



Universidad Nacional Autónoma de
México



Facultad de Ingeniería

División de Ingeniería Eléctrica (DIE)

Cómputo Móvil

Grupo: 3

Semestre: 2024-2

Fecha: 24 de mayo de 2024

Profesor: Ing. Marduk Pérez de Lara Domínguez

Cabrera Gaytán Jazmín Andrea

Trabajo Final

AthenaStrength

Reglas de Negocio:

Registro de Usuarios:

- Los usuarios pueden optar por crear una cuenta nueva o iniciar sesión.
- Los usuarios deben proporcionar información básica como nombre, correo electrónico, contraseña y usuario para poder registrarse en la aplicación.
- Se debe validar la dirección de correo electrónico para garantizar la autenticidad de la cuenta del usuario, esto a través de un link el cuál recibirá en su bandeja de entrada.
- Los usuarios contestarán un formulario para conocer a detalle su historia clínica y preferencias.

Privacidad y Seguridad:

- Implementar medidas de seguridad para la protección de los datos personales y de salud de los usuarios.

Monetización:

- La aplicación ofrece tres versiones de suscripciones: una premium la cual es el programa completo y dos suscripciones normales las cuales se escogen entre alimentación o entrenamiento.

Requerimientos Funcionales:

Fitness:

- Registro y seguimiento de entrenamientos, incluyendo grupo muscular, tipo de ejercicio, duración, intensidad y calorías quemadas.
- Ofrecer planes de entrenamiento personalizados basados en objetivos específicos y nivel de condición física.
- Funcionalidades de cronómetro y temporizador para guiar durante los entrenamientos.

Nutrición:

- Registro y seguimiento de la ingesta diaria de alimentos, incluyendo calorías, macronutrientes y micronutrientes.
- Sugerencias de comidas y planes de alimentación personalizados según objetivos de salud, preferencias dietéticas y grupo muscular que toque entrenar.
- Base de datos de alimentos con información nutricional detallada y búsqueda por nombre o categoría y recetas.

Desarrollo del Bienestar:

- Registro y análisis del sueño para mejorar la calidad del descanso.
- Herramientas de gestión del estrés, como ejercicios de relajación, lecturas y afirmaciones diarias.
- Sesiones guiadas de meditación y mindfulness para promover el bienestar mental.

Requerimientos No Funcionales

Usabilidad:

- La interfaz de usuario debe ser intuitiva y fácil de usar para usuarios de todas las edades y niveles de experiencia.
- La aplicación debe ser compatible con múltiples dispositivos y sistemas operativos móviles (iOS y Android).

Seguridad:

- La app debe cumplir con los estándares de seguridad y privacidad de datos.

Rendimiento:

- La aplicación debe ser rápida y receptiva, incluso cuando se trabaja con grandes cantidades de datos como registros de entrenamiento y alimentos.

Escalabilidad:

- La arquitectura de la aplicación debe ser escalable para manejar un gran número de usuarios y datos a medida que crece la base de usuarios, así como nueva implementación de entrenamientos, nutrición o los recursos de bienestar.

Disponibilidad:

- La aplicación debe estar disponible y funcionar correctamente en todo momento, con un tiempo de inactividad mínimo para mantenimiento planificado.

Consideremos la funcionalidad de "Integración con accesorios como Smartwatches" como parte de una aplicación enfocada en fitness, nutrición y bienestar. Esta funcionalidad implica permitir que la aplicación se sincronice con dispositivos como smartwatches, rastreadores de actividad física o básculas inteligentes.

Viabilidad

Mejora de la experiencia del usuario: La integración con estos dispositivos puede mejorar la experiencia del usuario al proporcionar una forma conveniente de rastrear

automáticamente su actividad física, ritmo cardíaco, calidad del sueño, etc. Esto elimina la necesidad de que los usuarios ingresen manualmente estos datos en la aplicación.

Precisión de los datos: Los dispositivos wearables suelen contar con sensores avanzados que proporcionan mediciones precisas de la actividad física, el ritmo cardíaco y otros parámetros relevantes para el fitness y el bienestar. Esto asegura que los datos recopilados sean confiables y precisos.

Motivación extra: Al sincronizar los datos del dispositivo con la aplicación, los usuarios pueden recibir análisis detallados de su progreso, establecer objetivos personalizados y recibir incentivos para mantenerse activos y saludables.

Inviabilidad

Privacidad y seguridad: Es crucial implementar medidas sólidas de seguridad y privacidad para proteger los datos sensibles del usuario.

Costo: Los dispositivos wearables pueden ser costosos, lo que puede limitar su accesibilidad para ciertos usuarios. Esto podría reducir el alcance de la aplicación si depende en gran medida de la integración con estos dispositivos.

Dependencia de la tecnología: La integración con dispositivos wearables hace que la aplicación sea más dependiente de la tecnología externa. Si hay problemas de compatibilidad o fallas en los dispositivos, los usuarios pueden experimentar interrupciones en la funcionalidad de la aplicación.

Integrar dispositivos wearables puede mejorar la experiencia del usuario y la calidad de los datos recopilados, pero puede presentar desafíos relacionados con el costo, la dependencia tecnológica y la seguridad de los datos.

