



Shared Family

Emilio Garcia del Arco

Diseño de Experiencia de usuario y dispositivos móviles

Alberto Zamarrón Pinilla

Carlos Caballero González

20/06/2016



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Shared Family
Nombre del autor:	Emilio García del Arco
Nombre del consultor:	Alberto Zamarrón Pinilla Carlos Caballero González
Fecha de entrega (mm/aaaa):	06/2016
Área del Trabajo Final:	Área de postgrado Informática, Multimedia y Telecomunicación
Titulación:	<i>Diseño de Experiencia de usuario y dispositivos móviles.</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Se trata de diseñar y desarrollar una App dirigida a los padres separados con hijos en custodia compartida, para ayudar a la gestión de los calendarios de custodia compartida, así como para gestionar y controlar los gastos correspondientes a cada hijo en común.</p> <p>La aplicación debería ser capaz de servir de herramienta a los padres para gestionar todo lo referente a la custodia compartida de sus hijos, haciendo más fácil y fluida la comunicación entre los cónyuges y la gestión de todo lo referente a ello.</p> <p>También podría servir de ayuda a los propios hijos para mantenerse en comunicación con sus padres.</p> <p>El trabajo consiste en el análisis de las necesidades de las personas en la situación descrita anteriormente, apoyándose en las mismas personas para definir los requerimientos y funcionalidades más importantes de las que debería disponer la misma.</p> <p>Una vez hecho esto, se llevaría a cabo el diseño y evaluación con los usuarios, para proceder al desarrollo de la misma.</p>	

Abstract (in English, 250 words or less):

The project is about the design and development of an application addressed to divorced parents with children on shared custody, to help them managing shared custody calendars and managing the expenses corresponding to every shared child.

The application should be able to be a tool to manage all the tasks referred to the children shared custody, making easy the communication between parents, and also to help the communication between children and parents.

The job consist in analyse the necessities of people in situation described above, taking into account the above described persons in order to define the most important requirements and functionalities that the application has to cover.

Once done, the next step will be the design and the evaluation with users, and finally gone with the application development.

Palabras clave (entre 4 y 8):

Custodia, compartida, padres, hijos, calendario custodia, gastos compartidos.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	5
2. Desarrollo del proyecto.....	7
2.1 Investigación con usuarios.....	7
2.2 Diseño de la interacción.....	9
2.3 Evaluación con usuarios.....	19
2.4 Desarrollo de la aplicación.....	24
2.4.1 Diagrama de clases.....	24
2.4.2 Diagrama de Base de Datos.....	25
2.4.3 Implementación.....	26
3. Conclusiones.....	54
4. Glosario.....	55
5. Bibliografía.....	56
6. Anexos.....	57

1. Introducción

1.1 Contexto y justificación del Trabajo

Se trata de una App dirigida a los padres separados con hijos en custodia compartida, para ayudar a la gestión de los calendarios de custodia compartida, así como para gestionar y controlar los gastos correspondientes a cada hijo en común.

El usuario podrá configurar varios perfiles de custodia compartida (hay personas que comparten custodia de hijos con diferentes exparejas). Para cada perfil se debería poder indicar el cónyuge, los hijos compartidos con cada cónyuge, las cuentas compartidas, el calendario de custodia, etc.

Se podrán gestionar eventos y tareas en el calendario, privados o compartidos con el cónyuge.

Se podrán gestionar los gastos compartidos, el usuario podrá ver los movimientos de la cuenta compartida, añadir movimientos y categorizarlos.

El usuario podrá controlar dichos gastos mediante informes de análisis de los mismos, viendo el saldo total de la cuenta y el saldo de cada cónyuge, pudiendo desglosar por hijos, categorías, etc.

La App dispondrá de un módulo de mensajería instantánea para permitir a los padres comunicarse de forma rápida.

La App también gestionará notificaciones en función de la configuración, informando al usuario de evento en el calendario y movimientos en las cuentas compartidas y mensajería instantánea.

El motivo principal de haber escogido esta propuesta para la realización de Trabajo Final del Master es que es una propuesta real de desarrollo que se me ha puesto encima de la mesa, la cual considero muy interesante, dado que va dirigida a un público muy concreto, con unas necesidades muy concretas.

Por otra parte, el número de parejas separadas con hijos en custodia compartida se ha duplicado en los últimos años, lo que le da un plus más de interés.

La idea es aprovechar todo el proceso de investigación y análisis para el desarrollo de la aplicación completa, aprovechando también el desarrollo que se realice dentro de este Trabajo Final de Master como primera versión de la App.

1.2 Objetivos del Trabajo

- Dar al usuario la posibilidad de mantener diferentes perfiles de custodia compartida, pudiendo indicar el cónyuge, enviándole invitación para utilizar la App, los hijos compartidos, las cuentas compartidas y definición del calendario de custodia.
- Que el usuario pueda gestionar eventos y tareas en el calendario, privados o compartidos con el cónyuge.
- Que el usuario pueda gestionar los gastos compartidos en cada cuenta compartida definida, pudiendo ver los gastos e ingresos del cónyuge y pudiendo introducir sus propios gastos e ingresos, pudiendo categorizarlos.
- Que el usuario pueda disponer de informes de análisis de gastos, por perfil de custodia, por hijo, por cuenta y por categoría.
- Que el usuario pueda enviar y recibir mensajería instantánea a cada cónyuge.

Dado el tiempo y los recursos disponibles para la ejecución del proyecto, el alcance del mismo será:

- Definición de un único perfil de custodia por usuario.
- Una única cuenta compartida por hijo.
- No se desarrollarán los informes de análisis de gastos.
- La funcionalidad de mensajería instantánea será lo más sencilla posible. No queremos desarrollar un WhatsApp^[2]
- La aplicación se desarrollará únicamente para dispositivos Android^[5] de última generación.

1.3 Enfoque y método seguido

La metodología a utilizar será una combinación de metodologías cuantitativas y cualitativas.

Se comenzará utilizando metodologías cuantitativas para determinar qué personas serían susceptibles de utilizar la aplicación, así como para conocer que personas utilizan ya la aplicación de referencia 2houses[1]. Posteriormente, después de analizar los datos obtenidos, se centrará el estudio en las personas que cumplan las características deseadas para realizar un estudio cualitativo al respecto.

Para la recogida de datos cuantitativa se utilizarán las encuestas.

Se diseñará una encuesta con preguntas orientadas a determinar la predisposición de las personas a utilizar nuestra aplicación y a participar en los estudios cualitativos posteriores, así como para determinar si ya están utilizando la aplicación 2houses[1]. Se han de recoger datos acerca de la edad, sexo, perfil social, perfil laboral, etc., que posteriormente ayudarán a clasificar la información para obtener nuestro perfil o perfiles de usuarios potenciales.

