



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

Asignatura: Cómputo Móvil

Proyecto Final

Grupo: 03

Profesor: Marduk Pérez de Lara Domínguez

Equipo 4

Integrantes:

Cruz Vázquez Zayra Sheccid

Gómez Bolaños Luis Aldo

Rivera Morales Alan Adrián

Semestre 2025-1

Fecha de entrega: 22 de noviembre del 2024

CONTENIDO

Introducción	2
Desarrollo	3
Detalle de requerimientos de la app	3
Viabilidad técnica	5
Wireframes de la app	8
Registro de ID	14
Registro Exitoso	14
Retiro con valor cero y mil	15
Confirmación de Retiro	15
Configuración (Activar notificaciones)	16
Promoción de boletos	16
Explicación del flujo de recorrido de las pantallas	17
Detalles de los lenguajes de programación, permisos y adecuaciones para su publicación	20
Componentes	21
Tiempo de trabajo y Equipo de trabajo	23
Conclusión	23
Referencias	24

Introducción

Miiles es una aplicación diseñada para potenciar a los emprendedores latinoamericanos en el competitivo mundo del marketing digital. Su historia comenzó como una agencia de marketing tradicional, pero, con una visión ambiciosa y un enfoque dinámico, evolucionó hacia una plataforma integral que impulsa el desarrollo de emprendedores y startups. Esta transformación refleja su compromiso de adaptarse a las necesidades específicas de cada cliente, convirtiéndose en un aliado estratégico para quienes buscan destacar en un entorno digital en constante cambio.

Lo que distingue a Miiles es su capacidad para ofrecer soluciones personalizadas y herramientas innovadoras, diseñadas para superar los desafíos particulares que enfrentan los emprendedores en la región. Desde estrategias de marketing efectivas hasta la implementación de tecnología avanzada, la plataforma brinda un ecosistema completo para que los negocios prosperen. Uno de sus pilares es el marketing de afiliados, un modelo que no solo aumenta la visibilidad de los emprendedores, sino que también les permite acceder a nuevos mercados y diversificar sus canales de ingreso.

Además, Miiles no se limita a ofrecer herramientas; fomenta una comunidad vibrante de colaboración y aprendizaje continuo. Esta red de apoyo, combinada con un enfoque centrado en el crecimiento y la innovación, crea un entorno donde los emprendedores pueden compartir conocimientos, generar alianzas estratégicas y escalar sus proyectos con confianza.

Desarrollo

Nombre de la App: Miiles

Reglas de Negocio

Generación de ID

Generación de Código de Afiliado (ID): Cada nuevo socio (vendedor) registrado en la aplicación recibirá un código único de identificación al momento de registrarse. Este ID es necesario para que los clientes puedan registrar su solicitud y establecer contacto con la empresa.

Estructura de Ganancias

Vendedor y Comisión: El vendedor gana una comisión por cada cliente que traiga a la empresa. Esta comisión puede ser de dos tipos:

Comisión Parcial: Si el cliente no cierra el trato, el vendedor recibe una pequeña ganancia por haber traído al cliente.

Comisión Completa: Si el cliente cierra un trato, el vendedor recibe una comisión completa de acuerdo con el acuerdo entre las partes.

Proceso de Registro

El cliente debe ingresar el ID del vendedor en la aplicación para generar una solicitud de cita.

Después de ingresar el ID, el cliente puede hacer una cita para hablar sobre sus necesidades de marketing, la cual será enviada al equipo interno para contactar con el cliente y discutir la propuesta.

Rechazo del Cliente

Si el cliente no decide cerrar el trato, el vendedor puede recibir una compensación parcial o sin compensación, dependiendo de las políticas internas.

Control de la Información

Los vendedores deben poder ver y gestionar sus ganancias y actividad dentro de la aplicación.

La aplicación debe llevar un registro completo de las interacciones y el estado de las citas generadas (pendiente, cerrado, cancelado, etc.).

Requerimientos Funcionales

Registro de Usuario

Cliente: El cliente debe poder registrarse con una cuenta básica (nombre, correo electrónico, contraseña). Al registrarse, debe ingresar un ID de vendedor que se utiliza para asociarlo con el vendedor correspondiente.

Vendedor: Los vendedores se registran igual que los clientes, pero tienen roles y privilegios adicionales en la aplicación para gestionar sus ganancias y sus usuarios.

Interfaz del Cliente

Ingreso de Código de Vendedor: Los clientes deben poder ingresar el código ID de su vendedor al momento de registrarse, para generar un ticket y hacer una cita.

Generación de Citas: Los clientes pueden seleccionar una fecha y hora para generar una cita. Esta cita es enviada a los vendedores y a los administradores.

Visualización de Citas: Los clientes podrán ver el estado de la cita: si está confirmada, pendiente o cancelada.

Interfaz del Vendedor

Visualización de Actividad: Los vendedores deben poder ver la actividad de las citas generadas con sus clientes (pendientes, confirmadas, canceladas).

Visualización de Ganancias: Los vendedores pueden ver cuánto han ganado por cada venta cerrada o intento de venta.

Gestión de Clientes: Los vendedores podrán gestionar la información de sus clientes y ver el historial de los clientes que han traído.

Interfaz Administrativa

Gestión de Socios/Vendedores: El administrador debe poder ver todos los vendedores registrados, verificar el estado de las citas y gestionar los pagos o comisiones.

Seguimiento de Comisiones: El administrador debe tener acceso a las ganancias generadas por los vendedores.

Propuestas y Citas: El administrador debe gestionar las citas pendientes, asegurándose de que se asignen al vendedor adecuado y se coordinen con los clientes.

Generación de Reportes

Reporte de Ganancias: La aplicación debe generar reportes mensuales o semanales de las ganancias de los vendedores, el número de citas generadas, y la tasa de conversión.

Soporte y Atención al Cliente

Chat en Vivo: La aplicación debe ofrecer una funcionalidad de chat en vivo para que los usuarios (tanto clientes como vendedores) puedan resolver dudas rápidamente con el equipo de atención al cliente.

Requerimientos No Funcionales

Rendimiento

La aplicación debe ser rápida y responsiva, con tiempos de carga de páginas menores a 3 segundos. La aplicación debe manejar un número alto de usuarios concurrentes sin que se vea comprometido su rendimiento.

Escalabilidad

La aplicación debe ser escalable para adaptarse a un creciente número de usuarios, vendedores y clientes. El sistema debe permitir expandir funcionalidades en el futuro, como integración con otros servicios de pago, sistemas de marketing, o plataformas de CRM.

Seguridad

Los datos de los usuarios deben ser encriptados tanto en tránsito (HTTPS) como en reposo (base de datos). Se debe garantizar el cumplimiento de normas de privacidad como GDPR.

Autenticación segura mediante autenticación de dos factores para vendedores y administradores.

El acceso de los usuarios debe ser limitado de acuerdo con su rol (cliente, vendedor, administrador).

Accesibilidad

La aplicación debe ser accesible para usuarios con discapacidades, cumpliendo con las pautas WCAG (Web Content Accessibility Guidelines).

Debe ser completamente funcional tanto en dispositivos móviles como en escritorio.

Mantenimiento

La aplicación debe tener un sistema de backups automáticos para evitar la pérdida de datos importantes.

Monitoreo constante del sistema para detectar problemas de rendimiento o caídas.

Usabilidad

La interfaz debe ser intuitiva y fácil de navegar, con un diseño amigable para usuarios de distintos niveles de experiencia.

La aplicación debe tener tutoriales o guías para nuevos usuarios, especialmente para los vendedores que tal vez no estén familiarizados con las herramientas digitales.

Multilingüe

La aplicación puede ser multilingüe para adaptarse a diferentes regiones y mercados.

Integración de Chat en Vivo para Atención al Cliente

Viable, pero con desafíos: El chat en vivo es una funcionalidad común en muchas aplicaciones. Hay múltiples opciones para integrar esta funcionalidad, ya sea mediante el desarrollo de una solución interna o mediante la integración de servicios de terceros como Intercom, Zendesk, Tawk.to, o Freshchat.

Los desafíos incluyen la necesidad de gestionar concurrencia en tiempo real, la escalabilidad del sistema, y asegurar que los usuarios reciban respuestas oportunas. Si la aplicación está orientada a un gran número de usuarios simultáneos, los requisitos de infraestructura aumentarán.

Aunque es una funcionalidad viable, su implementación puede requerir la integración con servicios de terceros o el desarrollo de un backend más robusto que soporte múltiples conexiones de usuarios en tiempo real, lo cual puede ser costoso en términos de infraestructura si no se optimiza correctamente.

