



**UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA Y CIENCIAS DE  
LA COMPUTACIÓN**



**Concepción y Desarrollo de aplicación móvil prototipo para la gestión cooperativa de finanzas en familia**

POR

**Cristian Emilio Contreras Moreno**

Memoria de Título presentada para optar al título profesional de Ingeniero Civil Informático

Profesor Guía  
Javier Vidal Valenzuela

Profesor Supervisor  
Erich Grüttner Díaz

Mayo 2023

© 2023 Cristian Emilio Contreras Moreno

© 2023 Cristian Emilio Contreras Moreno

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

## Resumen

La carencia de educación e inclusión financiera en familias chilenas, especialmente en niños y adolescentes, representa un desafío significativo en la sociedad actual. Este estudio aborda dicha problemática mediante la creación y desarrollo de un prototipo de aplicación móvil enfocado en la gestión cooperativa de finanzas familiares, con el propósito de potenciar las habilidades financieras de los menores en un entorno seguro. Se realizó una investigación en los contextos nacional e internacional para identificar oportunidades de mejora y definir el mercado objetivo. Basándose en estos hallazgos, se propuso una solución desarrollada con la metodología Canvas, que integra las mejores prácticas y tendencias internacionales, adaptándolas al escenario chileno. La aplicación móvil se implementó utilizando tecnologías punteras como React Native y AWS Amplify en una arquitectura serverless. Se diseñó una interfaz adaptativa para los distintos tipos de usuario, considerando las variaciones en la complejidad de gestos y navegación entre padres e hijos. Se establecieron casos de uso adecuados para garantizar una interacción segura, incorporando elementos de control parental y habilidades de gestión para los menores. Finalmente, se identifican áreas de trabajo futuro con el objetivo de fomentar la colaboración familiar y la integración tecnológica como medio de aprendizaje financiero práctico.

# Índice

<b>1. Introducción</b>	<b>5</b>
<b>2. Objetivos</b>	<b>7</b>
2.1 Objetivo General	7
2.2 Objetivos Específicos	7
<b>3. Problemática</b>	<b>8</b>
<b>4. Acceso a instrumentos financieros y uso de tecnología</b>	<b>9</b>
<b>5. Soluciones existentes</b>	<b>11</b>
5.1 Contexto Nacional	11
5.2 Contexto Internacional	13
<b>6. Mercado Objetivo</b>	<b>16</b>
6.1 Hijos	16
6.2 Padres	17
<b>7. Solución Propuesta</b>	<b>19</b>
7.1 Business Model Canvas	19
<b>8. Diseño e implementación del sistema</b>	<b>22</b>
8.1 Definición del sistema	23
8.2 Casos de uso	<b>23</b>
8.2.1 Diagrama de casos de uso	23
8.2.2 Padres	25
8.2.3 Hijos	26
8.3 Arquitectura de software	<b>27</b>
8.3.1 Elementos	29
8.3.2 Interacciones	31
8.4 Modelo de datos	<b>35</b>
8.4.1 Modelos	36
8.4.2 Tipos	37
8.5 Navegación móvil	<b>40</b>
8.6 Proceso de desarrollo	<b>43</b>
8.6.1 Repositorio Centralizado	44
8.6.2 Front-end	44
8.6.3 Back-end	47
<b>9. Presentación de la Implementación</b>	<b>48</b>
<b>10. Pruebas de casos de uso</b>	<b>51</b>
<b>11. Conclusiones</b>	<b>56</b>
<b>12. Trabajo futuro</b>	<b>57</b>
<b>Referencias</b>	<b>59</b>
<b>Anexos</b>	<b>61</b>



## 1. Introducción

La Organización para la Cooperación y el Desarrollo Económico (OCDE) define la educación financiera como "el conocimiento y las habilidades necesarias para tomar decisiones financieras informadas y responsables". Esto incluye el conocimiento de conceptos financieros básicos, como el ahorro, la inversión y el endeudamiento, así como la comprensión de cómo funcionan los mercados financieros y cómo utilizar adecuadamente productos y servicios financieros. (García, N., Grifoni, A., López, J. C., & Mejía, D. M., 2017).

La educación financiera es una habilidad clave que todo estudiante debe desarrollar para asegurar su futuro financiero. Por este motivo, la Organización para la Cooperación y el Desarrollo Económicos lanzó el proyecto Recomendación sobre los Principios y Buenas Prácticas de Educación y Concienciación Financiera, el cual establece estrategias para brindar a los estudiantes las habilidades y herramientas necesarias para tomar decisiones financieras informadas y responsables (OCDE, 2005).

Desde entonces, la OCDE ha ampliado su enfoque en educación financiera a través de la Red Internacional de Educación Financiera (INFE), que promueve la implementación efectiva de programas de educación financiera en todo el mundo. La INFE trabaja para desarrollar estándares de educación financiera, compartir mejores prácticas y realizar investigaciones comparativas para medir el impacto de los programas de educación financiera (OCDE, 2023).

Es esencial que los estudiantes desarrollen habilidades financieras desde una edad temprana para tomar decisiones informadas sobre su futuro financiero. A través de la educación financiera, los estudiantes pueden aprender a manejar su dinero de manera responsable, planificar su futuro financiero, evitar el endeudamiento excesivo y comprender mejor el funcionamiento del sistema financiero (OCDE, 2005).

El Programa Internacional para la Evaluación de Estudiantes (PISA, por sus siglas en inglés) es una evaluación a nivel internacional realizada por la OCDE. Esta prueba se lleva a cabo cada tres años y tiene como objetivo medir el rendimiento académico de los estudiantes de 15 años de edad en tres áreas clave: lectura, matemáticas y ciencias.

La prueba PISA incluye una sección sobre educación financiera, diseñada para evaluar la capacidad de los estudiantes para tomar decisiones financieras informadas y para desenvolverse de manera efectiva en el mundo financiero. Esta sección incluyó preguntas sobre temas como el manejo del dinero, la toma de decisiones financieras y el uso de instrumentos financieros, como tarjetas de crédito y cuentas bancarias. Los resultados de la sección sobre educación financiera de la prueba PISA se utilizan para evaluar el nivel de conocimiento y habilidades financieras de los estudiantes y para identificar áreas en las que es necesario mejorar la educación financiera.

Resultados	Chile	OCDE
Desempeño estudiantil en alfabetización financiera (promedio de puntuación)	451	505
Bajos rendidores en alfabetización financiera (porcentaje de estudiantes que obtienen una puntuación inferior al Nivel 2)	30.2	14.7
Altos rendidores en alfabetización financiera (porcentaje de estudiantes que obtienen una puntuación en el Nivel 5)	3.0	10.5
Porcentaje de estudiantes que obtienen información sobre asuntos monetarios (por ejemplo, gasto, ahorro, banca o inversión) de padres, tutores u otras relaciones adultas	93.8	94.4

*Tabla 1: Resultados relevantes en alfabetización financiera de la prueba PISA 2018 (OECD, 2019)*

En la tabla 1 se puede observar que los estudiantes chilenos obtuvieron una puntuación promedio de 451 en alfabetización financiera, inferior al promedio de la OCDE de 505. Además, el porcentaje de estudiantes chilenos con bajo rendimiento en alfabetización financiera (30.2%) es significativamente mayor que el promedio de la OCDE (14.7%), mientras que el porcentaje de estudiantes chilenos con alto rendimiento en alfabetización financiera (3.0%) es menor al promedio de la OCDE (10.5%).

Ante esta realidad, surge la necesidad de promover herramientas educativas que permitan mejorar la educación financiera de los estudiantes chilenos y, en particular, de las familias. Una solución posible es el desarrollo de una aplicación móvil que facilite la gestión cooperativa de las finanzas familiares. Esta aplicación prototipo permitiría a las familias administrar sus finanzas de manera eficiente y colaborativa, involucrando a los miembros de la familia en el proceso de toma de decisiones financieras. Esta memoria de título tiene como objetivo presentar el proceso de concepción y desarrollo de dicha aplicación prototipo, con el fin de contribuir a la mejora de la gestión financiera de las familias chilenas.

## **2. Objetivos**

Esta sección tiene como propósito establecer las metas y propósitos específicos que se buscan alcanzar con el desarrollo de esta memoria de título.

Los objetivos son de dos tipos: generales y específicos. El objetivo general abarca la meta principal, mientras que los objetivos específicos son pasos o etapas más detalladas que conducen a la consecución del objetivo general.

### **2.1 Objetivo General**

El objetivo principal de esta memoria de título es desarrollar un prototipo de software como solución innovadora a los desafíos persistentes de educación e inclusión financiera en familias de Chile, específicamente para niños y adolescentes. Esto se logrará mediante la concepción, diseño e implementación de una aplicación móvil multiplataforma que permita la gestión cooperativa de finanzas en el contexto familiar, proporcionando herramientas prácticas para el manejo del dinero desde una edad temprana con la supervisión de sus padres.

### **2.2 Objetivos Específicos**

- Profundizar y detallar la situación actual del problema detectado en el contexto nacional, incluyendo un análisis de las soluciones parciales existentes y su eficacia en la resolución de las problemáticas de educación e inclusión financiera.
- Definir rigurosamente el mercado objetivo y obtener cifras estimadas acerca de la magnitud de las poblaciones identificadas.
- Proponer una solución integral y sostenible a través de la elaboración de un 'Business Model Canvas'.
- Detallar a profundidad los casos de uso, la arquitectura de software, el modelo de datos, la navegación móvil y el proceso de desarrollo, incluyendo la selección y utilización de las tecnologías seleccionadas.
- Realizar pruebas internas para validar la correcta funcionalidad de la aplicación.

### **3. Problemática**

Los principales tópicos que manifiestan un problema a nivel macrosocial en Chile en el área de las finanzas y que tienen relación con el presente proyecto son la alfabetización financiera y la inclusión financiera.

La inclusión financiera se refiere al acceso de individuos y empresas a servicios financieros básicos, como cuentas de ahorro y crédito, a un precio justo y asequible. La inclusión financiera tiene como objetivo mejorar la calidad de vida de las personas al permitirles ahorrar, planificar y proteger sus finanzas, y también puede tener un impacto positivo en el desarrollo económico de una región al fomentar la creación de empleo y el crecimiento de las pequeñas empresas. (World Bank, 2022)

Por otra parte, la alfabetización financiera se define como el conocimiento y comprensión de conceptos financieros básicos, incluyendo el manejo del dinero y la toma de decisiones financieras informadas. Incluye habilidades como el uso de una banca en línea, la interpretación de estados financieros y la planificación de metas financieras a largo plazo. La alfabetización financiera es importante para permitir que las personas tomen decisiones financieras informadas y puedan utilizar servicios financieros de manera efectiva. (OECD, 2020).

Dichos tópicos son considerados como un requisito fundamental en la vida diaria de las personas a nivel mundial. Actualmente, 45 países en diferentes niveles de ingresos, incluyendo a Chile, avanzan significativamente en el diseño o implementación de una estrategia nacional de educación financiera. (Russia's G20 Presidency & OECD, 2013)

La OCDE/INFE (Red Internacional de Educación Financiera) ha desarrollado un conjunto de principios de alto nivel para la educación financiera que brinda orientación a los países que diseñan e implementan estrategias nacionales de educación financiera. Estos principios abarcan áreas como la necesidad de un enfoque holístico y a largo plazo para la educación financiera, la importancia de involucrar a múltiples partes interesadas en el desarrollo e implementación de iniciativas de educación financiera y la necesidad de garantizar que la educación financiera sea relevante y accesible para poblaciones diversas. (OECD, 2012)

En 2014, la Estrategia Nacional de Educación Financiera (ENEF) fue creada en Chile con el objetivo de promover una mejor comprensión de conceptos y productos financieros y previsionales por parte de la ciudadanía, fomentar el desarrollo de habilidades y actitudes que mejoren el bienestar personal, familiar y comunitario y difundir los derechos financieros de las personas.

Sin embargo, a pesar de los esfuerzos gubernamentales para mitigar las problemáticas nacionales asociadas a la inclusión y alfabetización financiera, carecen de estrategias que involucren considerablemente el nuevo paradigma tecnológico predominante en la sociedad moderna.

#### **4. Acceso a instrumentos financieros y uso de tecnología**

A partir del análisis de los resultados de la evaluación de contenidos, procesos, contexto y factores no cognitivos (Agencia de se extrapolan las siguientes conclusiones:

A pesar de que la mayoría de los estudiantes chilenos de 15 años manifiestan no haber recibido instrucción alguna sobre educación financiera, ya sea en una asignatura específica o en cualquier otra, resulta interesante constatar que aquellos estudiantes que no recibieron esta formación académica obtuvieron puntuaciones más elevadas en pruebas de conocimiento sobre Educación Financiera en los años 2015 y 2018, en comparación con aquellos que sí la recibieron.

Es importante destacar el hecho de que cada vez más estudiantes están adquiriendo habilidades y conocimientos en la gestión de sus finanzas mediante actividades extracurriculares. De manera significativa, aquellos estudiantes que están aprendiendo de esta forma presentan promedios más altos y se aprecia una tendencia creciente en esta línea de aprendizaje.

Los adolescentes en Chile afrontan desafíos significativos para acceder a instrumentos financieros, evidenciado por las estadísticas recientes. Solo un 37% de los jóvenes de 15 años en Chile tienen en su poder una cuenta bancaria, y un 27% posee una tarjeta de débito. Estos porcentajes son inferiores en comparación con las cifras de los Estados Unidos, donde el 40% de los adolescentes tienen tarjeta de débito (Next Gen Personal Finance, 2020).

Aquellos estudiantes que desconocen estos instrumentos financieros presentan las calificaciones más bajas en las pruebas de Educación Financiera. Por otro lado, aquellos estudiantes que sí poseen instrumentos bancarios (tarjeta de débito o cuenta bancaria) obtienen puntajes significativamente más altos en las evaluaciones de Educación Financiera que aquellos que no los poseen.

Los estudiantes chilenos manifiestan sentir menos seguridad en relación a la comprensión y completitud de documentos y formularios (contratos de compraventa, estados de cuenta, formularios) que implican operaciones presenciales.

Por otra parte, es relevante notar que aquellos estudiantes que manifiestan mayor seguridad en relación a operaciones virtuales tales como la planificación de gastos, seguimiento de saldos, realización de transferencias, comprensión de estados de cuenta, completitud de formularios en el banco y comprensión de contratos de compraventa, alcanzan mejores resultados en las pruebas de Educación Financiera.

Además, se observa que a medida que aumenta el nivel de seguridad declarado por los estudiantes en relación al uso de dispositivos digitales y electrónicos para llevar a cabo acciones concretas fuera del banco, su promedio en la escala de Educación Financiera es mayor.

Estas afirmaciones se pueden explicar en cierta medida con características asociadas a la inclusión digital y democratización de la tecnología para los individuos que aún no alcanzan la

mayoría de edad en Chile. Algunos estudios realizados en los últimos años nos otorgan datos ilustrativos:

El 90% de los adolescentes mayores de 13 años posee un celular. (VTR & CRITERIA, 2022) La tenencia de celulares es amplia desde edades tempranas y aumenta con la edad en todo el rango etario que involucra la minoría de edad. (CASEN & Ministerio del Desarrollo Social, 2017)

Desde una perspectiva conductual, el Censo Digital desarrollado el 2018 por la Subsecretaría de Telecomunicaciones y VTR indica que más de la mitad de los individuos encuestados entre 8 y 14 años se conecta a su celular desde las 7 AM. (Censo digital, 2018)

La Universidad de los Andes publicó en 2018 un estudio que plantea aspectos relevantes:

- En Chile, el promedio de edad en que los niños reciben su primer celular es de 10,3 años, y el 59% de los encuestados recibió un celular antes de los 10 años.
- En cuanto al tiempo que los niños pasan frente a la pantalla del teléfono, el promedio es de alrededor de 6,8 horas diarias, y uno de cada cuatro encuestados entre 10 y 18 años usa su celular más de 10 horas diarias.
- El 87% de los niños encuestados declara que lleva su teléfono al colegio todos los días y el 71% lo usa dentro de la sala de clases. (Universidad de los Andes, 2018)

La información presentada describe el contexto nacional que combina problemáticas, necesidades y oportunidades para la implementación de una solución digital adecuada.

## **5. Soluciones existentes**

La gestión cooperativa de finanzas en familia es un tema relevante en la actualidad, que busca facilitar el control y la organización del dinero en el hogar, a través de la colaboración y el trabajo en equipo. En este sentido, contar con soluciones eficientes y adaptadas a las necesidades actuales resulta fundamental para lograr una gestión financiera óptima y mejorar la calidad de vida de los miembros de la familia.

En este contexto, es esencial conocer las soluciones ya existentes para la gestión cooperativa de finanzas en familia, a fin de identificar las fortalezas y debilidades de estas soluciones, así como las oportunidades de mejora y diferenciación que podrían explorarse. Asimismo, a través de la investigación de soluciones existentes, es posible identificar tendencias y buenas prácticas que podrían ser adaptadas y aplicadas a nuevos desarrollos, para lograr una solución innovadora y efectiva.

Por lo tanto, en la presente sección, se abordarán las soluciones existentes en la gestión cooperativa de finanzas en familia, las cuales se clasificaron en dos categorías: soluciones nacionales e internacionales. Se profundizará en las soluciones internacionales más destacadas, que ofrecen una solución más acertada y adaptada a las necesidades actuales de los usuarios. Esta investigación permitirá contextualizar el trabajo y destacar la necesidad de desarrollar una nueva solución para la gestión cooperativa de finanzas en familia, que se adapte a las necesidades específicas de los usuarios y ofrezca un valor diferencial en el mercado nacional.

### **5.1 Contexto Nacional**

En el mercado nacional, existen algunas soluciones que abordan de manera parcial el problema de la inclusión financiera para los menores de edad, centrándose principalmente en la obtención de tarjetas de débito y acceso a aplicaciones móviles. Sin embargo, estas soluciones no profundizan lo suficiente en sus funcionalidades, lo que limita su capacidad para gestionar las finanzas y para transaccionar y cooperar con los miembros de la familia en un ambiente seguro. Además, estas soluciones no suelen destacarse por sus interfaces gráficas ni por ofrecer una experiencia de usuario satisfactoria, lo que es una limitación importante. Evidencia de esto son sus bajos puntajes de valoración en las tiendas de aplicaciones. Actualmente en Google Play Banco Estado posee una valoración de 3.6/5 y Mi Banco (Banco de Chile) una valoración de 2.7/5. Es por esta razón que no se incluye información detallada acerca de sus interfaces o experiencia de usuario.

Adicionalmente, las cuentas de ahorro son excluidas de estas soluciones, debido a que, aunque ofrecen un producto bancario accesible, su naturaleza limita el número de operaciones disponibles. Estas cuentas están diseñadas para que los usuarios realicen depósitos y retiros de manera esporádica, pero no ofrecen funcionalidades esenciales, como la realización de transferencias o pagos en línea. Esta falta de opciones limita significativamente la capacidad de los jóvenes para administrar sus finanzas de manera efectiva y completa.

## **Cuenta Rut**

Perteneciente al BancoEstado, permite a las mujeres mayores de 12 años y los hombres mayores de 14 años obtener una cuenta vista vinculada a una tarjeta de débito y acceso a la aplicación móvil. Es necesario que el representante legal firme el contrato de apertura presencialmente para concretar el registro.

Entre sus funcionalidades destacan: Consultar saldo, transferir entre cuentas propias y a contactos, configuración de tarjeta, entre otros. (*Banco Estado*, s. f.)

## **TuMACH**

Perteneciente al Banco BCI, permite a los menores entre 14 y 17 años obtener una tarjeta virtual prepago y acceso a la aplicación móvil. Es necesario que al menos un padre, madre o tutor legal autorice el registro.

Entre sus funcionalidades destacan: Ahorrar y recibir dinero; Compras en línea, Transferencia entre amigos registrados en MACH, División de cuenta entre amigos, entre otros. (*Tu MACH*, s. f.)

## **Cuenta Fan Clan**

Perteneciente al Banco de Chile, permite a los menores entre 14 y 17 años obtener una cuenta vista vinculada a una tarjeta de débito virtual y acceso a la aplicación móvil. Es necesario que ambos padres autoricen el registro, y al menos uno debe ser cliente vigente del Banco de Chile.

Entre sus funcionalidades destacan: Uso de la aplicación Mi Pago para pagar con códigos QR y cobrar a los contactos de la agenda del teléfono, pago sin contacto, entre otros. (*Banco de Chile*, s. f.)

Los software para dispositivos móviles de las cuentas listadas anteriormente establecen funcionalidades, estándares de diseño y experiencia de usuario fundamentales de una aplicación móvil bancaria, los cuales son esenciales para garantizar que la aplicación sea fácil de usar, accesible y segura para los usuarios.

- **Tipografía clara y sencilla:** Para mejorar la legibilidad de la aplicación, se recomienda utilizar tipografías claras y consistentes en toda la interfaz. Se deben evitar fuentes pequeñas y excesivamente estilizadas que puedan dificultar la lectura.
- **Diseño responsive:** La aplicación debe estar diseñada para ser responsive a diferentes tamaños y orientaciones de pantalla, garantizando que se vea bien y funcione correctamente en una variedad de dispositivos.
- **Accesibilidad:** La aplicación debe ser diseñada con características accesibles para usuarios con discapacidades. Esto puede incluir tamaños de fuente ajustables, modos de alto contraste y soporte para lectores de pantalla.



- Visualización de datos: Muchas aplicaciones bancarias utilizan gráficos, tablas y otras herramientas de visualización de datos para ayudar a los usuarios a comprender sus datos financieros y tomar decisiones informadas.
- Seguridad: La seguridad es una preocupación principal en las aplicaciones móviles bancarias, por lo que se deben implementar estándares de seguridad como la autenticación de dos factores, el cifrado y la detección de fraudes para garantizar la protección de los datos del usuario.
- Historial de transacciones: El historial de transacciones es una característica clave de las aplicaciones móviles bancarias. La interfaz de usuario debe facilitar a los usuarios ver y buscar su historial de transacciones, así como categorizar y etiquetar transacciones.
- Pagos de facturas: La aplicación debe permitir a los usuarios pagar facturas directamente desde sus cuentas. La interfaz de usuario debe facilitar la configuración y el manejo de los pagos de facturas, con características como programación y recordatorios de pago.
- Gestión de cuentas: Los usuarios deben poder ver y gestionar todas sus cuentas desde dentro de la aplicación, incluyendo cuentas corrientes, de ahorro y tarjetas de crédito. La interfaz de usuario debe facilitar la transferencia de fondos entre cuentas, así como la apertura y el cierre de cuentas.
- Alertas y notificaciones: La aplicación debe ofrecer alertas y notificaciones, como cuando los saldos de las cuentas caen por debajo de un cierto umbral o cuando se debe realizar un pago de factura. La interfaz de usuario debe facilitar a los usuarios configurar y gestionar alertas, así como ver y responder a las notificaciones.

Dichos patrones serán considerados durante el proceso de diseño y desarrollo de la aplicación para garantizar una experiencia satisfactoria para los usuarios.

## **5.2 Contexto Internacional**

A nivel internacional, existen diversas soluciones que abordan el problema de la inclusión financiera para los menores de edad de una manera más completa y efectiva. Sin embargo, no están disponibles para la población chilena.

Estas soluciones cuentan con un enfoque más amplio, ofreciendo funcionalidades adicionales y experiencias de usuario más satisfactorias. Algunas de las soluciones internacionales más destacadas incluyen GoHenry, Greenlight y Rooster Money.

## **GoHenry**

Originaria del Reino Unido, GoHenry es una aplicación que permite a los menores de edad gestionar su dinero de manera efectiva y aprender sobre finanzas personales. La plataforma ofrece una tarjeta de débito personalizada y una aplicación móvil con funcionalidades para padres y menores. Los padres pueden establecer límites de gasto, asignar tareas y recompensas, así como recibir notificaciones en tiempo real. Por otro lado, los menores pueden realizar compras, ahorrar y donar a organizaciones benéficas. (GoHenry, s. f.)

Ver Anexo 1: Interfaz gráfica para padres GoHenry y Anexo 2: Interfaz gráfica para hijos GoHenry

## **Greenlight**

Greenlight es una solución financiera estadounidense dirigida a menores de edad y sus familias. Ofrece una tarjeta de débito y una aplicación móvil con herramientas de control parental y educación financiera. Los padres pueden establecer límites de gasto, asignar tareas y recompensas, y recibir alertas de actividad. Los menores pueden utilizar la tarjeta para realizar compras, establecer objetivos de ahorro y aprender sobre finanzas a través de lecciones integradas en la aplicación. Además, Greenlight ofrece una función de inversión que permite a los menores invertir en acciones y fondos. (Greenlight, s. f.)

Ver Anexo 3: Interfaz gráfica para padres Greenlight

## **Rooster Money**

Rooster Money es una aplicación financiera del Reino Unido que se centra en la educación financiera para menores de edad. Ofrece una aplicación móvil que permite a los padres y menores llevar un registro de las asignaciones, gastos y ahorros. Los padres pueden asignar tareas y recompensas, establecer límites de gasto y recibir informes sobre la actividad financiera de sus hijos. (Rooster Money, s. f.)

Ver Anexo 4: Interfaz gráfica de bienvenida Rooster Money

Las aplicaciones GoHenry, Greenlight y Rooster Money establecen funcionalidades, estándares de diseño y experiencia de usuario fundamentales de una aplicación móvil bancaria dirigida a menores de edad, las cuales son esenciales para garantizar que la aplicación sea fácil de usar, accesible y segura para los usuarios y sus familias. Estos patrones y funcionalidades relevantes, no mencionados anteriormente, son:

- **Control parental:** Estas aplicaciones ofrecen herramientas de control parental que permiten a los padres supervisar y guiar las actividades financieras de sus hijos. La interfaz de usuario debe facilitar a los padres configurar y gestionar límites de gasto, asignar tareas y recompensas, y recibir alertas de actividad.
- **Educación financiera:** GoHenry, Greenlight y Rooster Money incorporan elementos educativos en sus aplicaciones, enseñando habilidades financieras básicas y promoviendo el ahorro y la responsabilidad financiera. La interfaz de usuario debe incluir lecciones y actividades que ayuden a los usuarios a aprender y aplicar conceptos financieros.
- **Personalización:** La personalización es una característica importante en estas aplicaciones, permitiendo a los usuarios adaptar su experiencia según sus preferencias y necesidades. La interfaz de usuario debe permitir a los usuarios personalizar aspectos como la apariencia de la aplicación, las notificaciones y las metas de ahorro.
- **Redes sociales y colaboración:** Estas aplicaciones facilitan la interacción entre amigos y familiares, permitiendo a los usuarios compartir sus logros financieros y colaborar en actividades como dividir cuentas o realizar transferencias entre amigos registrados. La interfaz de usuario debe facilitar estas interacciones sociales y de colaboración.
- **Diseño centrado en el usuario:** Es fundamental que el diseño de la aplicación esté centrado en el usuario, especialmente en menores de edad. Esto implica utilizar iconografía, colores y elementos visuales atractivos y apropiados para la edad que ayuden a los usuarios a navegar y comprender fácilmente las funciones y características de la aplicación.

## 6. Mercado Objetivo

A partir de los principales problemas y necesidades abordados previamente, se identifican dos grupos demográficos directamente relacionados. En primer lugar, se encuentra el grupo de los hijos, que son individuos residentes en Chile con edades entre 5 y 17 años y que tienen acceso a un smartphone. Por otro lado, se encuentra el grupo de los padres, que son individuos residentes en Chile que tienen al menos un hijo entre 5 y 17 años con acceso a un smartphone.

### 6.1 Hijos

Individuos residentes en Chile pertenecientes al rango etario que varía entre los 5 y 17 años, y que poseen acceso a un smartphone. Se describe el proceso para cuantificar de forma discreta y simplista la magnitud de esta población.

Edad	Población	Posee y utiliza celular	Total
5	238.867	13 %	31.052,71
6	247.819	17 %	42.129,23
7	246.261	23 %	56.640,03
8	241.525	31 %	74.872,75
9	235.717	41 %	96.643,97
10	230.340	50 %	115.170
11	228.005	61 %	139.083,05
12	225.539	71 %	160.132,69
13	227.584	79 %	179.791,36
14	235.947	84 %	198.195,48
15	239.492	89 %	213.147,88
16	247.940	93 %	230.584,2
17	247.973	93 %	230.614,89
5-17	3.093.009	57,3%	1.768.058,24

*Tabla 2: Elaboración propia (Resultados Censo 2017, Encuesta Casen 2017)*

Finalmente obtenemos que la estimación de la magnitud de los hijos es de aproximadamente 1.768.000 individuos.

## 6.2 Padres

Individuos residentes en Chile que son padres/madres de al menos un hijo entre 5 y 17 años con acceso a un smartphone. Inicialmente, calculamos el porcentaje de hogares chilenos con al menos un hijo de entre 5 y 17 años. El porcentaje es del 42,4%, lo que equivale a aproximadamente 2.792.001 hogares.

<b>Porcentaje de hogares con personas de 0 a 17 años (2020)</b>		
<i>(porcentaje, hogares)</i>		
	Estimación	N expandido
	2020	2020
Hogar sin NNA	57,6	3.786.325
Hogar con NNA	42,4	2.792.001
Total	100,0	6.578.326

*Tabla 3: Elaboración propia (Niños, niñas y adolescentes, Casen en Pandemia 2020)*

Posteriormente, analizamos la distribución de hogares monoparentales y biparentales.

<b>Distribución de los hogares según tipo de hogar (1992-2020)</b>		
<i>(porcentaje, hogares)</i>		
		Estimación
		2020
Hogar con NNA	Unipersonal	0,0
	Nuclear Monoparental	25,6
	Nuclear Biparental	47,4
	Extenso Monoparental	13,7
	Extenso Biparental	13,2
	Sin núcleo	0,0

*Tabla 4: Elaboración propia (Niños, niñas y adolescentes Encuesta Casen en Pandemia 2020)*

Considerando que en hogares monoparentales hay 1 padre, mientras que en hogares biparentales hay 2 padres, podemos calcular la cantidad de padres en estos hogares utilizando nuestra estimación previa de 2.792.001 hogares con hijos entre 5 y 17 años:

- Hogares Nuclear Monoparental:  $2.792.001 * 0.256 = 714.752$ 
  - Padres en hogares Nuclear Monoparental: 714.752
- Hogares Nuclear Biparental:  $2.792.001 * 0.474 = 1.323.408$ 
  - Padres en hogares Nuclear Biparental:  $1.323.408 * 2 = 2.646.816$
- Hogares Extenso Monoparental:  $2.792.001 * 0.137 = 382.504$ 
  - Padres en hogares Extenso Monoparental: 382.504
- Hogares Extenso Biparental:  $2.792.001 * 0.132 = 368.544$ 
  - Padres en hogares Extenso Biparental:  $368.544 * 2 = 737.088$

Total de padres/madres =  $714.752 + 2.646.816 + 382.504 + 737.088 \approx 4.481.160$  padres/madres

Para estimar cuántos de estos padres/madres tienen al menos un hijo con acceso a un smartphone, utilizamos la información proporcionada en la tabla de datos sobre el porcentaje de niños y adolescentes que poseen y utilizan un celular. En 2020, el 57,3% de los 3.093.009 niños y adolescentes de 5 a 17 años en Chile tenía acceso a un celular, lo que equivale a aproximadamente 1.768.058 niños y adolescentes.

Proporción de niños y adolescentes con celular =  $1.768.058 / 3.093.009 \approx 0,5718$  (57,18%)

Aplicamos esta proporción al número total de padres/madres estimados:

Cantidad de padres/madres con al menos un hijo con celular =  $4.481.160 * 0,5718 \approx 2.562.327$

Finalmente, se estima que en Chile hay aproximadamente 2.562.000 padres/madres con al menos un hijo entre 5 y 17 años que posee un celular.

## 7. Solución Propuesta

El foco de la solución propuesta radica en una aplicación móvil multiplataforma con una interfaz adaptativa para padres e hijos, junto con una tarjeta prepago para menores de edad.

Es el primer acercamiento de niños y adolescentes a las finanzas, enseñando lo que dichas herramientas financieras implican, y facilitando la gestión de gastos y ahorros de forma cooperativa.

A la vez, permite a los padres el control del dinero de sus hijos, proporcionando herramientas de supervisión sobre los movimientos de sus hijos, transferencias instantáneas, asignación de tareas con recompensa monetaria, entre otros. El uso de una tarjeta prepago segura en un entorno familiar permite a los padres tener mejor control sobre las finanzas de sus hijos.

### 7.1 Business Model Canvas

Para describir las proyecciones de negocio que surgen de la presente ideación, se utiliza la metodología de Business Model Canvas propuesta por Alexander Osterwalder (Osterwalder, A., & Pigneur, Y, 2010), el cual proporciona una plantilla de gestión estratégica utilizada para desarrollar y documentar modelos de negocio a partir de nueve elementos clave.

A continuación se describen los nueve bloques de construcción del Business Model Canvas asociado a la solución propuesta:

#### Propuesta de valor

- Padres y madres
  - Seguimiento y trazabilidad del comportamiento financiero de sus hijos.
  - Transferencias instantáneas a las cuentas de sus hijos.
  - Asignación de mesadas automáticas a sus hijos.
  - Asignación de tareas con recompensas a sus hijos.
  - Control parental de la tarjeta de sus hijos.
  - Medio centralizado para la gestión de las finanzas de sus hijos.
  - Aplicación móvil simple e intuitiva.
- Niños y adolescentes
  - Medio de pago para comercios presenciales y electrónicos.
  - Transferencias entre cuentas propias.
  - Gestión de metas de ahorro.
  - Medio de obtención de dinero a través de mesadas y tareas con recompensa.
  - Beneficios en comercios frecuentados.
  - Consciencia mediante la trazabilidad de transacciones y gastos.

## **Canales**

- Aplicación iOS/Android
- Tiendas de aplicaciones (Google Play Store, App Store, App Gallery)
- Sitio web
- Redes Sociales (Instagram, TikTok, Discord, Twitter, LinkedIn)
- Correo electrónico
- Blogs
- Streaming (Twitch)

## **Relación con clientes**

- Servicio al cliente (correo electrónico, redes sociales)
- Inducción interactiva del uso de la aplicación
- Manual de uso (texto y multimedia)
- FAQ (Preguntas más frecuentes)

## **Fuente de ingresos**

- Comisión por uso de tarjeta: Al emitir tarjetas prepago, la utilización de dichas tarjetas genera una fuente de ingresos que es una fracción de la comisión que el adquirente cobra a los comercios al utilizar un terminal de ventas para realizar pagos.
- Emisión de tarjeta física: A diferencia de la tarjeta virtual que es de carácter gratuito, la emisión, impresión y entrega de la tarjeta física contempla un cobro para el cliente.
- Movimientos desde cuentas de ahorro: Para fomentar el ahorro de los menores, se cobra una comisión al realizar movimientos de dinero desde cuentas de ahorro asociadas a metas que no han sido alcanzadas.
- Retiros de efectivo: Como precursores de la era cashless, los retiros libres en efectivo realizados en cajeros automáticos presentan un límite mensual, el cual al ser excedido comienza el cobro de comisiones.