Alcance

Entrenamiento:

Seguimiento de entrenamientos: Permitir a los usuarios registrar y hacer un seguimiento de sus entrenamientos con ejercicios personalizados.

Planes de entrenamiento: Proporcionar planes de entrenamiento basados en objetivos específicos, como pérdida de peso, ganancia muscular o recomposición corporal.

Registro de progreso: Permitir a los usuarios registrar su progreso a lo largo del tiempo, incluyendo medidas corporales, tiempos, repeticiones, etc.

Integración con dispositivos: Conexión con dispositivos wearables o aplicaciones de seguimiento de actividad para sincronizar datos automáticamente.

Comunidad: Incluir funciones sociales para que los usuarios puedan compartir sus logros, retos y progreso con otros usuarios en redes sociales.

Nutrición:

Registro de alimentos: Permitir a los usuarios llevar un registro de su ingesta diaria de alimentos, incluyendo calorías, macronutrientes, vitaminas, etc.

Planificación de comidas: Ofrecer sugerencias de comidas y planes de comidas personalizados según los objetivos de salud y preferencias dietéticas.

Base de datos de alimentos: Incluir una amplia base de datos de alimentos con información nutricional detallada.

Desarrollo del bienestar:

Seguimiento del sueño: Permitir a los usuarios registrar y analizar sus hábitos de sueño para mejorar la calidad del descanso.

Gestión del estrés: Proporcionar recursos y técnicas para ayudar a los usuarios a gestionar el estrés y mejorar su bienestar mental.

Meditación y mindfulness: Incluir sesiones guiadas de meditación y ejercicios de mindfulness para promover la relajación y la atención plena.

MVP (Producto Mínimo Viable):

Para el MVP, me centraría en las características más esenciales que proporcionen valor inmediato a los usuarios:

1. Registro básico de entrenamientos y progreso.
2. Funcionalidad de registro de alimentos con una base de datos limitada.
3. Un par de planes de entrenamiento y de comidas predefinidos.
4. Función social básica para compartir logros.
5. Integración mínima con dispositivos.

Nombre final de la app: AthenaStrenght

Wireframes y navegación de pantallas principales con detalle de diseño.

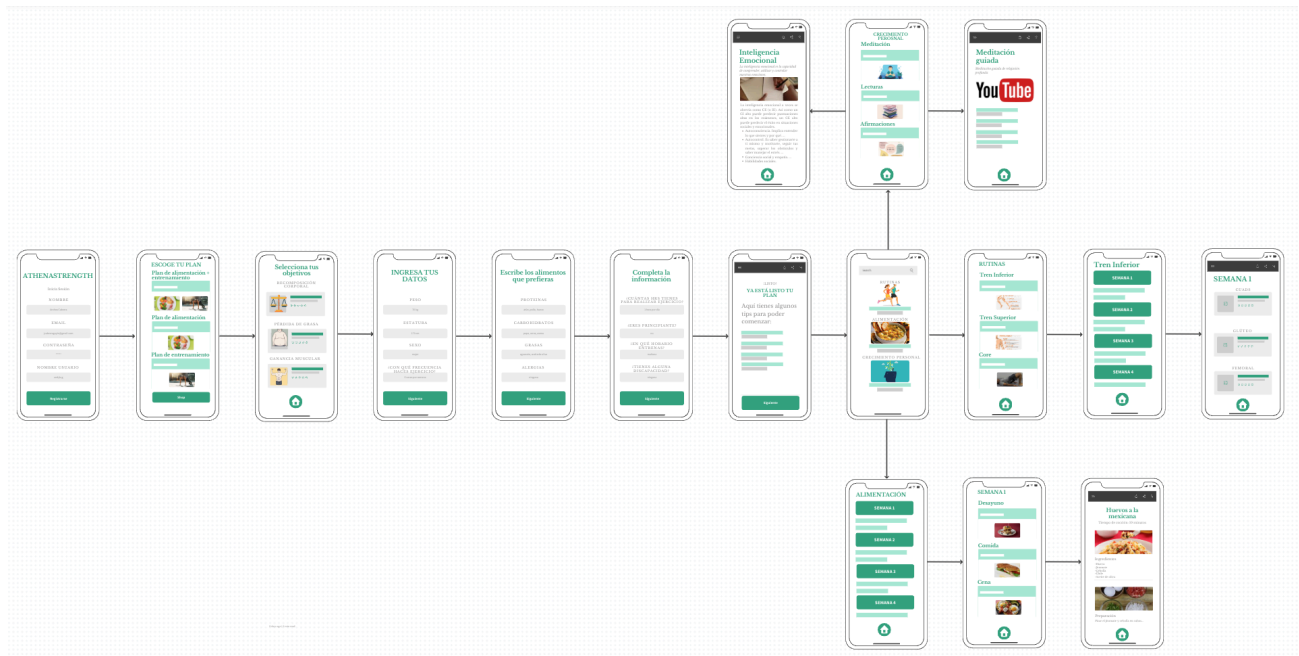


Imagen 1. Sección completa de Wireframes

En la Imagen 1 podemos ver a gran escala cómo estaría implementada nuestra aplicación.