Notificaciones Push en Tiempo Real para Confirmaciones de Citas

Viable: Las notificaciones push son una funcionalidad bien soportada por los frameworks modernos. Esto permitiría que los clientes y vendedores reciban actualizaciones en tiempo real sobre el estado de las citas (confirmación, cancelación, cambios en la hora, etc.).

Este tipo de funcionalidad también se puede combinar con notificaciones por correo electrónico y SMS, pero las notificaciones push serían más inmediatas.

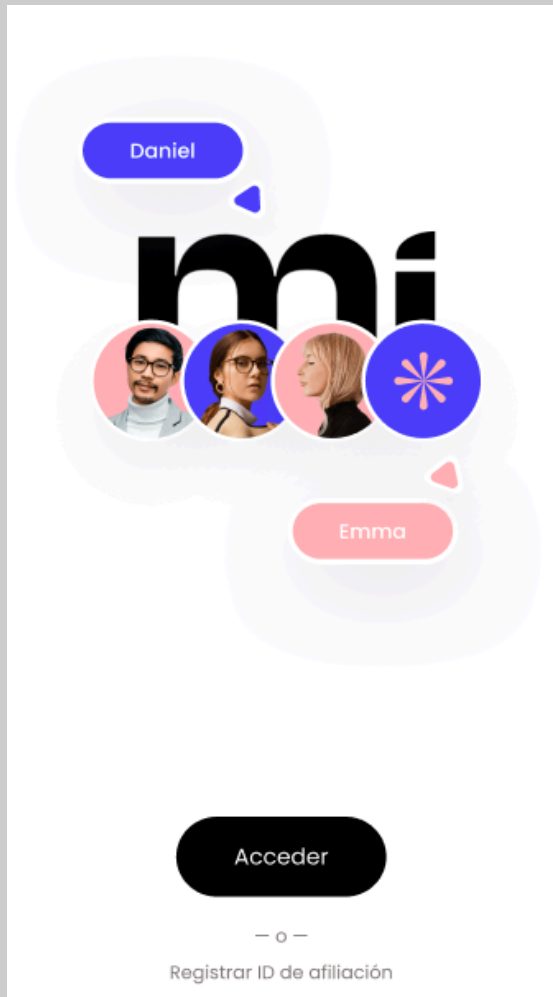
Los sistemas de notificación push son ampliamente utilizados en aplicaciones móviles y web, y existen SDKs disponibles para una integración sencilla. No obstante, puede ser necesario gestionar la sincronización en tiempo real y garantizar que los usuarios reciban las notificaciones de manera confiable, lo que requiere un sistema de backend robusto y confiable.

Multilingüismo

Viable: Implementar multilingüismo en la aplicación es técnicamente posible y comúnmente usado en aplicaciones que operan en múltiples regiones geográficas. Esto puede lograrse mediante el uso de archivos de traducción o servicios como i18n en plataformas de frontend como React, Angular, o Vue.js, y su integración con un sistema de backend que gestione las preferencias de idioma del usuario.

Los desafíos principales son la gestión de traducciones, especialmente si se soportan múltiples idiomas, y garantizar que todas las partes de la interfaz (textos, botones, mensajes) estén correctamente traducidas.

Dado que la mayoría de las aplicaciones móviles y web internacionales soportan varios idiomas, hay herramientas y bibliotecas que permiten implementar esta funcionalidad de forma eficiente, sin grandes complicaciones técnicas. Sin embargo, es importante contar con un equipo de traductores y testers para garantizar que todas las traducciones sean precisas y adecuadas al contexto.



Justificación de Viabilidad

El inicio de sesión de Miiles garantiza que solo usuarios autorizados puedan acceder a las funcionalidades de la plataforma. Además, al requerir un ID de afiliación, se asegura que los usuarios que desean utilizar la app para gestionar campañas publicitarias tengan un vínculo directo con el sistema de marketing de afiliados.

Funcionalidad Implementada

El botón principal "Acceder" está diseñado para que los usuarios con credenciales ingresen rápidamente. Debajo de este, existe una opción para quienes aún no tienen un ID de afiliación, lo que les permite crear uno y así comenzar a utilizar las herramientas de Miiles.

Información y Tipo de Datos

- ID de afiliación: Cadena de texto que identifica de manera única al usuario dentro del sistema.
- Credenciales de inicio de sesión: Usuario y contraseña.
- Botones interactivos: "Acceder" y "Registrar ID de afiliación".

Vigencia del Dato

El usuario y contraseña se mantienen vigentes hasta que el usuario decida cambiar su información de inicio de sesión en la configuración. El ID de afiliación se utiliza de forma continua para rastrear y gestionar las campañas de marketing.

Origen de los Datos

Los datos son ingresados directamente por el usuario durante el proceso de inicio de sesión o al registrar un nuevo ID de afiliación.

Operaciones Esperadas

- El usuario puede ingresar su nombre de usuario y contraseña, luego hacer clic en "Acceder" para entrar a la app.
- Si no tiene un ID de afiliación, puede seleccionar la opción de Registrar ID de afiliación, lo que lo llevará a un flujo de registro donde obtendrá su ID único.
- La aplicación verifica las credenciales ingresadas y, en caso de ser correctas, otorga acceso a las funcionalidades de la plataforma.

Servicios, Frameworks y Kits de Desarrollo

- React Native: Utilizado para la creación de la interfaz de usuario responsiva y adaptable tanto para iOS como para Android.
- Autenticación con Firebase: Implementación de un sistema seguro para gestionar el inicio de sesión y el registro.
- Backend en Node.js: Utilizado para la gestión de los datos de los usuarios, incluyendo la validación del ID de afiliación y las credenciales.

Justificación de Viabilidad

Etapas del proceso de autenticación, donde los usuarios que ya tienen una cuenta pueden ingresar sus credenciales para acceder a la plataforma.

Funcionalidad Implementada

La pantalla permite al usuario ingresar su correo electrónico y su nombre de usuario antes de presionar el botón "Acceder" para entrar en la app. Este método garantiza que solo los usuarios registrados previamente puedan utilizar las funciones exclusivas de la plataforma, como la gestión de campañas de afiliación y el análisis de resultados.

Información y Tipo de Datos

- Email: Campo de texto para que el usuario ingrese su dirección de correo electrónico registrada.
- Usuario: Campo de texto para ingresar el nombre de usuario asociado a su cuenta.
- Botón interactivo: "Acceder", que procesa la validación de las credenciales.

Bienvenido
de vuelta

Email

Usuario

Acceder

— o —

Registrar ID de afiliación

- Enlace para registrar un ID de afiliación: Ubicado en la parte inferior, que redirige a un flujo de registro si el usuario aún no tiene sus credenciales.

Vigencia del Dato

Los datos de inicio de sesión (email y nombre de usuario) se mantienen vigentes hasta que el usuario decida actualizar su información de cuenta en la configuración. La información se utiliza únicamente para verificar la identidad del usuario en el momento del acceso.

Origen de los Datos

Los datos son ingresados manualmente por el usuario cada vez que inicia sesión en la aplicación.

Operaciones Esperadas:

- Entrada de texto: El usuario debe completar ambos campos con su email y nombre de usuario.
- Acción del botón "Acceder": Al hacer clic, la aplicación valida las credenciales ingresadas. Si son correctas, el usuario será redirigido al dashboard de la app.
- Enlace de registro: Si el usuario no cuenta con las credenciales, puede hacer clic en "Registrar ID de afiliación" para iniciar el proceso de registro.

Servicios, Frameworks y Kits de Desarrollo

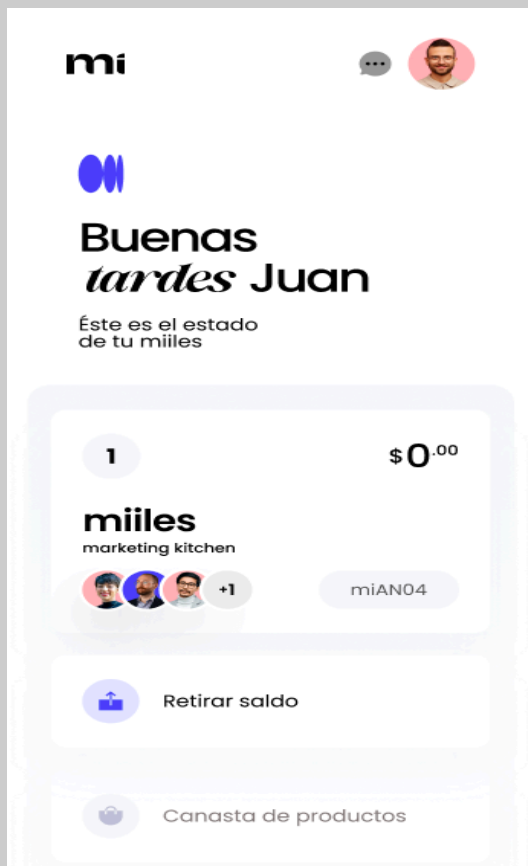
- React Native: Utilizado para diseñar esta interfaz de usuario responsiva.
- Firebase Authentication: Gestiona el inicio de sesión y la verificación de credenciales, ofreciendo un sistema seguro y fiable.
- Backend en Node.js: Interactúa con la base de datos para validar que el email y el usuario coincidan con una cuenta registrada.