## **Actividades claves**

- Etapa inicial
  - Definición de la interfaz y experiencia de usuario de la aplicación móvil.
  - Implementación del prototipo de la aplicación móvil.
  - Desarrollo del sitio web y captura de early adopters.
  - Publicación de prototipo en plataformas de pruebas.
  - Integración con emisor de tarjetas.
- Etapa operativa
  - Operaciones de producto.
  - Operaciones TI.
  - Creación de contenidos.
  - Operaciones de finanzas y contabilidad.
  - Operaciones de marketing y beneficios.



- Operaciones de recursos humanos.
- Operaciones de área legal.
- Operaciones de ventas y servicio al cliente.

### **Recursos claves**

- Proveedor de servicios de computación cloud (AWS).
- Infraestructura de transacciones.
- Equipo de investigación y desarrollo.
- Recursos intelectuales.
- Profesionales capacitados para las actividades claves de la etapa operativa.

### **Socios clave**

- Plataforma de emisión y distribución de tarjetas prepago.
- Agencia digital.
- Establecimientos educacionales.
- Medios de pago digitales.
- Inversionistas.
- Comercios con la capacidad de ofrecer beneficios.

### **Estructura de costos**

- Costos asociados a los servicios de nube.
- Presupuesto para investigación y desarrollo.
- Presupuesto para actividades clave de la etapa operativa.
- Estructura salarial.
- Licencias comerciales y legales.
- Publicidad.

### **Segmento de clientes**

- Padres y madres de individuos con edades entre 5 y 17 años.
- Menores de edad con edades entre 5 y 17 años con acceso a un smartphone.

## **8. Diseño e implementación del sistema**

El objetivo de esta sección es proporcionar una descripción detallada del diseño e implementación de la solución, con el propósito de comprender la definición, funcionamiento y la metodología de desarrollo del software.

Comenzamos con la definición del sistema, donde se describe el propósito, restricciones y los objetivos generales del software. A continuación, analizamos los casos de uso, que representan las diferentes interacciones entre los usuarios y el sistema. Presentamos el diagrama de casos de uso y describimos los casos de uso principales, distinguiéndolos según el actor principal.

Luego, abordamos la arquitectura de software del sistema, identificando sus elementos clave y cómo interactúan entre sí. Esta descripción proporciona una visión general de alto nivel de la estructura general del sistema, lo que facilita la comprensión de su diseño y funcionamiento.

A continuación, nos centramos en el modelo de datos, que es fundamental para el almacenamiento y la gestión de la información en el sistema. Explicamos el modelo conceptual y describimos los diferentes tipos de datos que se utilizan en el software.

Después de analizar el modelo de datos, discutimos la navegación móvil y cómo se diseña el sistema para garantizar una experiencia de usuario óptima en dispositivos móviles. La compatibilidad con dispositivos móviles es esencial en el mundo actual, y la navegación se diseña teniendo en cuenta esta premisa.

Por último, describimos el proceso de desarrollo del sistema, incluyendo los enfoques y las herramientas utilizadas en la implementación del front-end y el back-end. También se explica cómo se gestiona el código fuente mediante el uso de un repositorio centralizado.

## **8.1 Definición del sistema**

El presente trabajo contempla la implementación de un prototipo que ilustra los principios y operaciones fundamentales de la solución propuesta. El principal objetivo de la implementación consiste en demostrar los beneficios derivados de contar con una plataforma que aborde las problemáticas identificadas a través de una solución de software.

La solución consiste en una aplicación móvil multiplataforma, compatible con los sistemas operativos Android e iOS. La aplicación cuenta con una interfaz adaptativa, la cual varía según el perfil del usuario (padre o hijo) que esté utilizando el sistema. De esta manera, se ofrece una experiencia de usuario personalizada y enfocada en las necesidades específicas de cada tipo de usuario.

Además, la aplicación proporciona una actualización en tiempo real de los datos para ambos perfiles de usuario, permitiendo una interacción dinámica y cooperativa entre dos dispositivos distintos que empleen la misma aplicación.

Cabe mencionar que la principal restricción en el prototipo radica en la utilización de dinero ficticio, debido a las limitaciones inherentes a la integración de un sistema capaz de gestionar transacciones de dinero real en el contexto de este proyecto.

## **8.2 Casos de uso**

La definición de los casos de uso permiten modelar el comportamiento, capacidades y funcionalidades de la aplicación, además de las interacciones de los actores Padre e Hijo con el sistema.

La descripción de cada caso de uso incluye a detalle el comportamiento de la interfaz gráfica de la aplicación en el flujo principal y los flujos de excepción, con el objetivo de precisar la definición del front-end del software.

### **8.2.1 Diagrama de casos de uso**

A continuación se presenta el diagrama de casos de uso elaborado para modelar el comportamiento de la aplicación y las interacciones que tienen los actores involucrados, Padre e Hijo, con el sistema. Se utiliza la definición del diagrama de casos de uso proporcionada por Rational Software Architect 9.6.1 el cual emplea el Lenguaje de Modelado Unificado (UML). (IBM, s. f.)



Figura 1: Diagrama de Casos de uso

## 8.2.2 Padres

En esta sección se describen los casos de usos asociados a los Actores Padre, incluyendo una breve descripción. La documentación de los casos de uso se encuentra en los Anexos 5-19, los cuales incluyen detalles tales como: ID, Descripción, Actores, Pre-condiciones, Post-condiciones, Flujo Principal y Flujos de excepción.

**Agregar fondos:** Permite que un Padre agregue dinero a su cuenta. (Ver Anexo 5).

**Transferir a hijo:** Permite que un Padre transfiera dinero desde el saldo de su billetera a una cuenta de uno de sus hijos. (Ver Anexo 6).

**Ver actividad:** Permite que un Padre visualice un historial de la actividad suya y la de sus hijos. (Ver Anexo 7).

**Ver detalle general de hijo:** Permite que un Padre visualice el perfil de uno de sus hijos. (Ver Anexo 8).

**Ver detalle ahorros de hijo:** Permite que un Padre visualice el detalle de los ahorros de su hijo. (Ver Anexo 9).

**Ver detalle mesada de hijo:** Permite que un Padre visualice el detalle de la mesada de su hijo. (Ver Anexo 10).

**Asignar mesada a hijo:** Permite que un Padre cree y asigne una mesada a su hijo. (Ver Anexo 11).

**Editar mesada de hijo:** Permite que un Padre edite la mesada que asignó a su hijo. (Ver Anexo 12).

**Eliminar mesada de hijo:** Permite que un Padre elimine la mesada que asignó a su hijo. (Ver Anexo 13).

**Ver detalle tareas de hijo:** Permite que un Padre visualice el detalle de las tareas de su hijo. (Ver Anexo 14).

**Asignar tarea a hijo:** Permite que un Padre cree y asigne una tarea a su hijo. (Ver Anexo 15).

**Pagar tarea finalizada de hijo:** Permite que un Padre confirme la finalización de una tarea asignada a su hijo y efectúe el pago correspondiente. (Ver Anexo 16).

**Rechazar finalización de tarea de hijo:** Permite que un Padre rechace la finalización de una tarea asignada a su hijo. (Ver Anexo 17).

**Ver detalle tarjeta de hijo:** Permite que un Padre visualice el detalle de la tarjeta de su hijo. (Ver Anexo 18).

**Bloquear temporalmente tarjeta de hijo:** Permite que un Padre desactive temporalmente la tarjeta de su hijo. (Ver Anexo 19).

### 8.2.3 Hijos

En esta sección se describen los casos de usos asociados a los Actores Hijo, incluyendo una breve descripción. La documentación de los casos de uso se puede visualizar en los Anexos 20-30, los cuales incluyen detalles tales como: ID, Descripción, Actores, Pre-condiciones, Post-condiciones, Flujo Principal y Flujos de excepción.

**Mostrar información sensible de tarjeta:** Permite que un Hijo visualice los detalles de su tarjeta. (Ver Anexo 20).

**Ocultar información sensible de tarjeta:** Permite que un Hijo oculte los detalles de su tarjeta (Ver Anexo 21).

**Transferir entre cuentas:** Permite que un Hijo transfiera dinero entre sus cuentas. (Ver Anexo 22).

**Ver detalle de ahorros:** Permite que un Hijo visualice el detalle de sus ahorros. (Ver Anexo 23).

**Crear meta de ahorro:** Permite que un Hijo cree una meta de ahorro. (Ver Anexo 24).

**Editar meta de ahorro:** Permite que un Hijo edite una meta de ahorro existente. (Ver Anexo 25).

**Eliminar meta de ahorro:** Permite que un Hijo elimine una meta de ahorro existente. (Ver Anexo 26).

**Ver detalle de ganar dinero:** Permite que un Hijo visualice el detalle de sus medios para ganar dinero. (Ver Anexo 27).

**Completar tarea “Por hacer”:** Permite que un Hijo confirme la realización de una tarea en el estado “Por hacer”. (Ver Anexo 28).

**Simular compra:** Permite que un Hijo simule una compra con su tarjeta. (Ver Anexo 29).

**Ver detalle de tarjeta:** Permite que un Hijo visualice el detalle de su tarjeta. (Ver Anexo 30).

### 8.3 Arquitectura de software

En lo que concierne a la arquitectura de software de la implementación se utiliza como paradigma base la Arquitectura Serverless.

Este tipo de arquitectura nativo de la nube provee la capacidad de crear y ejecutar aplicaciones complejas sin la necesidad de administrar y mantener la infraestructura del servidor. Las aplicaciones se dividen en servicios individuales que se pueden invocar y escalar individualmente.

La elección de la arquitectura Serverless como base para el desarrollo del proyecto se sustenta en varios factores que a continuación se detallan:

**Escalamiento automático:** Permite un escalado dinámico y automático en función de la demanda del servicio. A pesar de que en la etapa de desarrollo no se esperan altos niveles de concurrencia, la capacidad de adaptarse a las demandas cambiantes es un aspecto crucial que proporciona flexibilidad y garantiza una respuesta óptima en caso de incrementos inesperados de la demanda en etapas posteriores.

**Eficiencia de costos:** En un modelo de servidor tradicional, los recursos del servidor están activos constantemente, incluso cuando la aplicación no está siendo utilizada, lo que conlleva a costos innecesarios. En contraposición, el modelo Serverless contempla un cobro por el tiempo de computación consumido. Es decir, no se paga por tiempo en servicios inactivos, lo que resulta en un ahorro significativo y optimiza el uso de los recursos, especialmente en etapas de desarrollo.

**Rapidez en desarrollo y despliegue:** La arquitectura Serverless mitiga la necesidad de administrar y mantener servidores. Esto permite enfocar los esfuerzos en la lógica de negocio y en el desarrollo de funcionalidades de la aplicación, acelerando así el proceso de desarrollo y despliegue. Este factor es particularmente relevante en el contexto de esta memoria de título, donde el desarrollo es realizado por un único individuo.

**Seguridad:** En el ámbito de la seguridad, la arquitectura Serverless sobresale debido a su infraestructura administrada. Los proveedores de servicios Serverless, como AWS, cuentan con una amplia gama de herramientas y servicios dedicados a la seguridad, que permiten configurar detalladamente las políticas de autenticación y autorización. De este modo, se puede garantizar que solo las entidades permitidas tienen acceso a los servicios y datos correspondientes, lo que proporciona una capa adicional de seguridad para la aplicación.

**Panel de Control y CLI:** Las soluciones Serverless ofrecen herramientas eficientes y potentes que simplifican la administración, el monitoreo y la depuración de las aplicaciones. Permiten una administración granular de los servicios, así como la capacidad de realizar ajustes y monitorear el desempeño de las aplicaciones en tiempo real.

Es relevante justificar la elección de Amazon Web Services (AWS) como el proveedor de esta infraestructura. AWS es uno de los principales proveedores de servicios en la nube en el mundo y ha sido seleccionado principalmente por las siguientes razones:

**Integración con React Native:** AWS ofrece una integración sólida y sin problemas con React Native, una de las tecnologías fundamentales utilizadas en este proyecto. AWS Amplify, en particular, es una biblioteca que proporciona una serie de capacidades, como autenticación, almacenamiento de archivos, API (GraphQL y REST), análisis, notificaciones push, entre otros. Estas capacidades son fácilmente accesibles a través de interfaces de programación claras y sencillas. Amplify también ofrece un conjunto de componentes de interfaz de usuario predefinidos que simplifican enormemente el desarrollo de aplicaciones móviles. Esta estrecha integración con React Native facilita y acelera el desarrollo y el despliegue de la aplicación.

**Documentación:** AWS ofrece una amplia gama de documentación detallada y de alta calidad que cubre todos los aspectos de su plataforma. Esto incluye guías de inicio rápido, tutoriales, documentación de referencia de API, ejemplos de código y mucho más. Esta documentación es invaluable para entender cómo utilizar correctamente los servicios y cómo integrarlos con la aplicación.

A continuación se presenta el diagrama de la arquitectura de software del sistema:

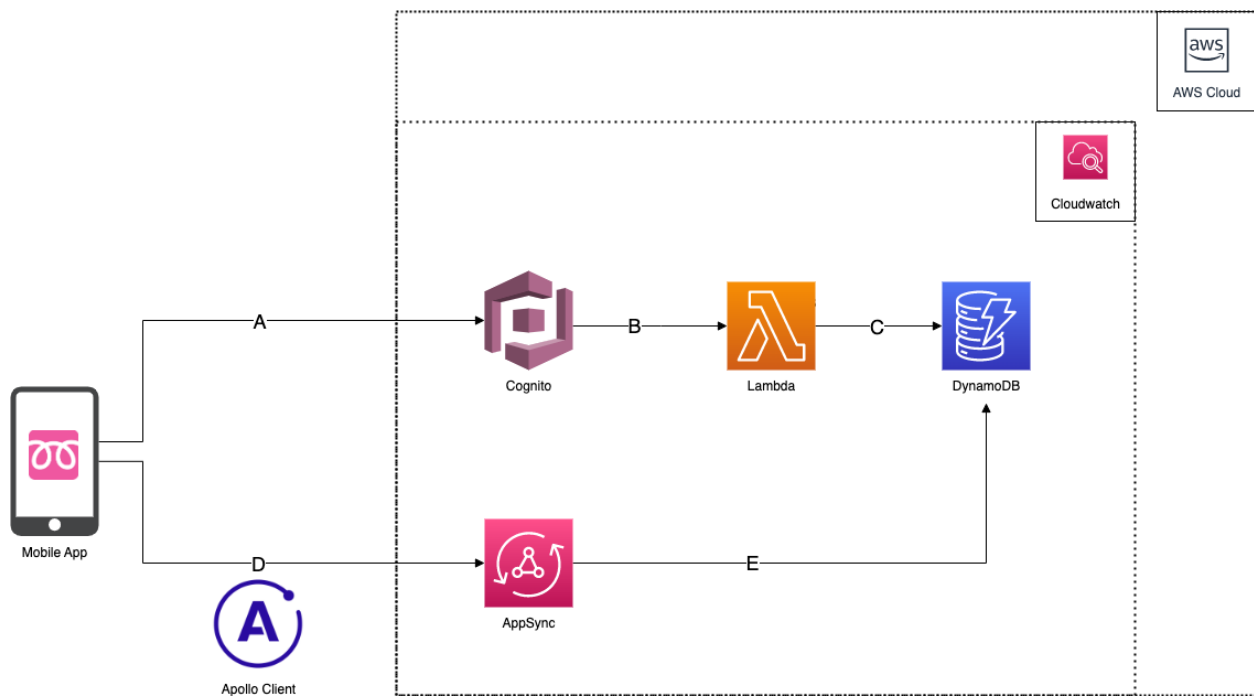


Figura 2: Diagrama de Arquitectura de software



### **8.3.1 Elementos**

En la sección de Elementos, se presentan los componentes y tecnologías fundamentales que constituyen la arquitectura de la solución propuesta en la memoria de título. Estos elementos trabajan en conjunto para proporcionar una infraestructura escalable, segura y de alto rendimiento, que respalda las funcionalidades y requisitos de la aplicación.

#### **Aplicación Móvil**

El concepto de aplicación móvil presente en el diagrama hace referencia a la versión en ejecución en un dispositivo Android o iOS, la cual es generada a partir del desarrollo de aquella con el framework React Native (React Native s.f.).

#### **Apollo Client**

Apollo Client es una biblioteca de gestión de estado integral para JavaScript que permite la gestión de datos tanto locales como remotos mediante GraphQL (Apollo GraphQL, s. f.). Se utiliza para obtener, almacenar en caché y modificar los datos de la aplicación, actualizando automáticamente la interfaz de usuario. La estructura de código es eficiente, predecible y declarativa, de acuerdo con las prácticas de desarrollo modernas.

#### **AWS Cloud**

Amazon Web Services (AWS) es una plataforma que ofrece soluciones de computación en la nube flexibles, confiables, escalables, fáciles de usar y de bajo costo (Amazon Web Services, s.f.). La plataforma está desarrollada con una combinación de servicios de infraestructura (IaaS), plataforma (PaaS) y paquetes de software como servicio (SaaS).

En la presente arquitectura, AWS Cloud hace referencia al ecosistema que integra y disponibiliza los servicios implementados en la aplicación, precisamente en la región US East (Ohio).

#### **Cognito**

Amazon Cognito es un servicio de AWS que proporciona autenticación, autorización y administración de usuarios (Amazon Web Services, s.f.). Ofrece un almacenamiento de identidades escalable a millones de usuarios, soporte de proveedores de identidad social y empresarial, y características de seguridad avanzadas.

## **Lambda**

AWS Lambda es un servicio de computación serverless proporcionado por AWS (Amazon Web Services, s.f.), el cual permite crear funciones y aplicaciones independientes escritas en uno de los lenguajes y entornos de ejecución compatibles, y desplegarlas en AWS Lambda, que ejecuta dichas funciones de manera eficiente, flexible y escalable.

Las funciones Lambda pueden realizar cualquier tarea informática, desde servir páginas web y procesar flujos de datos hasta comunicarse con APIs e integrarse con otros servicios de AWS.

Además permiten ser desencadenadas automáticamente a partir de eventos emitidos por servicios de AWS, o directamente desde cualquier aplicación.

## **AppSync**

AWS AppSync es un servicio administrado de AWS (Amazon Web Services, s.f.) que utiliza GraphQL para facilitar que las aplicaciones obtengan precisamente los datos que necesitan. Permite el desarrollo de aplicaciones escalables, incluyendo aquellas que requieren actualizaciones en tiempo real, en una variedad de fuentes de datos como bases de datos NoSQL, bases de datos relacionales, APIs y fuentes de datos personalizadas con AWS Lambda.

Además proporciona SDKs para aplicaciones móviles y web que soportan el acceso a los datos locales cuando los dispositivos están desconectados, y la sincronización de datos con resolución de conflictos personalizables al reestablecer la conexión a internet.

## **DynamoDB**

Amazon DynamoDB es un servicio serverless de base de datos NoSQL de AWS (Amazon Web Services, s.f.) que proporciona una solución de base de datos completamente autogestionada con robusta seguridad integrada. Permite manejar cargas de trabajo variables con un rendimiento consistente, escalabilidad flexible y escalado automático de capacidad según el tráfico de la aplicación. Además, se integra perfectamente con otros servicios de AWS para potenciar sus capacidades de registro, monitoreo y análisis.

## **CloudWatch**

Amazon CloudWatch es un servicio de monitoreo de AWS (Amazon Web Services, s.f.) que proporciona datos e insights accionables para aplicaciones y recursos de infraestructura de AWS. Permite recolectar y acceder a todos sus datos de rendimiento y operacionales en forma de registros y métricas desde una plataforma única, y utilizar alarmas y eventos para ejecutar acciones automatizadas.

### 8.3.2 Interacciones

La sección de Interacciones describe las comunicaciones esenciales que ocurren entre los elementos mencionados anteriormente. Estas interacciones aseguran el correcto funcionamiento de la aplicación y permiten que los componentes trabajen de manera eficiente y efectiva en conjunto. Al analizar las interacciones, se obtiene una comprensión más profunda de cómo se integran los componentes y cómo colaboran para cumplir con los objetivos de la solución propuesta.

#### A: Aplicación móvil - AWS Cognito

La aplicación móvil comienza el proceso de autenticación para validar su identidad en el grupo de usuarios de Cognito mediante el inicio de sesión. Una vez que se autentica un usuario, Cognito retorna un JSON Web Token del grupo de usuarios a la aplicación. Luego, la aplicación intercambia el token de autenticación por credenciales temporales de AWS para acceder a otros servicios de AWS desde el grupo de identidades de Cognito. Finalmente las credenciales obtenidas admiten las interacciones del tipo E.

El siguiente diagrama ilustra las intercomunicaciones involucradas en la interacción A:

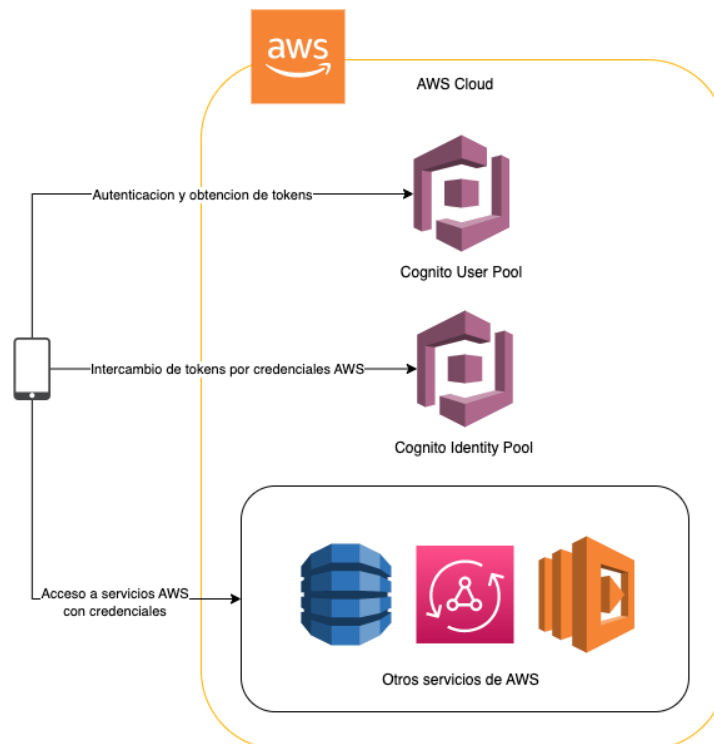


Figura 3: Diagrama de intercomunicaciones en Interacción A

## B: AWS Cognito - AWS Lambda

Cuando el usuario completa el registro en la plataforma, Cognito dispara el activador “ConfirmSignUp” después de que se confirma un nuevo usuario, lo que permite agregar una lógica personalizada.

En este caso, se desencadena una solicitud que contiene los atributos actuales del registro para el usuario confirmado a la función “Post Confirmation Lambda”.

A continuación se presenta el diagrama de la interacción B:

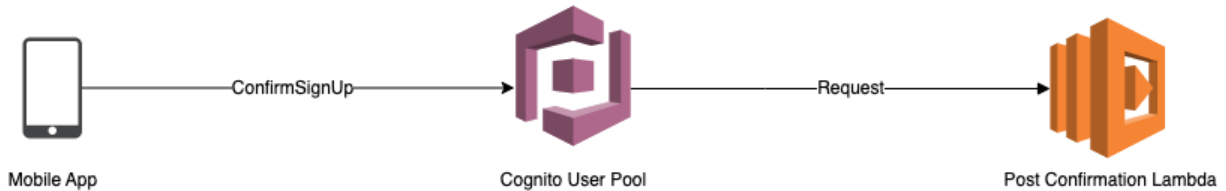


Figura 4: Diagrama de intercomunicaciones en Interacción B

## C: AWS Lambda - AWS DynamoDB

Tras el desencadenamiento de la función “Post Confirmation Lambda”, se diferencia el tipo de usuario que se registra en la plataforma.

Si el usuario es un Padre, se generan entradas en las tablas Parent, Family y Wallet, con sus respectivas asociaciones.

Si el usuario es un Hijo, se generan entradas en las tablas Child, Family, Card y SavingsAccount, con sus respectivas asociaciones.

El siguiente diagrama permite visualizar la interacción C:

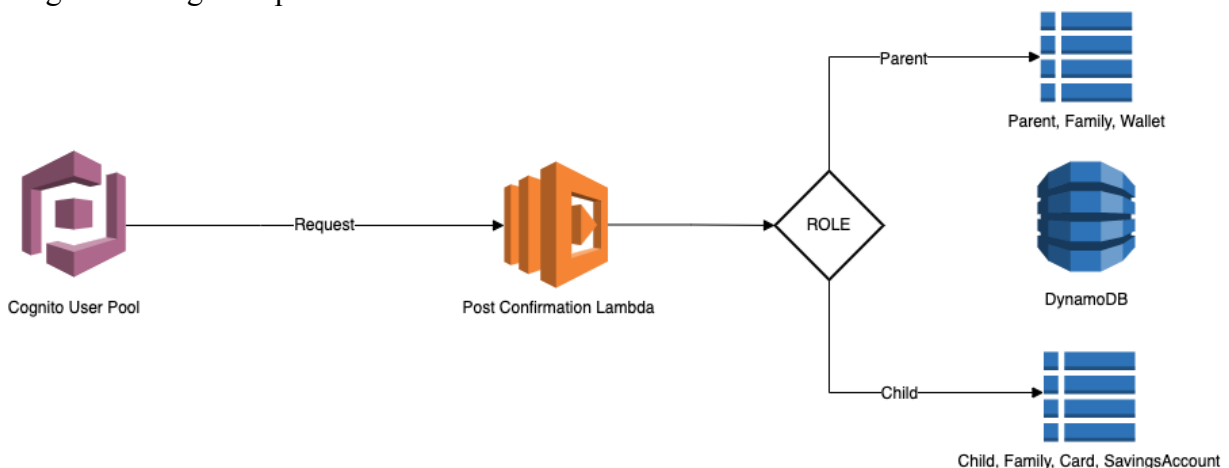


Figura 5: Diagrama de intercomunicaciones en Interacción C

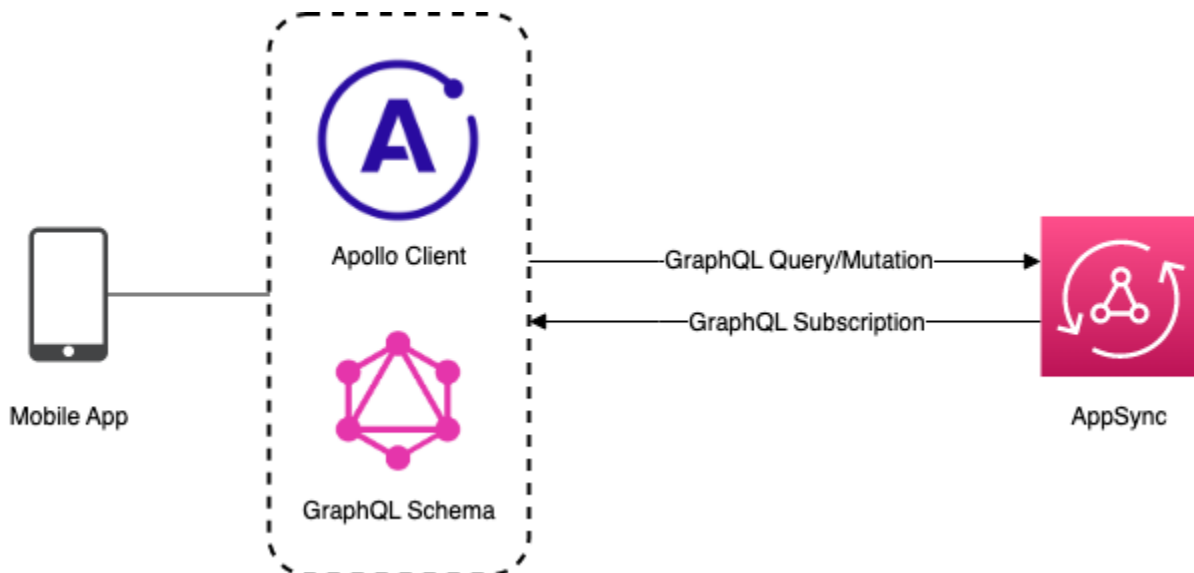
## D: Aplicación móvil - AWS AppSync

La aplicación móvil implementa un cliente Apollo para comunicarse con AWS AppSync, utilizando las directivas centralizadas del esquema GraphQL de la aplicación. Apollo Client permite ofrece almacenamiento en caché y soporte fuera de línea para obtener una aplicación completamente reactiva.

La implementación de las principales operaciones involucradas en los casos de uso definidos emplean este tipo de interacción.

- GraphQL Query: Solicitudes efectuadas por el cliente para obtener datos.
- GraphQL Mutation: Solicitudes efectuadas por el cliente para modificar o crear datos.
- GraphQL Subscription: Mecanismo a través del cual el cliente puede recibir actualizaciones en tiempo real de los datos.

La interacción D se visualiza en el siguiente diagrama:



*Figura 6: Diagrama de intercomunicaciones en Interacción D*

## E: AWS AppSync - DynamoDB

AppSync emplea resolutores GraphQL para almacenar y recuperar datos en tablas de DynamoDB existentes. Este resolutor permite transformar una solicitud entrante de GraphQL en una llamada de DynamoDB y, posteriormente, vuelva a asignar la respuesta de DynamoDB a GraphQL.

Las principales operaciones DynamoDB utilizadas son GetItem, PutItem, UpdateItem, DeleteItem, Query, Scan, Sync, BrachGetItem, BatchDeleteItem, BatchPutItem, TransactGetItems, y TransactWriteItems.



*Figura 7: Diagrama de intercomunicaciones en Interacción E*

## 8.4 Modelo de datos

Se define la estructura y modelo de datos de la implementación de la aplicación a partir del lenguaje de consulta GraphQL (GraphQL, s. f.), el cual a través de directivas y su sinergia automática con AWS Amplify permiten generar bases de datos no relacionales, modelos, atributos, resolutores, definiciones de operaciones CRUD, consultas, suscripciones, relaciones y estructuras de índice de base de datos subyacentes para la optimización de consultas.

A continuación se presenta el siguiente diagrama que ilustra los modelos implementados con el detalle de sus atributos y sus respectivos tipos, incluyendo las relaciones entre dichos modelos y sus cardinalidades correspondientes.

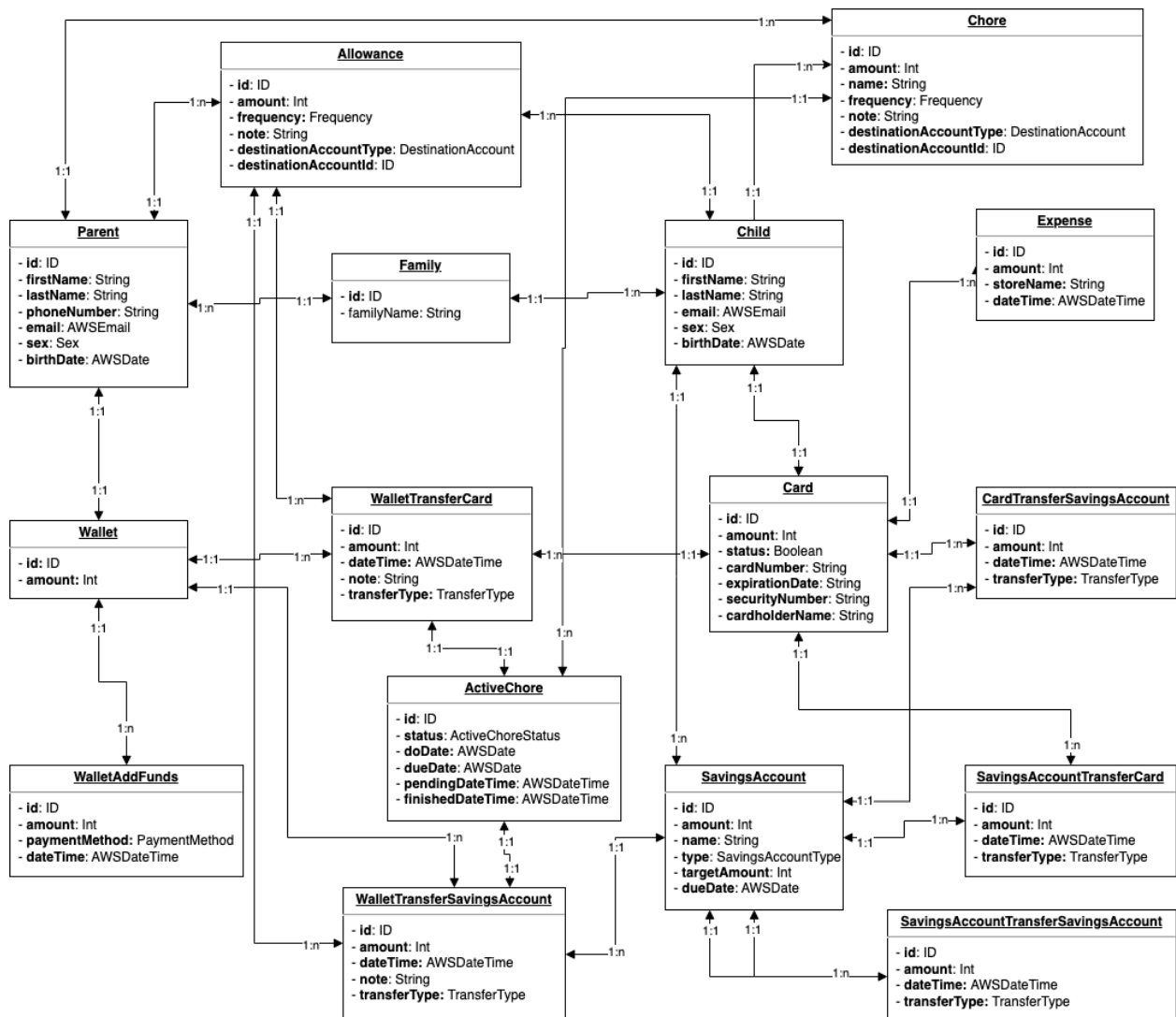


Figura 8: Diagrama de modelado de datos

### 8.4.1 Modelos

Se definen los modelos de datos mediante el lenguaje de definición de esquemas SDL (Schema Definition Language) de GraphQL, el cual gracias a su integración con AWS es transformado en una plantilla completamente descriptiva de AWS CloudFormation que implementa dichas especificaciones.

Los modelos definidos son los siguientes:

**Parent:** Modelo que representa las instancias de la entidad Padre. Ver Anexo 31: Detalle de modelo Parent.

**Child:** Modelo que representa las instancias de la entidad Hijo. Ver Anexo 32: Detalle de modelo Child.

**Family:** Modelo que representa las instancias de la entidad Familia. Ver Anexo 33: Detalle de modelo Family.

**Wallet:** Modelo que representa las instancias de la entidad Billetera. Ver Anexo 34: Detalle de modelo Wallet.

**SavingsAccount:** Modelo que representa las instancias de la entidad Cuenta de Ahorro. Ver Anexo 35: Detalle del modelo SavingsAccount.

**Card:** Modelo que representa las instancias de la entidad Tarjeta. Ver Anexo 36: Detalle de modelo Card.

**ActiveChore:** Modelo que representa las instancias de la entidad Tarea Activa. Ver Anexo 37: Detalle de modelo ActiveChore.