Para realizar dichas encuestas, se visitarán diferentes foros de padres con hijos en custodia compartida, contactando con ellos y enviándoles la encuesta por correo electrónico.

Para la recogida de datos cualitativa se utilizará:

- ‘Diario de uso’, mientras utilizan la aplicación 2houses[1] durante unos días, fijándose más de lo habitual en como la utilizan, que utilidades echan en falta, que funcionalidades no acaban de ser de su agrado, etc.
- Encuesta – cuestionario acerca de la aplicación, para extraer las funcionalidades que más valoran los usuarios, las que echan en falta y las que mejorarían.
- Focus Group con todos los usuarios donde se plantean todas las cuestiones observadas mediante el ‘Diario de uso’.

Como se ha indicado anteriormente, la población a la que se dirige el estudio son las personas con hijos menores de edad en custodia compartida que utilizan la aplicación 2houses[1].

Sobre dicha población, obtenida visitando diferentes foros de personas con hijos en custodia compartida, se lanzarán las encuestas para el estudio cuantitativo y a partir de las mismas se obtendrá el marco de referencia, que serán las personas que cumplan los requisitos y que estén dispuestos a participar en los estudios cualitativos.

Con estos datos y utilizando algún software estadístico se podrá calcular el tamaño de la muestra.

Para calcular el tamaño de la muestra, se necesitará el tamaño total de nuestra población, que sería el volumen total de usuarios de los diferentes foros visitados.

Una vez obtenido, se podrá extrapolar el volumen de nuestras encuestas al total de personas.

Finalmente, para obtener la muestra definitiva, se seleccionarán elementos aleatoriamente.

Los datos obtenidos para el análisis cuantitativo mediante las encuestas deben ser procesados mediante algún software de procesamiento estadístico, el cual ayudará a procesar la información obtenida, pero antes se debe preparar dicha información:

- Revisando y analizando la calidad de la información obtenida, depurando los posibles errores.
- Codificando los datos obtenidos para que el sistema de procesamiento estadístico los pueda entender.

Ahora ya se estará en disposición de procesar la información y poder obtener el perfil o perfiles de usuarios potenciales de nuestra aplicación, así como si utilizan la aplicación 2houses[1] y la muestra de usuarios para la recogida de datos cualitativos.

Para los datos obtenidos en los análisis cualitativos se utilizará el análisis de contenido. Analizando los datos obtenidos en la observación participante, actividades que el usuario realiza tanto previa, como durante como al final, contextos de utilización, interacción con la aplicación 2houses[1], etc., se podrán obtener los posibles requisitos de nuestra aplicación.

A partir de estos requisitos se podrá realizar una propuesta de las funcionalidades de nuestra aplicación, e incluso se podría realizar una primera propuesta conceptual de la misma.

También es importante utilizar el análisis del discurso.

Durante la observación participante, se habrán obtenido datos referentes al lenguaje que los usuarios utilizan para referirse a las diferentes acciones que realizan, para relacionarse con el contexto y para interactuar con la aplicación 2houses[1]. Analizando dicho discurso, se podrá determinar el lenguaje con el que los usuarios se refieren a cada tarea, flujo de información, relaciones con los elementos tecnológicos, lo que ayudará a eliminar posibles barreras de uso en nuestra nueva aplicación y a no cometer posibles errores ya cometidos en las aplicaciones ya existentes.

1.4 Planificación del Trabajo

Dado el carácter de las aplicaciones móviles y los recursos temporales y materiales de los que se dispone, se aplicará un método de desarrollo ágil, mediante el cual el usuario final forma parte activa del proceso de diseño y desarrollo, estando completamente involucrado en el proceso. Se irán realizando entregables sucesivos a los usuarios finales, con el fin de que sean evaluados por ellos y poder detectar lo antes posible posibles errores de diseño.

Después de evaluar las funcionalidades más importantes, dado que se ve claramente que no va a ser posible desarrollarlas todas para la fecha de entrega prevista, se determina dejar para una segunda fase el desarrollo del módulo de mensajería y del módulo de notificaciones.

Con ello tendremos la tabla de hitos que se puede ver a continuación.

<i>Id.</i>	<i>Nombre de tarea</i>	<i>Comienzo</i>	<i>Fin</i>	<i>Duración</i>
1	PEC1. Propuesta y planificación	24/02/2016	07/03/2016	12d
2	Propuesta	24/02/2016	24/02/2016	1d
3	Análisis de mercado	25/02/2016	26/02/2016	2d
4	Definición usuarios	27/02/2016	28/02/2016	2d
5	Objetivos y alcance	29/02/2016	01/03/2016	2d
6	Plan de investigación usuarios	02/03/2016	03/03/2016	2d
7	Planificación	04/03/2016	06/03/2016	3d
8	Entrega PEC1	07/03/2016	07/03/2016	0d
9	PEC2. Diseño interfaz y experiencia usuario	08/03/2016	29/03/2016	22d
10	Investigación con usuarios	08/03/2016	12/03/2016	5d
11	Análisis investigación	13/03/2016	14/03/2016	2d
12	Diseño conceptual	15/03/2016	18/03/2016	4d
13	Diseño prototipos	19/03/2016	20/03/2016	2d
14	Prototipado	21/03/2016	24/03/2016	4d
15	Inspección y test con usuarios	25/03/2016	29/03/2016	5d
16	Entrega PEC2	30/03/2016	30/03/2016	0d
17	PEC3. Desarrollo	31/03/2016	24/04/2016	25d
18	Reajustes prototipo	31/03/2016	01/04/2016	2d
19	Diseño base de Datos	31/03/2016	02/04/2016	3d
20	Arquitectura aplicación	03/04/2016	04/04/2016	2d
21	Diseño interfaces	05/04/2016	06/04/2016	2d
22	Desarrollo módulo acceso usuarios	07/04/2016	11/04/2016	5d
23	Entrega a usuarios módulo acceso	12/04/2016	12/04/2016	0d
24	Test usuario módulo acceso	12/04/2016	13/04/2016	2d
25	Desarrollo módulo registro nuevo usuario	12/04/2016	18/04/2016	7d
26	Entrega usuario módulo registro	19/04/2016	19/04/2016	0d
27	Test usuario módulo registro	19/04/2016	20/04/2016	2d
28	Ajustes test usuarios	21/04/2016	24/04/2016	4d
29	Entrega PEC3	25/04/2016	25/04/2016	0d
30	PEC4. Desarrollo final	26/04/2016	12/06/2016	48d
31	Desarrollo módulo configuración	26/04/2016	07/05/2016	12d
32	Entrega usuario módulo configuración	08/05/2016	08/05/2016	0d
33	Test usuario módulo configuración	08/05/2016	09/05/2016	2d
34	Desarrollo gestión de gastos	08/05/2016	19/05/2016	12d
35	Entrega usuario gestión de gastos	20/05/2016	20/05/2016	0d
36	Test usuario gestión de gastos	20/05/2016	21/05/2016	2d
37	Desarrollo eventos y tareas calendario	20/05/2016	31/05/2016	12d
38	Entrega usuario eventos y tareas	01/06/2016	01/06/2016	0d
39	Test usuario eventos y tareas	01/06/2016	02/06/2016	2d
40	Ajustes finales de desarrollo	03/06/2016	10/06/2016	8d
41	Entrega final usuario	10/06/2016	10/06/2016	0d
42	Validación final usuario	10/06/2016	12/06/2016	3d
43	Entrega PEC4	13/06/2016	13/06/2016	0d
44	PEC5. Documentación	14/06/2016	19/06/2016	6d
45	Entrega PEC5	20/06/2016	20/06/2016	0d
46	FASE II	21/06/2016	17/07/2016	27d
47	Desarrollo módulo de mensajería	21/06/2016	28/06/2016	8d
48	Entrega usuario mensajería	29/06/2016	29/06/2016	0d
49	Test usuario mensajería	29/06/2016	30/06/2016	2d
50	Desarrollo módulo de notificaciones	29/06/2016	06/07/2016	8d
51	Entrega usuario notificaciones	07/07/2016	07/07/2016	0d
52	Test usuario notificaciones	07/07/2016	08/07/2016	2d
53	Ajustes finales de desarrollo	09/07/2016	16/07/2016	8d
54	Entrega Final	17/07/2016	17/07/2016	0d