Justificación de Viabilidad

Proporciona a los usuarios una visión de su estado actual dentro de la plataforma. Al mostrar el saldo generado por las campañas publicitarias, el ID del usuario y las opciones para retirar fondos o gestionar productos, se garantiza que los usuarios puedan monitorear y gestionar sus ingresos de forma eficiente.

Funcionalidad Implementada

- Un saludo con el nombre del usuario.
- El estado actual de las ganancias generadas por las campañas, reflejadas en un saldo disponible.
- Botones de acción como "Retirar saldo" y "Canasta de productos" para gestionar sus fondos y productos afiliados.
- Información del ID de afiliación del usuario, junto con una breve explicación de cómo está compuesto el ID.



Información y Tipo de Datos

- Foto de perfil: Imagen del usuario para personalizar la experiencia.
- Nombre del usuario: Cadena de texto que muestra el nombre ingresado durante el registro.
- Saldo disponible: Monto en dólares que representa las comisiones generadas.

- Retirar saldo: Permite al usuario transferir sus ganancias a su cuenta bancaria.
- Canasta de productos: Accede a los productos que el usuario está promocionando.
- Mi ID: Código único compuesto por un prefijo, letras y números.

Vigencia del Dato

Los datos del saldo y las campañas se actualizan en tiempo real, reflejando las ganancias acumuladas al momento. El Mi ID es único para cada usuario y permanece constante mientras el usuario esté registrado en la plataforma.

Origen de los Datos

Los datos son generados por las actividades del usuario en la plataforma, como las campañas de marketing de afiliados y las transacciones de productos. El Mi ID es asignado automáticamente al usuario al momento de su registro.

Operaciones Esperadas

- El usuario puede ver su saldo actual y decidir retirar fondos haciendo clic en el botón.
- Puede acceder a la canasta de productos para revisar o gestionar los productos que está promocionando.
- El usuario puede ver y comprender la composición de su ID.

Servicios, Frameworks y Kits de Desarrollo:

- React Native: Utilizado para desarrollar la interfaz de usuario dinámica y responsiva.
- Node.js y MongoDB: Gestionan los datos relacionados con las ganancias, retiros y productos afiliados.
- AWS Lambda: Para procesar las solicitudes de retiro de saldo y sincronizar la información en tiempo real.
- Firebase Storage: Almacena y gestiona las imágenes de perfil de los usuarios.



Justificación de Viabilidad:

Esta pantalla es esencial para mejorar la experiencia del usuario, permitiéndole acceder de manera a opciones de configuración y soporte.

Funcionalidad Implementada:

La pantalla incluye varias opciones que permiten al usuario gestionar distintos aspectos de su cuenta y resolver dudas:

- Volver al inicio: Redirige al usuario al dashboard principal.

- Ayuda: Proporciona acceso a un centro de soporte con respuestas a preguntas frecuentes y recursos de autoayuda.
- Tu perfil: Permite al usuario editar información personal, como su nombre, foto de perfil.
- Seguridad: Brinda acceso a opciones para cambiar la contraseña y actualizar la CLABE Bancaria para el retiro de fondos.
- Cerrar sesión: Permite al usuario salir de la aplicación de forma segura.

Información y Tipo de Datos:

- Volver al inicio: Acción que redirige al dashboard.
- Ayuda: No requiere entrada de datos, solo acceso a contenido informativo.
- Tu perfil: Permite la edición de información personal.
- Seguridad: Campos para cambiar contraseña y actualizar la CLABE.
- Cerrar sesión: Acción que no requiere datos, solo confirma la salida.

Vigencia del Dato:

La información editada en las secciones de "Tu perfil" y "Seguridad" se actualiza de inmediato y permanece vigente hasta que el usuario decida realizar otro cambio. La opción de cerrar sesión invalida la sesión actual, protegiendo la seguridad del usuario.

Origen de los Datos:

Los datos mostrados provienen del perfil del usuario registrado en la base de datos de MILES. La edición de información se realiza directamente en la aplicación y se sincroniza con el backend.

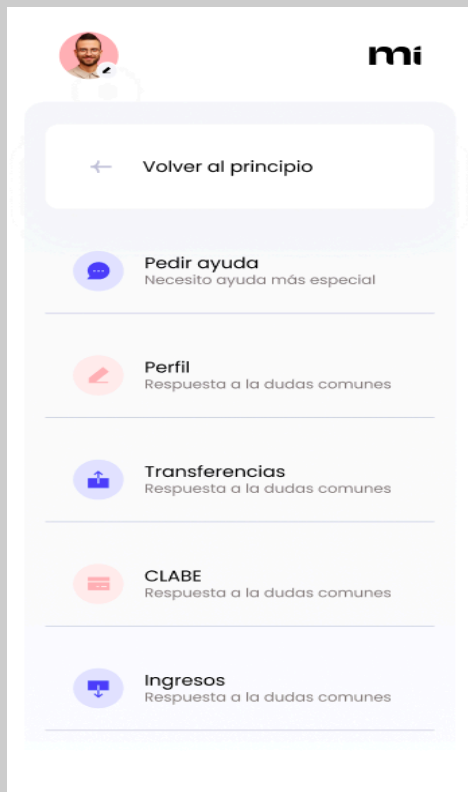
Operaciones Esperadas:

- Volver al inicio: El usuario regresa al dashboard con un solo clic.
- Ayuda: El usuario accede a artículos y guías para resolver dudas.
- Tu perfil: El usuario puede editar y guardar su información personal.
- Seguridad: El usuario puede cambiar su contraseña y la CLABE para el retiro de fondos.
- Cerrar sesión: Al seleccionar esta opción, la aplicación termina la sesión y regresa a la pantalla de inicio de sesión.

Servicios, Frameworks y Kits de Desarrollo:

- React Native: Utilizado para desarrollar la interfaz del menú.
- Node.js y MongoDB: Gestionan la actualización de datos del perfil y la seguridad.

- Firebase Authentication: Garantiza que el proceso de cambio de contraseña sea seguro.
- AWS Lambda: Facilita la gestión de las configuraciones y el cierre de sesión.



Justificación de Viabilidad:

Esta pantalla es crucial para permitir que los usuarios tengan un control total sobre su cuenta, ajustando sus datos personales y gestionando la configuración relacionada con transferencias y datos bancarios.

Funcionalidad Implementada:

- Pedir ayuda: Redirige al centro de soporte donde el usuario puede buscar asistencia en caso de tener problemas más específicos.
- Perfil: Permite al usuario editar su información personal como nombre, foto de perfil, y otros datos de contacto.
- Transferencias: Muestra información y respuestas a dudas comunes sobre la gestión de transferencias realizadas a la cuenta del usuario.

- CLABE: Permite al usuario actualizar su CLABE para asegurar que las transferencias de sus comisiones se realicen correctamente.
- Ingresos: Proporciona respuestas y detalles sobre cómo se gestionan las ganancias generadas a través de la plataforma.

Información y Tipo de Datos:

- Datos personales (perfil): Incluyen nombre, foto de perfil, y otros datos relevantes que el usuario puede actualizar.
- Información bancaria: Campo para la actualización del número de cuenta bancaria necesario para procesar los retiros de saldo.
- Transferencias e ingresos: Información basada en los datos históricos del usuario en la plataforma.

Vigencia del Dato:

Los cambios realizados en esta pantalla se reflejan inmediatamente en el sistema y se mantienen hasta que el usuario decida actualizarlos nuevamente. Esto incluye actualizaciones en la CLABE, perfil, y preferencias de contacto.