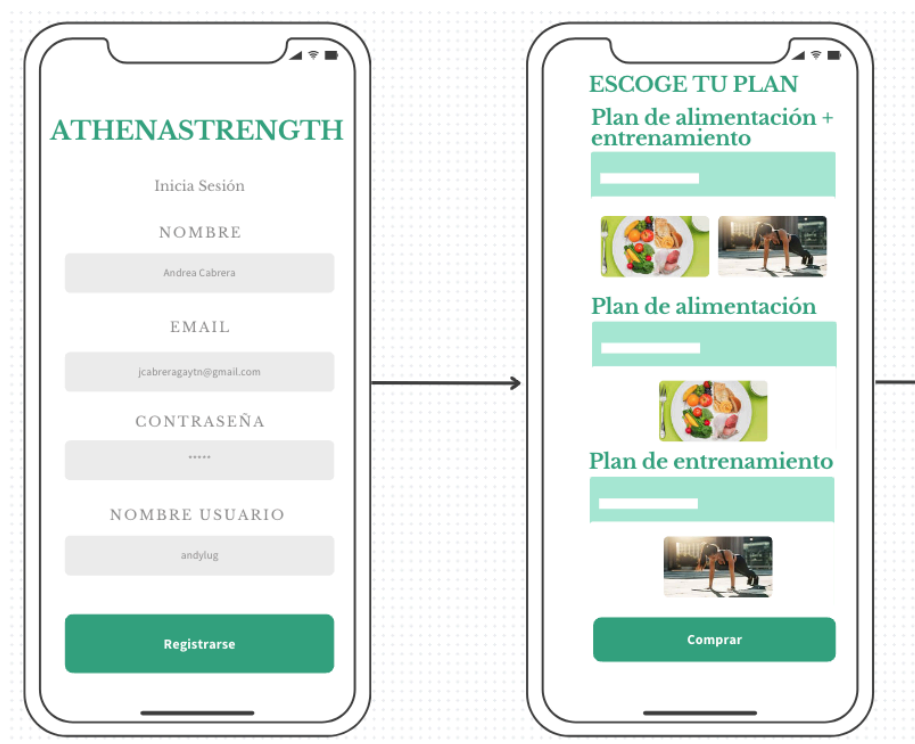
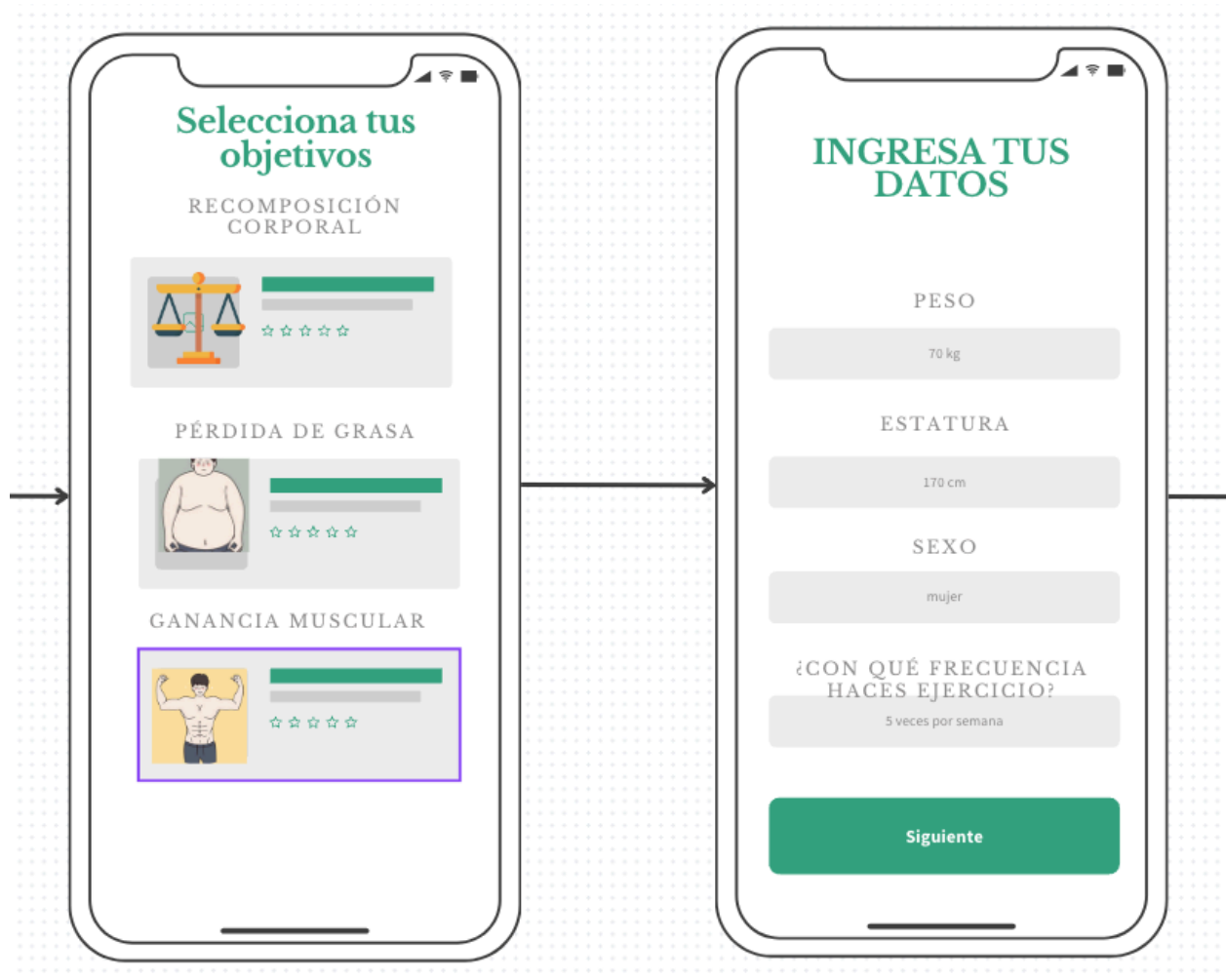