2. Desarrollo del proyecto

2.1 Investigación con usuarios

2.1.1 Trabajo de campo

Para la recogida de datos cuantitativa se han utilizado encuestas.

Se ha diseñado la encuesta que se puede ver a continuación para determinar la predisposición de las personas a utilizar nuestra aplicación y a participar en los estudios cualitativos posteriores, así como para determinar si ya están utilizando la aplicación 2houses[1].

Para realizar dichas encuestas, se han visitado diferentes foros de padres con hijos en custodia compartida, contactando con ellos y enviándoles la encuesta por correo electrónico.

Para la recogida de datos cualitativa se tenía previsto realizar un Diario de uso, un cuestionario acerca de la aplicación 2Houses[1] y finalmente un Focus Group para debatir las cuestiones observadas durante el diario de uso.

Pero dado el poco tiempo disponible, se han realizado entrevistas guiadas a dos usuarios seleccionados mediante la encuesta de captación de usuarios anterior, los cuales se escogerán entre los que utilicen de forma habitual la aplicación 2Houses[1].

Guion de las entrevistas guiadas.

- Qué funciones de la aplicación son las que valora de más utilidad.
 - Calendario.
 - Planificación de custodia.
 - Calendario de visitas.
 - Peticiones de cambio.
 - Gestión de gastos.
 - Diario.
 - Álbumes.
 - Banco de información.
 - General.
 - Cuentas bancarias.
 - Información médica.
 - Vacunas.
 - Seguros médicos.
 - Documentación.
 - Mensajería.
- Dificultad del proceso de registro.
- Utilidad de la función de planificación de calendario de custodia y visitas.
- Que les parece la gestión de gastos.
- Que funcionalidades cambiarían de la aplicación.

2.1.2 Análisis de la información obtenida

Después de realizar las entrevistas guiadas, obtenemos información muy valiosa. Tras analizar dicha información, vemos claramente que las funcionalidades más importantes para los usuarios en el siguiente orden son:

- Calendario.
Es muy importante la planificación del calendario de custodia, así como la planificación de visitas. Además, como buen calendario es muy útil tener centralizado en el mismo toda gestión de eventos.
- Gestión de gastos.
Se valora muchísimo poder llevar al día la gestión de gastos compartidos. También valora positivamente el hecho de poder enlazar automáticamente con los extractos bancarios on-line, para semi-automatizar dicha gestión.
- Notificaciones integradas en la aplicación.
Es lo que echan muy en falta en la aplicación 2Houses[1], siendo muy útil que la aplicación muestre al usuario notificaciones de calendario, así como de acciones realizadas por el cónyuge en tiempo real.

Es muy valorado por los usuarios que el proceso de registro sea fácil y pueda integrarse con sus cuentas en diferentes redes sociales y que se pueda invitar a registrarse en la aplicación al cónyuge fácilmente. Los usuarios también valoran positivamente otras utilidades, como el banco de información, aunque lo pondrían dentro de la configuración de cada perfil de hijo, no como opción independiente.

Sobre lo que echan en falta en la aplicación, se obtiene que sobre todo es el tema de las notificaciones integradas, ya que es muy poco operativo recibir las notificaciones solo por email y no integradas en la aplicación.