Origen de los Datos:

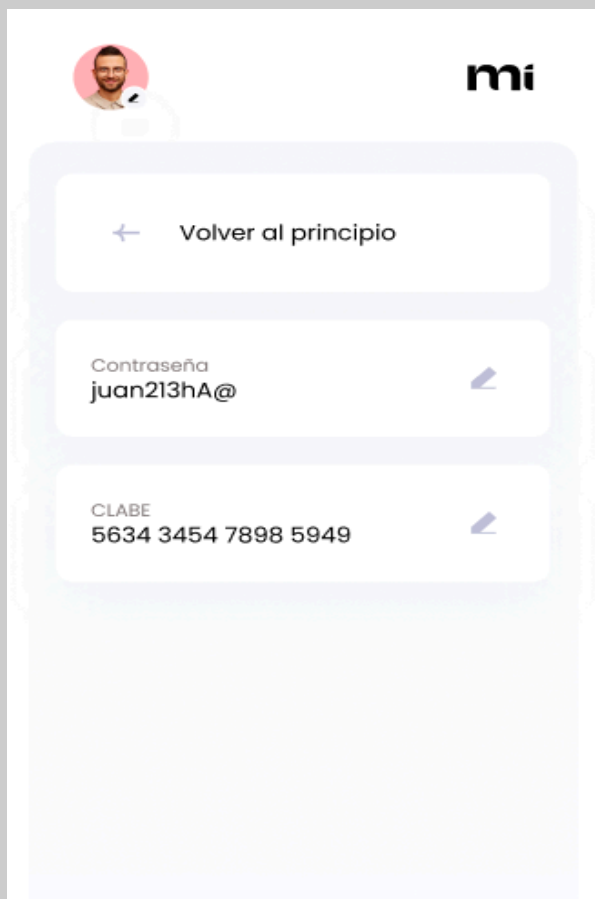
La información es proporcionada directamente por el usuario al completar o editar sus datos.

Operaciones Esperadas

- Edición de perfil: El usuario puede hacer clic en los campos para editarlos. Una vez que hace clic en el ícono de edición (lápiz), los campos se vuelven editables. Para confirmar los cambios, puede hacer clic en el ícono de verificación (check).
- Pedir ayuda: El usuario puede acceder a artículos y guías para resolver dudas.
- Actualizar CLAVE: Permite modificar el número de cuenta para los retiros de fondos.
- Gestión de transferencias: Proporciona acceso a las transferencias en la plataforma.
- Ver ingresos: Permite al usuario entender mejor cómo se generan y se gestionan sus comisiones.

Servicios, Frameworks y Kits de Desarrollo

- React Native: Desarrolla la interfaz y permite la edición dinámica de los campos.
- Node.js y MongoDB: Manejan la actualización y almacenamiento de datos del perfil y las configuraciones bancarias.
- Firebase Authentication: Asegura que solo los usuarios autenticados puedan modificar su información personal.
- AWS Lambda: Gestiona en tiempo real las actualizaciones de perfil y datos bancarios.



Seguridad

Justificación de Viabilidad

La pantalla aborda la gestión de datos sensibles relacionados con seguridad y pagos en una aplicación de marketing de afiliados.

Funcionalidad Implementada

Gestión de contraseña, Visualización de CLABE, Opción para regresar al menú principal o inicio de la app.

Datos Gestionados

Texto alfanumérico, cifrado en backend, Numérico, 18 dígitos estándar en México (CLABE).

Origen de los Datos

Contraseña: Creada y gestionada por el usuario; almacenada encriptada.

CLABE: Ingresada manualmente, validable mediante APIs bancarias.

Operaciones Clave

Visualización: Consultar información de CLABE y contraseña.

Edición: Actualizar ambas mediante controles intuitivos.

Validación: Verificación de la CLABE y otros datos críticos.

Seguridad: Autenticación, notificaciones de cambios y protección contra accesos no autorizados.

Herramientas y Tecnologías

- React Native, Flutter.
- Material UI, Ant Design.

Chat

Justificación de viabilidad:

Permitir la comunicación en tiempo real entre el usuario y el soporte

Funcionalidad implementada:

Chat en tiempo real donde los usuarios pueden enviar y recibir mensajes del equipo de soporte.

Información y tipo de datos:

Mensajes de texto (strings) enviados por el usuario y el soporte.

Vigencia del dato:

La vigencia es en tiempo real

Origen de los datos:

Los datos son proporcionados directamente por el usuario y el equipo de soporte.

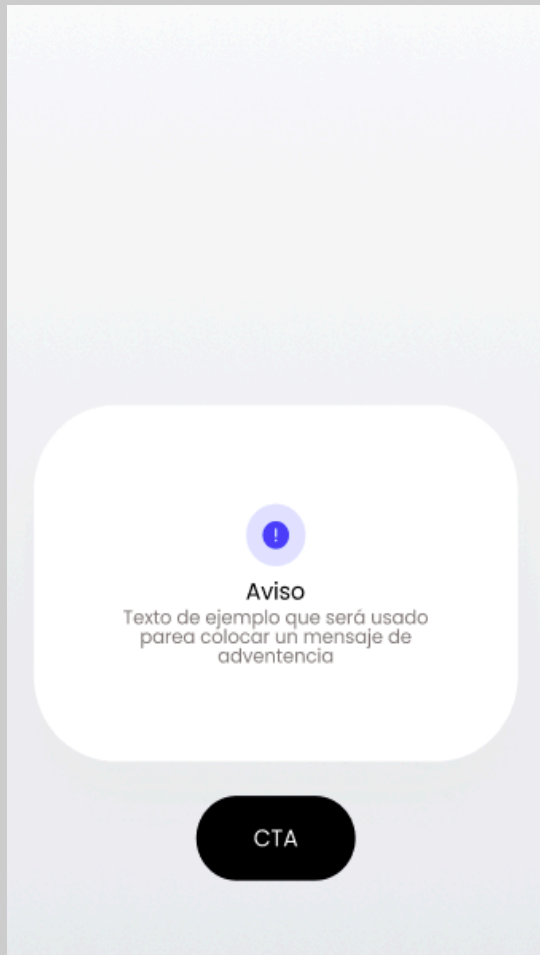
Operaciones esperadas:

Envío y recepción de mensajes, notificación de mensajes nuevos, carga de historial de conversación.

Servicios, frameworks y kits de desarrollo:

- Firebase Realtime Database o Firestore
- Firebase Cloud Messaging para notificaciones en tiempo real.
- React Native o Flutter





Pop-Up

Justificación de viabilidad:

Las ventanas emergentes son efectivas para informar rápidamente al usuario sobre situaciones importantes.

Funcionalidad implementada:

Muestra un mensaje informativo o de advertencia breve, con un botón de llamada a la acción (CTA) para responder o cerrar el pop-up.

Información y tipo de datos:

Texto de aviso (string) que explica la advertencia o notificación para el usuario.

Vigencia del dato:

Datos efímeros, diseñados para mostrarse en una sola ocasión o mientras dure la relevancia del aviso.

Origen de los datos:

Los datos provienen de la base de datos del sistema

Operaciones esperadas:

Mostrar y cerrar el pop-up, y permitir que el usuario interactúe con el botón CTA para continuar o realizar alguna acción específica.

Servicios, frameworks y kits de desarrollo:

- React Native o Flutter para el pop-up.
- Firebase Remote Config puede ser útil para gestionar avisos sin necesidad de actualizar la app.
- Expo y Material Design o Cupertino widgets en Flutter ayudan a crear pop-ups visualmente consistentes.



Loading

Justificación de viabilidad:

Una pantalla de carga mejora la experiencia del usuario, indicándole que la app está procesando una solicitud.

Funcionalidad implementada:

Muestra una animación de carga con el logotipo de la aplicación mientras se carga contenido o se realiza una solicitud en segundo plano.

Información y tipo de datos:

No se manejan datos específicos en esta pantalla; solo se muestra una animación visual.

Vigencia del dato:

La animación dura mientras se carga la información requerida.

Origen de los datos:

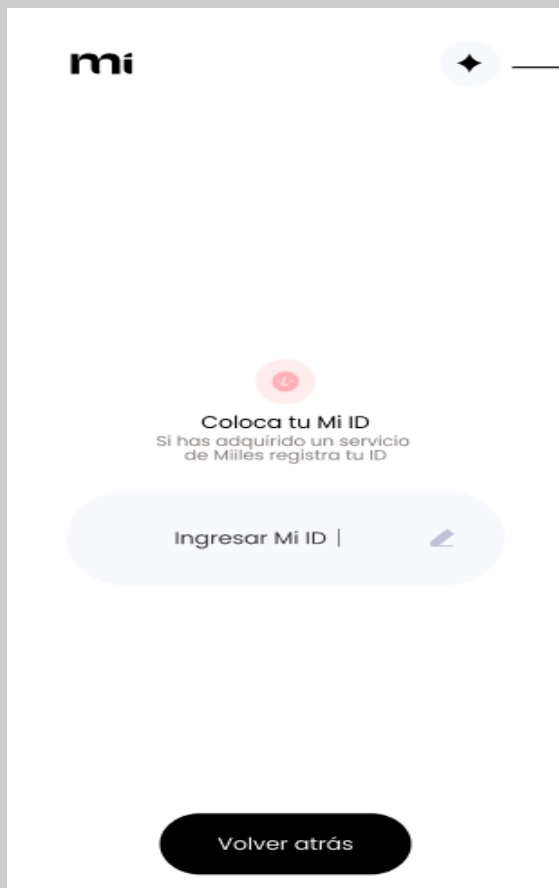
No hay datos externos; la animación se gestiona localmente en el dispositivo.