**Chore:** Modelo que representa las instancias de la entidad Tarea. Ver Anexo 38: Detalle de modelo Chore.

**Allowance:** Modelo que representa las instancias de la entidad Mesada. Ver Anexo 39: Detalle de modelo Allowance.

**Expense:** Modelo que representa las instancias de la entidad Gasto. Ver Anexo 40: Detalle de modelo Expense.

**WalletAddFunds:** Modelo que representa las instancias de la operación Agregar Fondos a Billetera. Ver Anexo 41: Detalle de modelo WalletAddFunds.

**WalletTransferCard:** Modelo que representa las instancias de la operación Transferir desde Billetera a Tarjeta. Ver Anexo 42: Detalle de modelo WalletTransferCard.



**WalletTransferSavingsAccount:** Modelo que representa las instancias de la operación Transferir desde Billetera a Cuenta de Ahorro. Ver Anexo 43: Detalle del modelo WalletTransferSavingsAccount.

**CardTransferSavingsAccount:** Modelo que representa las instancias de la operación Transferir desde Tarjeta a Cuenta de Ahorro. Ver Anexo 44: Detalle de modelo CardTransferSavingsAccount.

**SavingsAccountTransferCard:** Modelo que representa las instancias de la operación Transferir desde Cuenta de Ahorro a Tarjeta. Ver Anexo 45: Detalle del modelo SavingsAccountTransferCard.

**SavingsAccountTransferSavingsAccount:** Modelo que representa las instancias de la operación Transferir desde Cuenta de Ahorro a Cuenta de Ahorro. Ver Anexo 46: Detalle del modelo SavingsAccountTransferSavingsAccount.

## 8.4.2 Tipos

GraphQL es un lenguaje fuertemente tipado, por lo tanto se deben definir los tipos de datos que se emplearán en la aplicación. El sistema de tipos permite definir un esquema a partir de tipos heredados por defecto, y personalizados definidos en el proceso de desarrollo.

A continuación se describen los tipos implementados, incluyendo únicamente los que permiten una mayor comprensión del modelo de datos:

### Tipos enumeradores

Los enumeradores son un tipo de escalar que está restringido a un conjunto particular de valores permitidos, facilitando la validez de los argumentos y la comunicación con servicios externos como la aplicación móvil.

- **DestinationAccount:** Cuenta de destino
  - CARD: Tarjeta
  - SAVINGS\_ACCOUNT: Cuenta de ahorro
- **SavingsAccountType:** Tipo de cuenta de ahorro.
  - GENERAL: Cuenta por defecto llamada “Chanchito”
  - CUSTOM: Cuenta personalizada

- **FrequencyType:** Recurrencia de tareas o mesadas.
  - DAILY: Diariamente
  - WEEKLY: Semanalmente
  - MONTHLY: Mensualmente
  - ONE\_TIME: Evento único
  
- **Days:** Días de la semana.
  - MON: Lunes
  - TUE: Martes
  - WED: Miércoles
  - THU: Jueves
  - FRI: Viernes
  - SAT: Sábado
  - SUN: Domingo
  
- **Sex:** Sexo de un usuario.
  - MALE: Masculino
  - FEMALE: Femenino
  - OTHER: Otro
  
- **ActiveChoreStatus:** Estado de una tarea en particular.
  - PENDING: Pendiente de la confirmación y pago del padre.
  - TO\_DO: Aún no es completada por un hijo.
  - FINISHED: Completada y pagada.
  
- **TransferType:** Identifica la naturaleza de una transferencia entre cuentas.
  - TRANSFER: Transferencia manual realizada por un usuario.
  - CHORE: Pago por la finalización de una tarea con recompensa.
  - ALLOWANCE: Pago de la mesada asignada.
  - SAVINGS\_DELETE: Movimiento del dinero restante de una cuenta de ahorro eliminada.
  
- **PaymentMethod:** Corresponde al identificador del medio de pago utilizado para agregar fondos a una billetera.
  - SIMULATION\_MONEY: Dinero de simulación. Es la única opción habilitada en el prototipo.
  - BANK\_TRANSFER: Transferencia bancaria. Deshabilitado.
  - DEBIT\_CARD: Tarjeta de débito. Deshabilitado.
  - CREDIT\_CARD: Tarjeta de crédito. Deshabilitado.

## Tipos de objetos

Representan el tipo de objeto que puede ser obtenido del servicio y que campos incorpora. Son definidos de forma personalizada, ajustados a las necesidades particulares de la aplicación.

- **Frequency:** Detalles de la recurrencia de una Tarea o Mesada
  - [type: FrequencyType]
    - Corresponde al tipo de recurrencia de la tarea o mesada. Sus valores hacen referencia a una recurrencia diaria, semanal, mensual o evento único.
  - day: Days
    - Corresponde al día de la semana asignado en caso de que el tipo de recurrencia sea semanal. Sus valores hacen referencia a los días de la semana: Lunes, Martes, Miércoles, Jueves, Viernes, Sábado o Domingo.
  - date: AWSDate
    - Corresponde a la fecha asignada en caso de que el tipo de recurrencia sea mensual o evento único.

## Tipos escalares

Tipos que representan las hojas de consulta, pues no poseen subcampos y finalmente resuelven las consultas. Los tipos utilizados en la presente implementación, los cuales son definidos de forma predeterminada por GraphQL y AWS AppSync son:

- **ID:** Un identificador único para un objeto. Este escalar se serializa como un String, pero no está destinado a ser legible por humanos.
- **String:** Una secuencia de caracteres UTF-8.
- **Int:** Un valor entero entre  $-(2^{31})$  y  $2^{31}-1$ .
- **Boolean:** Un booleano valor, ya sea verdadero o falso.
- **AWSDate:** Fecha ISO 8601 (ISO, 2004) en el formato YYYY-MM-DD.
- **AWSDateTime:** Fecha y Tiempo ISO 8601 (ISO, 2004) en el formato YYYY-MM-DDThh:mm:ss.sssZ.

## 8.5 Navegación móvil

La implementación de la interfaz gráfica y la experiencia que tienen ambos tipos de usuarios al interactuar con la aplicación está directamente relacionada con el sistema de navegación móvil empleado. La navegación debe ser intuitiva y predecible tanto para usuarios principiantes como para quienes utilizan constantemente aplicaciones móviles.

La aplicación está enfocada en dos grupos de usuarios, padres e hijos, los cuales tienen distintas afinidades y experiencia respecto al uso de medios digitales.

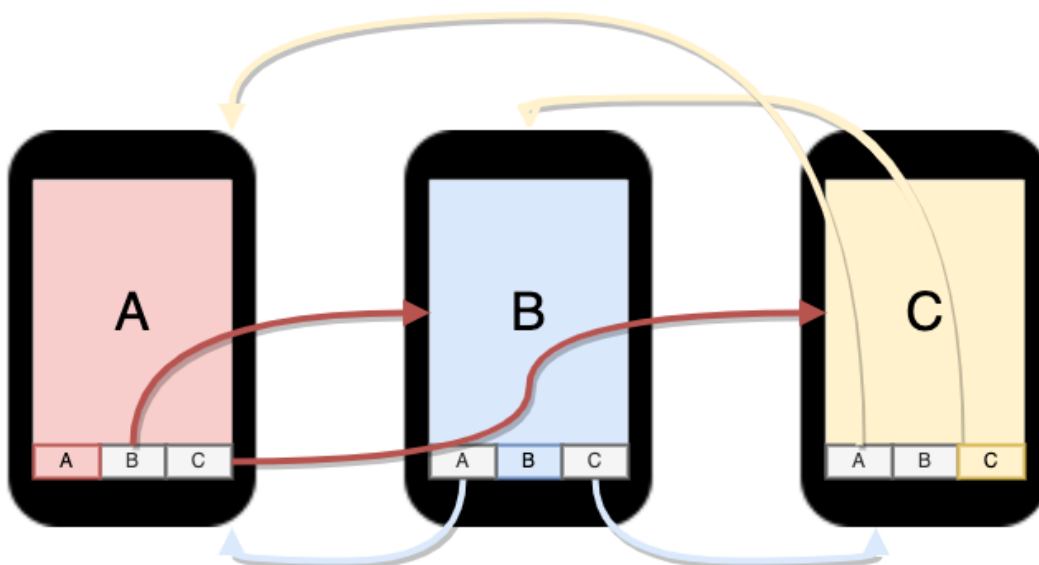
Los hijos pertenecen principalmente a la Generación Z y Generación Alfa, los cuales son grupos demográficos que nunca han conocido un mundo sin internet. Son denominados nativos digitales, y para ellos los dispositivos móviles son herramientas imprescindibles, pues los utilizan por una gran cantidad de horas al día. Aquellos individuos han desarrollado un alto grado de habilidad en el uso de plataformas digitales e intuitivamente comprenden patrones de navegación complejos.

Por su parte, los padres pertenecen a un segmento etario más adulto, en el cual no todos poseen un grado de experticia suficiente para el uso de software más sofisticado y complejo.

Estos puntos son fundamentales para la implementación del sistema de navegación móvil adaptativo de la aplicación, que varía dependiendo del usuario que haga uso de ésta.

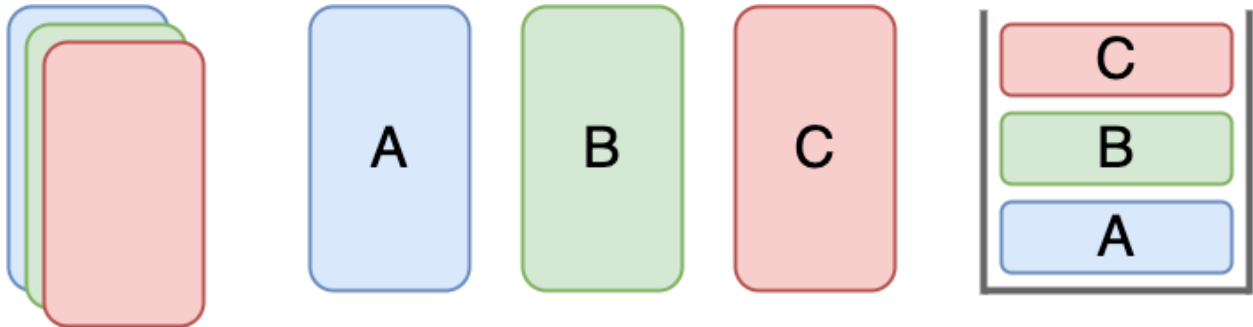
En la aplicación se utilizan dos tipos de navegadores:

- **Bottom Tab Navigator:** Las barras de navegación inferiores muestran de tres a cinco destinos en la parte inferior de una pantalla. Cada destino está representado por un icono y una etiqueta de texto opcional. Cuando se toca un icono de navegación inferior, se lleva al usuario al destino de navegación de nivel superior asociado con ese icono.



*Figura 11: Flujo de navegación Bottom Tab Navigator*

- Stack Navigator: En este tipo de navegación, se mantiene una pila de pantallas. Cada vez que visitamos una pantalla, se empuja a la pila, y al presionar Atrás, el elemento aparecerá de la pila y mostrará la última pantalla.



*Figura 12: Flujo de navegación Stack Navigator*

La navegación de los padres implementa únicamente un Stack Navigator, el cual facilita la navegación pues las únicas operaciones son push y pop de pantallas, las cuales están definidas para facilitar la navegación en función de los casos de usos descritos para este grupo de usuarios.

Por otra parte, la navegación de los hijos implementa como base un Bottom Tab Navigator, el cual posee tres Stack Navigators anidados que engloban distintos aspectos de los casos de uso del hijo: Tarjeta, Ahorros y Ganar dinero.

La siguiente figura presenta el diagrama de navegación que ilustra el sistema de navegación implementado para Padres e Hijos, incluyendo sus respectivos navegadores y pantallas anidadas.

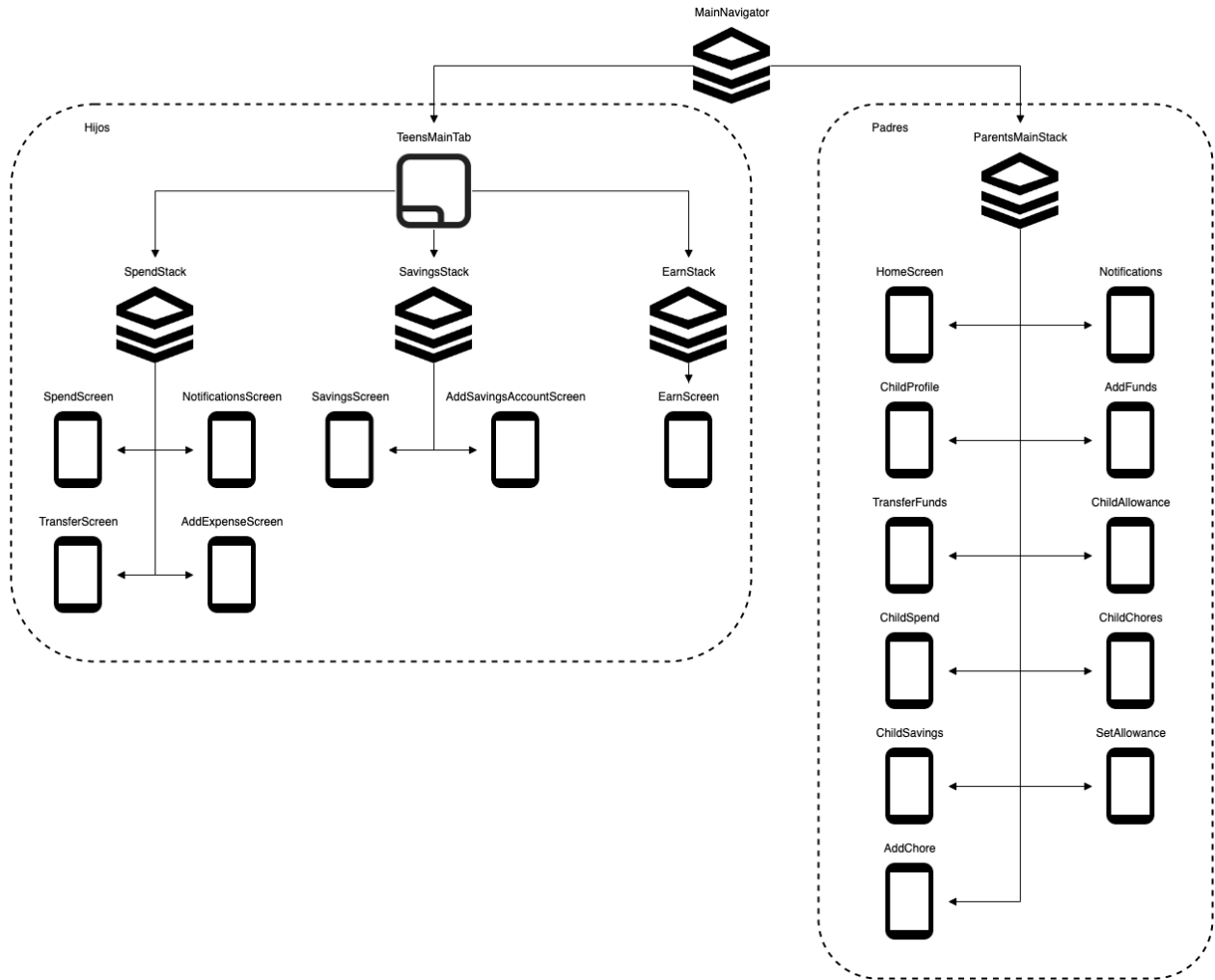


Figura 13: Diagrama de navegadores y pantallas

## 8.6 Proceso de desarrollo

La metodología empleada para el desarrollo de la aplicación prioriza la construcción e integración continua de software a partir de una única fuente de información. De esta manera, el desarrollo de los sistemas back-end y front-end son realizados en un mismo repositorio que integra ambas capas en virtud de las tecnologías utilizadas.

El siguiente diagrama ilustra el proceso de desarrollo continuo de los sistemas front-end y back-end, visualizando las tecnologías involucradas y sus respectivas interacciones:

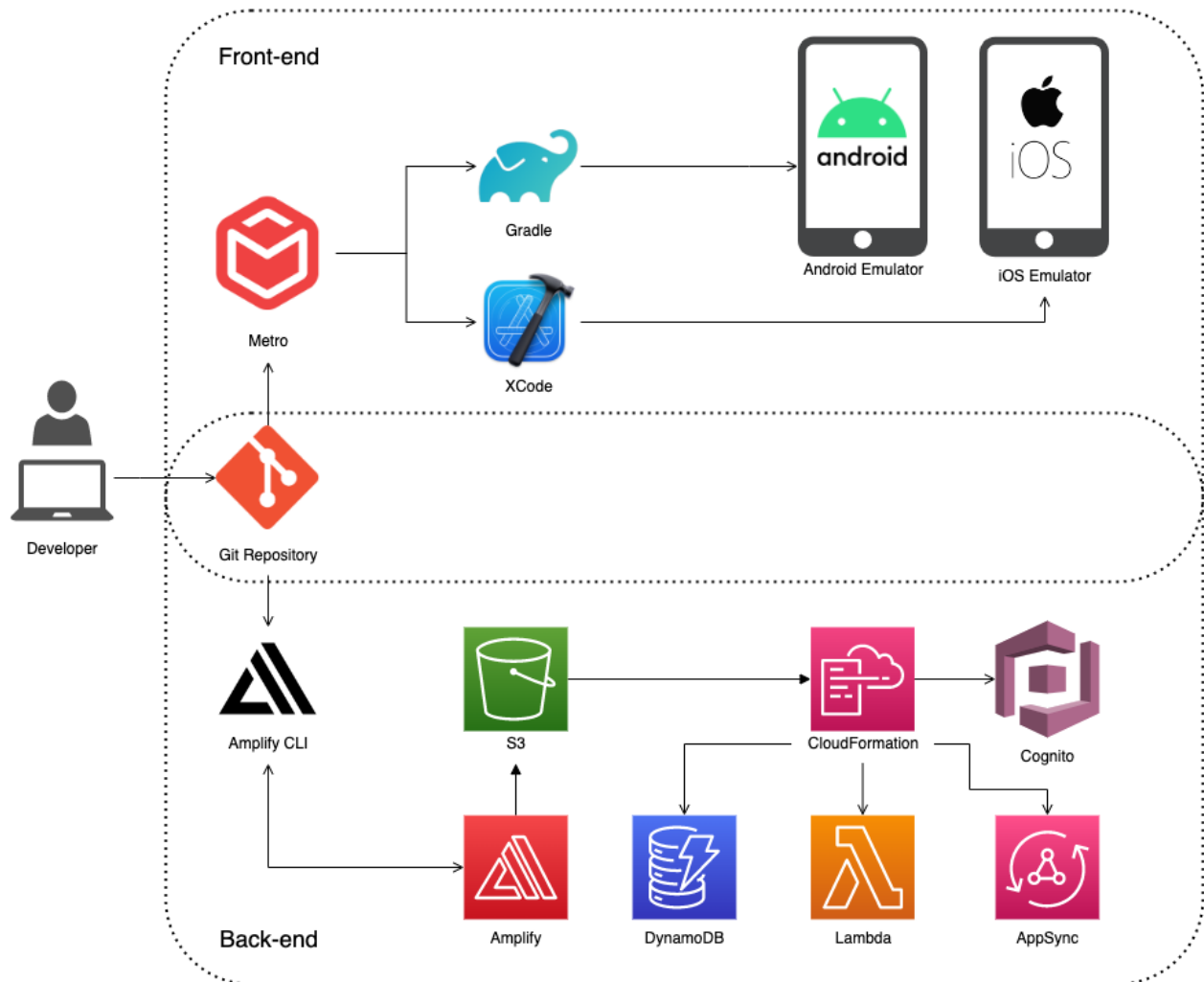


Figura 9: Diagrama de proceso de desarrollo

### **8.6.1 Repositorio Centralizado**

El repositorio centralizado emplea el sistema de control de versiones distribuido Git el cual permite realizar un seguimiento de los cambios del conjunto de archivos de código fuente del sistema. (Chacon, S., & Straub, B., 2014)

Los principales directorios del repositorio incluyen las directrices de AWS Amplify, y los proyectos de Android, iOS y React Native.

El principal beneficio de la incorporación de un repositorio centralizado para el control de las capas front-end y back-end radica en la consistencia al utilizar una única definición de modelo de datos y consultas de comunicación entre ambos sistemas.

Además, este enfoque proporciona una base sólida para la incorporación de pruebas automatizadas E2E y herramientas de supervisión. Estas herramientas podrían implantarse para controlar y verificar la integridad del código antes de su despliegue. Sin embargo, es importante señalar que estos mecanismos automatizados de prueba y supervisión aún no están incorporados en el prototipo actual.

### **8.6.2 Front-end**

El desarrollo del front-end de la aplicación móvil es realizado con React Native, el cual es un framework de código abierto utilizado para el desarrollo de aplicaciones multiplataforma con sus respectivas características nativas. En la presente implementación, se utiliza para el desarrollo de la aplicación móvil en sus versiones para Android e iOS empleando el lenguaje de programación JavaScript.

Metro es un paquete de JavaScript para React Native, el cual a partir de los archivos del código fuente, realiza un proceso de resolución, transformación y serialización, y finalmente genera un único paquete JavaScript para cada plataforma destino.

Dichos paquetes son los insumos para construir las aplicaciones ejecutables para Android e iOS. Gradle es responsable de configurar la compilación de la aplicación Android, mientras que XCode posee la misma responsabilidad para la plataforma iOS. (Gradle Inc, s.f.)

Finalmente, los ejecutables generados son enviados a los dispositivos móviles y ejecutados en un entorno de depuración conectado al servicio de Metro para una mayor flexibilidad en el proceso de desarrollo. El uso de emuladores de dispositivos móviles permite asegurar el correcto funcionamiento y visualización de la aplicación para distintas plataformas, modelos y resoluciones sin la necesidad de poseer dichos dispositivos físicamente.

El procedimiento descrito es ejecutado de forma automática con cada cambio relevante en el código fuente del proyecto de React Native, incorporando optimizaciones que permiten la visualización instantánea de las modificaciones realizadas.



## Librerías

Las principales librerías utilizadas para la aplicación móvil son:

`@apollo/client`: Se trata de una completa biblioteca de gestión de estados que permite gestionar datos locales y remotos con GraphQL. Se utiliza para obtener, almacenar en caché y modificar datos de aplicaciones, mientras se actualiza automáticamente la interfaz de usuario.

`@react-native-async-storage/async-storage`: Es un sistema de almacenamiento clave-valor en React Native. Se utiliza para almacenar el estado de la aplicación y otros datos que deben persistir localmente en el dispositivo.

`@react-native-community/netinfo`: Se utiliza para manejar el estado de la red dentro de la aplicación. Proporciona información acerca de la conexión del usuario (si están en línea o fuera de línea).

`@react-navigation/bottom-tabs`, `@react-navigation/native`, `@react-navigation/native-stack`: Estas librerías permiten la incorporación de sistemas de navegación entre las diferentes pantallas de la aplicación.

`amazon-cognito-identity-js`, `aws-amplify`, `aws-amplify-react-native`, `aws-appsync-auth-link`, `aws-appsync-subscription-link`: Se utilizan para gestionar la autenticación de usuarios y el acceso seguro a sus recursos de AWS.

`graphql`: Esta biblioteca permite construir consultas GraphQL para interactuar con su API. Se utiliza junto con Apollo Client.

`moment`: Se utiliza para analizar, validar, manipular y mostrar fechas y horas en JavaScript. Es útil en una aplicación bancaria para gestionar las marcas de tiempo de las transacciones y funciones similares.

`react`, `react-native`: Son las librerías fundamentales para desarrollar la aplicación. React es una librería JavaScript para construir interfaces de usuario, y React Native extiende esto a móviles.

`react-hook-form`: Esta biblioteca se utiliza para gestionar el estado y la validación de formularios escalables y performantes.

`react-native-ui-lib`: Disponibiliza herramientas y componentes de UI para React Native, que facilita el diseño de la interfaz de la app. Ofrece una amplia variedad de componentes y funciones de interfaz de usuario para acelerar el desarrollo y garantizar un aspecto coherente.

`react-native-vector-icons`: Proporciona una colección de iconos personalizables.

## Emuladores

Los emuladores empleados para ejecutar la aplicación móvil en el proceso de desarrollo y sus respectivas especificaciones son:

- Android

Plataforma	Android
Name	Pixel_2_API_31
CPU/ABI	Google Play ARM (arm64-v8a)
API Level	31
SD Card	512M
Width	1080p
Height	1920p
RAM Size	1536M
Heap Size	256M
LCD Density	420 dpi

*Tabla 5: Especificaciones del emulador Android*

- iOS

Plataforma	iOS
Name	iPhone 12
API Level	iOS 14.4
Width	1170 p
Height	2532 p
LCD Density	460 ppi

*Tabla 6: Especificaciones del emulador iOS*

### 8.6.3 Back-end

El proceso de implementación del back-end de la aplicación móvil utiliza principalmente AWS Amplify, el cual es una solución completa que permite la creación de aplicaciones back-end e interfaces de usuario front-end (AWS Amplify, s.f.).

El desarrollo del back-end de la aplicación es realizado a partir de la interfaz de línea de comandos AWS Amplify CLI, el cual es una colección de herramientas para crear, integrar y gestionar servicios de la nube de AWS (AWS Amplify, s.f.). La ejecución de dichos comandos generan cambios en el código fuente local, y son publicados al servicio remoto de AWS Amplify.

Tras la publicación de los cambios, AWS Amplify detecta las modificaciones en la infraestructura y servicios cloud particulares, genera plantillas JSON de CloudFormation y los envía al servicio de almacenamiento de objetos escalable, segura, de alta disponibilidad y seguridad Amazon S3 (Amazon Web Services, s.f.) y advierte al servicio AWS CloudFormation que existen cambios pendientes en la infraestructura.

AWS CloudFormation (Amazon Web Services, s.f.), servicio de infraestructura como código, obtiene las plantillas almacenadas en S3, revisa los cambios realizados y realiza todas las modificaciones descritas en el stack de servicios cloud de la aplicación. En la presente implementación, los servicios que son actualizados en el proceso de desarrollo son: Cognito, AppSync, Lambda y DynamoDB.

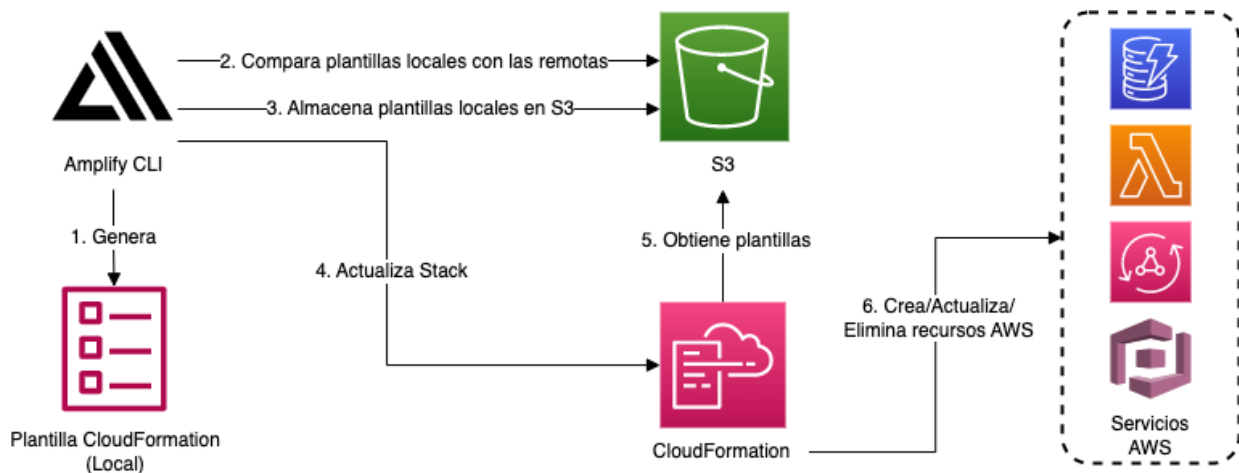


Figura 10: Diagrama del proceso de desarrollo back-end

Este proceso brinda flexibilidad y agilidad al momento del desarrollo de la solución completa, pues las modificaciones de los servicios son consistentes tanto en el ambiente cloud como en el repositorio local que posee las directrices de aquellos servicios.

## 9. Presentación de la Implementación

El propósito principal de esta sección es ofrecer una visión integral y representativa de la aplicación móvil diseñada durante este proyecto. Pretende facilitar una comprensión tanto de la interfaz de usuario como de la experiencia del usuario en un entorno de ejecución.

Para lograr esto, se han incluido capturas de pantalla que proporcionan una representación visual de la aplicación. Estas capturas permiten entender cómo un usuario se sitúa a través de los ciertos estados y secciones disponibles dentro de la aplicación.

Adicionalmente, se puede apreciar la flexibilidad de la interfaz, puesto que se presentan diferencias notables entre las interfaces diseñadas para los usuarios Padres y los usuarios Hijos. Estas diferencias demuestran la adaptabilidad de la aplicación a las necesidades específicas de cada grupo de usuarios.

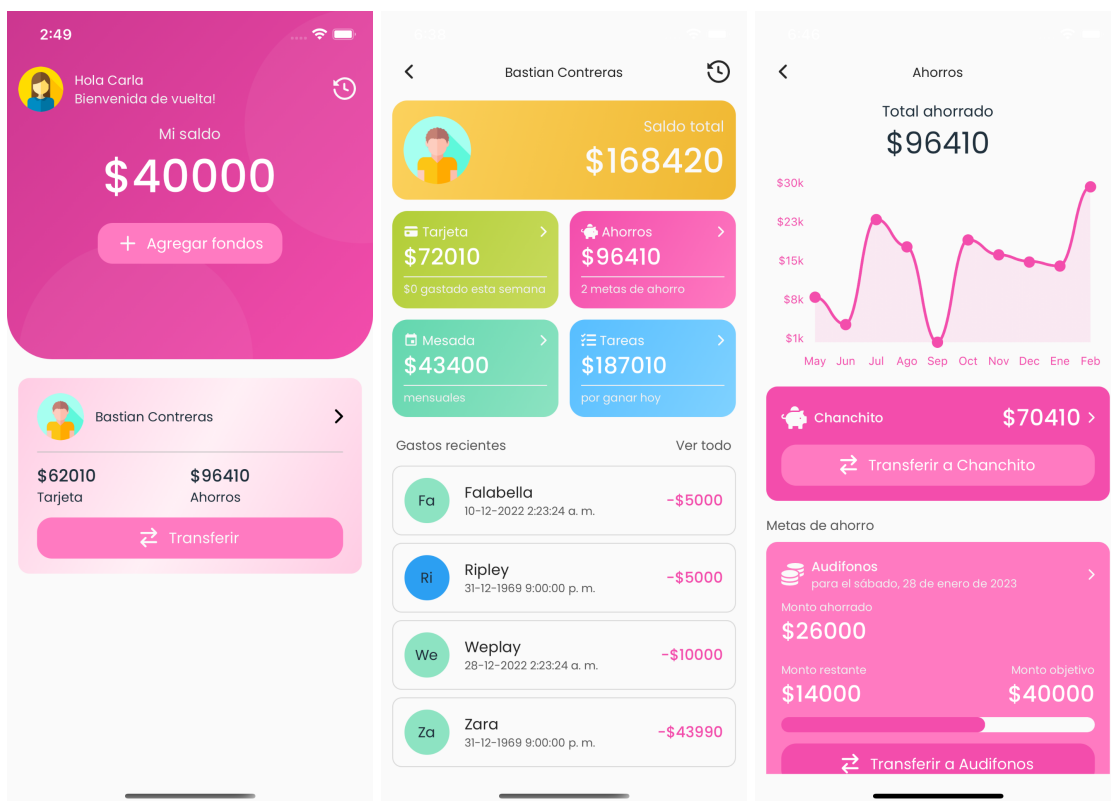


Figura 16: Interfaz Padres: Pantalla de inicio, Detalle del hijo y Detalle de los ahorros del hijo.

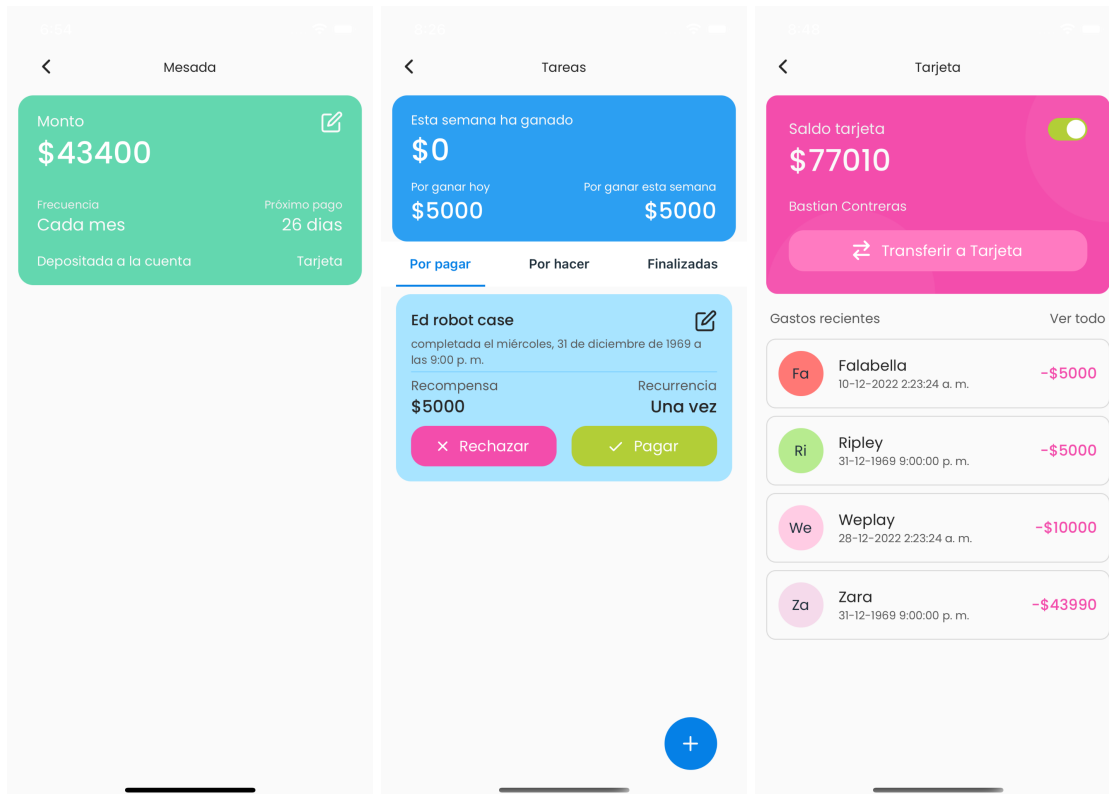


Figura 17: Interfaz Padres: Mesada configurada, Detalle de tareas y Detalle de tarjeta del hijo.