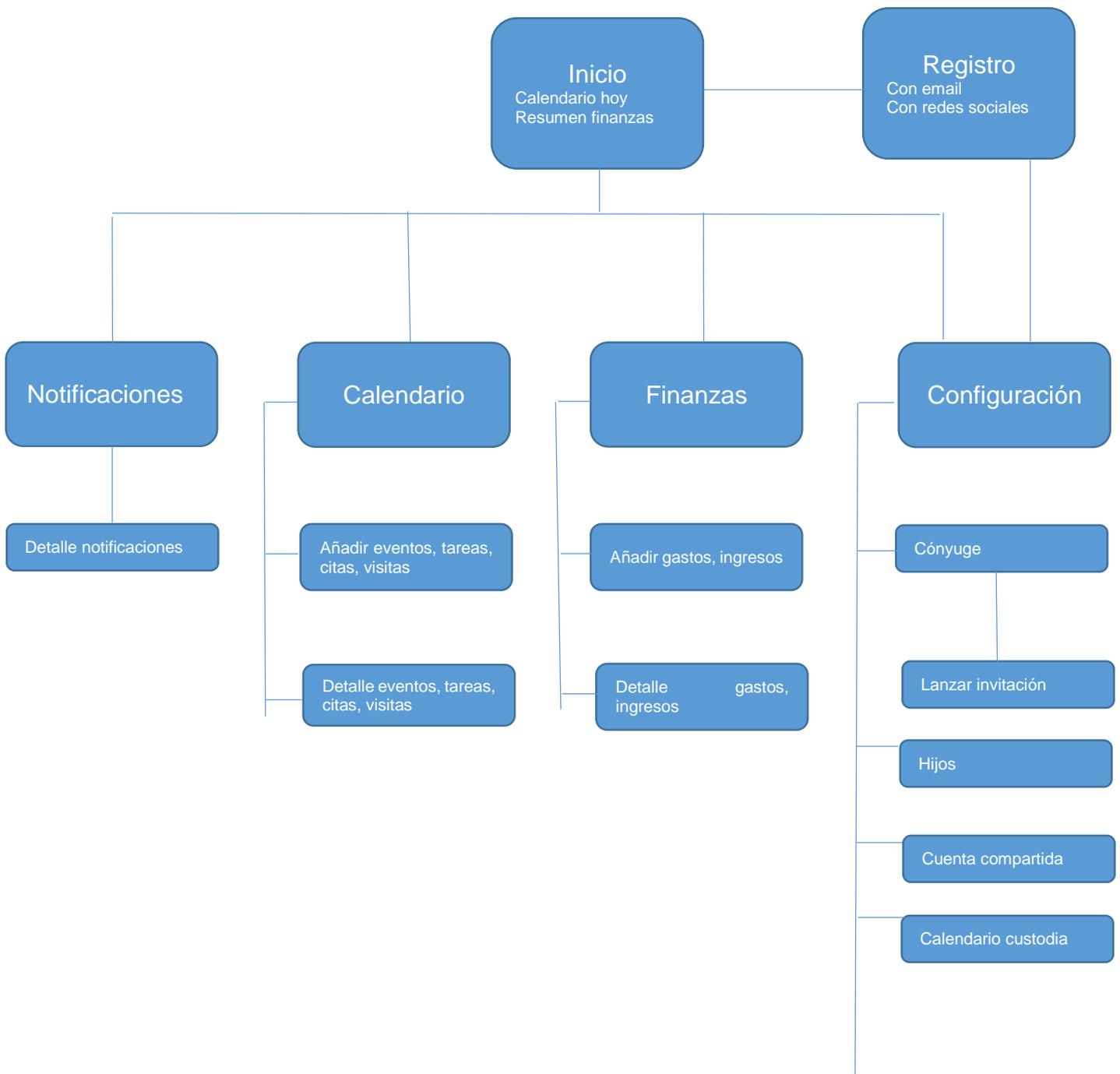
Finalmente se puede afirmar que los usuarios valoran bastante negativamente la funcionalidad de mensajería, ya que no es en tiempo real y no gestiona notificaciones. Ven mucho más útil utilizar una aplicación de mensajería externa, tipo WhatsApp[2].

2.1.3 Requisitos de usuario

- Registro con redes sociales
- Invitación cónyuge
- Configuración hijos compartidos
 - o Datos básicos
 - o Cuenta compartida
 - o Datos adicionales
- Calendario
 - o Custodia compartida
 - o Visitas
 - o Tareas
 - o Eventos
- Gestión de gastos
 - o Ingresos propios y del cónyuge
 - o Gastos propios y del cónyuge
 - o Saldo propio, del cónyuge y global
- Notificaciones integradas.