Operaciones esperadas:

Mostrar la animación de carga y ocultarla una vez que los datos se hayan cargado completamente.

Servicios, frameworks y kits de desarrollo:

- Lottie para animaciones
- React Native o Flutter para controlar la animación durante el proceso de carga.

**Mi ID****Justificación de viabilidad:**

Permitir el ingreso de un "Mi ID" ayuda a vincular la cuenta del usuario con el servicio de afiliación.

Funcionalidad implementada:

Campo de texto para ingresar el ID de usuario único. Incluye un botón que envía este ID a la web para su registro.

Información y tipo de datos:

ID del usuario (string o integer), que sirve como identificador único del servicio contratado.

Vigencia del dato:

Los datos tienen vigencia permanente, ya que el ID se mantiene mientras el usuario tenga activo el servicio.

Origen de los datos:

El "Mi ID" proviene del usuario y debe ser validado con una base de datos o API que confirme su existencia y validez.

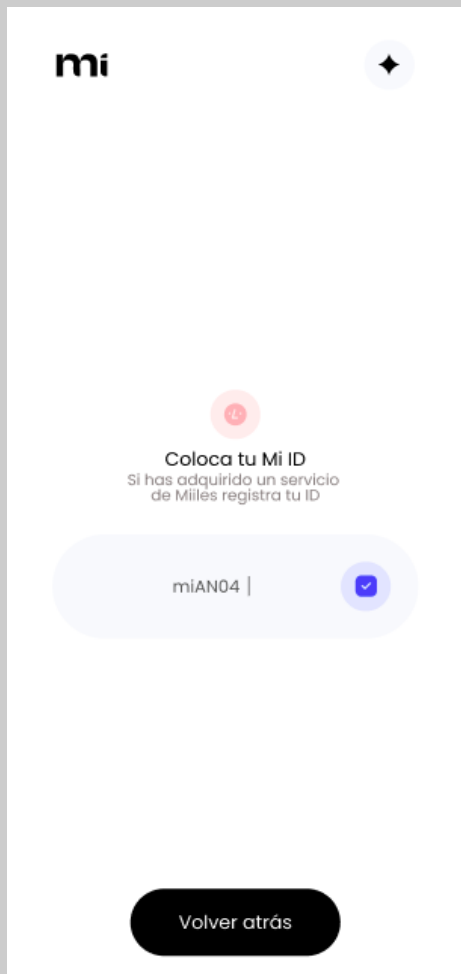
Operaciones esperadas:

Ingreso y validación del ID, envío a un servidor externo para asociar el ID con la cuenta de usuario.

Servicios, frameworks y kits de desarrollo:

- Firebase Authentication o OAuth
- Axios

- React Native o Flutter



Registro de ID

Justificación de Viabilidad: El uso de identificadores únicos facilita la personalización y seguimiento del usuario.

Funcionalidad Implementada:

- Ingresar y guardar el ID del usuario.

Información y Tipo de Datos:

- Campo de entrada para ID (cadena de texto).
- Botón de confirmación.

Vigencia del Dato: Permanente hasta que el usuario decida modificarlo.

Origen de los Datos: Ingresados por el usuario.

Operaciones Esperadas:

- El usuario introduce su ID.
- El sistema almacena el dato de forma segura.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Manejo eficiente de datos de usuario.



Registro Exitoso

Justificación de Viabilidad: Notificar al usuario sobre el éxito de una acción mejora la experiencia y da claridad.

Funcionalidad Implementada:

- Mostrar mensaje de confirmación.

Información y Tipo de Datos:

- Mensaje estático.
- Botón de navegación.

Vigencia del Dato: Solo mientras se visualiza la pantalla.

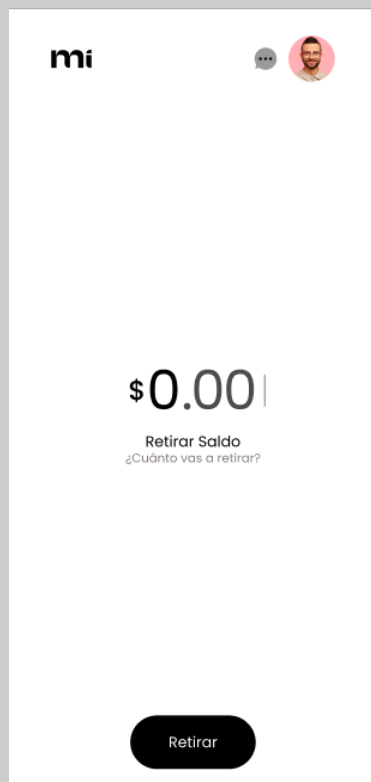
Origen de los Datos: Generado automáticamente tras la acción.

Operaciones Esperadas:

- Mostrar el mensaje.
- Usuario regresa al menú principal.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Renderizado rápido de mensajes estáticos.



Retiro con valor cero y mil

Justificación de Viabilidad: Manejar escenarios de error o valores inválidos asegura una experiencia robusta.

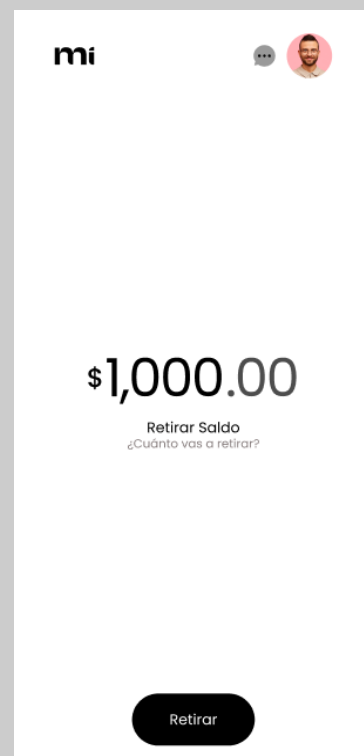
Funcionalidad Implementada:

- Validar entradas y mostrar mensajes acorde.

Información y Tipo de Datos:

- Mensaje de error.
- Entrada de monto (dato numérico).

Vigencia del Dato: Hasta que el usuario corrija el error.



Origen de los Datos: Ingresados por el usuario.

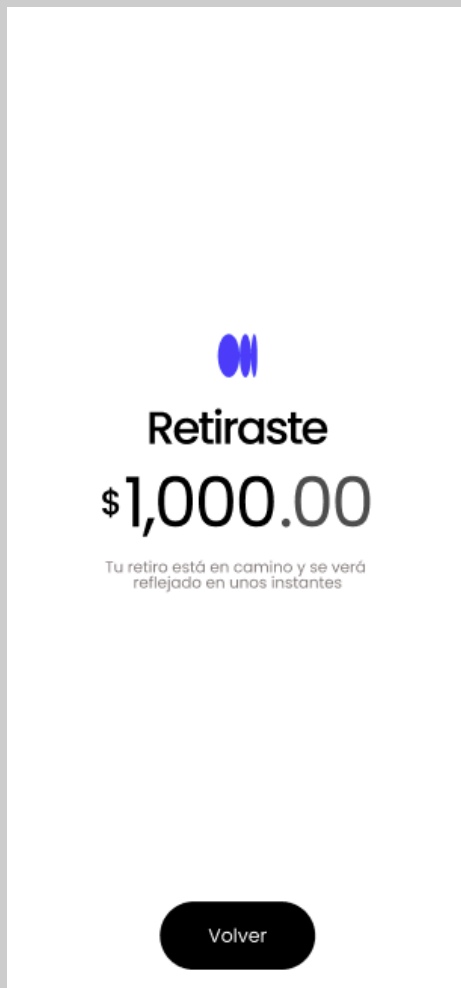
Operaciones Esperadas:

- El sistema valida el monto ingresado.
- Se notifica al usuario si el valor es incorrecto.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Manejo de validaciones y mensajes en tiempo real.

Confirmación de Retiro



Justificación de Viabilidad: Proporcionar confirmación visual refuerza la confianza del usuario en las transacciones.