Imagen 2. Pantalla de registro y selección/compra de plan

Al iniciar la aplicación podremos iniciar sesión, si no podemos registrarnos con nuestro nombre, usuario, correo y contraseña. Una vez registrados nos llegará un correo de confirmación, así podemos continuar con la compra de nuestro plan.



[Imagen 3. Selección de objetivos y registro de datos clínicos esenciales.](#)

Una vez hayamos pagado la suscripción, deberemos elegir entre tres objetivos, una vez hecho esto, tendremos que contestar varias preguntas a base de un formulario. (Imagen 3 y 4). Esto con la finalidad de conocer las preferencias del usuario y poder ofrecer un plan 100% personalizado.

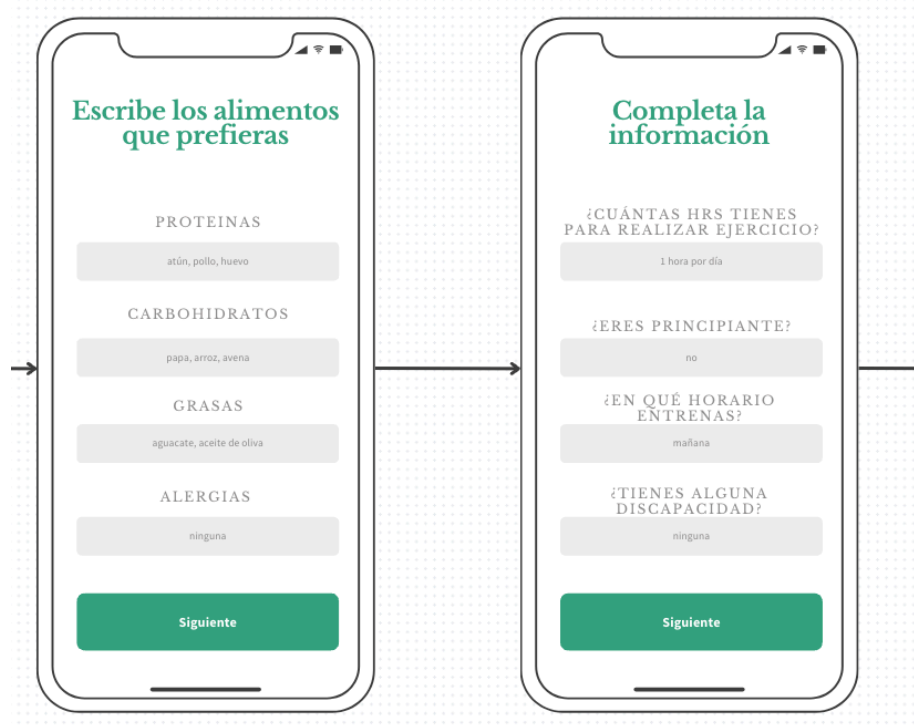


Imagen 4. Registro de datos clínicos importantes

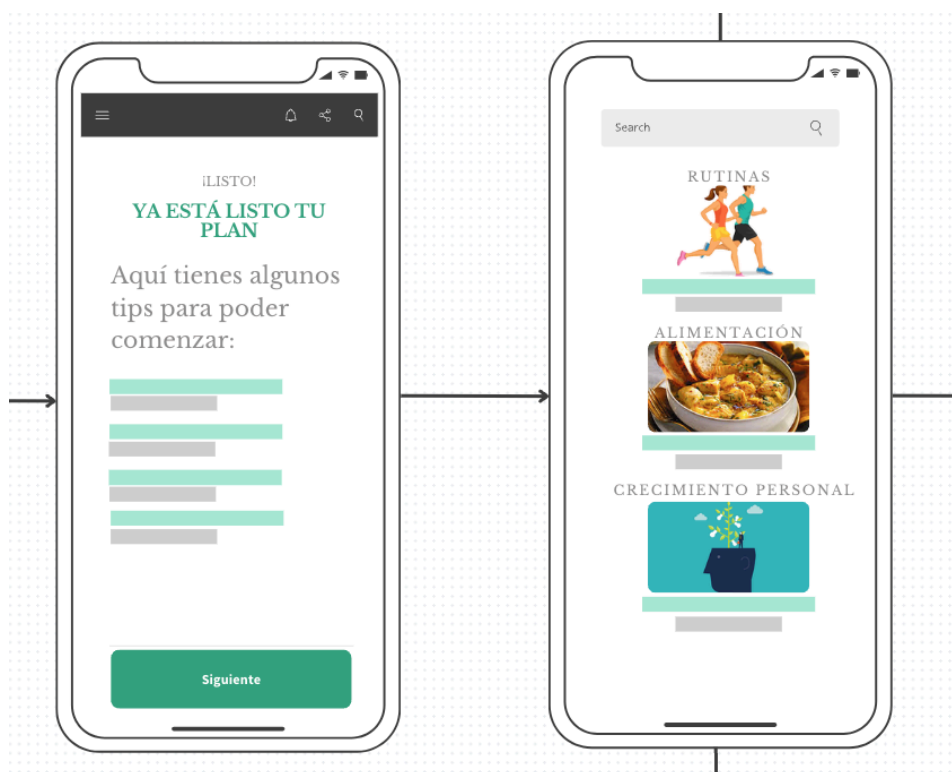
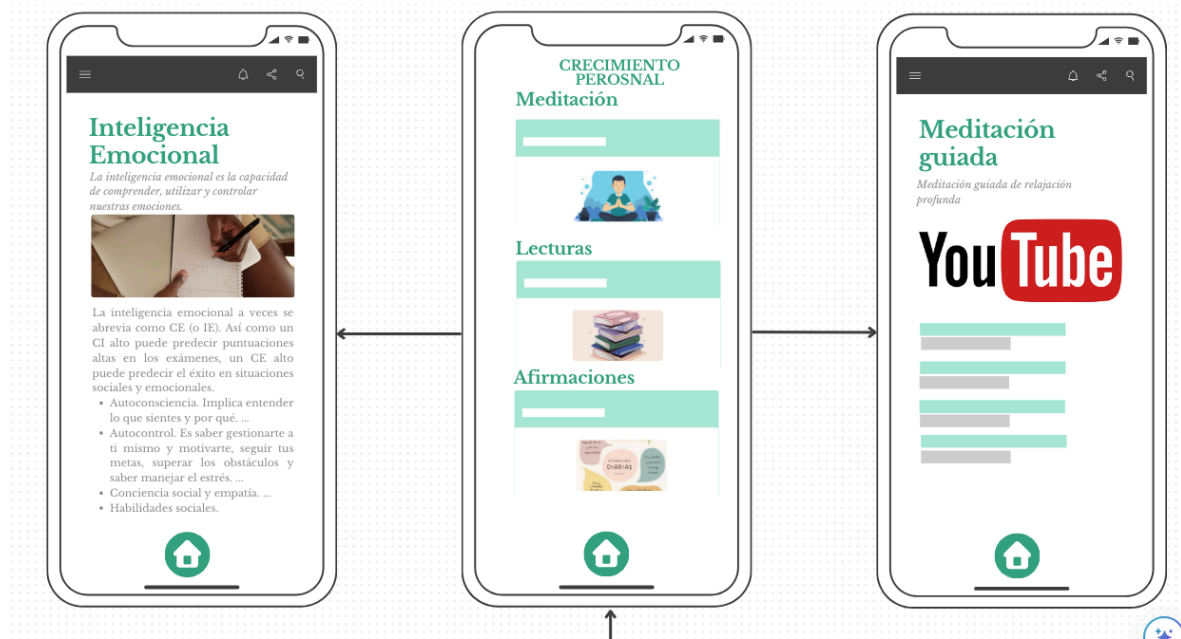


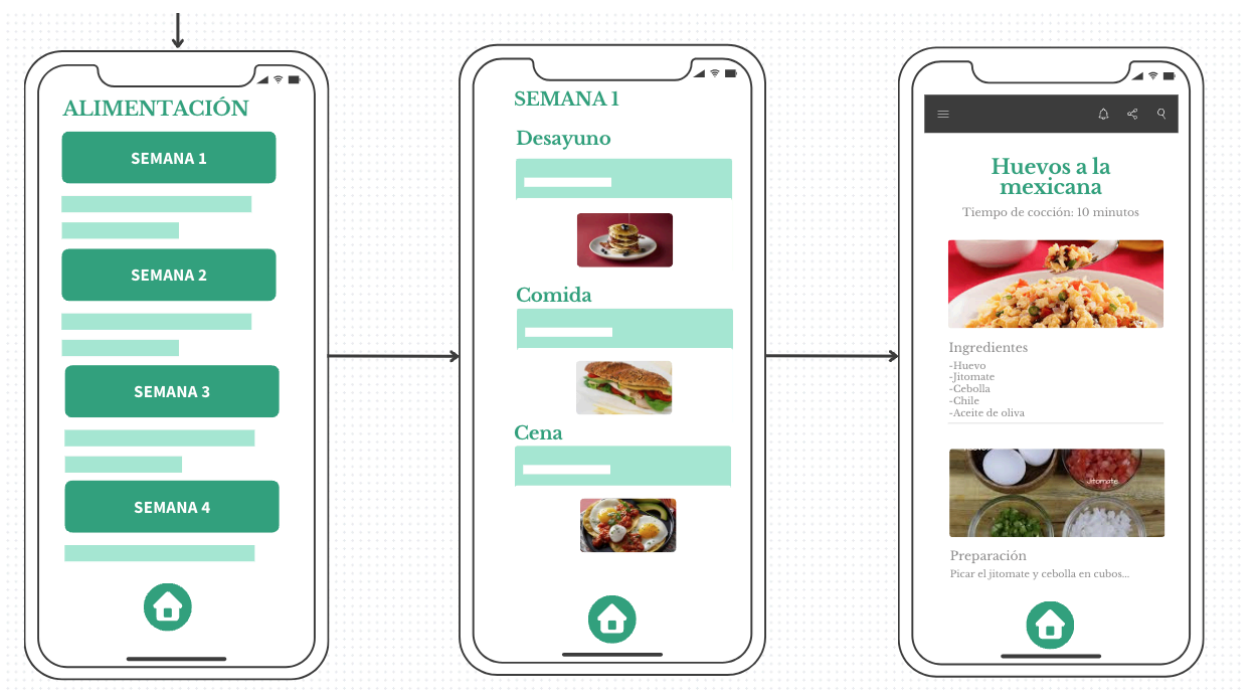
Imagen 5. Pantalla de Tips y Pantalla de las tres secciones principales de la app