2.2 Diseño de la interacción

2.2.1. Arquitectura de la información

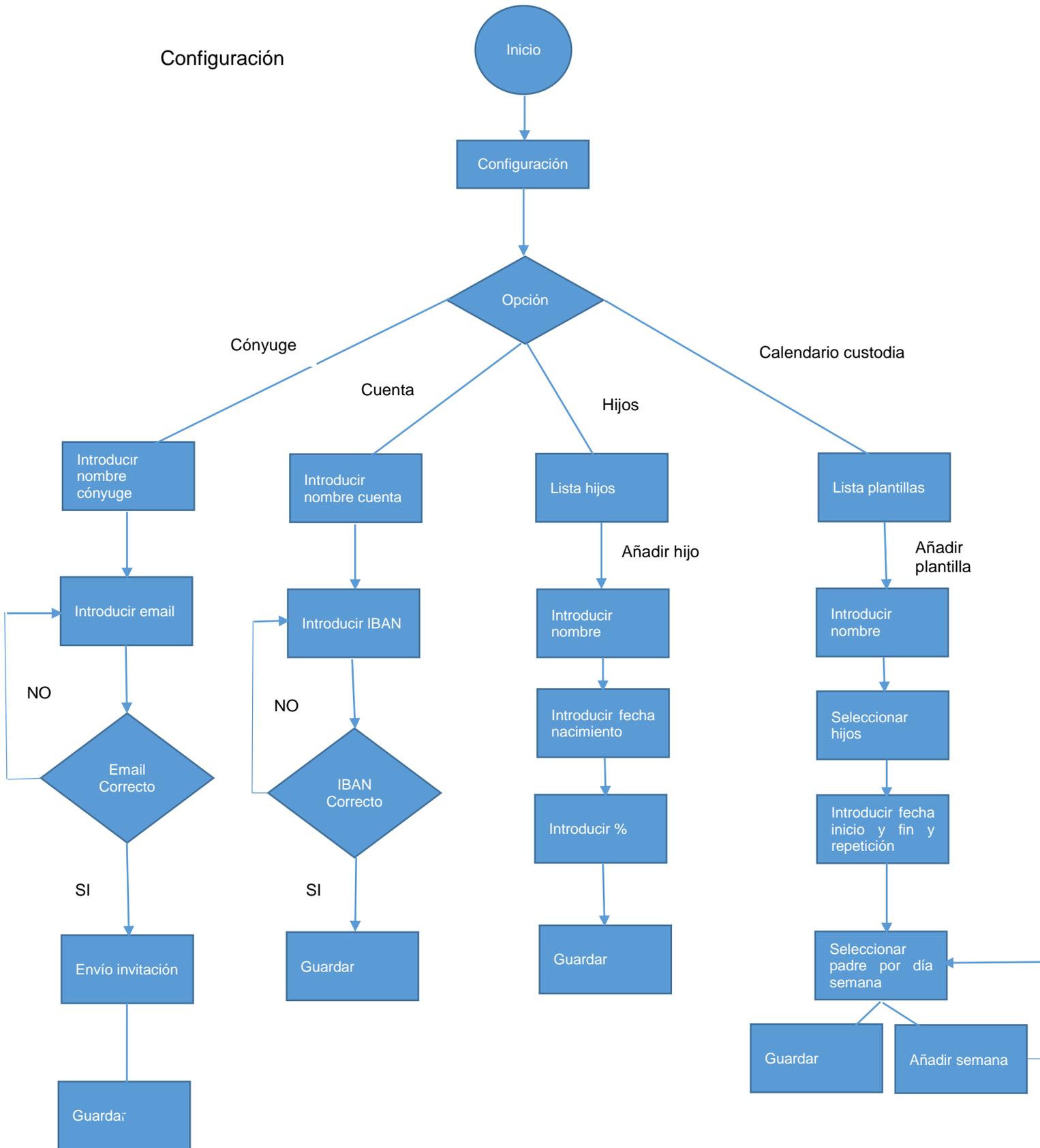


2.2.2. Funcionalidades

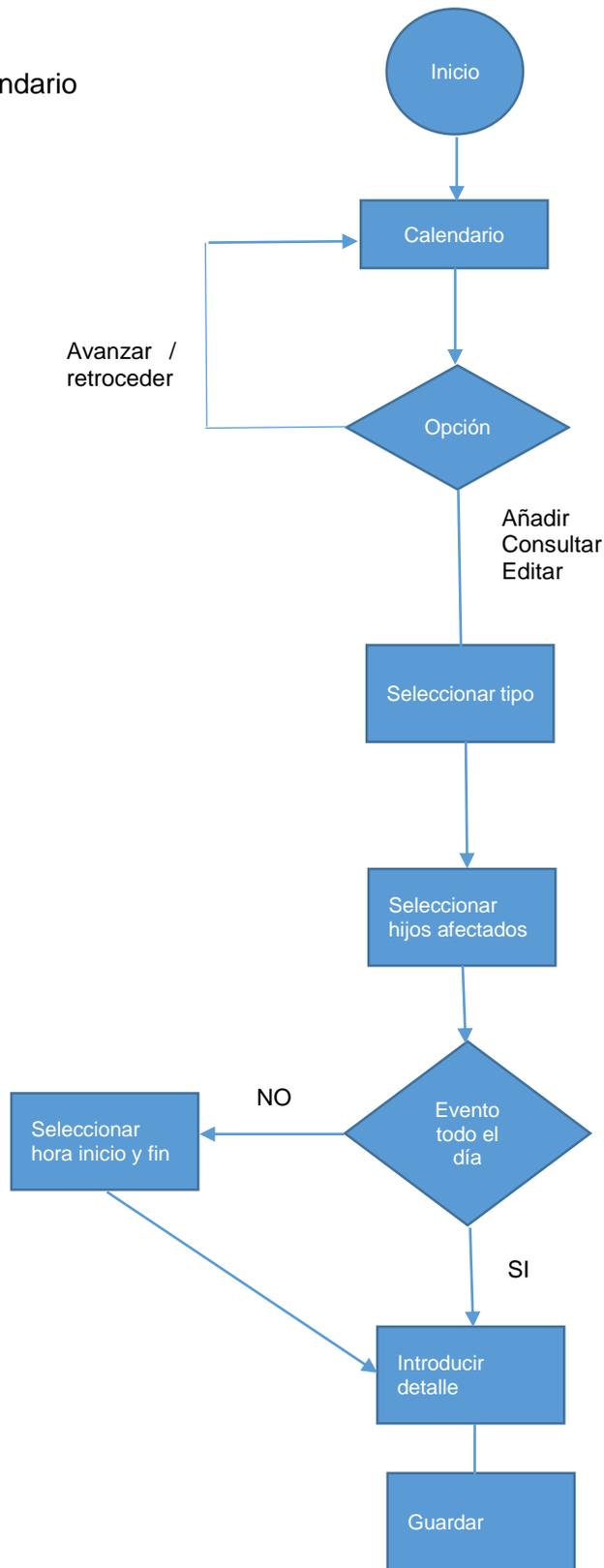
- Registro.
Nada más instalar la aplicación, el primer proceso será el de registro de usuario el cual podrá ser a partir de una cuenta de correo o mediante la integración con cuentas en diferentes redes sociales.
Una vez registrado, el usuario podrá completar el resto de la configuración o hacerlo más adelante.
- Configuración.
 - Cónyuge.
Se deberán indicar los datos básicos del cónyuge y lanzar la invitación para que se descargue e instale la App.
 - Cuenta compartida.
Se podrá configurar una única cuenta compartida para la gestión financiera.
 - Hijos compartidos.
Se indicarán los datos básicos de los hijos compartidos, así como el porcentaje de compartición de cada uno.
De forma opcional se podrán indicar datos adicionales por hijo, como datos médicos, tallas de ropa, etc.
 - Calendario de custodia.
Se definirá el calendario de custodia compartida para cada hijo.
- Pantalla inicial
En la pantalla inicial se mostrará un 'DashBoard' con un resumen del calendario de hoy y con un resumen de la información financiera.
Pulsando en cada uno de ellos se podrá acceder a su correspondiente detalle.
- Detalle Calendario.
Se mostrará el calendario del día actual, mostrando en el mismo la custodia compartida, así como todos los eventos, tareas, citas y visitas.
Pulsando sobre cada elemento del calendario se podrá acceder a su detalle.
Se podrán añadir nuevos eventos, tareas, citas y visitas al calendario.
- Finanzas.
Se mostrará el detalle de los movimientos de la cuenta compartida de más moderno a más antiguo, mostrando el saldo de la misma.
Se podrá optar por visión conjunta, visión propia o visión del cónyuge.
Pulsando sobre cada movimiento se podrá acceder al detalle del mismo.
Se podrán añadir nuevos movimientos, tanto de ingreso como de gasto.
- Notificaciones integradas.
Cada vez que el cónyuge realice alguna acción que afecte al usuario, como creación de eventos en calendario y movimientos financieros, el usuario recibirá una notificación en el dispositivo, pulsando sobre la cual accederá al detalle de lo que la ha provocado.
Dichas notificaciones se gestionarán integradas en la gestión de notificaciones del dispositivo.