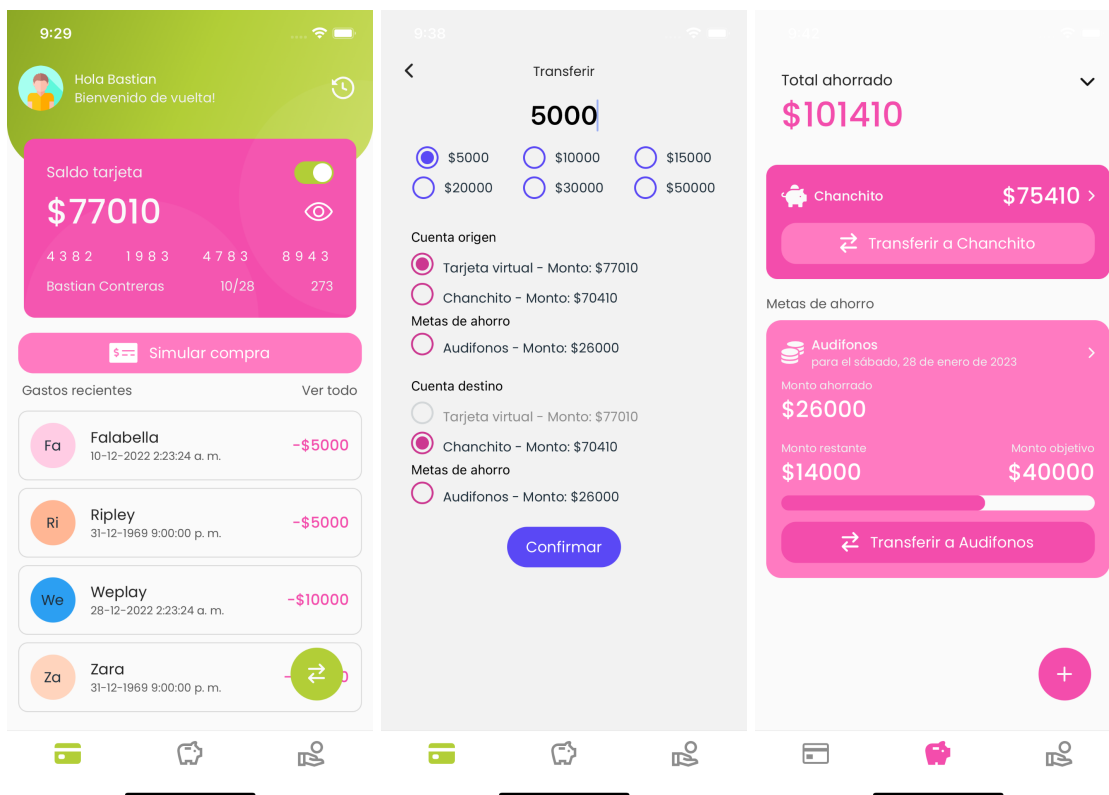


Figura 18: Interfaz Hijos: Pantalla inicial, Transferir entre cuentas y Cuentas de ahorro.

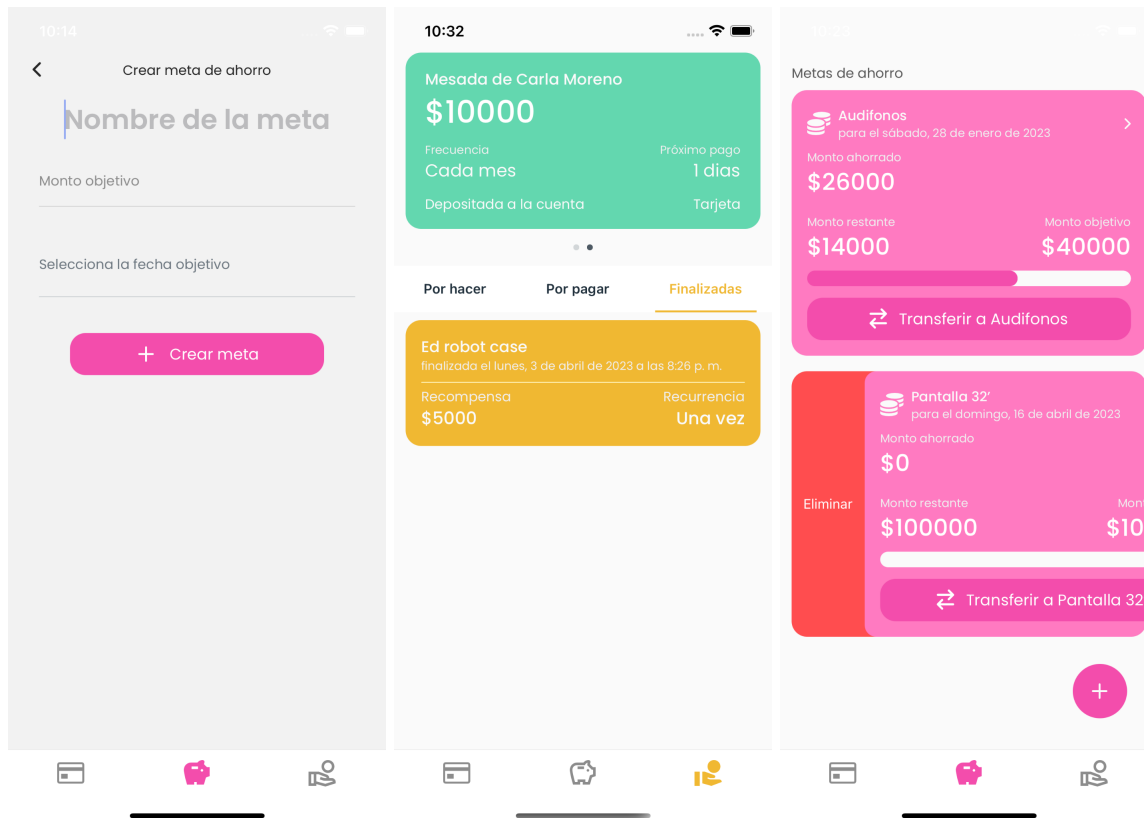


Figura 19: Interfaz Hijos: Crear meta de ahorro, Mesada y tareas y Eliminar meta de ahorro.

## 10. Pruebas de casos de uso

En el contexto del desarrollo de la aplicación móvil prototipo para la gestión cooperativa de finanzas en familia, las pruebas de casos de uso juegan un papel crucial para asegurar la calidad y el correcto funcionamiento de las funcionalidades implementadas (Cockburn, 2001). La validación de funcionalidades permite identificar y corregir posibles errores, así como mejorar la experiencia del usuario y garantizar que los requisitos iniciales se cumplan de manera efectiva.

Para llevar a cabo las pruebas, se han definido y documentado casos de uso que describen el comportamiento esperado de la aplicación, involucrando a los actores relevantes y las interacciones necesarias para completar cada tarea. Los casos de uso también incluyen pre-condiciones, post-condiciones, flujos principales y flujos de excepción que permiten evaluar la aplicación en diversos escenarios y situaciones.

Es importante mencionar que las pruebas se realizaron internamente y no con usuarios reales. Esta decisión fue tomada en base a ciertas consideraciones. La planificación y los recursos requeridos para las pruebas con usuarios reales excedían las capacidades disponibles, por lo que están fuera del alcance de este proyecto particular.

Adicionalmente, se consideró la sensibilidad de los datos manejados por la aplicación. Dado que la aplicación procesa información financiera y personal, se decidió no exponer a usuarios reales a posibles riesgos de privacidad y seguridad durante la fase de desarrollo.

Finalmente, dada la escala limitada del proyecto, las pruebas internas se consideraron suficientes para evaluar las funcionalidades implementadas.

En esta sección, se presentan los resultados de las pruebas de casos de uso, incluyendo las métricas recolectadas, sus respectivos análisis, visualizaciones de datos. Las métricas consideradas son:

- Resultado: Indica el estado de finalización de la prueba, categorizado en exitoso, exitoso con imprecisión y fallido.
- Interacciones para completar la tarea: Refleja la cantidad de interacciones que el usuario realiza con la interfaz para lograr la completitud de la tarea desde el contexto señalado en la precondición.
- Interacciones para completar la tarea desde la pantalla inicial: Similar a la métrica anterior, pero considerando el inicio desde la pantalla inicial.
- Tiempo de ejecución: Mide el tiempo transcurrido desde el comienzo de la tarea hasta su completitud, teniendo en cuenta que dentro de este tiempo se realizan capturas de pantalla y las interacciones son realizadas con una velocidad controlada simulando un usuario no experto.

Ver Anexo 47: Ejecución de pruebas de casos de uso, donde se detallan los resultados obtenidos para cada ejecución de prueba junto con capturas de pantalla.

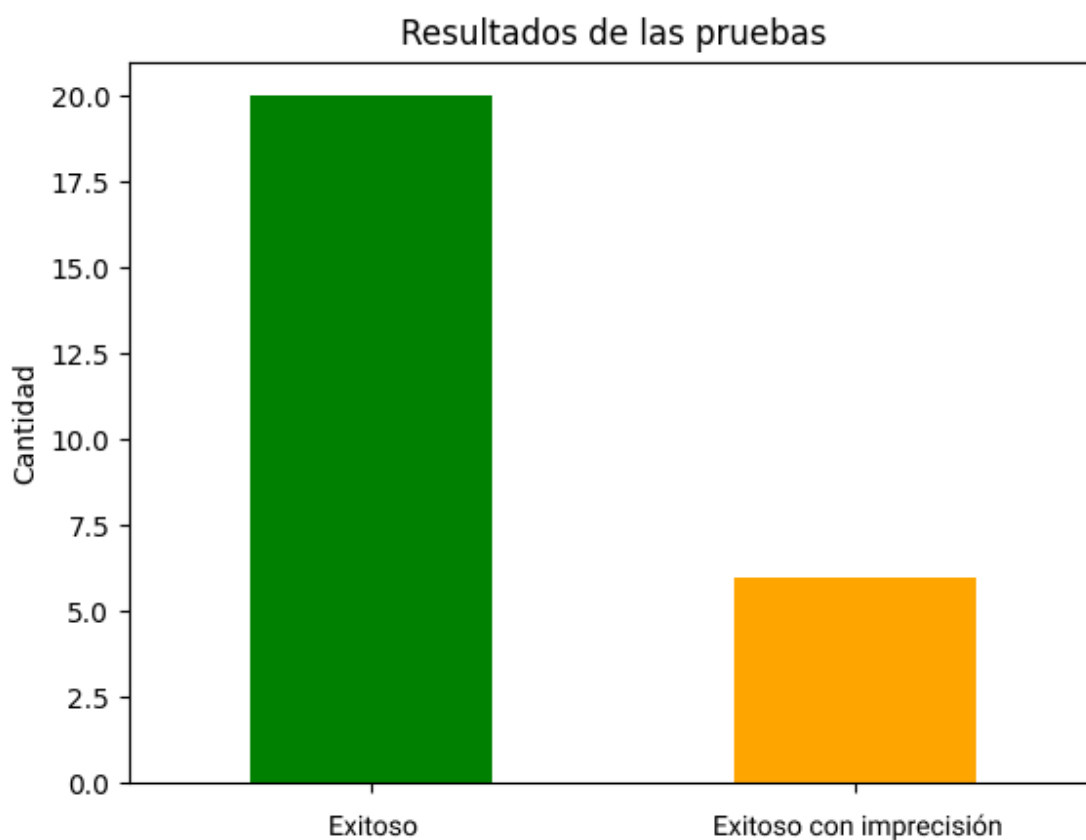
<b>Caso de prueba</b>	<b>Resultado</b>	<b>Tiempo de ejecución</b>	<b>Interacciones para completar la tarea</b>	<b>Interacciones desde pantalla inicial</b>
1	Exitoso	16 s	3	3
2	Exitoso	15 s	3	3
3	Exitoso	12 s	1	1
4	Exitoso	6 s	1	1
5	Exitoso	6 s	1	2
6	Exitoso	6 s	1	2
7	Exitoso con imprecisión	16 s	3	5
8	Exitoso con imprecisión	12 s	3	5
9	Exitoso	10 s	2	4
10	Exitoso	6 s	1	2
11	Exitoso con imprecisión	26 s	3	5
12	Exitoso	9 s	1	3
13	Exitoso	9 s	1	3
14	Exitoso	6 s	1	2
15	Exitoso	7 s	1	3
16	Exitoso	6 s	1	1
17	Exitoso	5 s	1	1
18	Exitoso	19 s	3	3
19	Exitoso	8 s	1	1
20	Exitoso con imprecisión	22 s	3	4
21	Exitoso con imprecisión	21 s	4	5
22	Exitoso	11 s	2	3
23	Exitoso	6 s	1	1
24	Exitoso con imprecisión	12 s	2	3
25	Exitoso	12 s	3	3
26	Exitoso	6 s	1	0

*Tabla 7: Resultados de las pruebas de casos de uso*



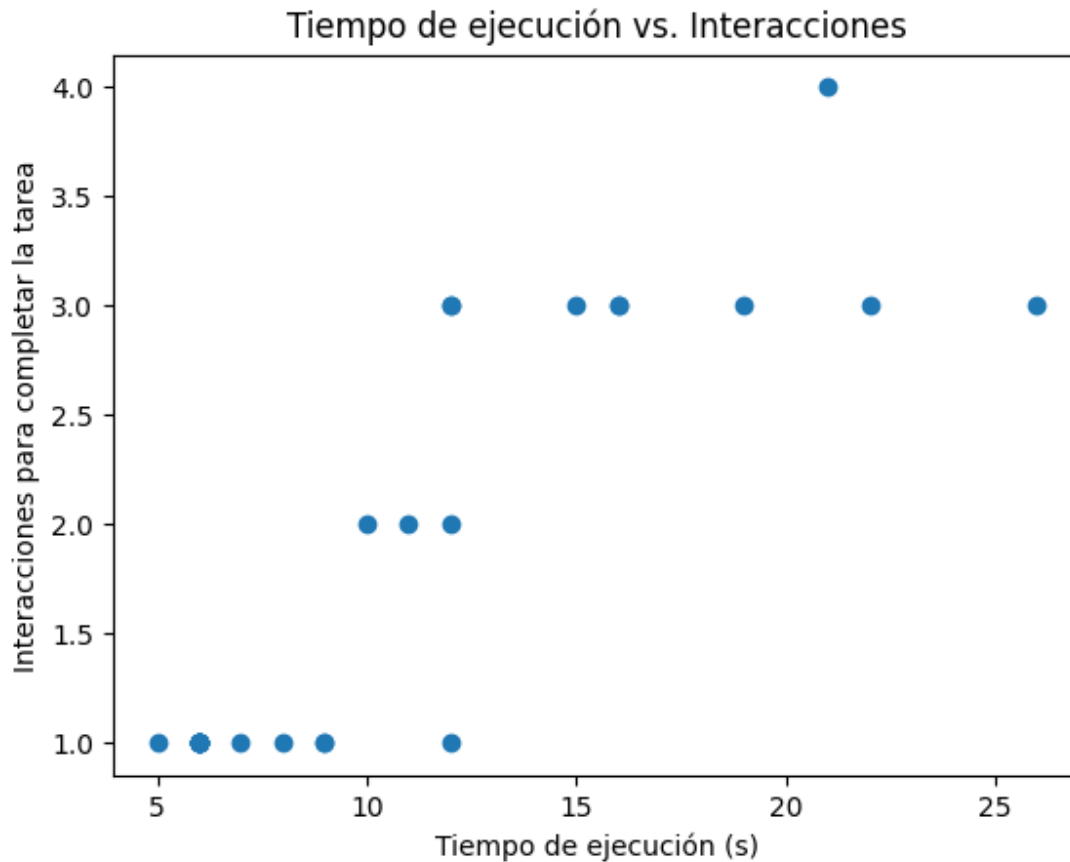
A continuación, se presenta el análisis de los resultados obtenidos para cada caso de prueba.

- Casos de prueba exitosos: La mayoría de los casos de prueba (18 de 26) resultaron en un éxito, lo que indica que las funcionalidades implementadas funcionan correctamente y cumplen con los requisitos iniciales.
- Casos de prueba exitosos con imprecisión: Se encontraron 5 casos de prueba con un resultado exitoso pero con imprecisión. Estos casos indican que, aunque las funcionalidades se completaron correctamente, existen oportunidades para mejorar la experiencia del usuario y ajustar el diseño de la interfaz.
- Tiempos de ejecución: Los tiempos de ejecución varían desde 5 segundos hasta 26 segundos. La mayoría de los casos de prueba tienen tiempos de ejecución cortos (menos de 10 segundos), lo que sugiere que la aplicación es ágil y permite a los usuarios realizar tareas rápidamente. Sin embargo, algunos casos de prueba con resultados exitosos con imprecisión presentan tiempos de ejecución más largos, lo que indica que se podrían realizar mejoras en la eficiencia de la aplicación.
- Interacciones para completar la tarea: El número de interacciones para completar la tarea varía entre 1 y 4. La mayoría de los casos de prueba requieren pocas interacciones, lo que sugiere una aplicación intuitiva y fácil de usar.
- Interacciones para completar la tarea desde la pantalla inicial: Los resultados muestran que, en general, las tareas se pueden completar con pocas interacciones desde la pantalla inicial. Esto indica que la aplicación está bien organizada y estructurada, permitiendo a los usuarios acceder a las funcionalidades de manera rápida y eficiente.



*Figura 14: Gráfico de barras de resultados de las pruebas*

Se puede observar que la mayoría de las pruebas fueron exitosas, lo que indica que la aplicación móvil prototipo cumple con las funcionalidades esperadas en gran medida. Sin embargo, también se presentaron casos de éxito con imprecisión y, aunque no se reportaron casos fallidos, es importante analizar y mejorar las situaciones en las que se presentaron imprecisiones para optimizar la experiencia del usuario.



*Figura 15: Gráfico de dispersión de tiempo de ejecución vs interacciones*

La Figura 15 muestra un gráfico de dispersión que relaciona el tiempo de ejecución con la cantidad de interacciones necesarias para completar la tarea. No se observa una correlación clara entre estas dos métricas, lo que sugiere que el tiempo de ejecución no depende únicamente del número de interacciones. Esto puede deberse a factores como la complejidad de la tarea, la familiaridad del usuario con la aplicación o las diferencias en la velocidad de interacción entre ejecuciones.

## 11. Conclusiones

La problemática de la educación e inclusión financiera en Chile, especialmente en el ámbito familiar y enfocado en niños y adolescentes, es un reto sustancial que requiere soluciones innovadoras. Esta memoria de título ha abordado precisamente este desafío, a través de la concepción y desarrollo de un prototipo de aplicación móvil orientada a la gestión cooperativa de finanzas en familia.

Hemos estudiado en detalle el contexto tanto nacional como internacional, identificando las soluciones existentes y evaluando su eficacia, así como también hemos detectado brechas y oportunidades de mejora. Este análisis nos ha permitido definir un mercado objetivo y estimar su magnitud, lo cual ha sido crucial para la elaboración de un 'Business Model Canvas' sólido y viable.

La solución propuesta combina la tecnología de vanguardia con una visión enfocada en las necesidades humanas. La aplicación móvil, construida con React Native y AWS Amplify en una arquitectura serverless, permite una alta escalabilidad y adaptabilidad, proporcionando al mismo tiempo una plataforma accesible y efectiva.

A lo largo del proceso de desarrollo, se ha trabajado con rigor en la definición de los casos de uso, la estructuración de la arquitectura de software, el modelado de datos y la navegación móvil, asegurando que cada componente contribuya a la funcionalidad y usabilidad de la aplicación. Se han realizado pruebas internas para validar su funcionamiento y garantizar que cumple con las expectativas planteadas.

Para finalizar, es fundamental recordar que la tecnología, en sí misma, no es la solución a los problemas de educación financiera e inclusión financiera. La verdadera transformación proviene de la integración de la tecnología con un enfoque humano y comprensivo, que permita a las familias construir un futuro sólido y sostenible, basado en la colaboración, el conocimiento y la responsabilidad financiera.

## 12. Trabajo futuro

Aunque el presente proyecto ha logrado la concepción y desarrollo de un prototipo de aplicación móvil para la gestión cooperativa de finanzas en familia, es necesario continuar trabajando en aspectos clave para optimizar y validar su efectividad. En esta sección, se presentan diversas áreas de mejora y acciones a abordar en futuras iteraciones del proyecto.

- Corregir imprecisiones en la visualización de fechas: Durante las pruebas de casos de uso, se encontraron discrepancias en la visualización de fechas, lo que sugiere la necesidad de mejorar la estrategia para trabajar con ellas. Se recomienda investigar y aplicar técnicas que permitan una correcta representación de las fechas en función de la zona horaria de Chile, considerando la diferencia con la zona horaria de los servicios de AWS utilizados en us-east-2 Ohio.
- Pulir interfaz gráfica: Se observaron inconsistencias en el diseño y la paleta de colores en algunas pantallas, especialmente en formularios. Para mejorar la experiencia del usuario, es fundamental mantener la coherencia en el diseño y los colores a lo largo de la aplicación adaptada al tipo de usuario, siguiendo las directrices del sistema de diseño empleado.
- Realizar pruebas con usuarios reales: Con el fin de validar la efectividad y usabilidad del prototipo, es crucial llevar a cabo pruebas con usuarios reales, abarcando aspectos como pruebas de usabilidad, donde se observen y registren las interacciones y dificultades de los usuarios al completar tareas específicas (Nielsen, 1993), y análisis de métricas de uso, implementando herramientas de seguimiento para recopilar datos como duración de sesiones y funcionalidades más utilizadas (Churchill, 2019). Estos métodos permitirán obtener retroalimentación valiosa y detectar oportunidades de mejora en la aplicación.
- Validar product-market fit: Para garantizar el éxito del proyecto y su adopción por parte de los usuarios, es esencial validar el ajuste entre el prototipo y las necesidades del mercado. Esto puede lograrse mediante la difusión de la solución a través de la creación de contenidos, demostraciones, promoción digital, entre otros, mostrando las capacidades del prototipo para gestionar dinero cooperativamente. De esta forma, se podrá obtener información sobre el interés y la aceptación del público objetivo, permitiendo ajustar las características y funcionalidades de la aplicación según las demandas y requerimientos específicos de los usuarios finales.
- Incorporar gamificación: Para incrementar la motivación y el compromiso de los usuarios jóvenes, se sugiere explorar la incorporación de elementos de gamificación en la aplicación. Estos elementos pueden incluir recompensas virtuales, insignias, logros, desafíos y tablas de clasificación, lo que fomentará la adopción de buenos hábitos financieros y la educación financiera de forma lúdica y entretenida.
- Ampliar la integración con instituciones financieras: Para mejorar la utilidad y practicidad de la aplicación, sería beneficioso ampliar su integración con otras

instituciones financieras y servicios, tales como bancos, cooperativas de ahorro y crédito, y fintechs. Esto permitiría a los usuarios tener acceso a una gama más amplia de productos financieros y herramientas de gestión, y posibilitará la transición de la gestión de dinero de simulación a dinero real, aumentando el impacto y alcance del proyecto.

- Implementar medidas de seguridad adicionales: Dado que la aplicación maneja información financiera, es crucial garantizar la seguridad y protección de los datos de los usuarios. Se recomienda investigar e implementar medidas de seguridad adicionales, como la autenticación de dos factores, cifrado de datos en tránsito y en reposo, y auditorías de seguridad periódicas para mantener la integridad de la plataforma y la confianza de los usuarios.
- Incorporar pruebas automatizadas: A pesar de las pruebas manuales realizadas hasta el momento, para garantizar la calidad del código y la funcionalidad correcta de la aplicación, es vital implementar pruebas automatizadas. Esto incluye pruebas unitarias, que se concentran en partes específicas del código para garantizar que funcionen como se espera; pruebas de integración, que se enfocan en asegurar que diferentes partes del código funcionen correctamente cuando se combinan; y pruebas end-to-end (e2e), que evalúan la funcionalidad de la aplicación desde la perspectiva del usuario final.
- Evaluación y seguimiento del impacto en la alfabetización financiera: Es importante establecer métricas y sistemas de seguimiento que permitan evaluar el impacto del prototipo en la adopción de habilidades y conocimientos financieros por parte de los usuarios. Estos indicadores pueden incluir la frecuencia de uso de la aplicación, la cantidad de objetivos financieros alcanzados, y la evolución en el comportamiento financiero de los usuarios a lo largo del tiempo.

Estas son algunas de las oportunidades para mejorar y validar el proyecto, lo cual permitirá reducir las incertidumbres respecto a su efectividad y adopción por parte de los usuarios finales.

## Referencias

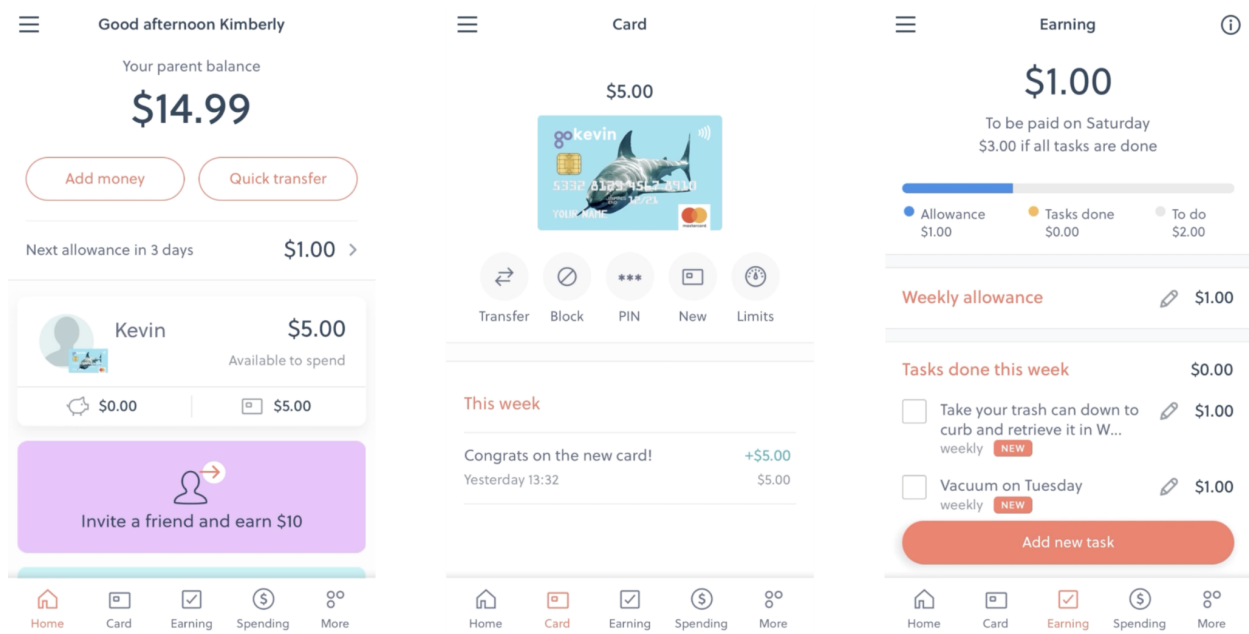
- García, N., Grifoni, A., López, J. C., Mejía, D. M. (2017). *La educación financiera en América Latina y el Caribe Situación actual y perspectivas*. [https://www.oecd.org/daf/fin/financial-education/OECD\\_CAF\\_Financial\\_Education\\_Latin\\_AmericaES.pdf](https://www.oecd.org/daf/fin/financial-education/OECD_CAF_Financial_Education_Latin_AmericaES.pdf)
- OCDE. (2005). *Recomendación sobre los Principios y Buenas Prácticas de Educación y Concienciación Financiera*. [https://www.oecd.org/daf/fin/financial-education/\[ES\]+Recomendaci%C3%B3n+Principios+de+Educaci%C3%B3n+Financiera+2005.pdf](https://www.oecd.org/daf/fin/financial-education/[ES]+Recomendaci%C3%B3n+Principios+de+Educaci%C3%B3n+Financiera+2005.pdf)
- OECD (2019). *PISA 2018 Results (Volume I): What Students Know and Can Do*. <https://doi.org/10.1787/5f07c754-en>
- OECD (2023). *OECD INTERNATIONAL NETWORK ON FINANCIAL EDUCATION*. <https://www.oecd.org/financial/education/oecd-infe-overview.pdf>
- World Bank. (2022). *Financial Inclusion*. <https://www.worldbank.org/en/topic/financialinclusion/overviewOECD>.
- (2020). *Recommendation of the Council on Financial Literacy*. <https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0461>
- Russia's G20 Presidency & OECD. (2013). *Advancing National Strategies for Financial Education*. [https://www.oecd.org/finance/financial-education/G20\\_OECD\\_NSFinancialEducation.pdf](https://www.oecd.org/finance/financial-education/G20_OECD_NSFinancialEducation.pdf)
- Agencia de Calidad de la Educación (2020). *PISA 2018 Educación Financiera Entrega de Resultados*. [https://archivos.agenciaeducacion.cl/PISA\\_2018\\_Educacion\\_Financiera.pdf](https://archivos.agenciaeducacion.cl/PISA_2018_Educacion_Financiera.pdf)
- VTR & CRITERIA. (2022). *RADIOGRAFÍA DIGITAL NIÑ@S Y ADOLESCENTES DE CHILE 2023*. <https://www.vtr.com/radiografia-digital>
- Censo digital (2018). Subsecretaría de Telecomunicaciones y VTR [https://www.subtel.gob.cl/wp-content/uploads/2019/02/PPT\\_Internet\\_Segura-1.pdf](https://www.subtel.gob.cl/wp-content/uploads/2019/02/PPT_Internet_Segura-1.pdf)
- Universidad de los Andes. (2018) “Efectos de la adicción al celular en los colegios”
- Next Gen Personal Finance (2020) “What percent of 13-17 year olds have a debit card”
- *Tu MACH*. (s. f.). <https://www.somosmach.com/tumach>
- Banco Estado. (s. f.). <https://www.bancoestado.cl/>
- Banco de Chile. (s. f.). *Cuenta FAN CLAN para jóvenes entre 14 y 17 años*. <https://portales.bancochile.cl/cuenta-fan-clan>
- GoHenry (s. f.). *GoHenry*. <https://www.gohenry.com/>
- Greenlight (s. f.). *Greenlight*. <https://greenlight.com/>
- RoosterMoney (s. f.). *RoosterMoney*. <https://roostermoney.com/>
- Instituto Nacional de Estadísticas INE. (2017) *Resultados Censo 2017*. <http://resultados.censo2017.cl/>
- Ministerio de Desarrollo Social y Familia (2017). *Encuesta Casen 2017*. <http://observatorio.ministeriodesarrollosocial.gob.cl/encuesta-casen-2017>

- Ministerio de Desarrollo Social y Familia (2020). *Niños, niñas y adolescentes Casen en Pandemia* 2020. <http://observatorio.ministeriodesarrollosocial.gob.cl/encuesta-casen-en-pandemia-2020>
- Osterwalder, A., & Pigneur, Y. (2010). *Business model generation*. John Wiley & Sons
- IBM. (s. f.). *Use-case diagrams*. <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>
- React Native. (s.f.). *React Native*. <https://reactnative.dev/>
- Apollo GraphQL. (s.f.). *Apollo Client*. <https://www.apollographql.com/apollo-client/>
- Amazon Web Services. (s.f.). *Amazon Web Services*. <https://aws.amazon.com>
- Amazon Web Services. (s.f.). *Amazon Cognito*. <https://aws.amazon.com/cognito/>
- Amazon Web Services. (s.f.). *Amazon Lambda*. <https://aws.amazon.com/lambda/>
- Amazon Web Services. (s.f.). *Amazon AppSync*. <https://aws.amazon.com/appsync/>
- Amazon Web Services. (s.f.). *Amazon DynamoDB*. <https://aws.amazon.com/dynamodb/>
- Amazon Web Services. (s.f.). *Amazon CloudWatch*. <https://aws.amazon.com/cloudwatch/>
- Amazon Web Services. (s.f.). *Amazon S3*. <https://aws.amazon.com/s3/>
- GraphQL. (s.f.). *GraphQL* <https://graphql.org/>
- International Organization for Standardization. (2004). *ISO 8601:2004 - Data elements and interchange formats - Information interchange - Representation of dates and times*. <https://www.iso.org/standard/40874.html>
- Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.
- Gradle Inc. (s. f.). *Gradle build tool*. <https://gradle.org/>
- AWS Amplify (s. f.). *Amplify*. <https://aws.amazon.com/es/amplify/>
- Cockburn, A. (2001). *Writing Effective Use Cases*. Addison-Wesley Professional.
- Churchill, E. F. (2019). *Foundations for Designing User-Centered Systems: What System Designers Need to Know about People*. Springer.
- Nielsen, J. (1993). *Usability Engineering*. Academic Press.