Una vez terminado el registro, ya podremos disfrutar de los servicios que ofrece la aplicación, podemos navegar por las secciones: Rutinas, Alimentación y Crecimiento personal. Estas a su vez se van desglosando según con lo que el cliente seleccione.



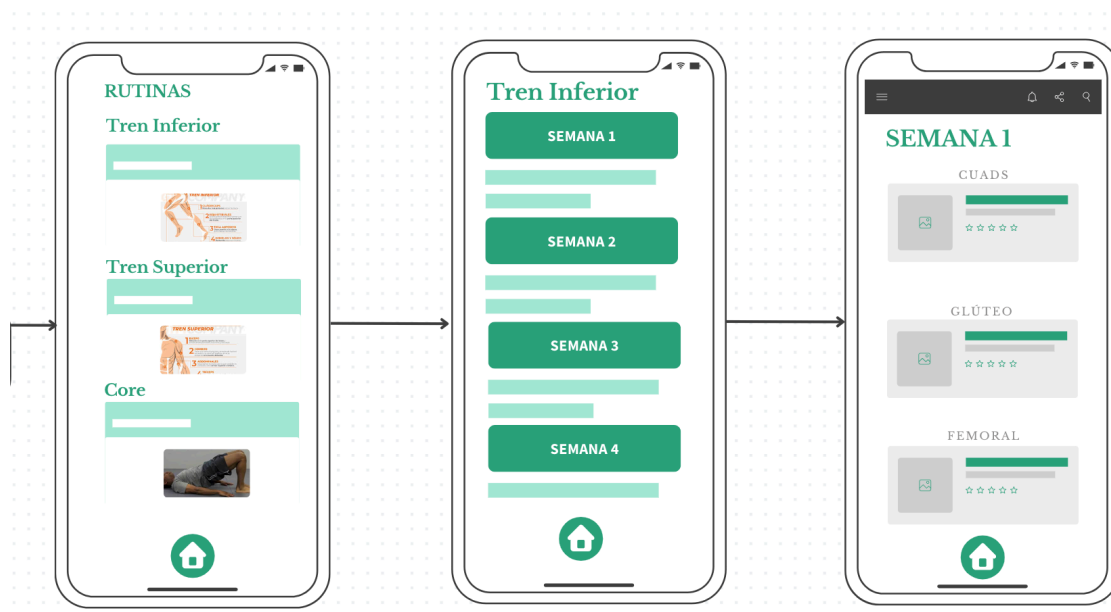
[Imagen 6. Sección de Crecimiento Personal](#)

Dentro de esta sección encontraremos tres subsecciones: Meditación que nos llevará a videos en YouTube de meditaciones guiadas, lecturas para cada día poder tener una sección donde podamos aprender cada día más sobre lo que estamos haciendo y mejorando.



[Imagen 7. Sección de Dieta](#)

En esta sección encontraremos dividida nuestra alimentación por semanas, cada una dividida en desayuno, comida y cena (esta sección podrá variar según la respuesta registrada del usuario en el formulario de entrada). una vez que seleccionemos el desayuno, comida o cena, nos llevará al platillo que esté programado, pudiendo observar la forma de preparación, sus calorías y tabla nutricional.



[Imagen 8. Sección de Rutinas](#)

Aquí podemos escoger el conjunto de grupos musculares divididos en: tren inferior (brazo, hombro, pecho y espalda), tren superior (cuádriceps, femorales, glúteos). Al escoger un grupo, avanzaremos a la siguiente ventana donde están divididas por semana las rutinas. Una vez seleccionado la semana, podremos encontrarnos ya con el músculo en específico y sus respectivas rutinas con sus tiempos y muestras de técnicas y ejecución de cada uno.

Algunas funcionalidades viables y los posibles servicios, frameworks o kits de desarrollo que podrían utilizarse serían:

Registro de Usuarios:

Servicio de autenticación: Se utilizarían servicios de autenticación como Firebase Authentication de Google o Auth0, que proporcionan una manera segura y fácil de implementar la autenticación de usuarios en la aplicación.