2.2.3. Flujos de usuario

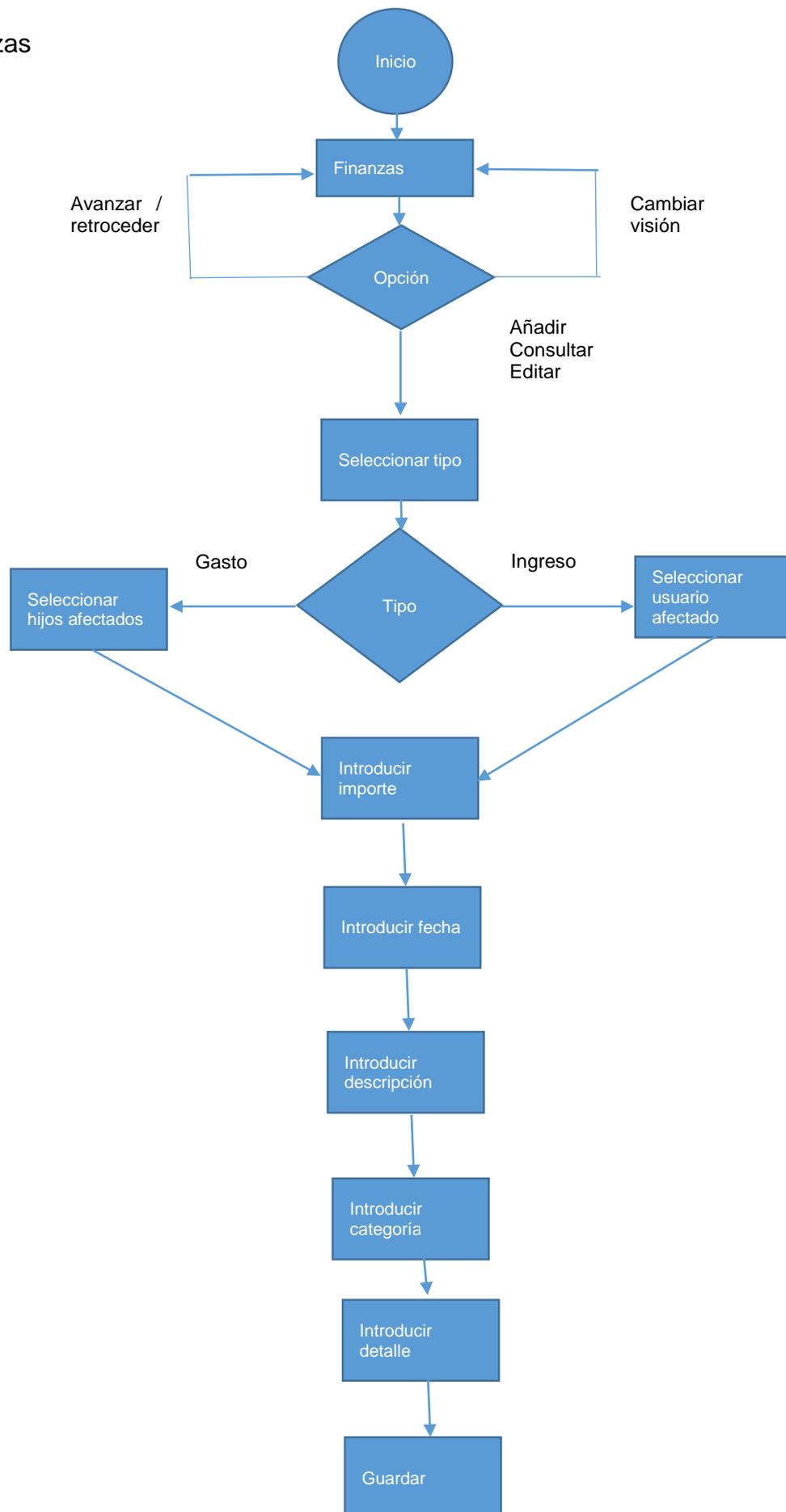
Configuración



Calendario



Finanzas

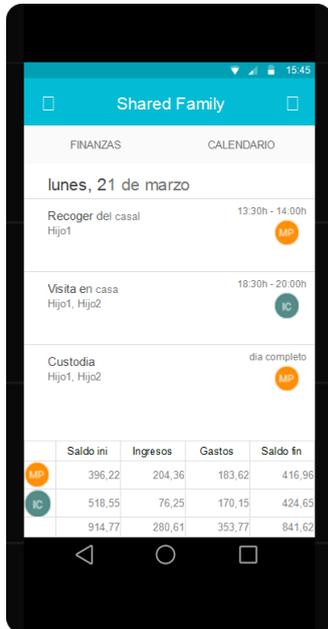


2.2.3. WireFrames

Debido a lo ajustado del tiempo disponible para la ejecución del proyecto, se decide no realizar los diseños en baja definición para pasar a realizar directamente el prototipo en alta definición.

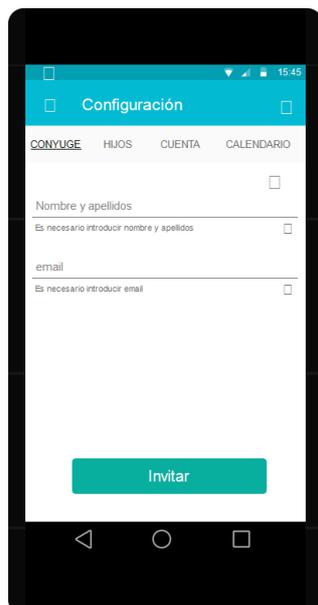
2.2.3. Prototipado

El prototipo en alta fidelidad se ha realizado mediante la herramienta just in mind^[3], teniendo el prototipo en la sección de anexos.

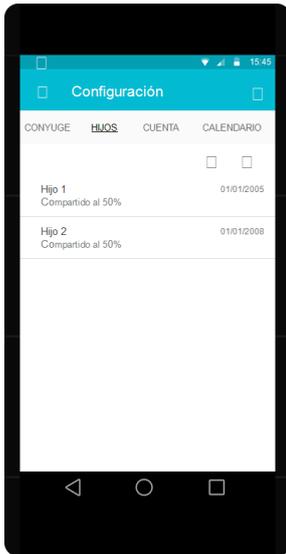


En la pantalla inicial se podrá ver el 'dashboard' con los últimos eventos del calendario y con el resumen de datos financieros.

Desde aquí se podrá acceder a la configuración, detalle de calendario y detalle de datos financieros.



En la pantalla de configuración se tendrá acceso mediante navegación por pestañas a la configuración del cónyuge, donde se podrán sus datos y podremos lanzarle la invitación para utilizar la aplicación.



En la sección de configuración de hijos, se podrá ver la lista de hijos configurados, editar ya añadir nuevos hijos.

Por cada hijo se deberán indicar nombre y apellidos, fecha nacimiento y porcentaje de compartición con el cónyuge.

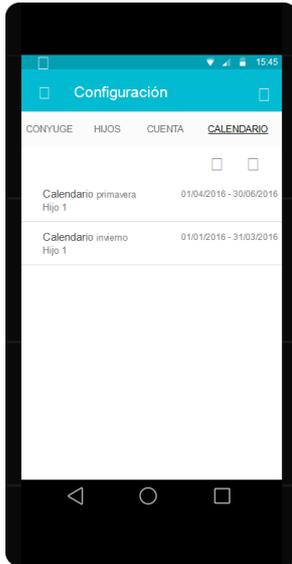
En versiones más avanzadas de la aplicación se planteará la posibilidad de añadir otros datos de detalle, como datos médicos, tallaje de ropa, etc.