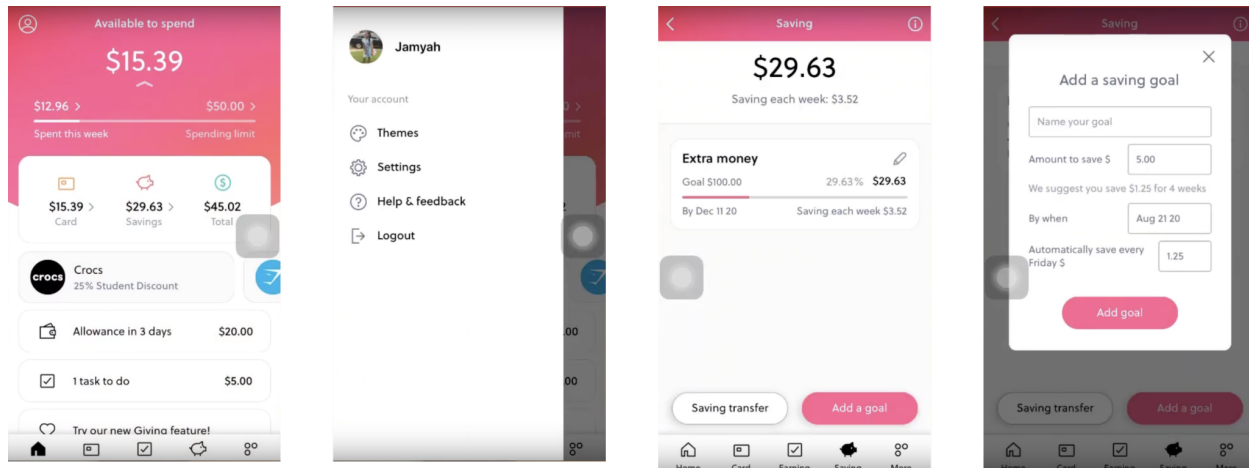


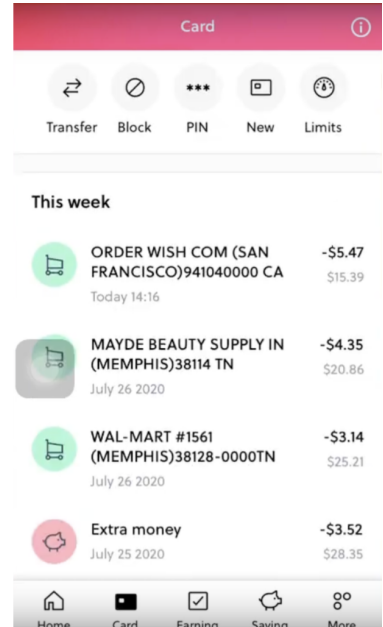
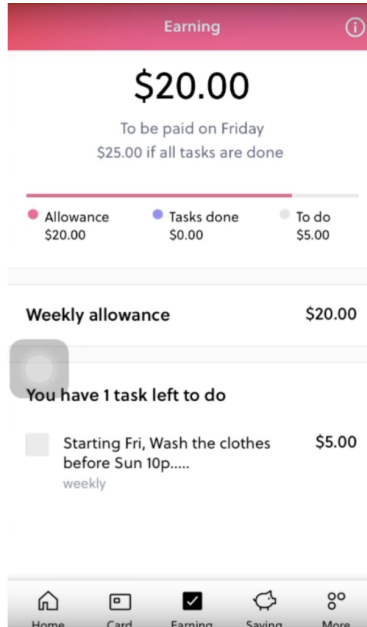
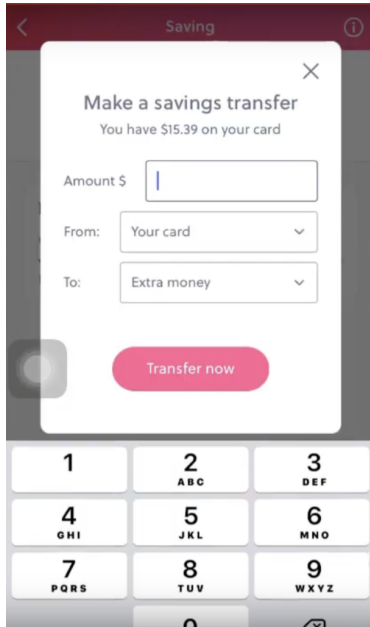
# Anexos

## Anexo 1: Interfaz gráfica para padres GoHenry

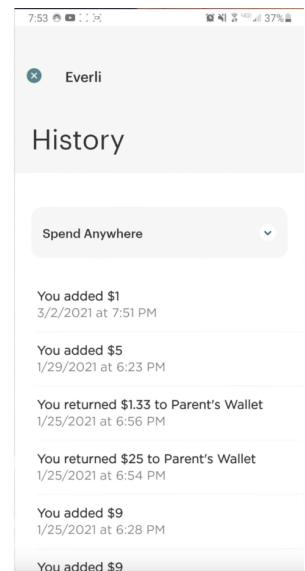
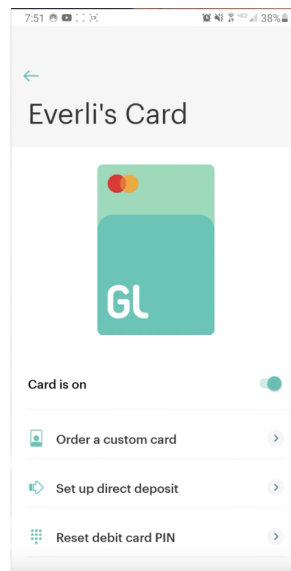
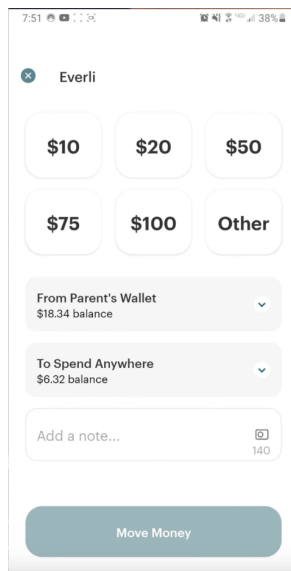
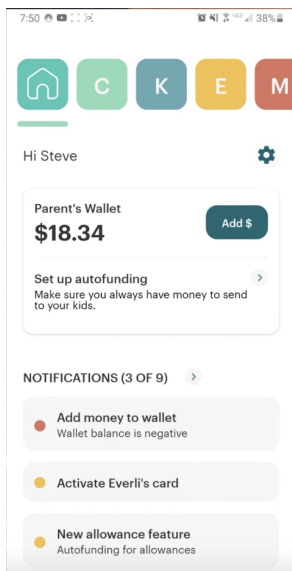


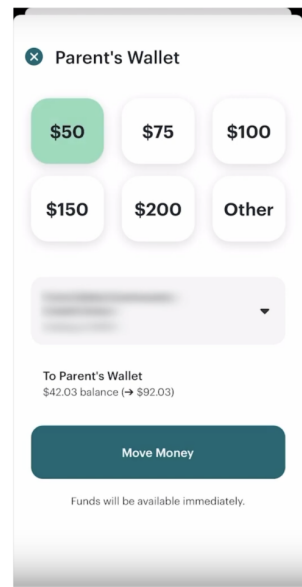
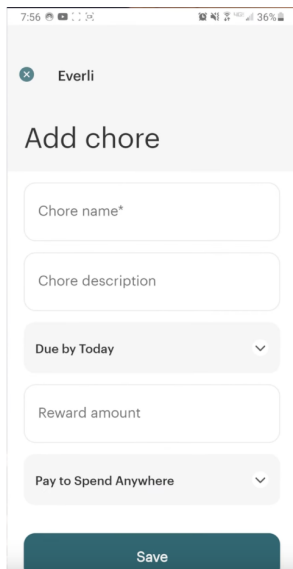
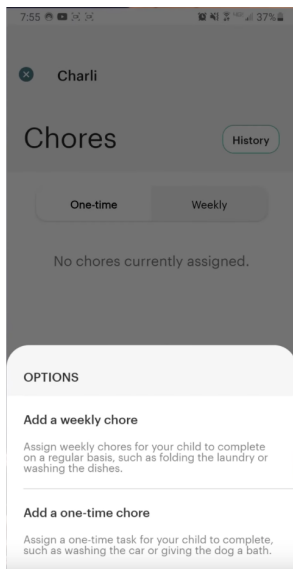
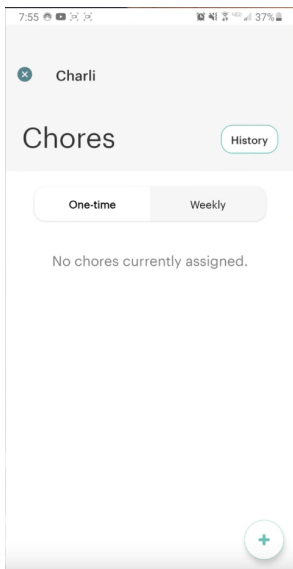
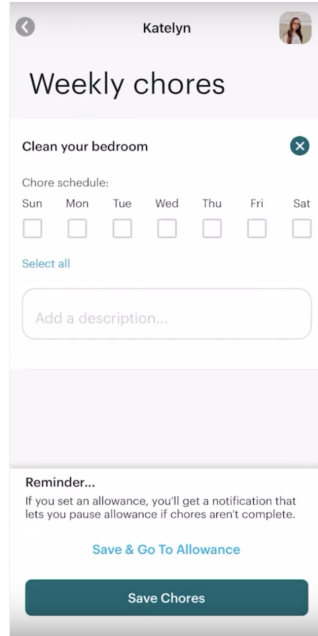
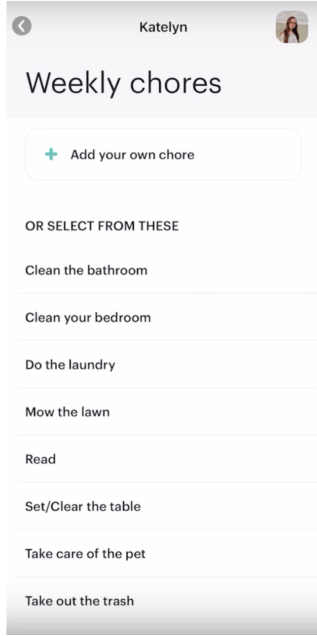
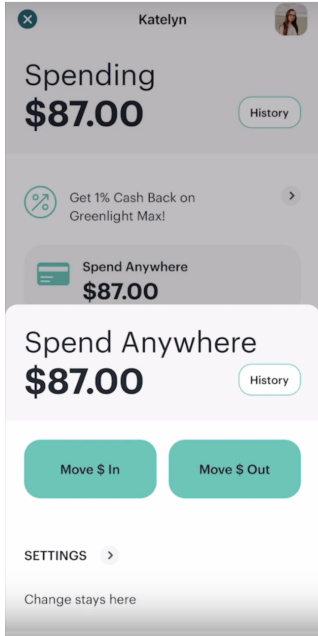
## Anexo 2: Interfaz gráfica para hijos GoHenry

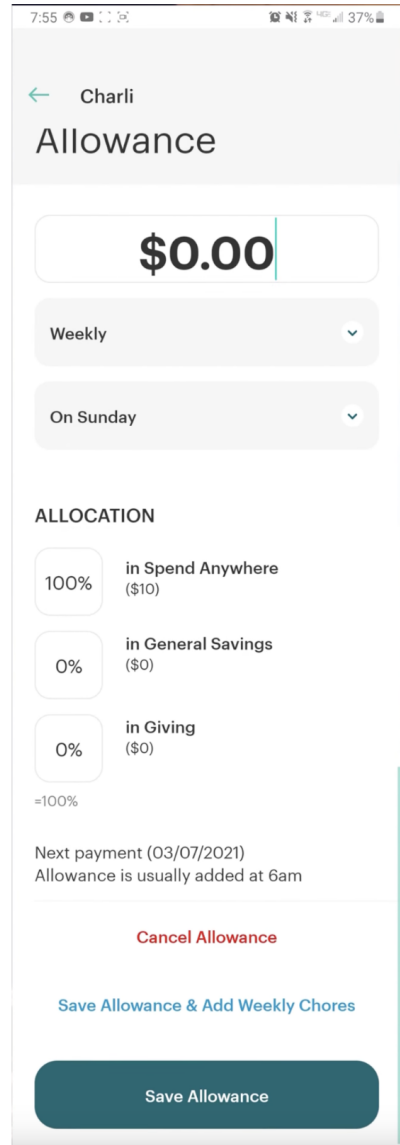
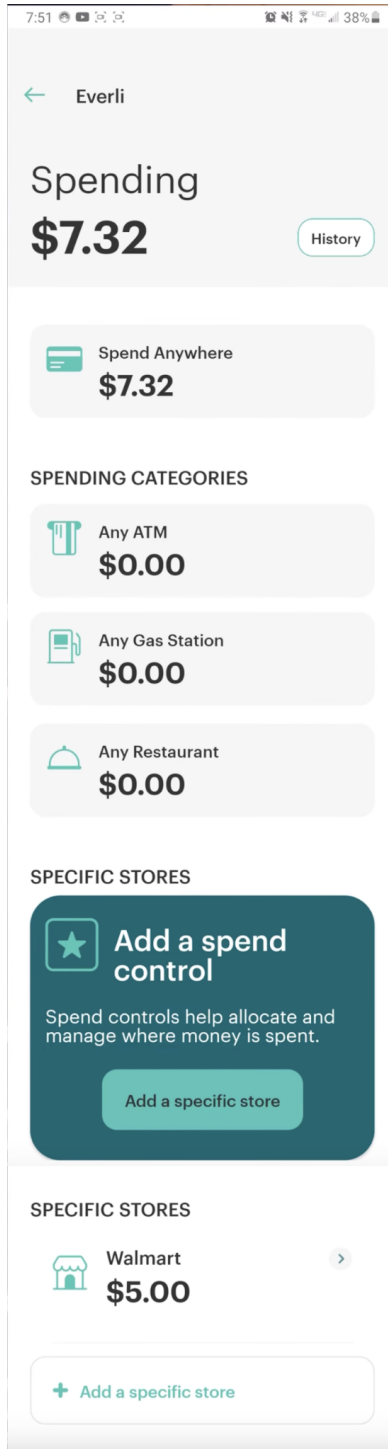
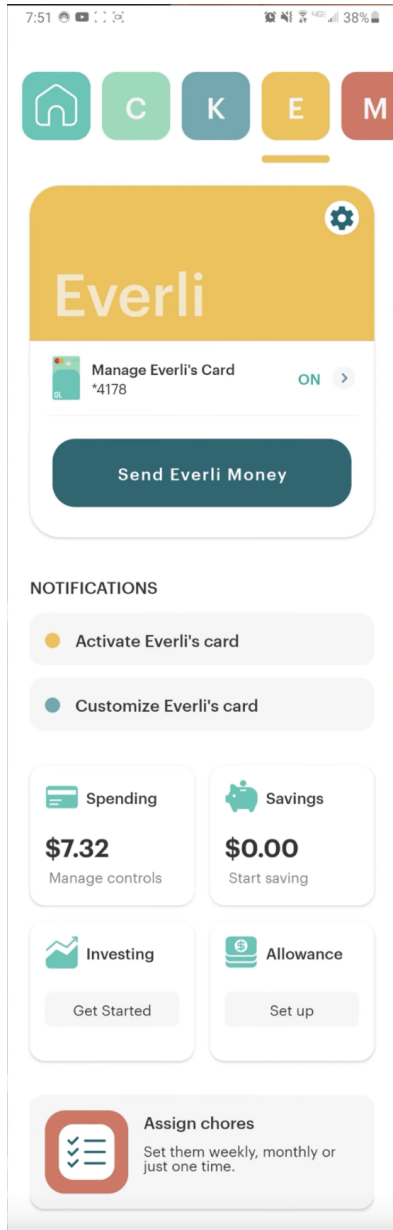




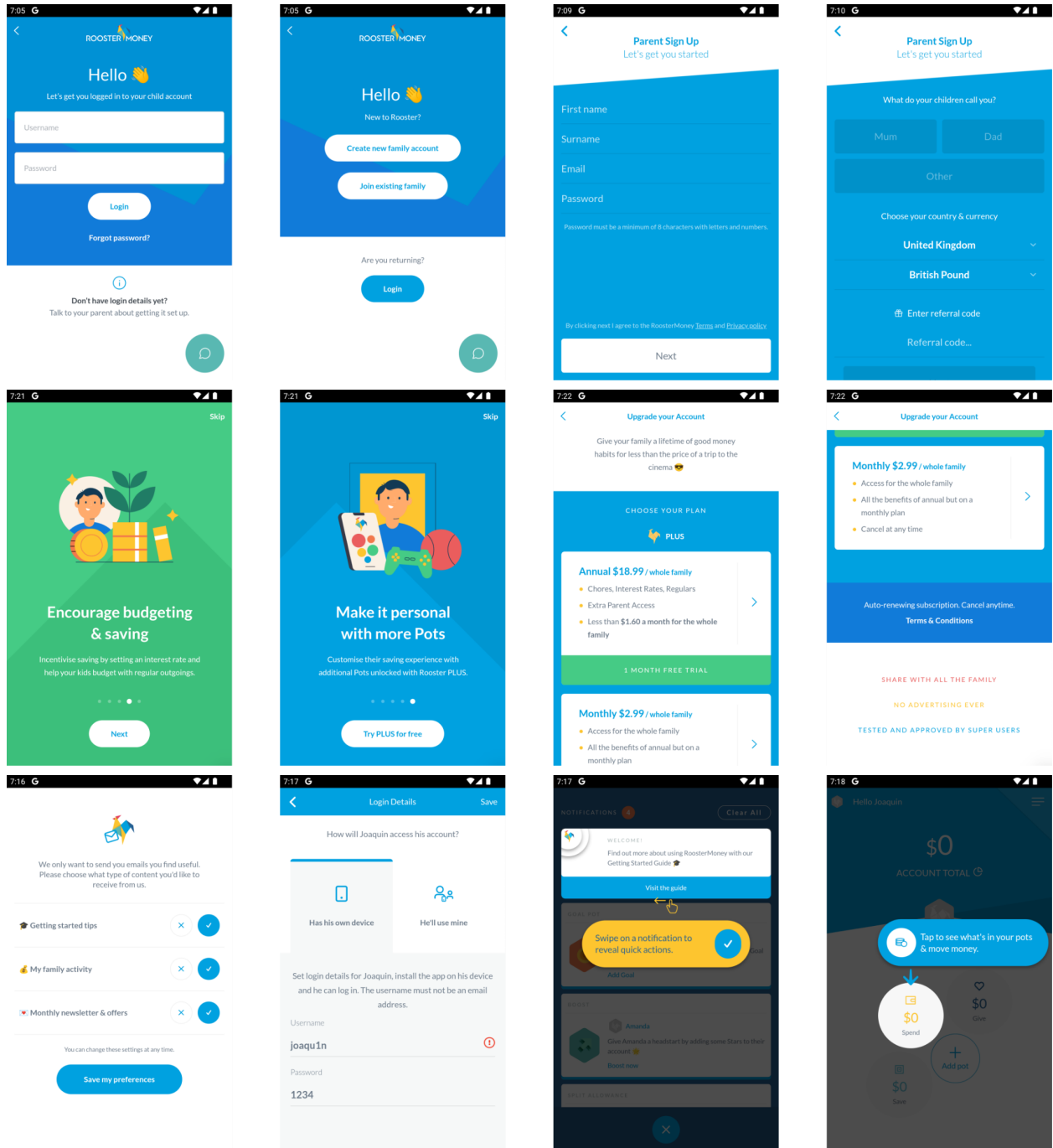
Anexo 3: Interfaz gráfica para padres Greenlight

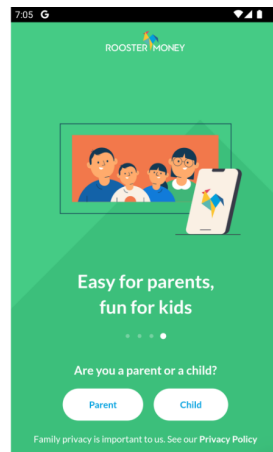
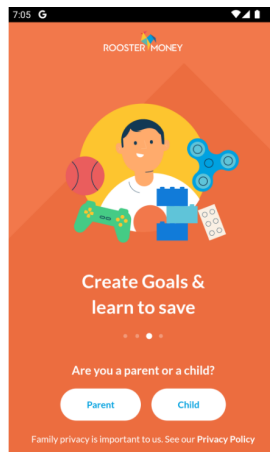
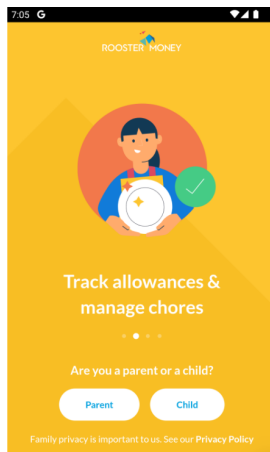
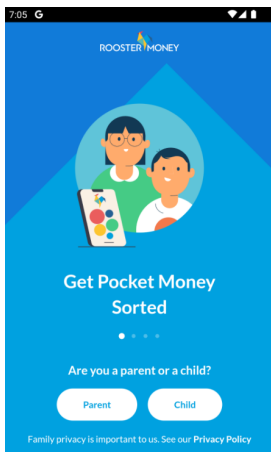
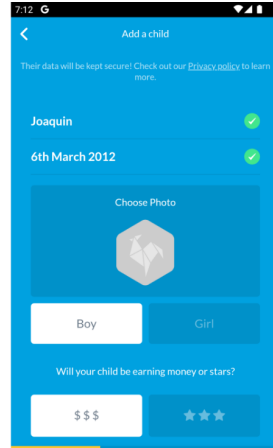
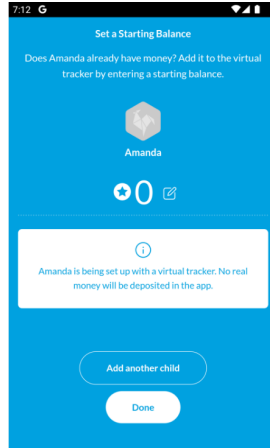
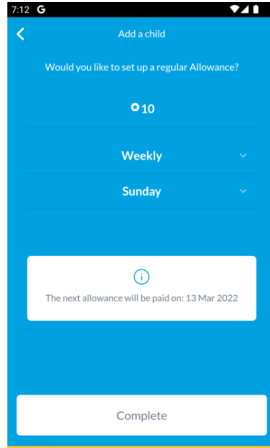
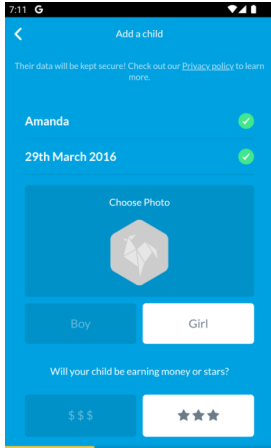






## Anexo 4: Interfaz gráfica de bienvenida RoosterMoney





Anexo 5: Documentación caso de uso Agregar fondos

<b>Caso de uso</b>	<b>Agregar fondos</b>
<b>ID</b>	1
<b>Descripción</b>	Permite que un Padre agregue dinero a su cuenta.
<b>Actores</b>	Padre
<b>Relaciones</b>	
<b>Pre-condiciones</b>	El Padre necesita tener una sesión activa y debe estar visualizando la pantalla principal.
<b>Post-condiciones</b>	Se actualiza el saldo de la cuenta del Padre, adicionando el monto añadido.
<b>Flujo Principal</b>	<p>P1. El Padre presiona el botón "Añadir fondos".</p> <p>P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario.</p> <p>P3. El Padre ingresa el monto a cargar y el método de pago.</p> <p>P4. El sistema muestra la comparativa entre el saldo actual y el nuevo saldo.</p> <p>P5. El Padre confirma y envía el formulario.</p> <p>P6. El sistema muestra una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, el cual muestra el saldo actualizado. (E1)</p>
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, el cual muestra el saldo anterior.

Anexo 6: Documentación caso de uso Transferir a hijo

<b>Caso de uso</b>	<b>Transferir a hijo</b>
<b>ID</b>	2
<b>Descripción</b>	Permite que un Padre transfiera dinero desde el saldo de su billetera a una cuenta de uno de sus hijos.
<b>Actores</b>	Padre
<b>Relaciones</b>	
<b>Pre-condiciones</b>	El Padre necesita tener una sesión activa y debe estar visualizando la pantalla principal. El Padre necesita pertenecer a una familia que tenga al menos un hijo.
<b>Post-condiciones</b>	Se actualiza el saldo de la billetera del Padre, restando el monto transferido. Se actualiza el saldo de la cuenta específica del Hijo seleccionado , sumando el monto recibido.
<b>Flujo Principal</b>	P1. El Padre presiona el botón "Transferir" contenida en la tarjeta del hijo al que desea transferir. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario. P3. El Padre ingresa el monto a transferir, el hijo seleccionado, la cuenta de destino y opcionalmente una nota. P4. El Padre confirma y envía el formulario. P5. El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, el cual muestra el saldo actualizado. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, el cual muestra el saldo anterior.



Anexo 7: Documentación caso de uso Ver Actividad

<b>Caso de uso</b>	<b>Ver actividad</b>
<b>ID</b>	3
<b>Descripción</b>	Permite que un Padre visualice un historial de la actividad suya y la de sus hijos.
<b>Actores</b>	Padre
<b>Relaciones</b>	
<b>Pre-condiciones</b>	El Padre necesita tener una sesión activa y debe estar visualizando la pantalla principal.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de actividad.
<b>Flujo Principal</b>	P1. El Padre presiona el botón de notificaciones. P2. El sistema agrega al stack de navegación una pantalla que presenta un listado de notificaciones, las cuales contienen el icono del usuario, detalles de la operación y fecha de la operación. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el listado de notificaciones.

Anexo 8: Documentación caso de uso Ver detalle ahorros hijo

<b>Caso de uso</b>	<b>Ver detalle general de hijo</b>
<b>ID</b>	4
<b>Descripción</b>	Permite que un Padre visualice el perfil de uno de sus hijos.
<b>Actores</b>	Padre
<b>Relaciones</b>	Es precondition para el caso de uso "Ver detalle ahorros de hijo" (ID 5). Es precondition para el caso de uso "Ver detalle mesada de hijo" (ID 6). Es precondition para el caso de uso "Ver detalle tareas de hijo" (ID 10). Es precondition para el caso de uso "Ver detalle tarjeta de hijo" (ID 14).
<b>Pre-condiciones</b>	El Padre necesita tener una sesión activa y debe estar visualizando la pantalla principal. El Padre necesita pertenecer a una familia que tenga al menos un hijo.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla del perfil de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona la tarjeta que contiene información básica de su hijo, como su foto de perfil, nombre, nivel, saldo en tarjeta, saldo de ahorros y un botón para transferir. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles del hijo, incluyendo el saldo total, saldo en tarjeta y el monto gastado esta semana; saldo en ahorros y la cantidad de metas de ahorro activas, monto y frecuencia de mesada; monto a ganar en tareas hoy y un listado de operaciones recientes. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el perfil del hijo seleccionado.

Anexo 9: Documentación caso de uso Ver detalle ahorros de hijo

<b>Caso de uso</b>	<b>Ver detalle ahorros hijo</b>
<b>ID</b>	5
<b>Descripción</b>	Permite que un Padre visualice el detalle de los ahorros de su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle general de hijo" (ID 4).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle general de su hijo.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de ahorros de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona la tarjeta de ahorros, el cual muestra el monto total ahorrado y la cantidad de metas de ahorro activas. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de los ahorros de su hijo, incluyendo el monto total ahorrado; un gráfico del movimiento histórico del monto total ahorrado; una tarjeta con el monto ahorrado en la cuenta por defecto 'Chanchito'; tarjetas con el detalle de las metas de ahorro incluyendo su nombre, fecha objetivo, monto ahorrado, monto restante, monto objetivo y un botón para transferir directamente fondos a dicha meta de ahorro. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de los ahorros del hijo.

Anexo 10: Documentación caso de uso Ver detalle mesada de hijo

<b>Caso de uso</b>	<b>Ver detalle mesada de hijo</b>
<b>ID</b>	6
<b>Descripción</b>	Permite que un Padre visualice el detalle de la mesada de su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle general de hijo" (ID 4). Es precondition para el caso de uso "Asignar mesada a hijo" (ID 7). Es precondition para el caso de uso "Editar mesada de hijo" (ID 8). Es precondition para el caso de uso "Eliminar mesada de hijo" (ID 9).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle general de su hijo.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de la mesada de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona la tarjeta de mesada, la cual muestra el monto de asignación recurrente y la frecuencia de pago de ésta. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de la mesada de su hijo, incluyendo el monto de asignación, la frecuencia, la próxima fecha de pago, la cuenta de destino en la cual se depositan estas asignaciones recurrentes y un botón para configurar la mesada. (E2)
<b>Flujos de excepción</b>	E2. El sistema muestra una alerta de error al cargar el detalle de la mesada del hijo.

Anexo 11: Documentación caso de uso Asignar mesada a hijo

<b>Caso de uso</b>	<b>Asignar mesada a hijo</b>
<b>ID</b>	7
<b>Descripción</b>	Permite que un Padre cree y asigne una mesada a su hijo.
<b>Actores</b>	Padre
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de la mesada de su hijo. El Padre no debe tener una mesada configurada para dicho hijo.
<b>Relaciones</b>	Depende del caso de uso "Ver detalle mesada de hijo" (ID 6).
<b>Post-condiciones</b>	Se asigna la mesada con las configuraciones establecidas al hijo seleccionado.
<b>Flujo Principal</b>	P1. El Padre presiona el botón para crear la mesada. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino. P3. El Padre completa, confirma y envía el formulario. P4. El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, el cual muestra los detalles de la mesada configurada. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al asignar la mesada del hijo.

Anexo 12: Documentación caso de uso Editar mesada de hijo

<b>Caso de uso</b>	<b>Editar mesada de hijo</b>
<b>ID</b>	8
<b>Descripción</b>	Permite que un Padre edite la mesada que asignó a su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle mesada de hijo" (ID 6).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de la mesada de su hijo. El Padre debe tener una mesada configurada para dicho hijo.
<b>Post-condiciones</b>	Se actualiza la mesada con las configuraciones establecidas al hijo seleccionado.
<b>Flujo Principal</b>	P1. El Padre presiona el botón para editar la mesada previamente configurada. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino. Los campos del formulario están previamente completados con la configuración actual de la mesada. P3. El Padre edita, confirma y envía el formulario. P4. El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, el cual muestra los detalles actualizados de la mesada configurada. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al actualizar la mesada del hijo.

Anexo 13: Documentación caso de uso Eliminar mesada de hijo

<b>Caso de uso</b>	<b>Eliminar mesada de hijo</b>
<b>ID</b>	9
<b>Descripción</b>	Permite que un Padre elimine la mesada que asignó a su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle mesada de hijo" (ID 6).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de la mesada de su hijo. El Padre debe tener una mesada configurada para dicho hijo.
<b>Post-condiciones</b>	Se elimina la mesada configurada del hijo seleccionado.
<b>Flujo Principal</b>	P1. El Padre presiona el botón para editar la mesada previamente configurada. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino. Los campos del formulario están previamente completados con la configuración actual de la mesada. P3. El Padre presiona el botón para eliminar la mesada previamente configurada. P4. El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, el cual advierte que no existe una mesada configurada. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al eliminar la mesada del hijo.

Anexo 14: Documentación caso de uso Ver detalle tareas de hijo

<b>Caso de uso</b>	<b>Ver detalle tareas de hijo</b>
<b>ID</b>	10
<b>Descripción</b>	Permite que un Padre visualice el detalle de las tareas de su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle general de hijo" (ID 4). Es precondition para el caso de uso "Asignar tarea a hijo" (ID 11). Es precondition para el caso de uso "Pagar tarea finalizada de hijo" (ID 12). Es precondition para el caso de uso "Rechazar finalización de tarea de hijo" (ID 13).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle general de su hijo.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de las tareas de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona la tarjeta de tareas, el cual muestra el monto por ganar esta semana y la cantidad de tareas pendientes de confirmación. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de las tareas de su hijo, incluyendo el monto ganado esta semana, el monto por ganar hoy, el monto por ganar esta semana, un botón flotante para crear una tarea y una lista de las tareas del hijo, permitiendo filtrar por tareas por pagar, tareas por hacer y tareas finalizadas. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de las tareas del hijo.



Anexo 15: Documentación caso de uso Asignar tarea a hijo

<b>Caso de uso</b>	<b>Asignar tarea a hijo</b>
<b>ID</b>	11
<b>Descripción</b>	Permite que un Padre cree y asigne una tarea a su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle tareas de hijo" (ID 10).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de tareas de su hijo.
<b>Post-condiciones</b>	Se agrega la nueva tarea a la categoría 'Por hacer' en el detalle de tareas de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona el botón flotante para crear una tarea. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de recompensa, el nombre de la tarea, la recurrencia de la tarea, la cuenta de destino y una nota opcional. P3. El Padre completa, confirma y envía el formulario. P4. El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de tareas, el cual presenta la nueva tarea en la categoría 'Por hacer'. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al crear la tarea del hijo seleccionado.

Anexo 16: Documentación caso de uso Pagar tarea finalizada de hijo

<b>Caso de uso</b>	<b>Pagar tarea finalizada de hijo</b>
<b>ID</b>	12
<b>Descripción</b>	Permite que un Padre confirme la finalización de una tarea asignada a su hijo y efectúe el pago correspondiente.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle tareas de hijo" (ID 10).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Hijo debe tener asignada una tarea en estado 'Por pagar'. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de tareas de su hijo.
<b>Post-condiciones</b>	Se actualiza el estado de la tarea del hijo, pasando del estado 'Por Pagar' a 'Finalizado'. Se actualiza el saldo de la billetera del Padre, restando al saldo el monto de la recompensa. Se actualiza el saldo de la cuenta de destino seleccionada del Hijo, sumando al saldo el monto de recompensa.
<b>Flujo Principal</b>	P1. El Padre presiona el botón 'Pagar' dentro de la tarjeta con los detalles de la tarea 'Por pagar'. P2. El sistema presenta al Padre una alerta de éxito, se mueve la tarea desde la pestaña de tareas 'Por pagar' hacia la pestaña de tareas 'Finalizadas'. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al confirmar la finalización de la tarea y efectuar el pago.

Anexo 17: Documentación caso de uso Rechazar finalización de tarea de hijo

<b>Caso de uso</b>	<b>Rechazar finalización de tarea de hijo</b>
<b>ID</b>	13
<b>Descripción</b>	Permite que un Padre rechace la finalización de una tarea asignada a su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle tareas de hijo" (ID 10).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Hijo debe tener asignada una tarea en estado 'Por pagar'. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de tareas de su hijo.
<b>Post-condiciones</b>	Se actualiza el estado de la tarea del hijo, pasando del estado 'Por Pagar' a 'Por hacer'.
<b>Flujo Principal</b>	P1. El Padre presiona el botón 'Rechazar' dentro de la tarjeta con los detalles de la tarea 'Por pagar'. P2. El sistema presenta al Padre una alerta de éxito, se mueve la tarea desde la pestaña de tareas 'Por pagar' hacia la pestaña de tareas 'Por hacer'. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al rechazar la finalización de la tarea.

Anexo 18: Documentación caso de uso Ver detalle tarjeta de hijo

<b>Caso de uso</b>	<b>Ver detalle tarjeta de hijo</b>
<b>ID</b>	14
<b>Descripción</b>	Permite que un Padre visualice el detalle de la tarjeta de su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle general de hijo" (ID 10). Es precondition para el caso de uso "Bloquear temporalmente tarjeta de hijo" (ID 15).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle general de su hijo.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de la tarjeta de su hijo.
<b>Flujo Principal</b>	P1. El Padre presiona la tarjeta de tarjeta, el cual muestra el saldo actual de su tarjeta y el monto gastado esta semana. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de la tarjeta de su hijo, incluyendo saldo actual, el estado de la tarjeta, un botón para transferir directamente a la tarjeta y un listado de los gastos recientes efectuados con la tarjeta. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de la tarjeta del hijo.

Anexo 19: Documentación caso de uso Bloquear temporalmente tarjeta de hijo

<b>Caso de uso</b>	<b>Bloquear temporalmente tarjeta de hijo</b>
<b>ID</b>	15
<b>Descripción</b>	Permite que un Padre desactive temporalmente la tarjeta de su hijo.
<b>Actores</b>	Padre
<b>Relaciones</b>	Depende del caso de uso "Ver detalle tarjeta de hijo" (ID 14).
<b>Pre-condiciones</b>	El Padre necesita pertenecer a una familia que tenga al menos un hijo. El Hijo debe tener su tarjeta en estado activo. El Padre necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de la tarjeta de su hijo.
<b>Post-condiciones</b>	Se actualiza el estado de la tarjeta del hijo, pasando del estado 'Activa' a 'Bloqueada'.
<b>Flujo Principal</b>	P5. El Padre presiona el slider activo dentro de la tarjeta del hijo. P6. El sistema presenta una animación exitosa del slider, el cual pasa desde el estado habilitado al estado deshabilitado, confirmando el bloqueo temporal de la tarjeta. (E1)
<b>Flujos de excepción</b>	E1. El sistema no cambia el estado del slider, lo mantiene encendido y no se efectúa el bloqueo temporal de la tarjeta.

Anexo 20: Documentación caso de uso Mostrar información sensible de tarjeta

<b>Caso de uso</b>	<b>Mostrar información sensible de tarjeta</b>
<b>ID</b>	16
<b>Descripción</b>	Permite que un Hijo visualice los detalles de su tarjeta
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de tarjeta" (ID 26).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla principal. La tarjeta debe tener los detalles ocultos.
<b>Post-condiciones</b>	El sistema muestra los detalles de su tarjeta.
<b>Flujo Principal</b>	P1. El Hijo presiona el botón con el ícono de un ojo cerrado dentro de su tarjeta. P2. El sistema cambia el ícono del botón por un ojo abierto, y se muestran los detalles de su tarjeta previamente ocultos por seguridad, que incluyen: Número de tarjeta, fecha de expiración y código de seguridad. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al mostrar los detalles de la tarjeta.