Gestión de Perfiles:

Base de Datos: Almacenar la información de los perfiles de usuario utilizando alguna base de datos como Firebase Realtime Database o Firestore (también de Google), que ofrecen una integración sencilla con otros servicios de Firebase y son escalables para manejar grandes cantidades de datos.

Entrenamiento:

API de Fitness: Integraría APIs de fitness como Google Fitness API o Apple HealthKit, que permiten acceder a los datos de actividad física y salud del usuario almacenados en sus dispositivos móviles. Estas APIs proporcionan acceso a información como pasos, actividad deportiva, frecuencia cardíaca, etc., para personalizar los planes de entrenamiento y realizar un seguimiento del progreso del usuario.

Nutrición:

API de Nutrición: Para obtener información nutricional de alimentos, integrar APIs como la base de datos de alimentos de USDA o la API de Nutrionix. Ofrecen datos detallados sobre el contenido nutricional de una amplia variedad de alimentos, lo que facilita el registro y seguimiento de la ingesta de alimentos por parte de los usuarios.

Desarrollo del Bienestar:

Servicios de Bienestar y Salud Mental: Como Calm o Headspace para ofrecer sesiones guiadas de meditación, ejercicios de respiración y técnicas de relajación. Suelen ofrecer APIs que permiten integrar su contenido directamente en tu aplicación.

Usabilidad y Rendimiento:

Frameworks de Desarrollo de Apps Móviles: Para garantizar una interfaz de usuario intuitiva y un rendimiento óptimo, podría utilizar frameworks como React Native (para desarrollo multiplataforma) o Flutter (de Google) que permiten desarrollar aplicaciones móviles nativas con un solo código base. Estos frameworks ofrecen herramientas y componentes predefinidos que facilitan el desarrollo y la optimización de la aplicación para diferentes dispositivos y sistemas operativos móviles.

Al elegir los servicios, frameworks o kits de desarrollo adecuados, se busca optimizar el proceso de desarrollo, garantizar la seguridad y la escalabilidad de la aplicación, así como ofrecer una experiencia de usuario óptima y un rendimiento sólido.

Características y Funcionalidades:

Gestión de Usuarios:

- Registro de nuevos usuarios con validación de correo electrónico.
- Inicio de sesión de usuarios existentes.
- Gestión de perfiles de usuario, incluyendo información personal, preferencias y datos de salud.

Almacenamiento de Datos:

- Almacenamiento de registros de entrenamiento, datos de actividad física y progreso del usuario.
- Base de datos para la gestión de alimentos, planes de comidas y datos nutricionales.

Integración con APIs Externas:

- Integración con APIs de fitness y nutrición para obtener datos de actividad física y nutricionales, respectivamente.
- Autenticación y autorización para acceder a las APIs externas de manera segura.

Seguridad y Privacidad:

- Implementación de medidas de seguridad para proteger los datos del usuario, como el cifrado de contraseñas y la autenticación de dos factores.
- Control de acceso basado en roles para garantizar que solo los usuarios autorizados puedan acceder a ciertos recursos.

Escalabilidad y Rendimiento:

- Arquitectura escalable que pueda manejar un gran volumen de usuarios y datos.

- Optimización de consultas y operaciones de base de datos para garantizar un rendimiento óptimo de la aplicación.

Detalles de Implementación:

Tecnología:

- Lenguaje de programación: como Node.js, Python, Java.
- Framework: Express.js (Node.js), Flask (Python) o Spring Boot (Java) para facilitar el desarrollo del backend y gestionar las solicitudes HTTP.

Base de Datos:

- Base de datos relacional como MongoDB, PostgreSQL, MySQL, etc.
- Diseño de esquema de base de datos para almacenar información de usuarios, registros de entrenamiento, datos nutricionales, etc.

API y Endpoints:

- Definición de endpoints RESTful para acceder y manipular recursos como usuarios, registros de entrenamiento, datos nutricionales, etc.
- Implementación de operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para cada recurso, como /usuarios, /entrenamientos, /alimentos, etc.

Autenticación y Autorización:

- Utilización de protocolos de autenticación como JWT (JSON Web Tokens) para autenticar a los usuarios y proporcionar acceso seguro a los recursos protegidos.
- Implementación de middleware de autorización para verificar los permisos de acceso de los usuarios a los recursos.

Escalabilidad:

- Arquitectura modular y escalable que permita agregar nuevos servicios y funcionalidades según sea necesario.
- Implementación de técnicas de escalabilidad horizontal, como el uso de balanceadores de carga y la distribución de la carga entre múltiples servidores.

El flujo de datos entre los usuarios y los servicios backend se gestiona de manera segura y eficiente, asegurando la persistencia de los datos, la disponibilidad de la aplicación en situaciones sin conexión y la protección de la seguridad de los datos mediante técnicas de cifrado, autenticación y autorización.

Flujo de Datos:

Desde los Usuarios:

- Proporcionan información personal durante el registro, como nombre, correo electrónico y contraseña.
- Registran datos de entrenamiento, incluyendo tipo de ejercicio, duración, intensidad, etc.
- Registran su ingesta de alimentos, incluyendo detalles nutricionales.
- Pueden interactuar con funcionalidades adicionales como seguimiento del sueño, gestión del estrés, etc.