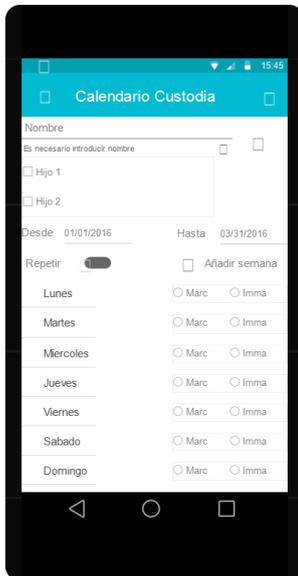
En la configuración de cuenta se introducirán los datos de la cuenta compartida utilizada en la sección de finanzas.

Se indicará el nombre de la cuenta y su IBAN.

En versiones más avanzadas de la aplicación se puede plantear la posibilidad de gestionar más de una cuenta, e incluso poder gestionar movimientos automáticos conectando con la información de los mismos proporcionada por las diferentes entidades financieras.



En la configuración de calendario de custodia, se podrán ver todos los patrones de calendario existentes, así como crear nuevos. Se pueden definir tantos patrones como se desee, teniendo siempre en cuenta que si unos patrones solapan a otros por fechas, prevalecerán los días del último patrón definido.



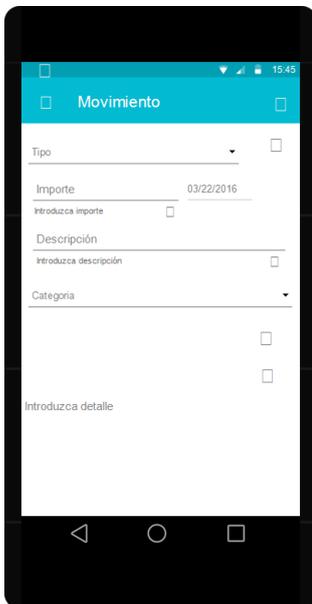
Cada patrón de calendario se define con un nombre, los hijos a los que afecta y un periodo de validez. Dentro de cada patrón, por defecto se configura una semana, indicando para cada día de la misma, que padre es el que tiene la custodia. Se pueden ir añadiendo más semanas al patrón, hasta definir el patrón completamente. Si se opta por repetir las semanas, se irá repitiendo el patrón de semanas hasta completar el periodo de validez del patrón.



En la sección de datos financieros, se podrán ver la lista de movimientos en la cuenta configurada, pudiendo hacerlo en visión conjunta de ambos cónyuges o visión independiente de cada uno de ellos.

Para cada movimiento se podrá ver la fecha, descripción, categoría e importe.

Se podrá consultar / editar cada movimiento y añadir nuevos.



En la consulta / edición / añadir movimientos se indicará si el movimiento es de ingreso o de gasto. Si es de gasto se deberán indicar los hijos afectados, mientras que es de ingreso se deberá indicar el cónyuge que lo realiza.

Se indicará el importe y la descripción.

También se indicará la categoría, pudiendo añadir nuevas categorías, así como un detalle descriptivo del mismo.

Además de podrán adjuntar archivos al movimiento, así como imágenes.

Cada cónyuge solo podrá editar sus movimientos y solo a futuro.



El calendario tendrá una barra lateral a la izquierda con acceso rápido a los días del mes y una lista de detalle de eventos a la derecha, agrupada por días, indicándose para cada día los eventos del mismo.

Para cada evento se podrá ver la descripción, la duración, los hijos afectados y el cónyuge al que se refiere.

Desde aquí se podrá consultar / editar / añadir eventos.



En la consulta / edición / añadir eventos se indicará si el evento es una visita, recordatorio, nota, evento, etc.

Se indicarán los hijos afectados, la duración del mismo y el detalle textual.

Cada cónyuge solo podrá editar sus movimientos y solo a futuro.

2.3 Evaluación con usuarios

2.3.1 Evaluación heurística

Debido a lo ajustado del tiempo disponible para la ejecución del proyecto, se decide no realizar la evaluación heurística del prototipo, con lo que se pasará directamente a realizar la evaluación con usuarios.

2.3.2 Muestra de usuarios

Para la evaluación con usuarios se utiliza a los mismos usuarios con los que se realizaron las entrevistas guiadas.

- Persona 1
Hombre de unos 45 años, separada legalmente con 5 hijos en custodia compartida. Está conviviendo con una nueva pareja con la que no tiene ningún otro hijo. La separación de su ex pareja se produjo de mutuo acuerdo y mantienen una buena relación.
Posee teléfono inteligente desde hace mucho tiempo, con el que utiliza muchas otras aplicaciones de todo tipo.
- Persona 2
Mujer de unos 35 años, separada legalmente con un hijo en custodia compartida. Está conviviendo con una nueva pareja con la que tiene otro hijo.
La separación de su ex pareja se produjo sin mutuo acuerdo, con lo que existió una sentencia judicial de por medio y no se mantiene una buena relación.
Usuario habitual de teléfonos inteligentes desde hace mucho tiempo, utilizándolo para todo tipo de tareas, mediante diferentes aplicaciones.

2.3.3 Evaluación con usuarios

Para la evaluación con usuarios no se definen escenarios específicos, sino que se insta a las personas a realizar las tareas definidas en cualquier escenario.

Las tareas definidas son:

- Tienes que definir y consensuar con el cónyuge el calendario de custodia de alguno de los hijos para un periodo determinado.
- Añadir en el calendario una visita a tu hijo para el próximo lunes de 15:00 a 16:00, momento en que no tienes la custodia del mismo.
- Añadir en el calendario un recordatorio para recoger mañana a tu hijo del taller de pintura a las 16:00h.
- Necesitas saber quién tiene la custodia de uno de tus hijos la semana que viene.
- Introducir un gasto de 40€ por haber realizado la compra de unas zapatillas de deporte para uno de tus hijos.
- Has realizado un ingreso de 100€ en la cuenta compartida.
- Necesitas conocer tu saldo en la cuenta compartida.

