 semana

Tiempo total estimado: 8 a 10 semanas de desarrollo.

Estimación de costos

- Cuenta de desarrollador Apple: $99 USD/año
- Herramientas de diseño (Figma/Canva): gratuitas o con versiones premium opcionales
- Repositorios y control de versiones (GitHub): gratuito
- Costos de mantenimiento anual:
- Actualizaciones menores
- Ajustes por cambios de iOS
- Corrección de bugs: aproximadamente $300–$500 USD al año (estimado profesional).

Si en el futuro se incorporan APIs externas, IA o infraestructura en la nube, los costos aumentarían significativamente, pero no forman parte del alcance actual.

Referencias

SwiftUI. (2024). Apple Developer Documentation.

https://developer.apple.com/documentation/swiftui

Core Data. (2024). Apple Developer Documentation.

https://developer.apple.com/documentation/coredata

Human Interface Guidelines. (2024). Apple Developer Documentation.

https://developer.apple.com/design/human-interface-guidelines/

App Store Review Guidelines. (2024). Apple Developer Documentation.

https://developer.apple.com/app-store/review/guidelines/

Privacy and Data Use. (2024). Apple Developer Documentation.

https://developer.apple.com/app-store/user-privacy-and-data-use/

Healthy Diet. (2023). World Health Organization.

https://www.who.int/news-room/fact-sheets/detail/healthy-diet