Funcionalidad Implementada:

- Confirmar y visualizar los detalles de la transacción.

Información y Tipo de Datos:

- Monto retirado (dato numérico).
- Mensaje informativo.

Vigencia del Dato: Permanente para ese registro, accesible en el historial.

Origen de los Datos: Generados automáticamente tras la acción del usuario.

Operaciones Esperadas:

- Mostrar los detalles del retiro.
- Usuario regresa al menú principal.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Generación y presentación de datos transaccionales de manera dinámica.

Configuración (Activar notificaciones)

Justificación de Viabilidad: Permitir a los usuarios personalizar la experiencia aumenta el nivel de satisfacción y control sobre la app.

Funcionalidad Implementada:

- Habilitar o deshabilitar notificaciones.

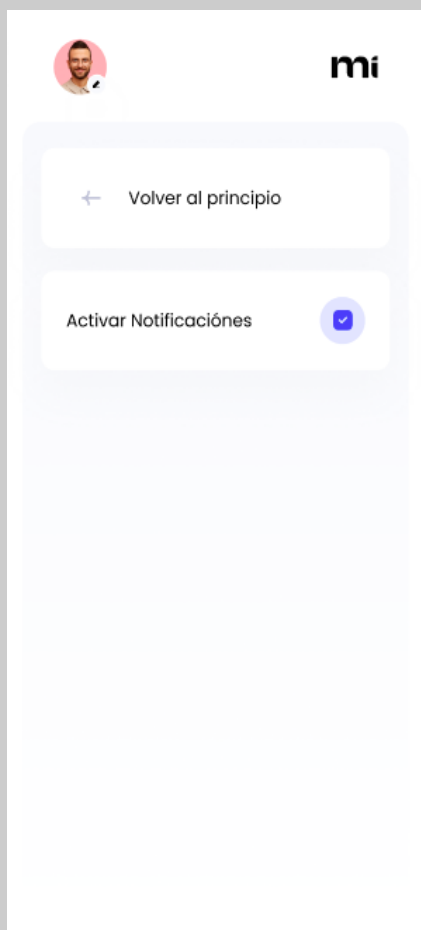
Información y Tipo de Datos:

- Estado de notificaciones (booleano: activado/desactivado).
- Botón de navegación para volver.

Vigencia del Dato: Hasta que el usuario realice un cambio en la configuración.

Origen de los Datos: Ingresados por el usuario.

Operaciones Esperadas:



- El usuario selecciona activar/desactivar.
- La configuración se guarda automáticamente.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Soporte para componentes interactivos y persistencia de datos en tiempo real.



Promoción de boletos

Justificación de Viabilidad: Presentar ofertas claras y llamativas es esencial para captar la atención de los usuarios y fomentar su interés en la aplicación.

Funcionalidad Implementada:

- Mostrar promociones disponibles de manera visual y atractiva.
- Fomentar la acción del usuario con precios destacados.

Información y Tipo de Datos:

- Texto informativo (ofertas y detalles).
- Elementos visuales (colores llamativos y formatos).
- Precio como dato numérico.

Vigencia del Dato: Hasta que la promoción sea modificada o finalizada.

Origen de los Datos: Ingresados por el equipo de contenido o sistema de promociones.

Operaciones Esperadas:

- El usuario visualiza la promoción.
- Puede decidir continuar explorando o tomar acción.

Servicios, Frameworks y Kits de Desarrollo:

- Framework: React Native.
- Justificación: Ideal para renderizar contenido dinámico y atractivo en múltiples plataformas.

Explicación del flujo de recorrido de las pantallas

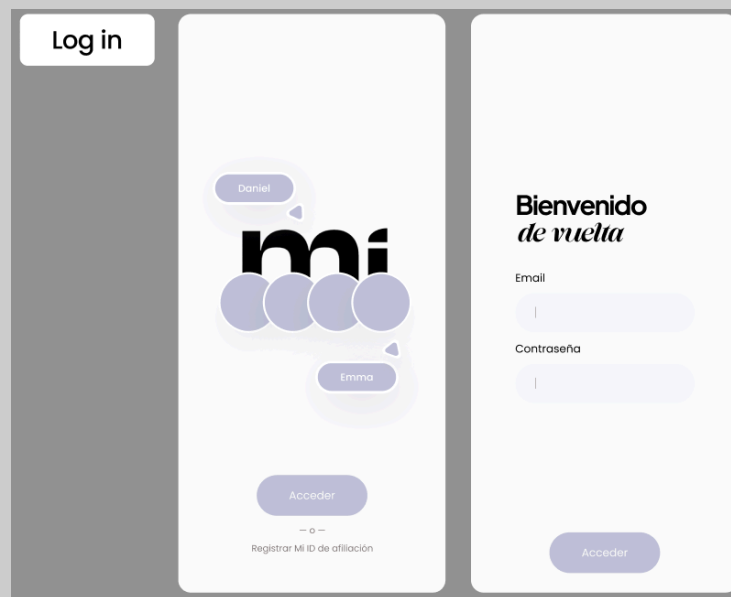
1. Pantalla de Inicio – "Log In"

- **Propósito:** Es la pantalla inicial que introduce al usuario a la app. Muestra opciones claras para iniciar sesión o registrarse.
- **Acciones esperadas del usuario:**

- Seleccionar su perfil (en caso de usuarios ya registrados, como "Daniel" o "Emma").
- Pulsar el botón "Acceder" para iniciar sesión.
- Alternativamente, puede seleccionar "Registrar Mi ID de afiliación" para iniciar el proceso de registro.

2. Pantalla de Bienvenida – "Bienvenido de vuelta"

- **Propósito:** Esta pantalla solicita las credenciales (email y contraseña) para validar al usuario y permitirle ingresar a la plataforma.
- **Acciones esperadas del usuario:**
 - Ingresar su email y contraseña en los campos correspondientes.
 - Pulsar el botón "Acceder" para proceder al sistema.
- **Validaciones incluidas:**
 - Confirmar que el email tiene un formato válido.
 - Validar que la contraseña cumple con las políticas de seguridad de la app.



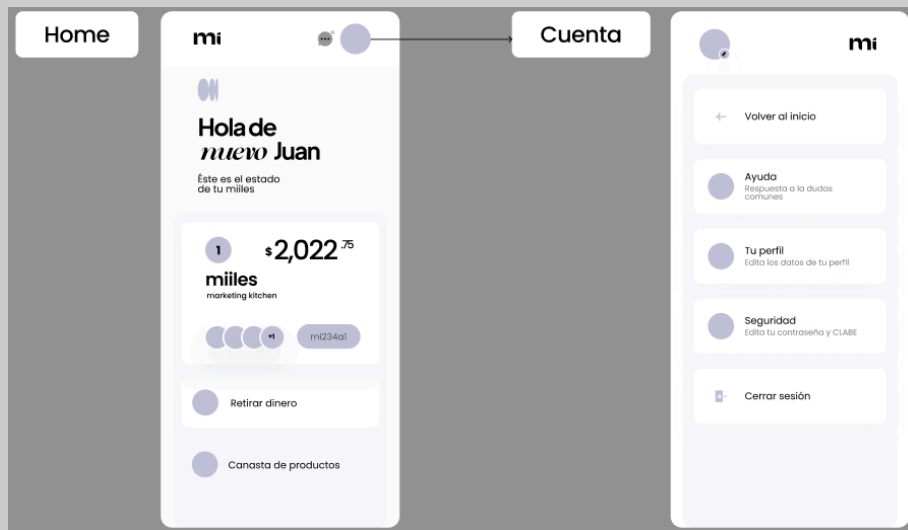
3. Pantalla: Home

- **Propósito:** Mostrar el estado actual del usuario en la plataforma (saldo, ID de afiliación, opciones rápidas).
- **Navegación:**
 - Botón para acceder a la **Pantalla Cuenta**.
 - Opciones directas para "Retirar dinero" o gestionar la "Canasta de productos".

4. Pantalla: Cuenta

- **Propósito:** Permitir al usuario acceder a configuraciones y ajustes de su cuenta.
- **Navegación:**
 - Opción "Volver al inicio" para regresar a la pantalla **Home**.
 - Opciones para:
 1. **Ayuda:** Accede a información y soporte.
 2. **Tu perfil:** Modifica datos personales.
 3. **Seguridad:** Gestiona contraseña y datos bancarios.

4. "Cerrar sesión".