Anexo 21: Documentación caso de uso Ocultar información sensible de tarjeta

<b>Caso de uso</b>	<b>Ocultar información sensible de tarjeta</b>
<b>ID</b>	17
<b>Descripción</b>	Permite que un Hijo oculte los detalles de su tarjeta
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de tarjeta" (ID 26).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla principal. La tarjeta debe tener los detalles visibles.
<b>Post-condiciones</b>	El sistema oculta los detalles de su tarjeta.
<b>Flujo Principal</b>	P1. El Hijo presiona el botón con el ícono de un ojo abierto dentro de su tarjeta. P2. El sistema cambia el ícono del botón por un ojo cerrado, y se ocultan los detalles de su tarjeta que incluyen: Número de tarjeta, fecha de expiración y código de seguridad. Cada caracter es reemplazado por un símbolo “*”. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al ocultar los detalles de la tarjeta.

Anexo 22: Documentación caso de uso Transferir entre cuentas

<b>Caso de uso</b>	<b>Transferir entre cuentas</b>
<b>ID</b>	18
<b>Descripción</b>	Permite que un Hijo transfiera dinero entre sus cuentas.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de tarjeta" (ID 26).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla principal.
<b>Post-condiciones</b>	Se actualiza el saldo de la cuenta origen del Hijo, restando el monto transferido. Se actualiza el saldo de la cuenta destino del Hijo, sumando el monto recibido.
<b>Flujo Principal</b>	P1. El Hijo presiona el botón flotante representado por un ícono de flechas opuestas. P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario. P3. El Hijo ingresa el monto a transferir, la cuenta de origen y la cuenta de destino. P4. El Hijo confirma y envía el formulario. P5. El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal.(E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal.



Anexo 23: Documentación caso de uso Ver detalle de ahorros

<b>Caso de uso</b>	<b>Ver detalle de ahorros</b>
<b>ID</b>	19
<b>Descripción</b>	Permite que un Hijo visualice el detalle de sus ahorros.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Es precondition para el caso de uso "Crear meta de ahorro" (ID 20). Es precondition para el caso de uso "Editar meta de ahorro" (ID 21). Es precondition para el caso de uso "Eliminar meta de ahorro" (ID 22).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando una pantalla que incluya la barra de navegación inferior.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de sus ahorros.
<b>Flujo Principal</b>	P1. El Hijo presiona el ícono de una alcancía con forma de cerdo en la barra de navegación inferior. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de sus ahorros, incluyendo el monto total ahorrado; un gráfico del movimiento histórico del monto total ahorrado; una tarjeta con el monto ahorrado en la cuenta por defecto 'Chanchito'; tarjetas con el detalle de las metas de ahorro incluyendo su nombre, fecha objetivo, monto ahorrado, monto restante, monto objetivo; una barra de progreso comparando el monto ahorrado con el monto objetivo y un botón para transferir directamente fondos a dicha meta de ahorro; y un botón flotante para crear una meta de ahorro. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de sus ahorros.

Anexo 24: Documentación caso de uso Crear meta de ahorro

<b>Caso de uso</b>	<b>Crear meta de ahorro</b>
<b>ID</b>	20
<b>Descripción</b>	Permite que un Hijo cree una meta de ahorro.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de ahorros" (ID 19).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de sus ahorros.
<b>Post-condiciones</b>	Se crea una meta de ahorro con las configuraciones establecidas.
<b>Flujo Principal</b>	<p>P1. El Hijo presiona el botón flotante con un símbolo “+” para crear la meta de ahorro.</p> <p>P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la meta, el monto objetivo y la fecha objetivo.</p> <p>P3. El Hijo completa, confirma y envía el formulario.</p> <p>P4. El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de sus ahorros, el cual muestra una nueva tarjeta con los detalles de la meta de ahorro creada. (E1)</p>
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al crear la meta de ahorro.

Anexo 25: Documentación caso de uso Editar meta de ahorro

<b>Caso de uso</b>	<b>Editar meta de ahorro</b>
<b>ID</b>	21
<b>Descripción</b>	Permite que un Hijo edite una meta de ahorro existente.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de ahorros" (ID 19).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de sus ahorros. El Hijo debe tener al menos una meta de ahorro creada.
<b>Post-condiciones</b>	Se actualiza la meta de ahorro seleccionada con las configuraciones establecidas.
<b>Flujo Principal</b>	<p>P1. El Hijo desliza hacia la izquierda la tarjeta de la meta de ahorro que desea editar.</p> <p>P2. El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón "Editar".</p> <p>P3. El Hijo presiona el botón "Editar"</p> <p>P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la meta, el monto objetivo y la fecha objetivo. Los campos del formulario están previamente completados con la configuración actual de la meta de ahorro.</p> <p>P3. El Hijo modifica, confirma y envía el formulario.</p> <p>P4. El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de sus ahorros, el cual muestra la tarjeta con los detalles actualizados de la meta de ahorro editada. (E1)</p>
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al editar la meta de ahorro.

Anexo 26: Documentación caso de uso Eliminar meta de ahorro

<b>Caso de uso</b>	<b>Eliminar meta de ahorro</b>
<b>ID</b>	22
<b>Descripción</b>	Permite que un Hijo elimine una meta de ahorro existente.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de ahorros" (ID 19).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de sus ahorros. El Hijo debe tener al menos una meta de ahorro creada.
<b>Post-condiciones</b>	Se elimina la meta de ahorro seleccionada. Si la meta de ahorro eliminada tenía un monto ahorrado, se suma dicho monto a su tarjeta.
<b>Flujo Principal</b>	P1. El Hijo desliza hacia la derecha la tarjeta de la meta de ahorro que desea eliminar. P2. El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón "Eliminar". P3. El Hijo presiona el botón "Eliminar" P4. El sistema presenta al Hijo una alerta de éxito y remueve la meta de ahorro y transfiere el saldo ahorrado a su tarjeta. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al eliminar la meta de ahorro.

Anexo 27: Documentación caso de uso Ver detalle de ganar dinero

<b>Caso de uso</b>	<b>Ver detalle de ganar dinero</b>
<b>ID</b>	23
<b>Descripción</b>	Permite que un Hijo visualice el detalle de sus medios para ganar dinero.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Es precondition para el caso de uso "Completar tarea "Por hacer"" (ID 24).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando una pantalla que incluya la barra de navegación inferior.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de ganar dinero.
<b>Flujo Principal</b>	<p>P1. El Hijo presiona el ícono de una mano con una moneda en la barra de navegación inferior.</p> <p>P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de sus medios para ganar dinero, incluyendo un carrusel que incluye la tarjeta que sintetiza la información de sus tareas con recompensa (monto ganado esta semana, monto por ganar hoy y monto por ganar esta semana) y tarjetas de las mesadas que tiene asignadas (monto de mesada, frecuencia, próximo pago y cuenta de destino). Por otra parte, se presenta listado de las tareas que tiene asociadas, con la capacidad de filtrarlas mediante las pestañas "Por hacer", "Por pagar" y "Finalizadas".(E1)</p>
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de ganar dinero.

Anexo 28: Documentación caso de uso Completar tarea “Por hacer”

<b>Caso de uso</b>	<b>Completar tarea “Por hacer”</b>
<b>ID</b>	24
<b>Descripción</b>	Permite que un Hijo confirme la realización de una tarea en el estado “Por hacer”
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de ganar dinero" (ID 23).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla del detalle de ganar dinero. El Hijo debe tener al menos una tarea en la pestaña “Por hacer”.
<b>Post-condiciones</b>	Se actualiza el estado de la tarea desde “Por hacer” a “Por pagar”.
<b>Flujo Principal</b>	P1. El Hijo desliza hacia la izquierda la tarjeta de la tarea “Por hacer” que desea completar. P2. El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón “Completar”. P3. El Hijo presiona el botón “Completar”. P4. El sistema presenta al Hijo una alerta de éxito, cambia el estado de la tarea de “Por hacer” a “Por pagar” y se mueve la tarjeta de la tarea desde la pestaña “Por hacer” a “Por pagar”.(E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al completar la tarea.

Anexo 29: Documentación caso de uso Simular compra

<b>Caso de uso</b>	<b>Simular compra</b>
<b>ID</b>	25
<b>Descripción</b>	Permite que un Hijo simule una compra con su tarjeta
<b>Actores</b>	Hijo
<b>Relaciones</b>	Depende del caso de uso "Ver detalle de tarjeta" (ID 26).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando la pantalla principal.
<b>Post-condiciones</b>	Se crea una un gasto asociado a su tarjeta con los detalles de la transacción.
<b>Flujo Principal</b>	<p>P1. El Hijo presiona el botón “Simular compra”.</p> <p>P2. El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la tienda, el monto de la compra y la fecha de transacción.</p> <p>P3. El Hijo completa, confirma y envía el formulario.</p> <p>P4. El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, la cual muestra una nueva entrada con los detalles de la transacción. (E1)</p>
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al simular la compra.

Anexo 30: Documentación caso de uso Ver detalle de tarjeta

<b>Caso de uso</b>	<b>Ver detalle de tarjeta</b>
<b>ID</b>	26
<b>Descripción</b>	Permite que un Hijo visualice el detalle de su tarjeta.
<b>Actores</b>	Hijo
<b>Relaciones</b>	Es precondition para el caso de uso "Mostrar información sensible de tarjeta" (ID 16). Es precondition para el caso de uso "Ocultar información sensible de tarjeta" (ID 17). Es precondition para el caso de uso "Transferir entre cuentas" (ID 18). Es precondition para el caso de uso "Simular compra" (ID 25).
<b>Pre-condiciones</b>	El Hijo necesita tener una sesión activa y debe estar visualizando una pantalla que incluya la barra de navegación inferior.
<b>Post-condiciones</b>	El stack de navegación presenta la pantalla de detalle de su tarjeta (pantalla principal).
<b>Flujo Principal</b>	P1. El Hijo presiona el ícono de una tarjeta en la barra de navegación inferior. P2. El sistema agrega al stack de navegación una pantalla que presenta los detalles de su tarjeta, incluyendo saldo actual, el estado de la tarjeta, un botón para mostrar/ocultar los detalles de la tarjeta, un listado de los gastos recientes efectuados con la tarjeta, un botón flotante para transferir entre sus cuentas, un botón para ver su actividad, su foto de perfil y un mensaje de bienvenida. (E1)
<b>Flujos de excepción</b>	E1. El sistema muestra una alerta de error al cargar el detalle de su tarjeta.



## Anexo 31: Detalle de modelo Parent

### - Atributos

- [id: ID!]
  - Identificador Cognito y DynamoDB
- [firstName: String]
  - Nombre
- [lastName: String]
  - Apellido
- [email: AWSEmail]
  - Correo Electrónico
- [birthDate: AWSDate]
  - Fecha de nacimiento
- [sex: Sex]
  - Sexo
- [phoneNumber: String]
  - Número de teléfono

### - Relaciones

- [Family: Family @belongsTo]
  - Relación de pertenencia 1:1 con Familia. Corresponde a la Familia a la que pertenece el Padre.
- [Wallet: Wallet @hasOne]
  - Relación 1:1 con Billetera. Corresponde a la Billetera que es de la propiedad del Padre.
- [Chores: [Chore] @hasMany]
  - Relación 1:n con Tarea. Corresponde a las tareas creadas por el Padre.
- [Allowances: [Allowance] @hasMany] Relación 1:n con Mesada
  - Relación 1:n con Mesada. Corresponde a las mesadas creadas por el Padre.

## Anexo 32: Detalle de modelo Child

### - Atributos

- [id: ID!]
  - Identificador Cognito y DynamoDB
- [firstName: String]
  - Nombre
- [lastName: String]
  - Apellido
- [email: AWSEmail]
  - Correo Electrónico
- [birthDate: AWSDate]
  - Fecha de nacimiento
- [sex: Sex]

- Sexo
- [phoneNumber: String]
  - Número de teléfono
- **Relaciones**
  - [Family: Family @belongsTo]
    - Relación de pertenencia 1:1 con Familia. Corresponde a la Familia a la que pertenece el Hijo.
  - [Card: Card @hasOne]
    - Relación 1:1 con Tarjeta. Corresponde a la Tarjeta que es de la propiedad del Hijo.
  - [SavingsAccounts: [SavingsAccount] @hasMany]
    - Relación 1:n con Cuenta de Ahorro. Corresponden a las cuentas de ahorro asociadas al Hijo.
  - [Chores: [Chore] @hasMany]
    - Relación 1:n con Tarea. Corresponde a las tareas asignadas al Hijo.
  - [Allowances: [Allowance] @hasMany]
    - Relación 1:n con Mesada. Corresponde a las mesadas asignadas al Hijo.

#### Anexo 33: Detalle de modelo Family

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [familyName: String]
    - Nombre de la familia
- **Relaciones**
  - [Child: Child @hasMany]
    - Relación 1:n con Hijo. Corresponde a los Hijos que pertenecen a la Familia.
  - [Parent: Parent @hasMany]
    - Relación 1:n con Hijo. Corresponde a los Hijos que pertenecen a la Familia.

#### Anexo 34: Detalle de modelo Wallet

- **Atributos**
  - [id: ID!]
    - Identificador único de la billetera.
  - [amount: Int]
    - Cantidad de fondos que contiene la billetera.
- **Relaciones**
  - [Parent: Parent @belongsTo]

- Relación de pertenencia 1:1 con el modelo Parent. Corresponde al padre al que pertenece la billetera.
- [WalletAddFundsRecords: WalletAddFunds @hasMany]
  - Relación 1:n con el modelo WalletAddFunds. Corresponde a los registros de la operación Agregar Fondos que se han realizado en la billetera.
- [WalletTransferCardRecords: WalletTransferCard @hasMany]
  - Relación 1:n con el modelo WalletTransferCard. Corresponde a los registros de la operación Transferir Fondos desde la billetera a una tarjeta que se han realizado en la billetera.
- [WalletTransferSavingsAccountRecords: WalletTransferSavingsAccount @hasMany]
  - Relación 1:n con el modelo WalletTransferSavingsAccount. Corresponde a los registros de la operación Transferir Fondos desde la billetera a una cuenta de ahorro que se han realizado en la billetera.

#### Anexo 35: Detalle de modelo SavingsAccount

##### - Atributos

- [id: ID!]
  - Identificador DynamoDB
- [amount: Int]
  - Cantidad de fondos que contiene la cuenta de ahorros.
- [name: String]
  - Nombre de la cuenta de ahorros.
- [type: SavingsAccountType]
  - Tipo de cuenta de ahorros.
- [targetAmount: Int]
  - Monto objetivo que se desea alcanzar en la cuenta de ahorros.
- [dueDate: AWSDate]
  - Fecha límite para alcanzar el monto objetivo en la cuenta de ahorros.

##### - Relaciones

- [Child: Child @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Child. Corresponde al hijo al que pertenece la cuenta de ahorros.
- [SavingsAccountTransferCardRecords: SavingsAccountTransferCard @hasMany]
  - Relación 1:n con el modelo SavingsAccountTransferCard. Corresponde a los registros de las transferencias realizadas desde la cuenta de ahorros a una tarjeta.
- [SavingsAccountTransferSavingsAccountRecords: SavingsAccountTransferSavingsAccount @hasMany]
  - Relación 1:n con el modelo SavingsAccountTransferSavingsAccount. Corresponde a los registros de las transferencias realizadas entre cuentas de ahorro.

- [WalletTransferSavingsAccountRecords: [WalletTransferSavingsAccount] @hasMany]
  - Relación 1:n con el modelo WalletTransferSavingsAccount. Corresponde a los registros de las transferencias realizadas desde una billetera a la cuenta de ahorros.
- [CardTransferSavingsAccountRecords: [CardTransferSavingsAccount] @hasMany]
  - Relación 1:n con el modelo CardTransferSavingsAccount. Corresponde a los registros de las transferencias realizadas desde una tarjeta a la cuenta de ahorros.

#### Anexo 36: Detalle de modelo Card

##### - **Atributos**

- [id: ID!]
  - Identificador DynamoDB
- [amount: Int]
  - Cantidad de fondos que contiene la tarjeta.
- [status: Boolean]
  - Estado de la tarjeta (activa o inactiva).
- [cardNumber: String]
  - Número de la tarjeta.
- [expirationDate: String]
  - Fecha de vencimiento de la tarjeta.
- [securityNumber: String]
  - Número de seguridad de la tarjeta.
- [cardholderName: String]
  - Nombre del titular de la tarjeta.

##### - **Relaciones**

- [Child: Child @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Child. Corresponde al hijo al que pertenece la tarjeta.
- [SavingsAccountTransferCardRecords: [SavingsAccountTransferCard] @hasMany]
  - Relación 1:n con el modelo SavingsAccountTransferCard. Corresponde a los registros de las transferencias realizadas desde una cuenta de ahorros a la tarjeta.
- [WalletTransferCardRecords: [WalletTransferCard] @hasMany]
  - Relación 1:n con el modelo WalletTransferCard. Corresponde a los registros de las transferencias realizadas desde una billetera a la tarjeta.
- [CardTransferSavingsAccountRecords: [CardTransferSavingsAccount] @hasMany]
  - Relación 1:n con el modelo CardTransferSavingsAccount. Corresponde a los registros de las transferencias realizadas desde la tarjeta a una cuenta de ahorros.

- [Expenses: [Expense] @hasMany]
  - Relación 1:n con el modelo Expense. Corresponde a los gastos realizados con la tarjeta.

#### Anexo 37: Detalle de modelo ActiveChore

##### - Atributos

- [id: ID!]
  - Identificador DynamoDB
- [status: ActiveChoreStatus]
  - Estado de la tarea activa (pendiente, completada, etc.).
- [doDate: AWSDate]
  - Fecha en la que se debe realizar la tarea.
- [dueDate: AWSDate]
  - Fecha límite para completar la tarea.
- [pendingDateTime: AWSDateTime]
  - Fecha y hora en la que la tarea quedó pendiente.
- [finishedDateTime: AWSDateTime]
  - Fecha y hora en la que la tarea se completó.

##### - Relaciones

- [Chore: Chore @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Chore. Corresponde a la tarea asociada a la tarea activa.
- [WalletTransferCard: WalletTransferCard @hasOne]
  - Relación 1:1 con el modelo WalletTransferCard. Corresponde al registro de transferencia realizada a una tarjeta al completar la tarea activa.
- [WalletTransferSavingsAccount: WalletTransferSavingsAccount @hasOne]
  - Relación 1:1 con el modelo WalletTransferSavingsAccount. Corresponde al registro de transferencia realizada a una cuenta de ahorros al completar la tarea activa.

#### Anexo 38: Detalle de modelo Chore

##### - Atributos

- [id: ID!]
  - Identificador DynamoDB
- [name: String]
  - Nombre de la tarea.
- [amount: Int]
  - Cantidad de fondos asignados por completar la tarea.
- [frequency: Frequency]
  - Frecuencia con la que se debe realizar la tarea (semanal, mensual, etc.).
- [note: String]

- Nota o descripción adicional sobre la tarea.
- [destinationAccountType: DestinationAccount]
  - Tipo de cuenta destino de los fondos al completar la tarea (tarjeta, cuenta de ahorros, etc.).
- [destinationAccountId: ID]
  - Identificador de la cuenta destino de los fondos al completar la tarea.
- **Relaciones**
  - [Child: Child @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Child. Corresponde al hijo al que se asigna la tarea.
  - [Parent: Parent @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Parent. Corresponde al padre que asigna la tarea.
  - [ActiveChores: [ActiveChore] @hasMany]
    - Relación 1:n con el modelo ActiveChore. Corresponde a las tareas activas asociadas a la tarea.

#### Anexo 39: Detalle de modelo Allowance

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [amount: Float]
    - Cantidad de fondos asignados en la mesada.
  - [frequency: Frequency]
    - Frecuencia con la que se asigna la mesada (semanal, mensual, etc.).
  - [startDate: AWSDate]
    - Fecha de inicio de la asignación de la mesada.
  - [note: String]
    - Nota o descripción adicional sobre la mesada.
  - [destinationAccountType: DestinationAccount]
    - Tipo de cuenta destino de la mesada (tarjeta, cuenta de ahorros, etc.).
  - [destinationAccountId: ID]
    - Identificador de la cuenta destino de la mesada.
- **Relaciones**
  - [Child: Child @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Child. Corresponde al hijo al que se asigna la mesada.
  - [Parent: Parent @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Parent. Corresponde al padre que asigna la mesada.
  - [WalletTransferCard: [WalletTransferCard] @hasMany]

- Relación 1:n con el modelo WalletTransferCard. Corresponde a los registros de transferencias de mesadas realizadas a tarjetas.
- [WalletTransferSavingsAccount: [WalletTransferSavingsAccount] @hasMany]
  - Relación 1:n con el modelo WalletTransferSavingsAccount. Corresponde a los registros de transferencias de mesadas realizadas a cuentas de ahorros.

#### Anexo 40: Detalle de modelo Expense

##### - **Atributos**

- [id: ID!]
  - Identificador DynamoDB
- [amount: Int]
  - Cantidad de fondos gastados.
- [storeName: String]
  - Nombre de la tienda o comercio donde se realizó el gasto.
- [dateTime: AWSDateTime]
  - Fecha y hora en que se realizó el gasto.
- [category: String]
  - Categoría del gasto (opcional).
- [note: String]
  - Nota o descripción adicional sobre el gasto (opcional).

##### - **Relaciones**

- [Card: Card @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Card. Corresponde a la tarjeta con la que se realizó el gasto.

#### Anexo 41: Detalle de modelo WalletAddFunds

##### - **Atributos**

- [id: ID!]
  - Identificador DynamoDB
- [amount: Int]
  - Cantidad de fondos agregados.
- [paymentMethod: PaymentMethod]
  - Método de pago utilizado para agregar los fondos (tarjeta de crédito, transferencia bancaria, etc.).
- [dateTime: AWSDateTime]
  - Fecha y hora en que se agregaron los fondos.

##### - **Relaciones**

- [Wallet: Wallet @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Wallet. Corresponde a la billetera en la que se agregaron los fondos.

## Anexo 42: Detalle de modelo WalletTransferCard

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [amount: Int]
    - Cantidad de fondos transferidos.
  - [dateTime: AWSDateTime]
    - Fecha y hora en que se realizó la transferencia.
  - [note: String]
    - Nota o descripción adicional sobre la transferencia (opcional).
  - [transferType: TransferType]
    - Tipo de transferencia (por ejemplo, mesada, recompensa por tarea, etc.).
- **Relaciones**
  - [Wallet: Wallet @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Wallet. Corresponde a la billetera desde la que se realizaron los fondos.
  - [Card: Card @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Card. Corresponde a la tarjeta a la que se transfirieron los fondos.
  - [Allowance: Allowance @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Allowance. Corresponde a la mesada asociada a la transferencia (si corresponde).
  - [ActiveChore: ActiveChore @belongsTo]
    - Relación de pertenencia 1:1 con el modelo ActiveChore. Corresponde a la tarea activa asociada a la transferencia (si corresponde).

## Anexo 43: Detalle de modelo WalletTransferSavingsAccount

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [amount: Int]
    - Cantidad de fondos transferidos.
  - [dateTime: AWSDateTime]
    - Fecha y hora en que se realizó la transferencia.
  - [note: String]
    - Nota o descripción adicional sobre la transferencia (opcional).
  - [transferType: TransferType]
    - Tipo de transferencia (por ejemplo, mesada, recompensa por tarea, etc.).
- **Relaciones**
  - [Wallet: Wallet @belongsTo]



- Relación de pertenencia 1:1 con el modelo Wallet. Corresponde a la billetera desde la que se realizaron los fondos.
- [SavingsAccount: SavingsAccount @belongsTo]
  - Relación de pertenencia 1:1 con el modelo SavingsAccount. Corresponde a la cuenta de ahorro a la que se transfirieron los fondos.
- [Allowance: Allowance @belongsTo]
  - Relación de pertenencia 1:1 con el modelo Allowance. Corresponde a la mesada asociada a la transferencia (si corresponde).
- [ActiveChore: ActiveChore @belongsTo]
  - Relación de pertenencia 1:1 con el modelo ActiveChore. Corresponde a la tarea activa asociada a la transferencia (si corresponde).

#### Anexo 44: Detalle de modelo CardTransferSavingsAccount

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [amount: Int]
    - Cantidad de fondos transferidos.
  - [dateTime: AWSDateTime]
    - Fecha y hora en que se realizó la transferencia.
  - [transferType: TransferType]
    - Tipo de transferencia (por ejemplo, mesada, recompensa por tarea, etc.).
- **Relaciones**
  - [Card: Card @belongsTo]
    - Relación de pertenencia 1:1 con el modelo Card. Corresponde a la tarjeta desde la que se realizaron los fondos.
  - [SavingsAccount: SavingsAccount @belongsTo]
    - Relación de pertenencia 1:1 con el modelo SavingsAccount. Corresponde a la cuenta de ahorro a la que se transfirieron los fondos.

#### Anexo 45: Detalle de modelo SavingsAccountTransferCard

- **Atributos**
  - [id: ID!]
    - Identificador DynamoDB
  - [amount: Int]
    - Cantidad de fondos transferidos.
  - [dateTime: AWSDateTime]
    - Fecha y hora en que se realizó la transferencia.
  - [transferType: TransferType]
    - Tipo de transferencia (por ejemplo, mesada, recompensa por tarea, etc.).
- **Relaciones**
  - [Card: Card @belongsTo]

- Relación de pertenencia 1:1 con el modelo Card. Corresponde a la tarjeta a la que se transfirieron los fondos.
- [SavingsAccount: SavingsAccount @belongsTo]
  - Relación de pertenencia 1:1 con el modelo SavingsAccount. Corresponde a la cuenta de ahorro desde la que se realizaron los fondos.

Anexo 46: Detalle de modelo SavingsAccountTransferSavingsAccount

- **Atributos**

- [id: ID!]
  - Identificador DynamoDB
- [amount: Int]
  - Cantidad de fondos transferidos.
- [dateTime: AWSDateTime]
  - Fecha y hora en que se realizó la transferencia.
- [transferType: TransferType]
  - Tipo de transferencia (por ejemplo, mesada, recompensa por tarea, etc.).

- **Relaciones**

- [OriginSavingsAccount: SavingsAccount @belongsTo]
  - Relación de pertenencia 1:1 con el modelo SavingsAccount. Corresponde a la cuenta de ahorro desde la que se realizaron los fondos.
- [DestinationSavingsAccount: SavingsAccount @hasOne]
  - Relación 1:1 con el modelo SavingsAccount. Corresponde a la cuenta de ahorro a la que se transfirieron los fondos.

## Anexo 47: Ejecución de pruebas de casos de uso

### Caso de prueba 1: Agregar fondos

Escenario: Padre agrega fondos con éxito

Descripción: Se prueba el caso de uso "Agregar fondos" en el escenario en el que el Padre agrega fondos a su cuenta de forma exitosa. En este caso, el proceso se completó en 20 segundos y se requirieron 3 pasos para finalizar la tarea.

Secuencia:

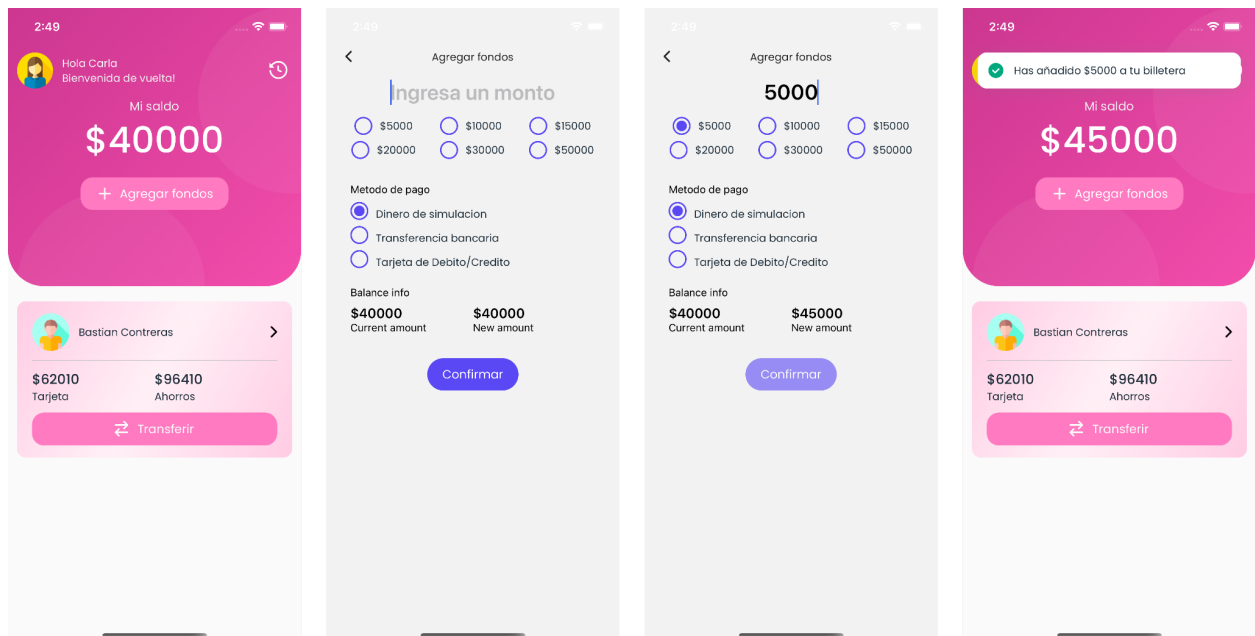
- 1: El Padre se encuentra en la pantalla principal con \$40.000 de saldo en su billetera..
- 2: El Padre presiona el botón "Añadir fondos", accediendo al formulario para ingresar el monto y el método de pago.
- 3: El Padre ingresa el monto a cargar equivalente a \$5.000 y el método de pago Dinero de simulación.
- 4: El Padre presiona confirmar y envía el formulario.

Resultado: Exitoso. La aplicación muestra una alerta de éxito, actualiza el saldo de la cuenta del Padre y regresa a la pantalla principal con el saldo actualizado equivalente a \$45.000.

Tiempo de ejecución: 16 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 2: Transferir a hijo

Escenario: Transferencia exitosa a un hijo

Descripción: Se prueba el caso de uso "Transferir a hijo" en el escenario en el que el Padre realiza una transferencia exitosa a la cuenta de uno de su hijo Bastian Contreras. En este caso, el proceso se completó en 30 segundos y se requirieron 3 interacciones para finalizar la tarea.

Secuencia:

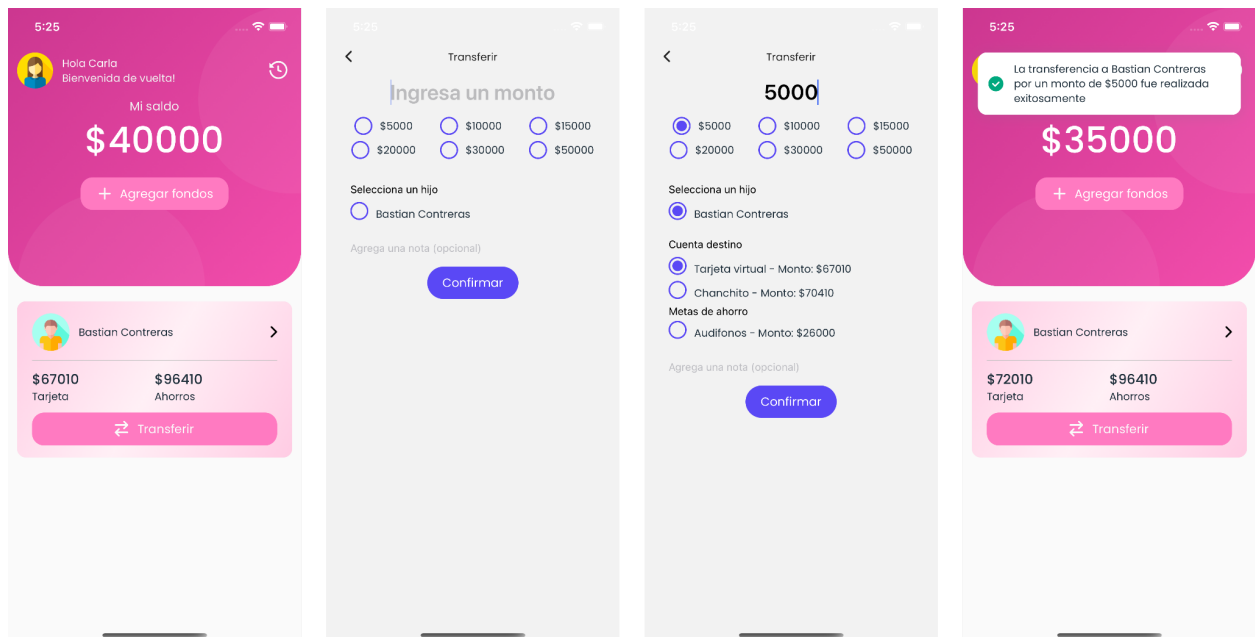
- 1: El Padre se encuentra en la pantalla principal con \$40.000 de saldo en su billetera.
- 2: El Padre presiona el botón "Transferir" contenida en la tarjeta del hijo Bastian Contreras al que desea transferir, accediendo al formulario de transferencia.
- 3: El Padre ingresa el monto a transferir equivalente a \$5.000 , el hijo seleccionado Bastian Contreras, la cuenta de destino Tarjeta Virtual que posee un monto de \$67.010 y omite la nota.
- 4: El Padre presiona confirmar y envía el formulario.
- 5: La aplicación muestra una alerta de éxito y regresa a la pantalla principal con los saldos actualizados tanto del Padre como del hijo seleccionado.

Resultado: Exitoso. La aplicación muestra una alerta de éxito, actualiza el saldo de la cuenta del Padre y regresa a la pantalla principal con el saldo. El Padre ahora posee \$35.000 de saldo en su billetera y la Tarjeta del hijo ahora cuenta con \$72.010.

Tiempo de ejecución: 15 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



### Caso de prueba 3: Ver actividad

Escenario: Visualizar el historial de actividad exitosamente

Descripción: Se prueba el caso de uso "Ver actividad" en el escenario en el que el Padre visualiza el historial de actividad suya y la de sus hijos de forma exitosa. En este caso, el proceso se completó en 12 segundos y se requirió 1 interacción para finalizar la tarea.

Secuencia:

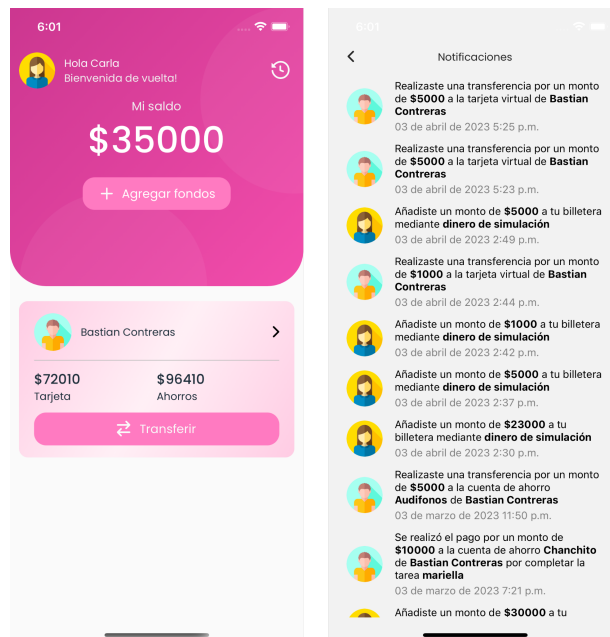
- 1: El Padre se encuentra en la pantalla principal.
- 2: El Padre presiona el botón de notificaciones ubicado en la parte superior derecha de la pantalla.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un listado de notificaciones. Cada notificación contiene el icono del usuario, detalles de la operación y fecha de la operación. La última notificación corresponde a una transferencia por un monto de \$5.000 a la tarjeta virtual de Bastian Contreras.