Hacia los Usuarios:

- Reciben información personalizada, como planes de entrenamiento, sugerencias de comidas, recordatorios, etc.
- Acceden a análisis y seguimiento de su progreso en fitness, nutrición y bienestar.
- Reciben notificaciones sobre eventos importantes, como recordatorios de entrenamiento o cumplimiento de objetivos.

Comunicación y Persistencia de Datos:

Comunicación:

- La comunicación entre la aplicación móvil y los servicios backend se realiza a través de solicitudes HTTP/HTTPS utilizando el protocolo RESTful.
- Los datos se envían en formato JSON para facilitar la interoperabilidad y el procesamiento en ambos extremos.

Persistencia de Datos:

- Los datos de los usuarios se persisten en una base de datos, ya sea relacional o no relacional, según los requisitos de la aplicación.
- Se pueden utilizar técnicas de almacenamiento en caché para mejorar el rendimiento y reducir la carga en la base de datos principal.

Manejo de Situaciones sin Conexión o Servicio de Datos:

Almacenamiento Local en Dispositivos Móviles:

- Los dispositivos móviles pueden almacenar datos localmente en la memoria del dispositivo, utilizando bases de datos internas como SQLite.
- La aplicación móvil puede funcionar en modo offline, permitiendo a los usuarios registrar datos localmente incluso cuando no hay conexión a internet.

Sincronización Asincrónica:

- La aplicación móvil implementa un mecanismo de sincronización asincrónica que verifica periódicamente si hay datos pendientes de sincronización.
- Cuando se restablece la conexión, los datos locales se sincronizan con los servicios backend de manera automática.

Seguridad de los Datos:

Cifrado de Datos en Reposo y en Tránsito:

- Los datos se cifran tanto en reposo (almacenados en la base de datos) como en tránsito (durante la transmisión entre la aplicación y los servicios backend) utilizando algoritmos de cifrado sólidos.

Autenticación y Autorización:

- Se implementa un sistema de autenticación como JWT (JSON Web Tokens), para verificar la identidad de los usuarios.
- Se establecen roles y permisos de acceso para controlar qué usuarios pueden acceder a qué datos y funcionalidades.

Protección contra Ataques:

- Se aplican medidas de seguridad adicionales, como la validación de datos de entrada, para prevenir ataques como inyección de SQL y XSS (Cross-Site Scripting).

Para publicar una aplicación en tiendas de aplicaciones como la App Store de Apple o Google Play Store, es importante considerar una serie de permisos y adecuaciones para garantizar que la aplicación cumpla con los requisitos y políticas establecidos por cada plataforma. Aquí hay algunas consideraciones importantes:

Políticas de Contenido:

- Asegúrate de que se cumplen con las políticas de contenido de la tienda de aplicaciones, lo que incluye no incluir contenido inapropiado, ilegal o que viole los derechos de autor.

Derechos de Autor y Propiedad Intelectual:

- Verificar que no se infrinjan los derechos de autor, marcas comerciales u otros derechos de propiedad intelectual de terceros.

Privacidad y Seguridad:

- Implementa medidas de seguridad y privacidad para proteger los datos de los usuarios.
- Proporciona una política de privacidad clara que explique cómo se recopilan, utilizan y comparten los datos del usuario.
- Obtén el consentimiento explícito de los usuarios antes de recopilar cualquier tipo de información personal.

Requisitos Técnicos:

- La app debe cumplir con los requisitos técnicos y las directrices de diseño establecidas por la plataforma de la tienda de aplicaciones.
- Realizar pruebas exhaustivas para garantizar que la aplicación funcione correctamente en diferentes dispositivos y versiones de sistemas operativos.

Monetización:

- Cumplir con las políticas de monetización de la tienda de aplicaciones.
- Proporcionar información clara sobre cualquier tipo de pago que requiera la aplicación.

Soporte y Mantenimiento:

- Proporcionar información de contacto y un mecanismo de soporte para los usuarios en caso de que necesiten ayuda o quieran informar sobre problemas técnicos.

Se busca desarrollar una aplicación híbrida, por lo que los posibles y potenciales lenguajes de programación serían:

- HTML (HyperText Markup Language) para definir la estructura de la interfaz de usuario de la aplicación.
- CSS (Cascading Style Sheets) para estilizar y dar formato a los elementos HTML.
- JavaScript para la lógica de la aplicación, la interactividad y la manipulación del DOM (Document Object Model).

Plataformas de Desarrollo en la Nube:

- Servicios como Firebase de Google o AWS Amplify de Amazon para utilizarlos para proporcionar funcionalidades backend como autenticación de usuarios, almacenamiento de datos, notificaciones push, análisis, etc.

Control de Versiones:

- Usar un control de versiones como Git y plataformas como GitHub o Bitbucket para colaborar en el desarrollo de la aplicación y mantener un historial de cambios.

Al elegir los lenguajes de programación y servicios para una aplicación híbrida, es importante considerar los requisitos del proyecto, la experiencia del equipo de desarrollo y la compatibilidad con las plataformas objetivo (iOS, Android, etc.).