5. Pantalla: Ayuda

- **Propósito:** Proporcionar soporte con preguntas frecuentes y acceso a ayuda personalizada.
- **Navegación:**
 - Botón "Volver al principio" para regresar a la pantalla **Cuenta**.
 - Opciones internas como:
 1. "Pedir ayuda" para contacto directo.
 2. Información sobre perfil, transferencias, CLABE e ingresos.

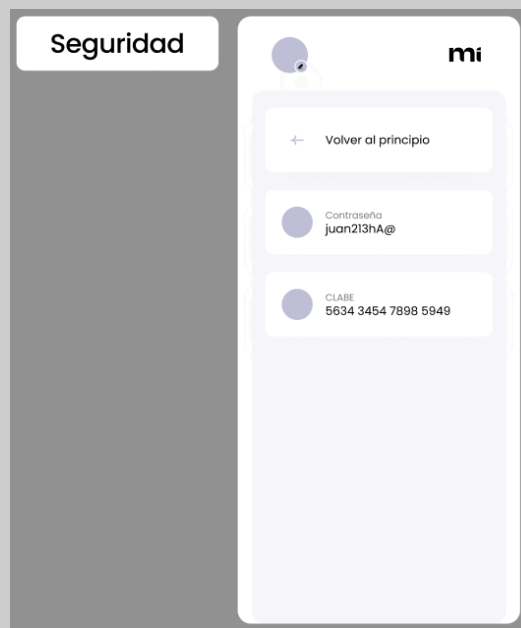
6. Pantalla: Tu perfil

- **Propósito:** Permitir al usuario visualizar y editar su información personal (nombre, email y teléfono).
- **Navegación:**
 - Botón "Volver al principio" para regresar a la pantalla **Cuenta**.



7. Pantalla: Seguridad

- **Propósito:** Permitir la gestión de datos sensibles como la contraseña y la CLABE.
- **Navegación:**
 - Botón "Volver al principio" para regresar a la pantalla **Cuenta**.

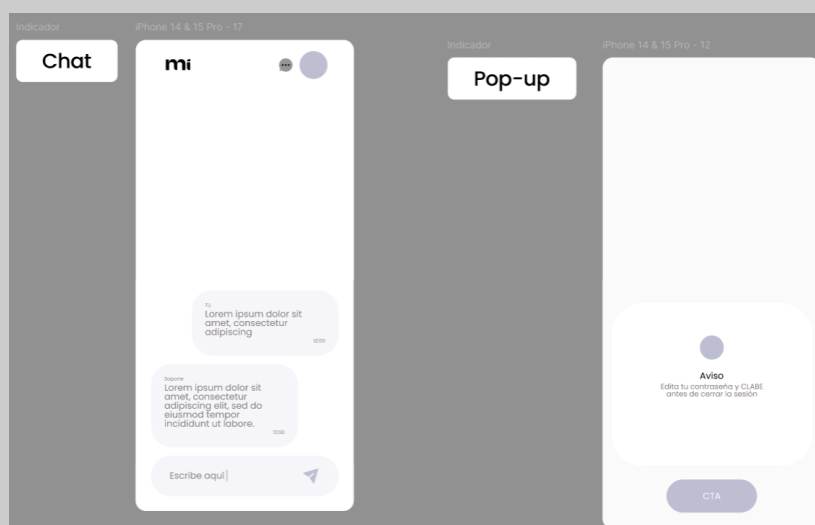


8. Pantalla: Chat

- **Propósito:** Ofrecer soporte directo mediante un chat interactivo en tiempo real.
- **Navegación:**
 - Botón para salir o regresar al flujo principal desde cualquier vista de soporte.

7. Pantalla: Pop-Up

- **Propósito:** Mostrar mensajes o alertas contextuales sobre acciones importantes (por ejemplo, recomendaciones antes de cerrar sesión).
- **Navegación:**
 - Botón de Call To Action (CTA) que redirige al usuario según el contexto.

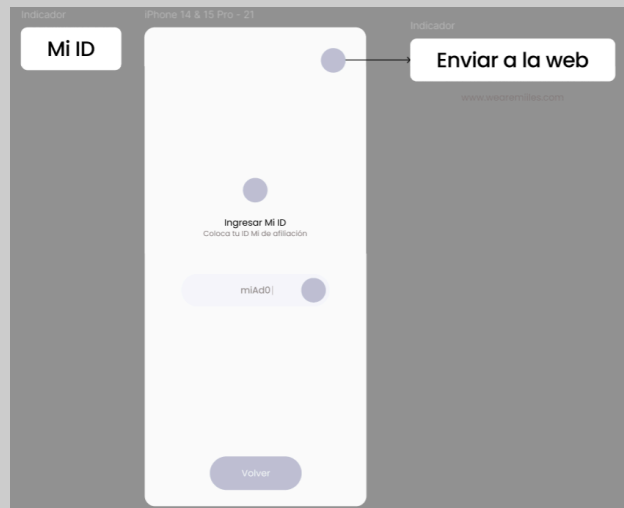


8. Pantalla: Loading

- **Propósito:** Proporcionar feedback visual al usuario durante tiempos de espera en el sistema.
- **Navegación:**
 - Automática al completar la carga hacia la pantalla correspondiente (ejemplo: **Home** o **Cuenta**).

9. Pantalla: Mi ID

- **Propósito:** Permitir al usuario ingresar su ID único para registro o afiliación.
- **Navegación:**
 - Botón "Volver" para regresar al flujo principal.



Detalles de los lenguajes de programación, permisos y adecuaciones para su publicación

Lenguajes de Programación y Herramientas de Desarrollo

Para el desarrollo de la app **Miiles**, utilizamos tecnologías que permiten crear aplicaciones móviles multiplataforma, asegurando compatibilidad con **iOS y Android** y optimizando el tiempo de desarrollo. Los detalles técnicos son los siguientes:

- **Frontend móvil:**
 - Se emplea **React Native o Flutter**, tecnologías que facilitan el desarrollo de aplicaciones de alto rendimiento para ambos sistemas operativos desde una sola base de código, acelerando el tiempo de implementación.
- **Backend:**
 - Usamos **Node.js o Python** para gestionar la comunicación entre la base de datos y la aplicación móvil de manera eficaz.
 - **Firebase o MongoDB** son las opciones preferidas para la base de datos, permitiendo un almacenamiento seguro y escalable para los datos de usuarios y transacciones.
- **Autenticación y seguridad:**
 - Se implementa **OAuth 2.0** y **JWT (JSON Web Tokens)** para una autenticación segura de los usuarios.

- También consideramos **Firestore Authentication** o **Auth0** como opciones para gestionar de manera segura los inicios de sesión y permisos de acceso.

Herramientas adicionales de desarrollo:

- **GitHub o GitLab** para el control de versiones.
- **Postman** para probar y validar las API del backend.
- **Google Analytics o Mixpanel** para el análisis de métricas de uso de la app.

Permisos y Adecuaciones para la Publicación en Tiendas de Apps

Para asegurar que Miiles funcione adecuadamente y cumpla con los requisitos de las tiendas de apps, se han considerado los siguientes permisos y normativas de privacidad:

1. Permisos requeridos:

- **Acceso a Internet** para que la app pueda conectarse con los servidores y procesar las transacciones en tiempo real.
- **Notificaciones push** para mantener a los usuarios informados sobre nuevas oportunidades y actualizaciones importantes.
- **Acceso al almacenamiento local** (en caso de ser necesario) para almacenar datos temporales y mejorar la experiencia de usuario.

2. Adecuaciones y requisitos para Google Play y App Store:

- **Política de privacidad:** Ambas plataformas exigen una política de privacidad que informe claramente a los usuarios sobre cómo se recopilan, utilizan y protegen sus datos personales.
- **Cumplimiento de GDPR y CCPA:** La app cumple con los requisitos de protección de datos, como GDPR para usuarios en Europa y CCPA para usuarios en California, asegurando la privacidad de los datos personales.
- **Clasificación de contenido y categorización adecuada:** Se asegura que la app esté correctamente clasificada en las tiendas para facilitar su descubrimiento por el público objetivo.