Resultado: Exitoso. El Padre puede visualizar el historial de actividad de su cuenta y la de sus hijos de forma exitosa, sin errores al cargar el listado de notificaciones.

Tiempo de ejecución: 12 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 4: Ver detalle general de hijo

Escenario: Visualizar el perfil del hijo exitosamente

Descripción: Se prueba el caso de uso "Ver detalle general de hijo" en el escenario en el que el Padre visualiza el perfil de uno de sus hijos de forma exitosa. En este caso, el proceso se completó en 6 segundos y se requirió 1 interacción para finalizar la tarea.

Secuencia:

1: El Padre se encuentra en la pantalla principal, viendo la tarjeta que contiene información básica de su hijo Bastian Contreras, como su foto de perfil, nombre Bastian Contreras, saldo en tarjeta \$72.010, saldo de ahorros \$96.410 y un botón para transferir.

2: El Padre presiona la tarjeta de su hijo Bastian Contreras.

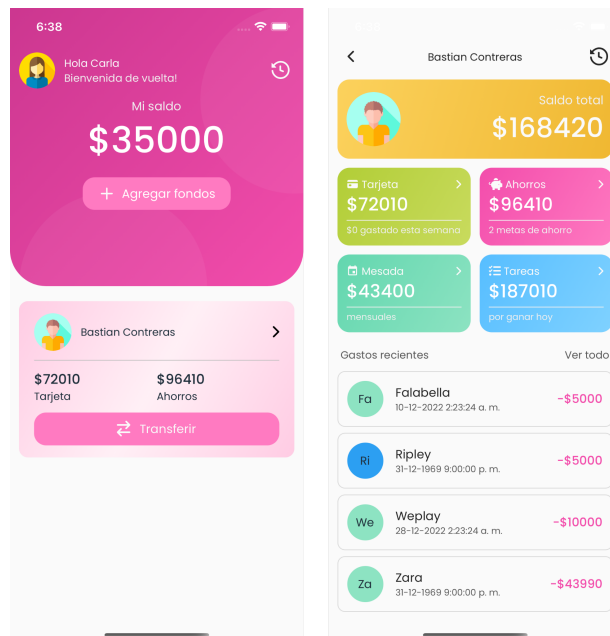
3: El sistema agrega al stack de navegación una pantalla que presenta los detalles del hijo, incluyendo el saldo total de \$168.420, \$72.010 de saldo en tarjeta y \$0 gastado esta semana; \$96.410 de saldo en ahorros y 2 de metas de ahorro activas, \$43.400 mensuales de mesada; monto a ganar en tareas hoy de \$187.010 y un listado de gastos recientes.

Resultado: Exitoso. El Padre puede visualizar el perfil de su hijo Bastian Contreras de forma exitosa, sin errores al cargar los detalles del hijo.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 5: Ver detalle ahorros hijo

Escenario: Visualizar el detalle de los ahorros del hijo exitosamente

Descripción: Se prueba el caso de uso "Ver detalle ahorros hijo" en el escenario en el que el Padre visualiza el detalle de los ahorros de su hijo de forma exitosa. En este caso, el proceso se completó en 6 segundos y se requirió 1 interacción para finalizar la tarea.

Secuencia:

1: El Padre se encuentra en la pantalla del detalle general de su hijo Bastian Contreras, donde puede ver la tarjeta de ahorros, que muestra el monto total ahorrado (\$96.410) y la cantidad de metas de ahorro activas.

2: El Padre presiona la tarjeta de ahorros.

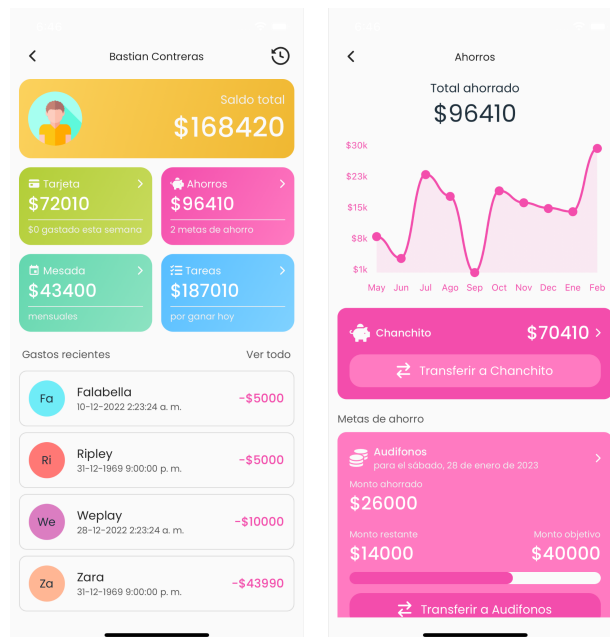
3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de los ahorros de su hijo, incluyendo el monto total ahorrado (\$96.410); un gráfico del movimiento histórico del monto total ahorrado; una tarjeta con el monto ahorrado en la cuenta por defecto 'Chanchito' (\$70.410); tarjetas con el detalle de las metas de ahorro, en este caso, una meta llamada "Audífonos" con fecha objetivo, monto ahorrado (\$26.000), monto restante (\$14.000), monto objetivo (\$40.000) y un botón para transferir directamente fondos a dicha meta de ahorro.

Resultado: Exitoso. El Padre puede visualizar el detalle de los ahorros de su hijo Bastian Contreras de forma exitosa, sin errores al cargar los detalles de los ahorros.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 2 interacciones



## Caso de prueba 6: Ver detalle mesada de hijo

Escenario: Visualizar el detalle de la mesada del hijo exitosamente

Descripción: Se prueba el caso de uso "Ver detalle mesada de hijo" en el escenario en el que el Padre visualiza el detalle de la mesada de su hijo de forma exitosa. En este caso, el proceso se completó en 6 segundos y se requirió 1 interacción para finalizar la tarea.

Secuencia:

1: El Padre se encuentra en la pantalla del detalle general de su hijo Bastian Contreras, donde puede ver la tarjeta de mesada, que muestra el monto de asignación recurrente (\$43.400) y la frecuencia de pago (Cada mes).

2: El Padre presiona la tarjeta de mesada.

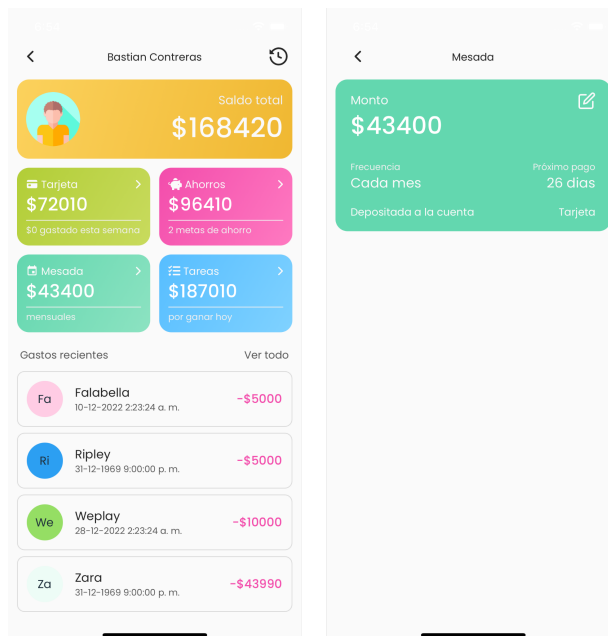
3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de la mesada de su hijo, incluyendo el monto de asignación (\$43.400), la frecuencia (Cada mes), la próxima fecha de pago (en 26 días), la cuenta de destino en la cual se depositan estas asignaciones recurrentes (Tarjeta) y un botón para configurar la mesada.

Resultado: Exitoso. El Padre puede visualizar el detalle de la mesada de su hijo Bastian Contreras de forma exitosa, sin errores al cargar los detalles de la mesada.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 2 interacciones





## Caso de prueba 7: Asignar mesada a hijo

Escenario: Configuración exitosa de mesada.

Descripción: Se prueba el caso de uso "Asignar mesada a hijo" en el escenario en el que el Padre crea y asigna una mesada a su hijo de forma exitosa. En este caso, el proceso se completó en 16 segundos y se requirieron 3 interacciones para finalizar la tarea.

Secuencia:

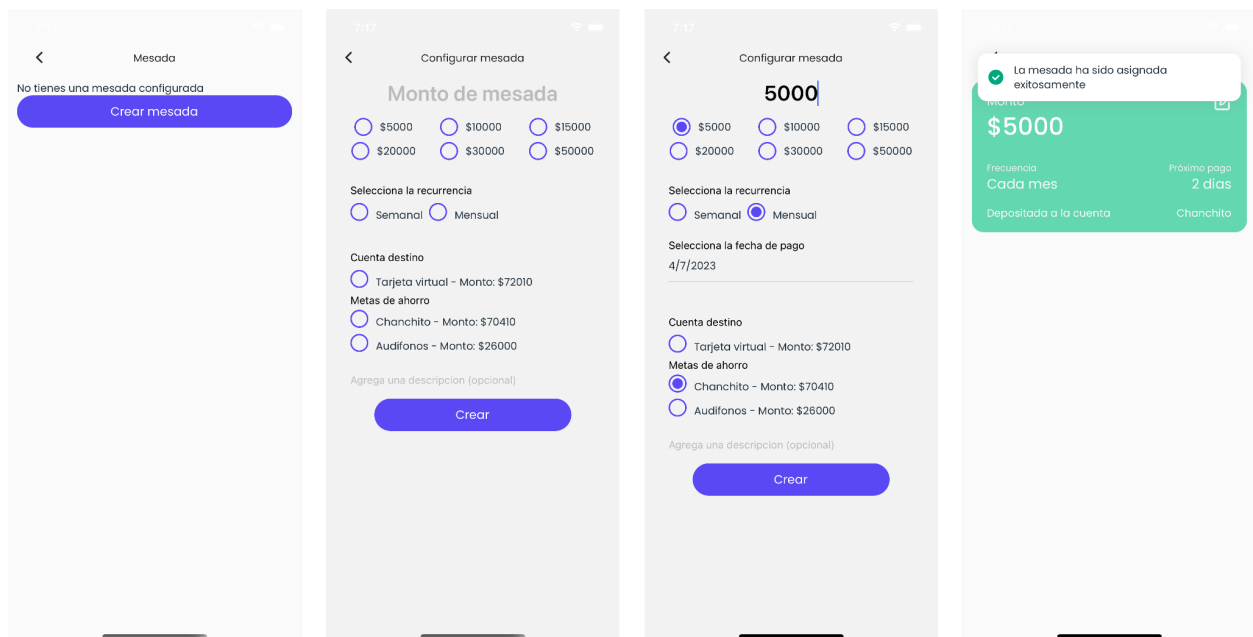
- 1: El Padre se encuentra en la pantalla del detalle de la mesada de su hijo Bastian Contreras, donde puede ver el botón para crear la mesada.
- 2: El Padre presiona el botón para crear la mesada.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino.
- 4: El Padre completa el formulario con un monto de \$5.000, frecuencia mensual, cuenta de destino Chanchito y fecha de pago el 7 de cada mes. Luego, confirma y envía el formulario.
- 5: El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, la cual muestra los detalles de la mesada configurada, pero con una fecha de próximo pago imprecisa, indicando que el próximo pago es en 2 días.

Resultado: Exitoso con imprecisión. El Padre configura con éxito la mesada para su hijo Bastian Contreras, pero la fecha de próximo pago mostrada no coincide con la fecha esperada.

Tiempo de ejecución: 16 segundos

Interacciones para completar la tarea: 3

Interacciones para completar la tarea desde la pantalla inicial: 5 interacciones



## Caso de prueba 8: Editar mesada de hijo

Escenario: Edición exitosa de mesada.

Descripción: Se prueba el caso de uso "Editar mesada de hijo" en el escenario en el que el Padre edita la mesada asignada a su hijo de forma exitosa.

Secuencia:

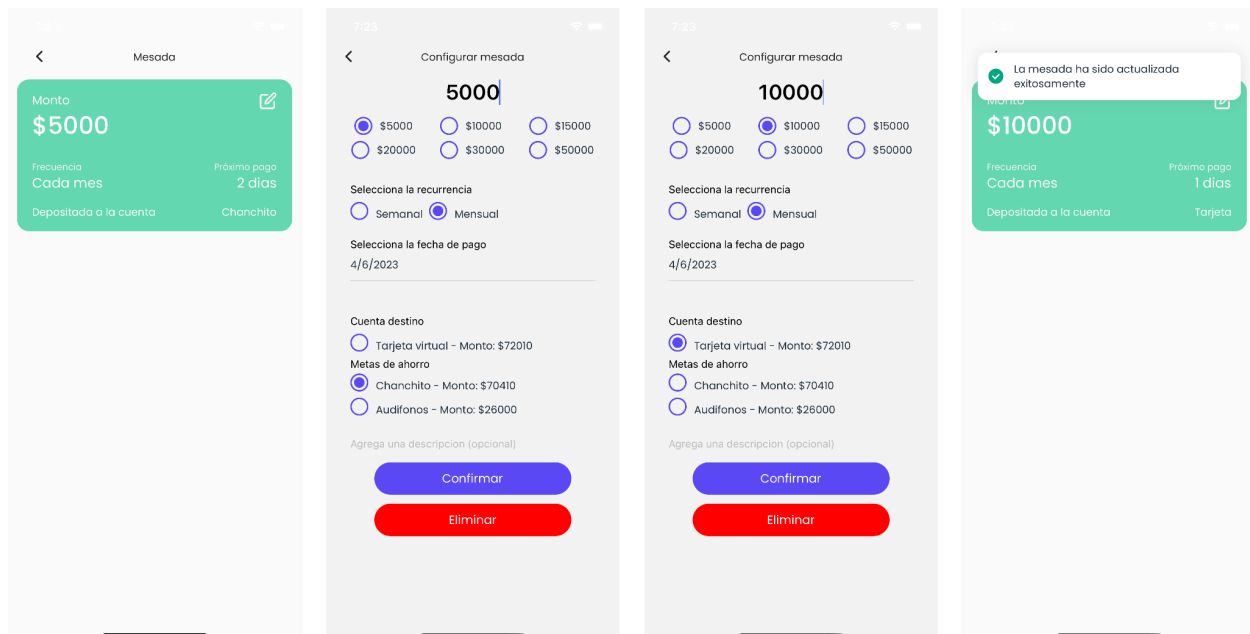
- 1: El Padre se encuentra en la pantalla del detalle de la mesada de su hijo Bastian Contreras, donde puede ver el botón para editar la mesada previamente configurada.
- 2: El Padre presiona el botón para editar la mesada previamente configurada.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino. Los campos del formulario están previamente completados con la configuración actual de la mesada (\$5.000, frecuencia mensual, cuenta de destino Chanchito, próximo pago en 2 días).
- 4: El Padre edita el formulario, cambiando el monto a \$10.000 y la cuenta de destino a Tarjeta. Luego, confirma y envía el formulario.
- 5: El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, la cual muestra los detalles actualizados de la mesada configurada, pero con una fecha de próximo pago imprecisa, indicando que el próximo pago es en 1 día.

Resultado: Exitoso con imprecisión. El Padre edita con éxito la mesada para su hijo Bastian Contreras, pero la fecha de próximo pago mostrada no coincide con la fecha esperada.

Tiempo de ejecución: 12 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 5 interacciones



## Caso de prueba 9: Eliminar mesada de hijo

Escenario: Eliminación exitosa de la mesada del hijo

Descripción: Se prueba el caso de uso "Eliminar mesada de hijo" en el escenario en el que el Padre elimina la mesada asignada a su hijo de forma exitosa. En este caso, el proceso se completó en 10 segundos y se requirieron 2 interacciones para finalizar la tarea.

Secuencia:

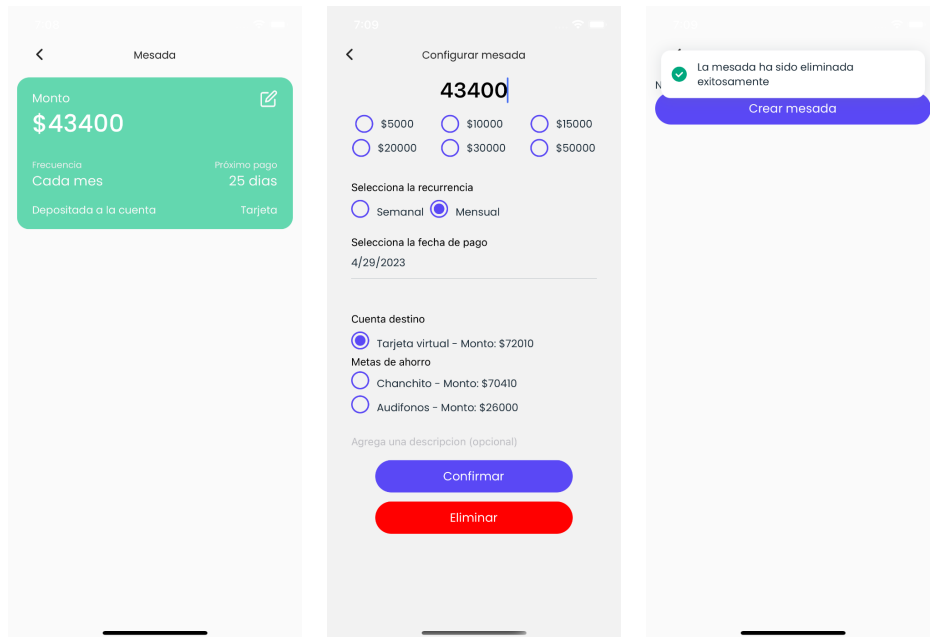
- 1: El Padre se encuentra en la pantalla del detalle de la mesada de su hijo Bastian Contreras, donde puede ver el botón para editar la mesada previamente configurada.
- 2: El Padre presiona el botón para editar la mesada.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de la asignación recurrente, la recurrencia del pago y la cuenta de destino. Los campos del formulario están previamente completados con la configuración actual de la mesada.
- 4: El Padre presiona el botón para eliminar la mesada previamente configurada.
- 5: El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de mesada, la cual advierte que no existe una mesada configurada.

Resultado: Exitoso. El Padre elimina con éxito la mesada asignada a su hijo Bastian Contreras, sin errores al eliminar la mesada.

Tiempo de ejecución: 10 segundos

Interacciones para completar la tarea: 2 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 4 interacciones



## Caso de prueba 10: Ver detalle tareas de hijo

Escenario: Visualización exitosa del detalle de tareas sin tareas asignadas

Descripción: Se prueba el caso de uso "Ver detalle tareas de hijo" en el escenario en el que el Padre visualiza el detalle de las tareas de su hijo sin tener tareas asignadas. En este caso, el proceso se completó con éxito.

Secuencia:

1: El Padre se encuentra en la pantalla del detalle general de su hijo Bastian Contreras, donde puede ver la tarjeta de tareas que muestra el monto por ganar esta semana y la cantidad de tareas pendientes de confirmación.

2: El Padre presiona la tarjeta de tareas.

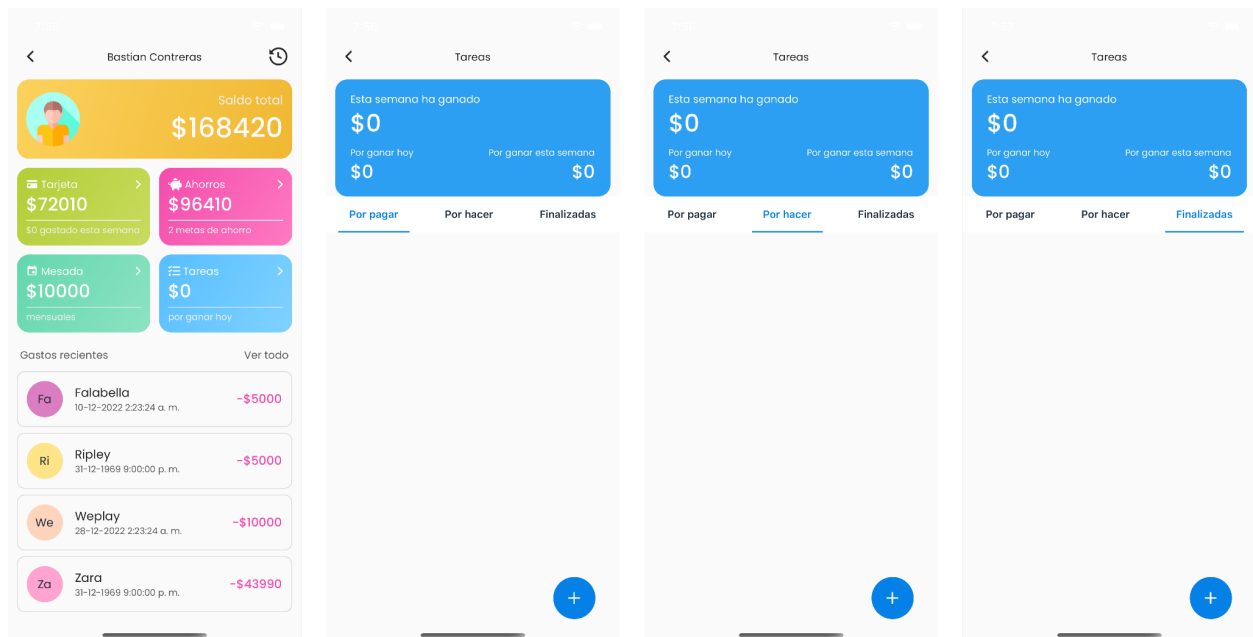
3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de las tareas de su hijo Bastian Contreras, incluyendo el monto ganado esta semana (\$0), el monto por ganar hoy (\$0), el monto por ganar esta semana (\$0), un botón flotante para crear una tarea y una lista vacía de las tareas del hijo, permitiendo filtrar por tareas por pagar, tareas por hacer y tareas finalizadas.

Resultado: Exitoso. El Padre visualiza con éxito el detalle de las tareas de su hijo Bastian Contreras, sin tareas asignadas en ninguna categoría (por pagar, por hacer y finalizadas).

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 2 interacciones



## Caso de prueba 11: Asignar tarea a hijo

Escenario: Creación y asignación exitosa de una tarea a hijo

Descripción: Se prueba el caso de uso "Asignar tarea a hijo" en el escenario en el que el Padre crea y asigna una tarea a su hijo. En este caso, el proceso se completó con éxito.

Secuencia:

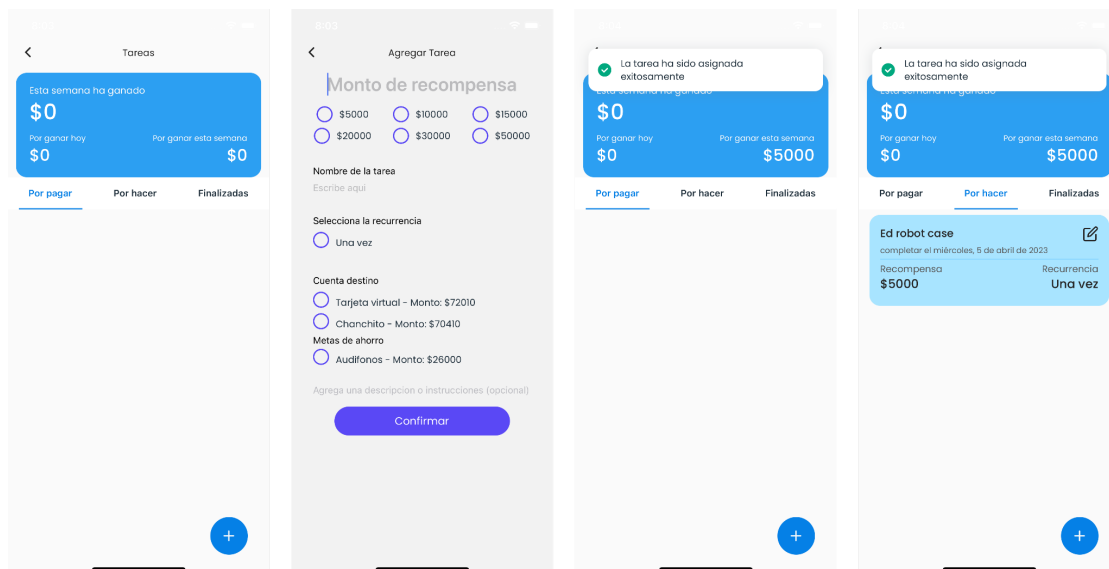
- 1: El Padre se encuentra en la pantalla del detalle de tareas de su hijo Bastian Contreras, donde puede ver el monto por ganar esta semana y la cantidad de tareas pendientes de confirmación, así como el botón flotante para crear una tarea.
- 2: El Padre presiona el botón flotante para crear una tarea.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el monto de recompensa, el nombre de la tarea, la recurrencia de la tarea, la cuenta de destino y una nota opcional.
- 4: El Padre completa el formulario con los siguientes datos: nombre de la tarea "Ed robot case", recurrencia "Una vez", monto de recompensa "\$5.000", fecha "6 de abril de 2023", cuenta de destino "Tarjeta virtual" y sin descripción.
- 5: El Padre confirma y envía el formulario.
- 6: El sistema presenta al Padre una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de tareas, donde se muestra la nueva tarea en la categoría 'Por hacer'.

Resultado: Exitoso con imprecisión. El Padre crea y asigna con éxito una tarea a su hijo Bastian Contreras, que aparece en la categoría 'Por hacer' en el detalle de tareas. Sin embargo, la fecha mostrada posteriormente en la tarjeta de la tarea señala 5 de abril de 2023.

Tiempo de ejecución: 26 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 5 interacciones



## Caso de prueba 12: Pagar tarea finalizada de hijo

Escenario: Pago exitoso de tarea finalizada y actualización de resumen de tareas

Descripción: Se prueba el caso de uso "Pagar tarea finalizada de hijo" en el escenario en el que el Padre confirma la finalización de una tarea asignada a su hijo y efectúa el pago correspondiente. En este caso, el proceso se completó con éxito y el resumen de las tareas se actualizó correctamente.

Secuencia:

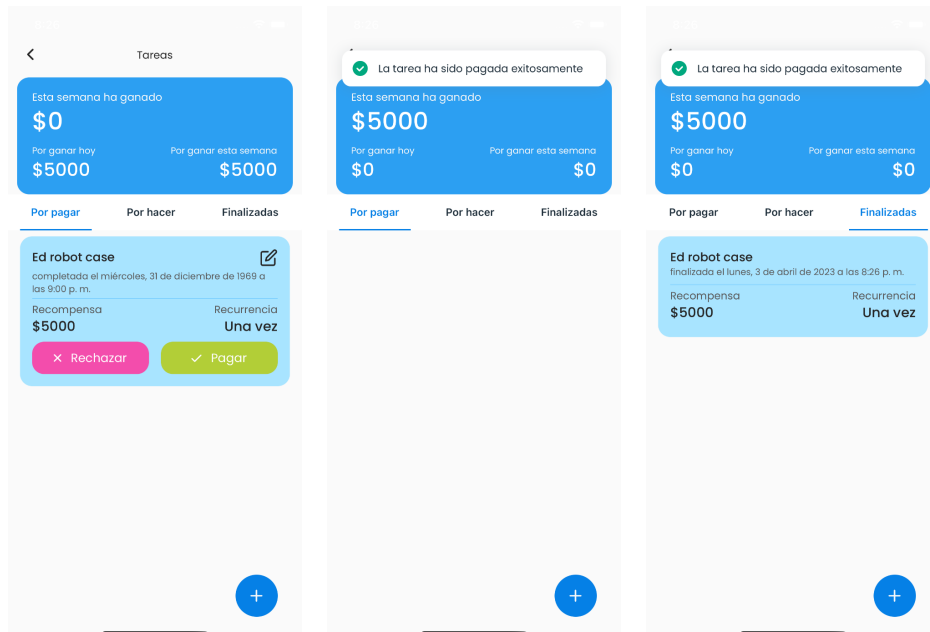
- 1: El Padre se encuentra en la pantalla del detalle de tareas de su hijo Bastian Contreras. Visualiza el resumen de las tareas, el cual indica que esta semana ha ganado \$0, tiene \$5.000 por ganar hoy y \$5.000 por ganar esta semana.
- 2: El Padre presiona el botón 'Pagar' dentro de la tarjeta con los detalles de la tarea "Ed robot case" en estado 'Por pagar'.
- 3: El sistema presenta al Padre una alerta de éxito, se mueve la tarea desde la pestaña de tareas 'Por pagar' hacia la pestaña de tareas 'Finalizadas'.
- 4: El resumen de las tareas se actualiza correctamente, indicando que esta semana ha ganado \$5.000, tiene \$0 por ganar hoy y \$0 por ganar esta semana.

Resultado: Exitoso. El Padre paga con éxito la tarea finalizada y el resumen de las tareas se actualiza correctamente.

Tiempo de ejecución: 9 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 13: Rechazar finalización de tarea de hijo

Escenario: Rechazo exitoso de la finalización de una tarea y actualización de estado de tarea

Descripción: Se prueba el caso de uso "Rechazar finalización de tarea de hijo" en el escenario en el que el Padre rechaza la finalización de una tarea asignada a su hijo. En este caso, el proceso se completó con éxito y la tarea se movió correctamente desde la categoría 'Por pagar' a la categoría 'Por hacer'.

Secuencia:

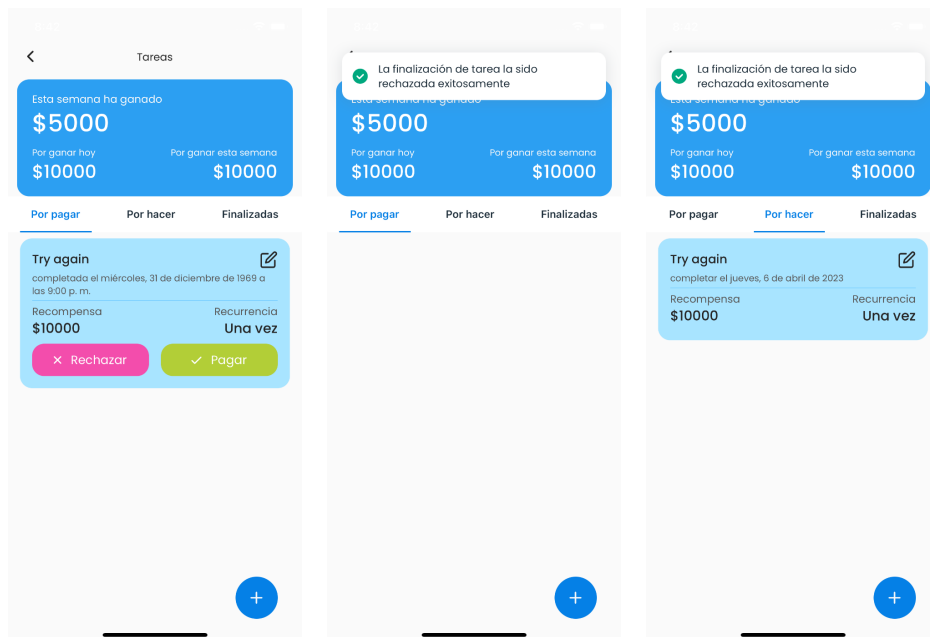
- 1: El Padre se encuentra en la pantalla del detalle de tareas de su hijo Bastian Contreras. Visualiza las tareas en la pestaña 'Por pagar'.
- 2: El Padre presiona el botón 'Rechazar' dentro de la tarjeta con los detalles de la tarea "Try again" en estado 'Por pagar'.
- 3: El sistema presenta al Padre una alerta de éxito, se mueve la tarea desde la pestaña de tareas 'Por pagar' hacia la pestaña de tareas 'Por hacer'.
- 4: El estado de la tarea del hijo se actualiza correctamente, pasando del estado 'Por Pagar' a 'Por hacer'.

Resultado: Exitoso. El Padre rechaza con éxito la finalización de la tarea y el estado de la tarea se actualiza correctamente.

Tiempo de ejecución: 9 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 14: Ver detalle tarjeta de hijo

Escenario: Visualización exitosa del detalle de la tarjeta del hijo

Descripción: Se prueba el caso de uso "Ver detalle tarjeta de hijo" en el escenario en el que el Padre visualiza el detalle de la tarjeta de su hijo. En este caso, el proceso se completó con éxito y se mostraron los detalles de la tarjeta correctamente.

Secuencia:

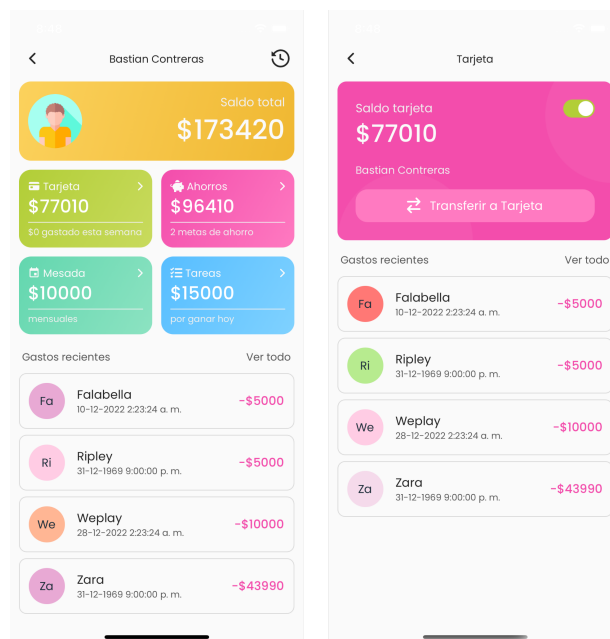
- 1: El Padre se encuentra en la pantalla del detalle general de su hijo Bastian Contreras. Visualiza la tarjeta de tarjeta con el saldo actual (\$77.010) y el monto gastado esta semana (\$0).
- 2: El Padre presiona la tarjeta de tarjeta.
- 3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de la tarjeta de su hijo, incluyendo saldo actual (\$77.010), el estado de la tarjeta (activa), un botón para transferir directamente a la tarjeta y un listado de los 4 gastos recientes efectuados con la tarjeta.

Resultado: Exitoso. El Padre accede con éxito a la pantalla de detalle de la tarjeta de su hijo y visualiza la información correspondiente.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 2 interacciones





## Caso de prueba 15: Bloquear temporalmente tarjeta de hijo

Escenario: Bloqueo exitoso de la tarjeta del hijo

Descripción: Se prueba el caso de uso "Bloquear temporalmente tarjeta de hijo" en el escenario en el que el Padre desactiva temporalmente la tarjeta de su hijo. En este caso, el proceso se completó con éxito y se bloqueó la tarjeta correctamente.