Icono

Texto

Texto de ejemplo

Texto de ejemplo

x22

x18

x12

Texto de ejemplo

Boton

CTA

Contenedor

Volver al inicio

Texto de ejemplo

Texto de ejemplo

Cerrar sesión

Texto de ejemplo

Texto de ejemplo

Contenedor editable

Campo

Campo editable

Campo

Campo editable

Campo

Campo

Mi ID

Ingresar Mi ID

Chat

Escribe aquí

24

mi

Volver al inicio

Ayuda

Respuesta a la dudas comunes

Tu perfil

Edita los datos de tu perfil

Seguridad

Edita tu contraseña y CLABE

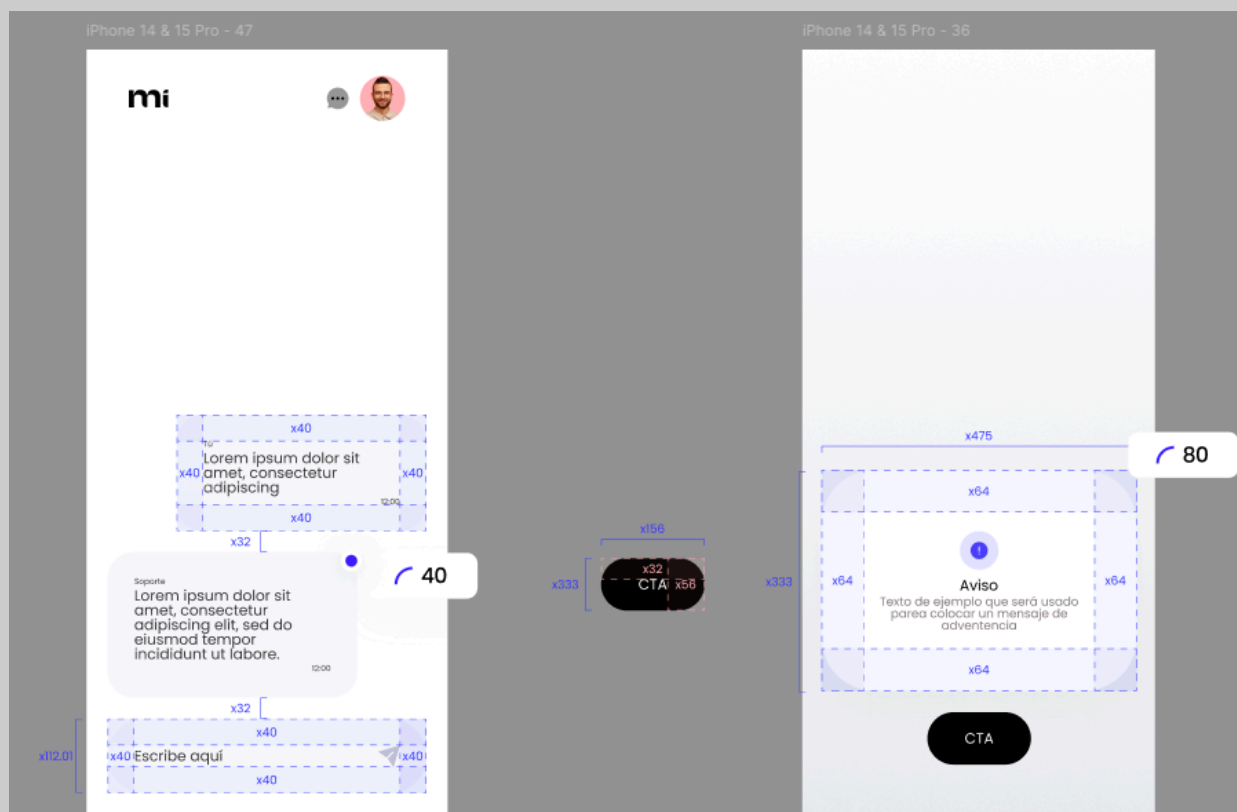
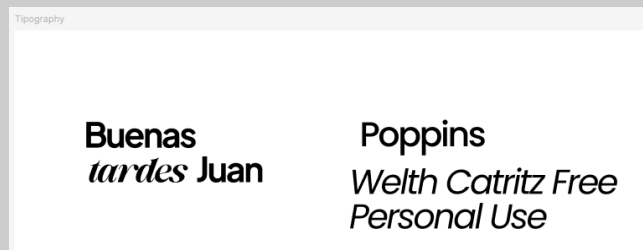
Notificaciones

Configura las notificaciones

Cerrar sesión

16

26



Tiempo de trabajo y Equipo de trabajo

Equipo de Trabajo y Roles

El desarrollo y la implementación de la app **Miiles** están a cargo de un equipo inicial de tres personas, con una visión de crecimiento a medida que la plataforma avance. Los roles principales dentro de este equipo incluyen:

- **Desarrollador Frontend:** Responsable de la creación de la interfaz móvil de la app y de garantizar una experiencia de usuario óptima, implementando diseño y funcionalidad en **React Native** o **Flutter**.
- **Desarrollador Backend:** Encargado de diseñar y mantener el servidor, bases de datos y API necesarias para la funcionalidad de la app, asegurando seguridad y eficiencia en el manejo de datos mediante **Node.js** o **Python** y **Firebase** o **MongoDB**.
- **Líder de Proyecto / Estratega de Producto:** Responsable de coordinar el desarrollo, definir las prioridades, realizar pruebas de usabilidad y planificar las mejoras y expansiones futuras de la plataforma, asegurando que se cumplan los objetivos de negocio.

A medida que el proyecto crezca, se planea la integración de más desarrolladores y especialistas en áreas como marketing digital, análisis de datos y soporte técnico.

Estimaciones de Tiempo de Desarrollo y Costos

Para lanzar una primera versión funcional de Miiles, estimamos las siguientes fases y tiempos de desarrollo, considerando que el equipo inicial de tres personas se mantiene enfocado en tareas críticas. Los tiempos se ajustarán y se ampliará el equipo conforme se incluyan nuevas funcionalidades y se aumente el alcance.

1. Desarrollo inicial (MVP – Producto Mínimo Viable):

- **Duración:** Aproximadamente 4-6 meses.
- **Actividades clave:** Desarrollo de la interfaz de usuario, configuración de backend, integración de API y sistemas de seguridad básicos, y pruebas iniciales de funcionalidad.
- **Costos estimados:** Entre \$10,000 y \$20,000 USD, considerando salarios del equipo, herramientas de desarrollo y recursos tecnológicos.

2. Implementación y lanzamiento en tiendas de apps:

- **Duración:** 1-2 meses adicionales para las pruebas finales y la publicación.
- **Costos estimados:** \$3,000 – \$5,000 USD adicionales, que incluyen la preparación para el lanzamiento en tiendas, licencias, y creación de materiales promocionales iniciales.

3. Mantenimiento y actualizaciones posteriores:

- **Duración:** Continuo, con un mantenimiento programado mensual.
- **Costos estimados mensuales:** \$1,000 – \$3,000 USD, considerando actualizaciones de seguridad, mejoras de funcionalidad y soporte técnico para los usuarios.

Conclusión

En este proyecto se diseñó una aplicación enfocada en la gestión de afiliaciones y finanzas personales, priorizando la experiencia del usuario mediante un flujo de pantallas intuitivo y organizado. Se incluyeron funcionalidades clave como la visualización del estado del usuario, opciones para retirar dinero, soporte a través de un chat en tiempo real y herramientas de personalización, como la edición de datos personales y configuraciones de seguridad.

El diseño de las pantallas garantiza simplicidad, accesibilidad y claridad, permitiendo que los usuarios interactúen fácilmente con la plataforma. Además, se priorizó la seguridad en el manejo de datos sensibles como contraseñas y CLABE, asegurando la confianza del usuario en el sistema. La inclusión de funciones como ayuda y notificaciones asegura un soporte constante, facilitando la resolución de dudas y mejorando la experiencia general.

En resumen, este proyecto representa un avance significativo en la creación de herramientas tecnológicas útiles para los usuarios, logrando un balance entre funcionalidad, diseño y seguridad. Este trabajo demuestra el valor de una buena planeación y diseño centrado en el usuario, cumpliendo con los objetivos establecidos de manera eficiente y profesional.

Referencias

- American Psychological Association. (2020). *Publication Manual of the American Psychological Association* (7.^a ed.). Washington, DC: American Psychological Association.
- Nielsen, J. (1994). *Usability engineering*. San Francisco, CA: Morgan Kaufmann Publishers.
- Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6.^a ed.). Pearson.
- Wroblewski, L. (2011). *Mobile first*. New York, NY: A Book Apart.
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9.^a ed.). McGraw-Hill Education.
- Cooper, A., Reimann, R., Cronin, D., Noessel, C., & Csizmadi, C. (2014). *About face: The essentials of interaction design* (4.^a ed.). Wiley.
- Krug, S. (2014). *Don't make me think, revisited: A common sense approach to web usability* (3.^a ed.). New Riders.
- ISO/IEC 27001. (2013). *Information technology — Security techniques — Information security management systems — Requirements*. Geneva, Switzerland: International Organization for Standardization.
- Preece, J., Sharp, H., & Rogers, Y. (2015). *Interaction design: Beyond human-computer interaction* (4.^a ed.). Wiley.