Secuencia:

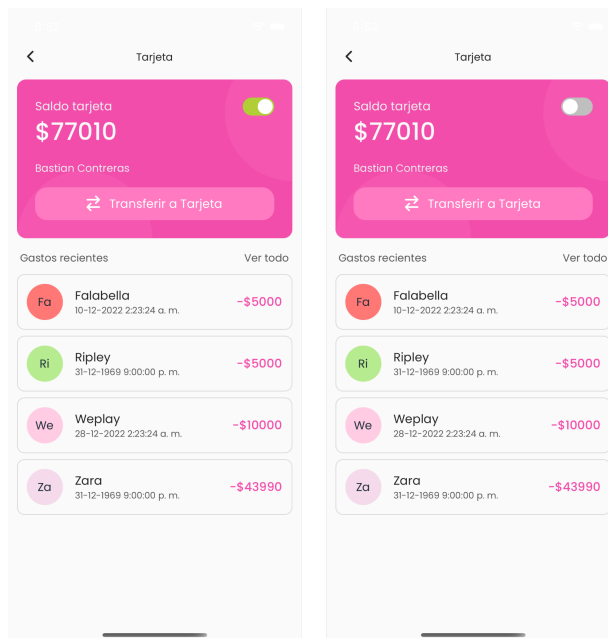
- 1: El Padre se encuentra en la pantalla del detalle de la tarjeta de su hijo Bastian Contreras. Visualiza el estado de la tarjeta como activa y el slider en estado habilitado.
- 2: El Padre presiona el slider activo dentro de la tarjeta del hijo.
- 3: El sistema presenta una animación exitosa del slider, el cual pasa desde el estado habilitado al estado deshabilitado, confirmando el bloqueo temporal de la tarjeta.

Resultado: Exitoso. El Padre bloquea con éxito la tarjeta de su hijo, cambiando el estado de la tarjeta de 'Activa' a 'Bloqueada'.

Tiempo de ejecución: 7 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 16: Mostrar información sensible de tarjeta

Escenario: Visualización exitosa de los detalles de la tarjeta

Descripción: Se prueba el caso de uso "Mostrar información sensible de tarjeta" en el escenario en el que el Hijo visualiza los detalles de su tarjeta. En este caso, el proceso se completó con éxito y se mostraron los detalles de la tarjeta correctamente.

Secuencia:

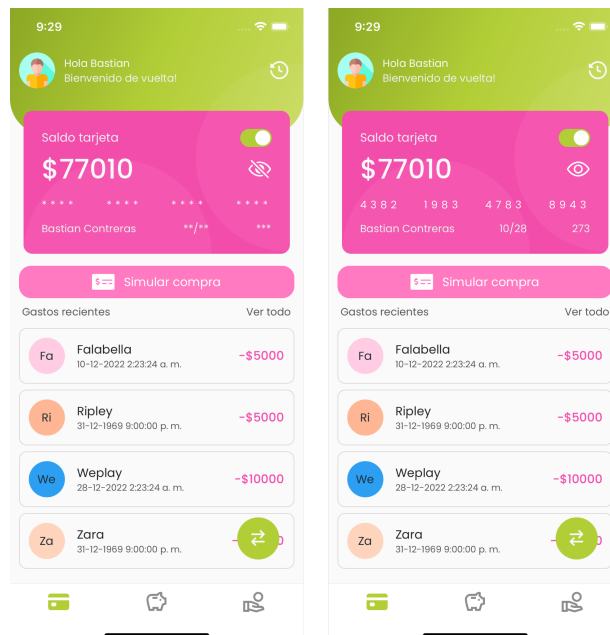
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla principal y ve su tarjeta con los detalles ocultos (número de tarjeta, fecha de caducidad y CVV) mostrados como asteriscos.
- 2: El Hijo presiona el botón con el ícono de un ojo cerrado dentro de su tarjeta.
- 3: El sistema cambia el ícono del botón por un ojo abierto, y se muestran los detalles de su tarjeta previamente ocultos por seguridad: Número de tarjeta (4382 1983 4783 8943), fecha de expiración (10/28) y código de seguridad (273).

Resultado: Exitoso. El Hijo visualiza con éxito los detalles de su tarjeta, incluyendo número de tarjeta, fecha de caducidad y código de seguridad.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 17: Ocultar información sensible de tarjeta

Escenario: Ocultar exitosamente los detalles de la tarjeta

Descripción: Se prueba el caso de uso "Ocultar información sensible de tarjeta" en el escenario en el que el Hijo oculta los detalles de su tarjeta. En este caso, el proceso se completó con éxito y se ocultaron los detalles de la tarjeta correctamente.

Secuencia:

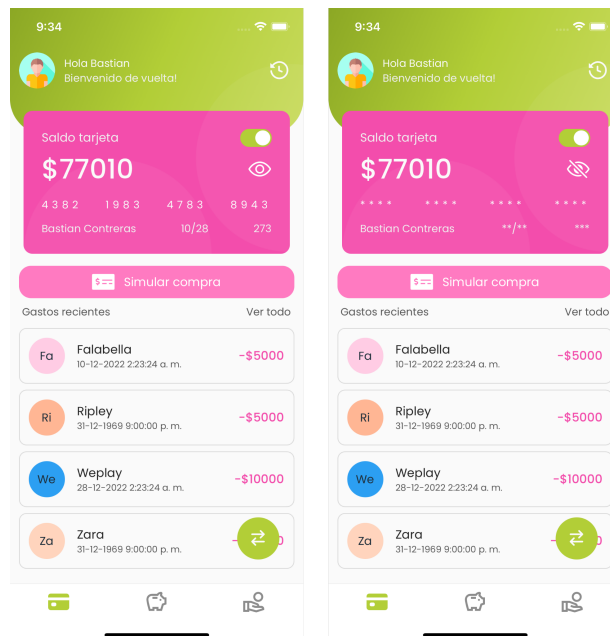
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla principal y ve su tarjeta con los detalles visibles: Número de tarjeta (4382 1983 4783 8943), fecha de expiración (10/28) y código de seguridad (273).
- 2: El Hijo presiona el botón con el ícono de un ojo abierto dentro de su tarjeta.
- 3: El sistema cambia el ícono del botón por un ojo cerrado, y se ocultan los detalles de su tarjeta, reemplazando cada caracter de número de tarjeta, fecha de expiración y código de seguridad por un símbolo “\*”.

Resultado: Exitoso. El Hijo oculta con éxito los detalles de su tarjeta, incluyendo número de tarjeta, fecha de caducidad y código de seguridad.

Tiempo de ejecución: 5 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 18: Transferir entre cuentas

Escenario: Transferencia exitosa entre cuentas

Descripción: Se prueba el caso de uso "Transferir entre cuentas" en el escenario en el que el Hijo realiza una transferencia entre sus cuentas. En este caso, la transferencia se completó con éxito y los saldos de las cuentas se actualizaron correctamente.

Secuencia:

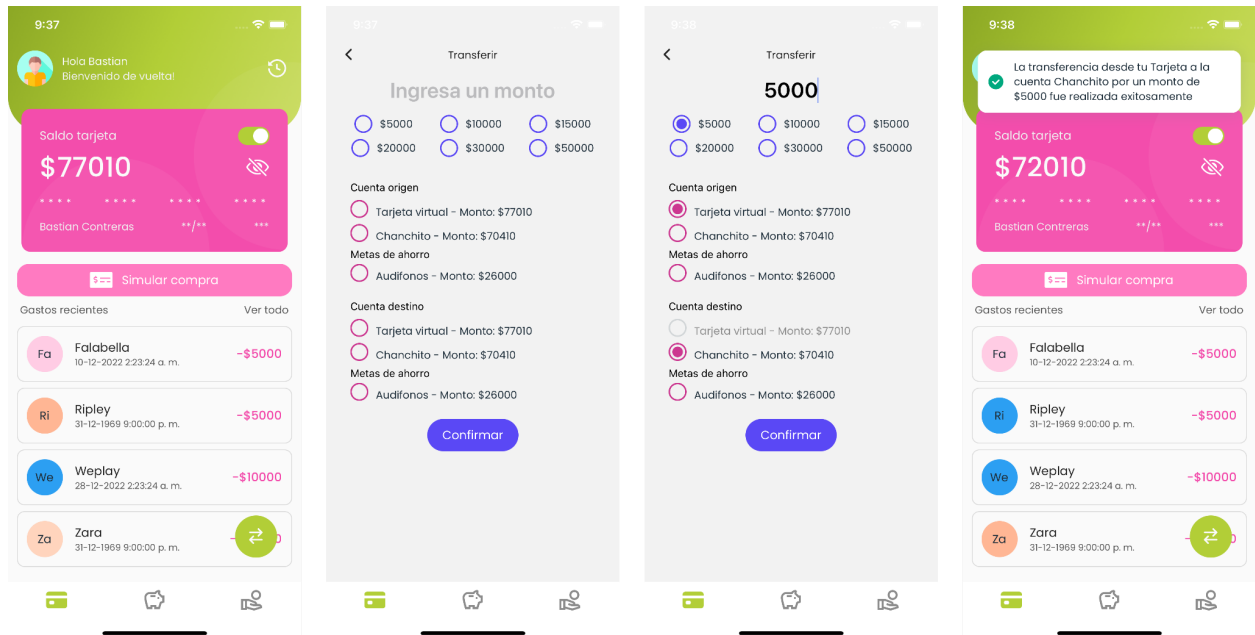
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla principal y ve su tarjeta con un saldo de \$77.010.
- 2: El Hijo presiona el botón flotante representado por un ícono de flechas opuestas.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario.
- 4: El Hijo ingresa un monto de \$5.000, selecciona como cuenta origen la tarjeta virtual y como cuenta destino la cuenta Chanchito.
- 6: El Hijo confirma y envía el formulario.
- 7: El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal.

Resultado: Exitoso. La transferencia se realiza con éxito y los saldos de las cuentas se actualizan correctamente. El saldo en la tarjeta virtual se reduce a \$72.010 y el saldo en la cuenta Chanchito aumenta en \$5.000.

Tiempo de ejecución: 19 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 19: Ver detalle de ahorros

Escenario: Visualizar el detalle de los ahorros del Hijo

Descripción: Se prueba el caso de uso "Ver detalle de ahorros" en el escenario en el que el Hijo visualiza el detalle de sus ahorros.

Secuencia:

1: El Hijo, Bastian Contreras, se encuentra en una pantalla que incluye la barra de navegación inferior.

2: El Hijo presiona el ícono de una alcancía con forma de cerdo en la barra de navegación inferior.

3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de sus ahorros: Monto total ahorrado: \$101.410.

Cuenta Chanchito: \$75.410.

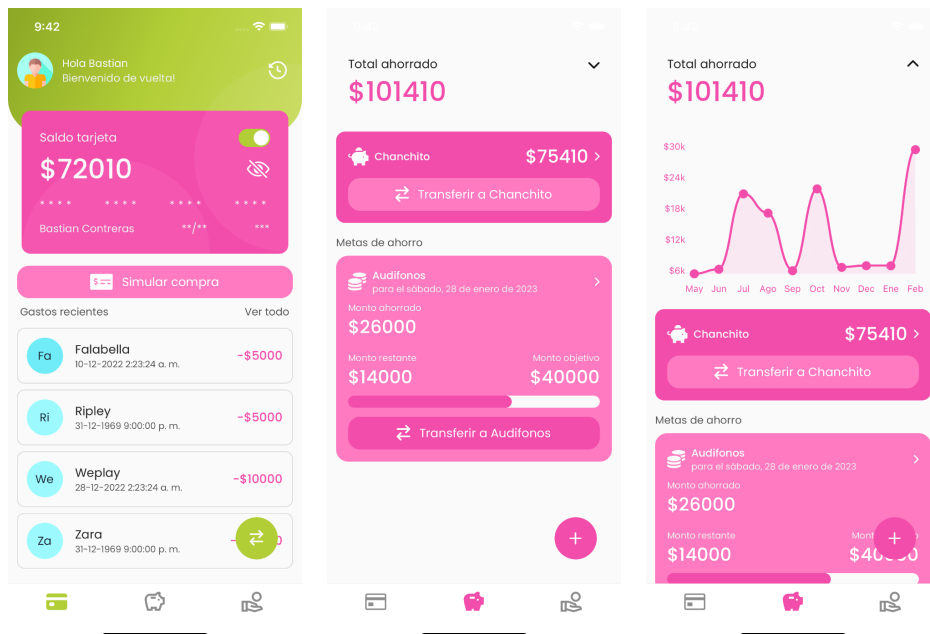
Meta de ahorro "Audifonos": monto ahorrado de \$26.000, monto restante de \$14.000 y monto objetivo de \$40.000.

Resultado: Exitoso. La pantalla de detalle de ahorros se muestra correctamente con la información actualizada del Hijo, incluyendo los montos de las cuentas y metas de ahorro.

Tiempo de ejecución: 8 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 20: Crear meta de ahorro

Escenario: Crear una nueva meta de ahorro

Descripción: Se prueba el caso de uso "Crear meta de ahorro" en el escenario en el que el Hijo crea una nueva meta de ahorro.

Secuencia:

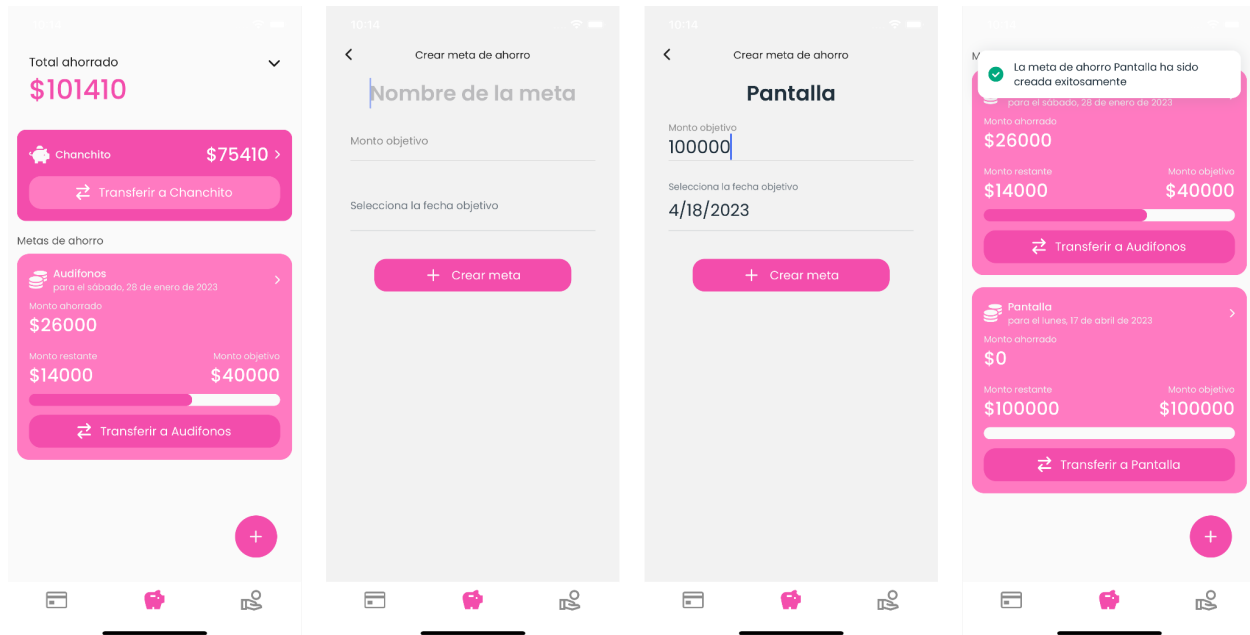
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla del detalle de sus ahorros.
- 2: El Hijo presiona el botón flotante con un símbolo "+" para crear la meta de ahorro.
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la meta, el monto objetivo y la fecha objetivo.
- 4: El Hijo completa el formulario con los siguientes datos: Nombre de la meta: "Pantalla", Monto objetivo: \$100.000, Fecha objetivo: 18 de abril del 2023.
- 5: El Hijo confirma y envía el formulario.
- 6: El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de sus ahorros, el cual muestra una nueva tarjeta con los detalles de la meta de ahorro creada.

Resultado: Exitoso con imprecisión. La meta de ahorro "Pantalla" se crea correctamente con el monto objetivo de \$100.000, monto ahorrado de \$0 y monto restante de \$100.000. Sin embargo, se muestra la fecha objetivo con el valor 17 de abril del 2023.

Tiempo de ejecución: 22 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 4 interacciones



## Caso de prueba 21: Editar meta de ahorro

Escenario: Editar el nombre de una meta de ahorro existente

Descripción: Se prueba el caso de uso "Editar meta de ahorro" en el escenario en el que el Hijo edita el nombre de una meta de ahorro existente.

Secuencia:

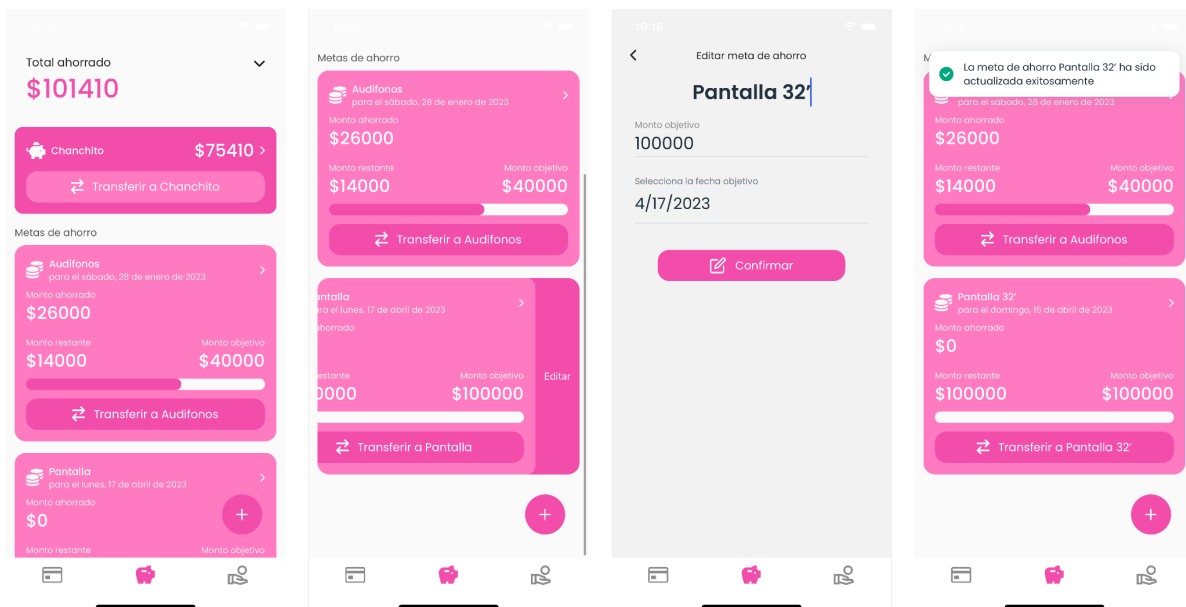
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla del detalle de sus ahorros.
- 2: El Hijo desliza hacia la izquierda la tarjeta de la meta de ahorro "Pantalla" que desea editar, con un monto objetivo de \$100.000 y la fecha objetivo 16 de abril de 2023.
- 3: El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón "Editar".
- 4: El Hijo presiona el botón "Editar".
- 5: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la meta, el monto objetivo y la fecha objetivo. Los campos del formulario están previamente completados con la configuración actual de la meta de ahorro.
- 6: El Hijo modifica el nombre de la meta de ahorro "Pantalla" a "Pantalla 32" y confirma y envía el formulario.
- 7: El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla de detalle de sus ahorros, el cual muestra la tarjeta con los detalles actualizados de la meta de ahorro editada.

Resultado: Exitoso con imprecisión. La meta de ahorro se edita correctamente, cambiando su nombre de "Pantalla" a "Pantalla 32". Sin embargo, tras la edición la fecha objetivo de la meta de ahorro indica 15 de abril de 2023.

Tiempo de ejecución: 21 segundos

Interacciones para completar la tarea: 4 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 5 interacciones



## Caso de prueba 22: Eliminar meta de ahorro

Escenario: Eliminar una meta de ahorro existente

Descripción: Se prueba el caso de uso "Eliminar meta de ahorro" en el escenario en el que el Hijo elimina una meta de ahorro existente.

Secuencia:

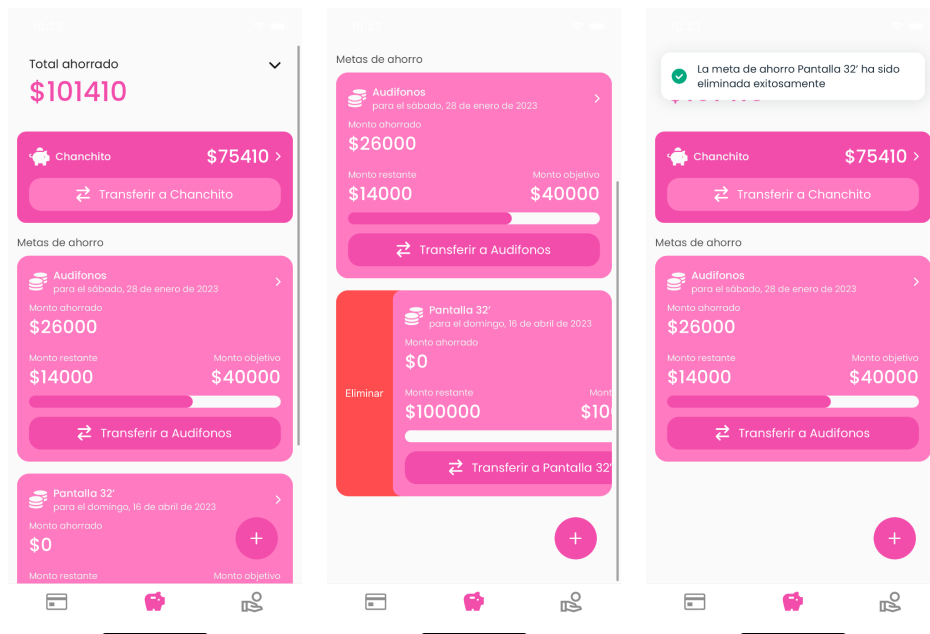
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla del detalle de sus ahorros.
- 2: El Hijo desliza hacia la derecha la tarjeta de la meta de ahorro "Pantalla 32" que desea eliminar.
- 3: El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón "Eliminar".
- 4: El Hijo presiona el botón "Eliminar".
- 5: El sistema presenta al Hijo una alerta de éxito y remueve la meta de ahorro "Pantalla 32" y transfiere el saldo ahorrado a su tarjeta.

Resultado: Exitoso. La meta de ahorro "Pantalla 32" se elimina correctamente y es eliminada del listado de metas de ahorro. El saldo ahorrado se transfiere a la tarjeta de Bastian Contreras. En este caso el saldo ahorrado era de \$0, por lo que se mantiene el saldo en la tarjeta.

Tiempo de ejecución: 11 segundos

Interacciones para completar la tarea: 2 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones





## Caso de prueba 23: Ver detalle de ganar dinero

Escenario: Visualizar el detalle de ganar dinero

Descripción: Se prueba el caso de uso "Ver detalle de ganar dinero" en el escenario en el que el Hijo visualiza el detalle de sus medios para ganar dinero.

Secuencia:

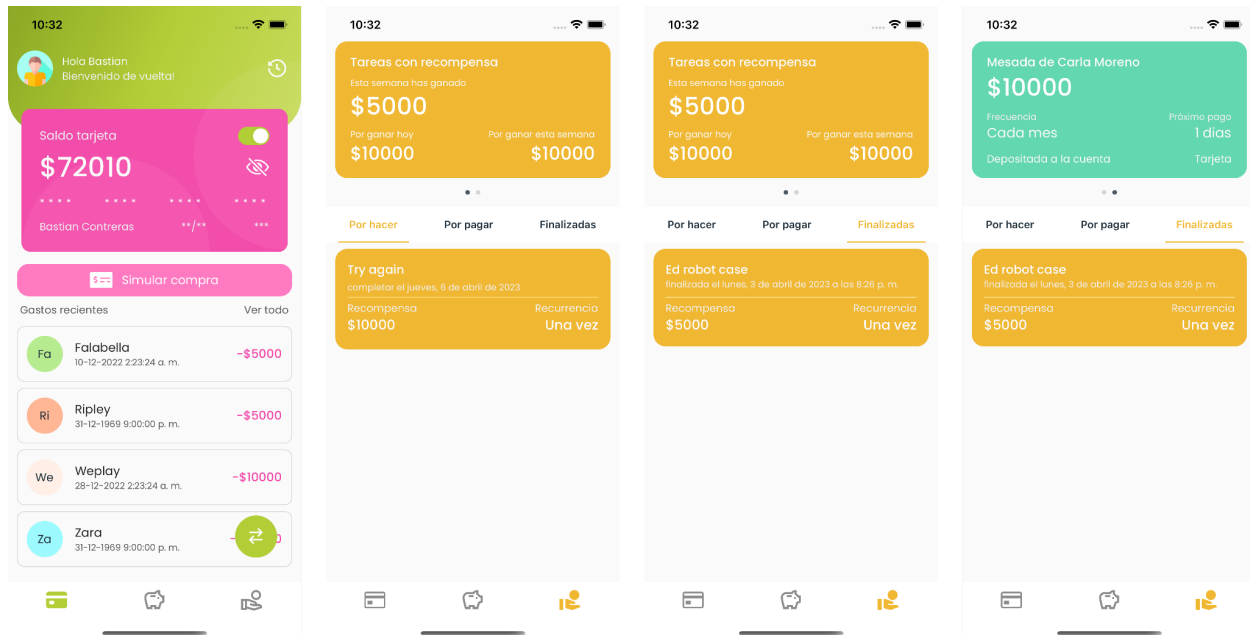
- 1: El Hijo, Bastian Contreras, se encuentra en una pantalla que incluye la barra de navegación inferior.
- 2: El Hijo presiona el ícono de una mano con una moneda en la barra de navegación inferior.
- 3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de sus medios para ganar dinero, incluyendo un carrusel que incluye la tarjeta que sintetiza la información de sus tareas con recompensa (monto ganado esta semana: \$5.000, monto por ganar hoy: \$10.000 y monto por ganar esta semana: \$10.000) y tarjetas de las mesadas que tiene asignadas (monto de mesada: \$10.000 mensuales depositada a la cuenta Tarjeta). Además, se presenta listado de las tareas que tiene asociadas, con la capacidad de filtrarlas mediante las pestañas "Por hacer" (tarea Try Again con una recompensa de \$10.000), "Por pagar" y "Finalizadas" (tarea Ed robot case con una recompensa de \$5.000).

Resultado: Exitoso. Se visualiza correctamente el detalle de sus medios para ganar dinero, incluyendo el resumen de tareas con recompensa y las mesadas asignadas.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 1 interacción



## Caso de prueba 24: Completar tarea "Por hacer"

Escenario: Completitud exitosa de tarea "Por hacer".

Descripción: Se prueba el caso de uso "Completar tarea 'Por hacer'" en el escenario en el que el Hijo confirma la realización de una tarea en el estado "Por hacer". En este caso, el proceso se completó con éxito, pero con una imprecisión en la fecha de completitud.

Secuencia:

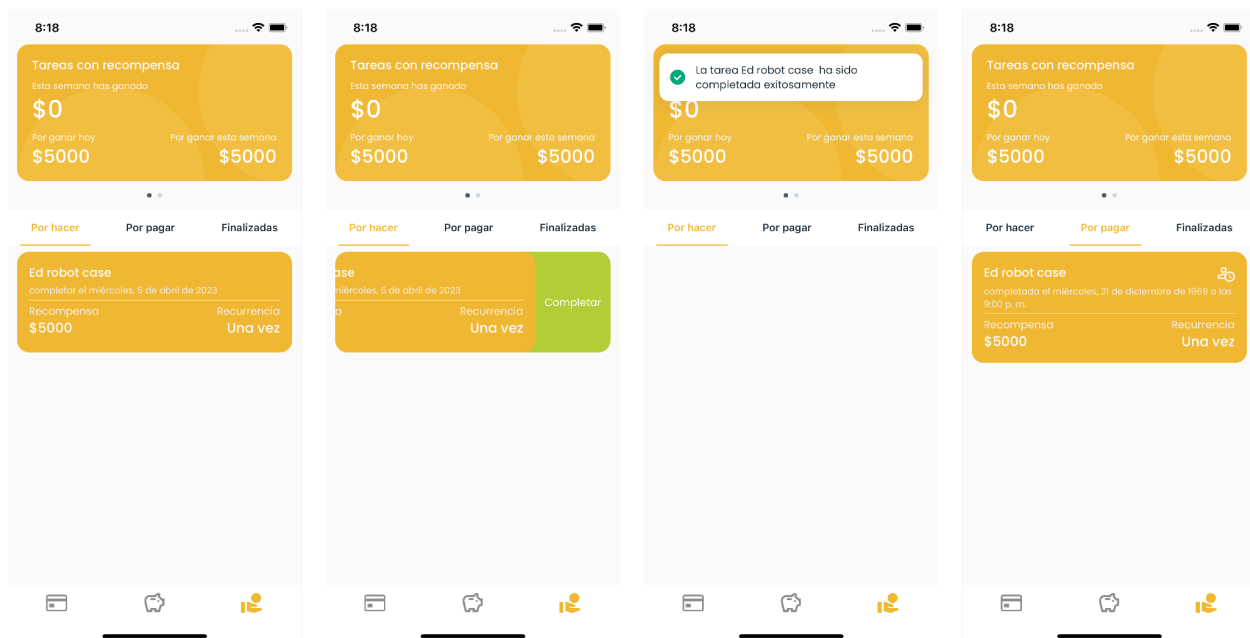
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla del detalle de ganar dinero, donde puede ver las tareas en la pestaña "Por hacer", incluyendo la tarea "Ed robot case".
- 2: El Hijo desliza hacia la izquierda la tarjeta de la tarea "Ed robot case" que desea completar.
- 3: El sistema presenta una animación de carrusel de la tarjeta y se visualiza un botón "Completar".
- 4: El Hijo presiona el botón "Completar".
- 5: El sistema presenta al Hijo una alerta de éxito, cambia el estado de la tarea de "Por hacer" a "Por pagar", y se mueve la tarjeta de la tarea desde la pestaña "Por hacer" a "Por pagar". Sin embargo, la fecha de completitud indica que fue el miércoles 31 de diciembre de 1969, lo que es una imprecisión.

Resultado: Exitoso con imprecisión. El Hijo completa con éxito una tarea en el estado "Por hacer", pero con una imprecisión en la fecha de completitud indicando 31 de diciembre de 1969.

Tiempo de ejecución: 12 segundos

Interacciones para completar la tarea: 2 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 25: Simular compra

Escenario: Simular una compra en la tienda Ripley

Descripción: Se prueba el caso de uso "Simular compra" en el escenario en el que el Hijo simula una compra por \$10.000 en la tienda Ripley.

Secuencia:

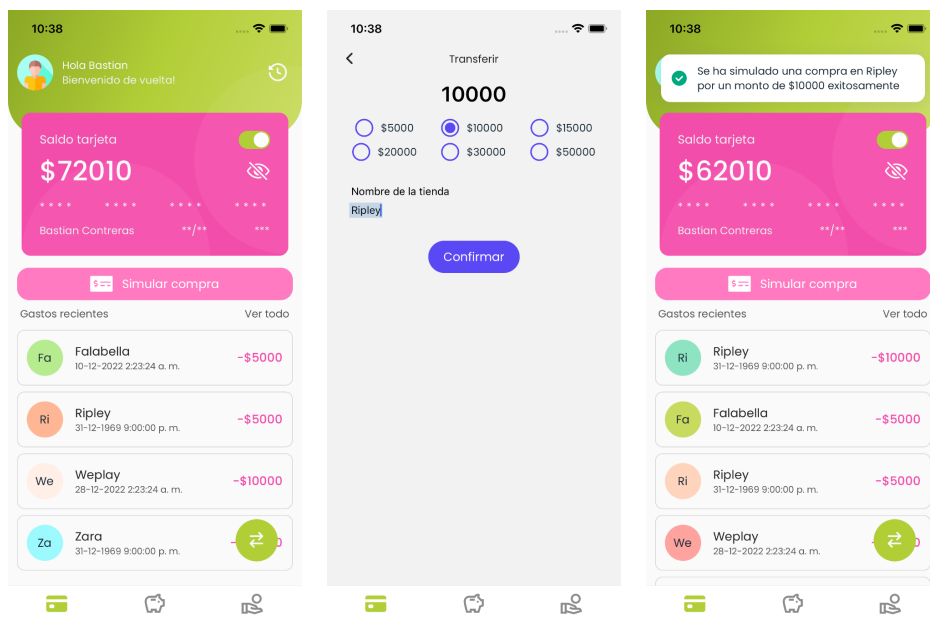
- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla principal.
- 2: El Hijo presiona el botón "Simular compra".
- 3: El sistema agrega al stack de navegación una pantalla que presenta un formulario solicitando el nombre de la tienda (Ripley), el monto de la compra (\$10.000) y la fecha de transacción.
- 4: El Hijo completa, confirma y envía el formulario.
- 5: El sistema presenta al Hijo una alerta de éxito y remueve del stack de navegación la pantalla del formulario, navegando a la pantalla principal, la cual muestra una nueva entrada con los detalles de la transacción (tienda Ripley, monto \$10.000).

Resultado: Exitoso con imprecisión. Simula correctamente una compra por \$10.000 en la tienda Ripley, y esta se muestra en la lista de gastos recientes. Sin embargo, la fecha de la compra se sitúa en el 31 de diciembre de 1969.

Tiempo de ejecución: 12 segundos

Interacciones para completar la tarea: 3 interacciones

Interacciones para completar la tarea desde la pantalla inicial: 3 interacciones



## Caso de prueba 26: Ver detalle de tarjeta

Escenario: Ver detalle de la tarjeta desde la pantalla de detalle de ahorros

Descripción: Se prueba el caso de uso "Ver detalle de tarjeta" en el escenario en el que el Hijo se encuentra en la pantalla de detalle de sus ahorros y navega hacia el detalle de su tarjeta.

Secuencia:

- 1: El Hijo, Bastian Contreras, se encuentra en la pantalla de detalle de sus ahorros.
- 2: El Hijo presiona el ícono de una tarjeta en la barra de navegación inferior.
- 3: El sistema agrega al stack de navegación una pantalla que presenta los detalles de su tarjeta, incluyendo saldo actual (\$62.010), el estado de la tarjeta (habilitada), un botón para mostrar/ocultar los detalles de la tarjeta (datos sensibles ocultos), un listado de los gastos recientes efectuados con la tarjeta (última transacción del gasto realizado en Ripley por \$10.000), un botón flotante para transferir entre sus cuentas, un botón para ver su actividad, su foto de perfil y un mensaje de bienvenida.

Resultado: Exitoso. Comienza en la pantalla de detalle de sus ahorros, navega hacia el detalle de su tarjeta y visualiza correctamente el saldo de \$62.010, tarjeta habilitada, datos sensibles ocultos y un listado de gastos recientes con la última transacción del gasto realizado en Ripley por \$10.000.

Tiempo de ejecución: 6 segundos

Interacciones para completar la tarea: 1 interacción

Interacciones para completar la tarea desde la pantalla inicial: 0 interacciones