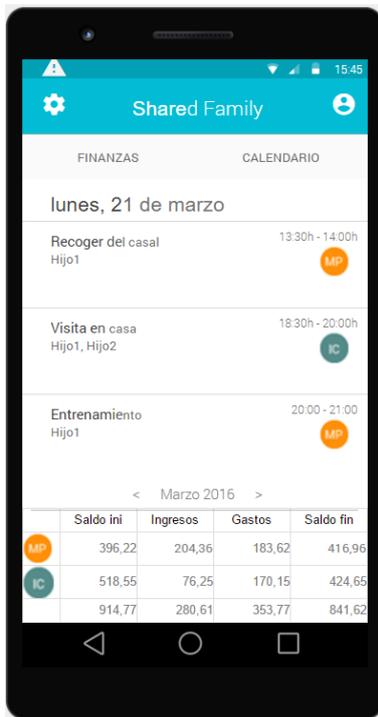
2.3.3 Resultados evaluación con usuarios

En general, lo primero que se detecta claramente es que existe una diferencia importante entre parejas separadas de mutuo acuerdo (a buenas) de las separadas sin acuerdo (a malas). Las separadas de mutuo acuerdo consideran que las validaciones por parte del otro de las acciones del uno no son necesarias e incluso molestan. Por el contrario, las parejas separadas sin acuerdo, necesitan irremediamente estas validaciones y constancia de las mismas. Ya en detalle, se obtiene.

- En la pantalla de inicio.
En el dashboard del calendario, no aporta nada mostrar quien tiene la custodia, sino que es mejor dar prioridad a mostrar otros eventos, ya que cada uno ya sabe quién tiene la custodia en el día actual y próximos. También es indicado por los usuarios que el término custodia es inadecuado, siendo mejor utilizar algo así como 'pernoctación'.
En el dashboard de finanzas falta indicar al mes que corresponde, se intuye que es el actual, pero no está de más indicarlo.
- En la configuración de Calendario de custodia.
Como se indicaba anteriormente, se debería cambiar el concepto de 'custodia' por 'pernoctación'.
Es necesario establecer un sistema de validación y consenso de los calendarios de pernoctación por parte de ambos cónyuges, quedando además registro de ello, para que pueda ser aportado como prueba en caso de conflicto.
- En el calendario.
Se considera que no es necesario indicar a quien corresponde la pernoctación como una cita más del calendario, sino que bastaría con indicarlo con un icono.
La lista de acceso rápido de días de la izquierda se debería cambiar por semanas o meses, ya que ya se dispone del detalle por días a la derecha. Se considera importante que quede bien claro quién podrá ver los eventos del calendario. No queda claro y el usuario cree que cualquier evento definido es visible por el otro conyugue, cuando solo lo son las visitas. También se cree necesario que los hijos también puedan compartir eventos del calendario. Para hijos de ciertas edades, en las que ya pueden ser usuarios de Smartphone, puede ser muy interesante que ellos también dispongan de la aplicación para tener acceso al calendario compartido con sus padres.
- En finanzas.
Se cree conveniente sustituir el concepto de 'ingresos' por 'aportaciones'. Los usuarios manifiestan que se deberían añadir colores para distinguir las aportaciones de los gastos más fácilmente. También se indica que debería ser más visible el hecho de que un movimiento no esté categorizado. Se indica también que la selección de vista (global/conyuge1/conyuge2) sería más útil situada en la parte superior, como cabecera de la lista de movimientos.
Sería de utilidad poder navegar entre meses directamente, en lugar de tener solo el desplazamiento vertical.
Se indica que sería interesante poder tener la posibilidad de gestionar movimientos 'favoritos' ya que existen movimientos que son mensuales y serían más fáciles de gestionar.

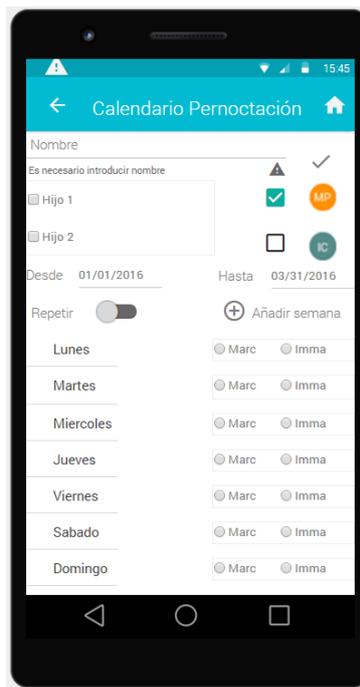
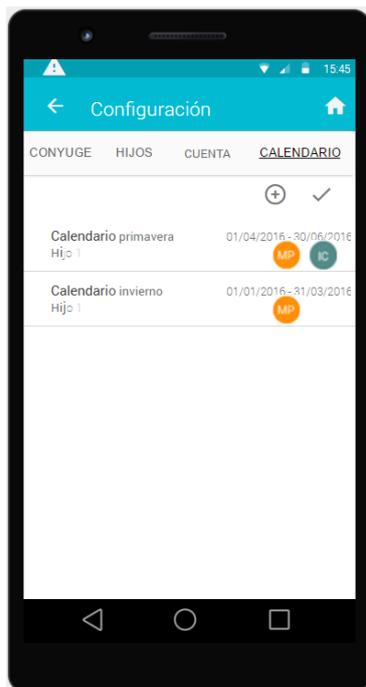
2.3.4 Ajustes prototipo en base a la evaluación con usuarios.

Pantalla de Inicio.



En el dashboard del calendario, solo se muestran los eventos, citas, etc., pero no las pernoctaciones. Se añade en el dashboard de finanzas el mes que corresponde y la navegación entre meses.

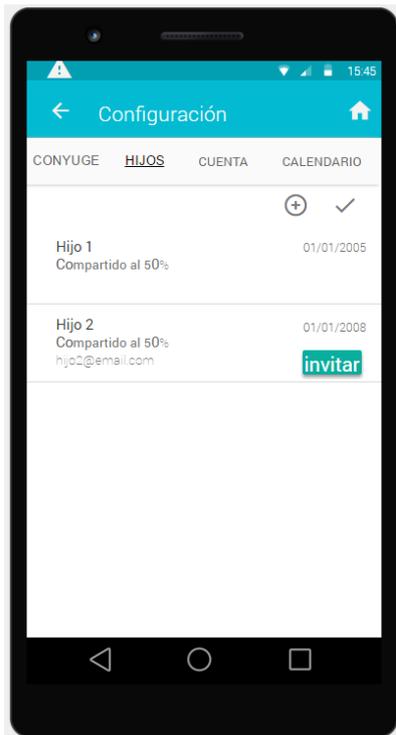
Configuración calendario pernoctación.



Se añade sistema de validación de calendarios por parte de los cónyuges.

En la configuración de calendario, se indica mediante los iconos de cada cónyuge, si están validados por los mismos. En el detalle de calendario se añade la posibilidad de la validación del mismo por parte de cada cónyuge. Pantalla de Inicio.

Configuración hijos.



Para que los hijos que ya posean Smartphone, puedan utilizar la aplicación para compartir eventos con sus padres, se añade la posibilidad de configurar el correo electrónico de los hijos y lanzar invitación de uso de la aplicación.

Para los usuarios de los hijos, solo estará disponible el calendario de eventos.

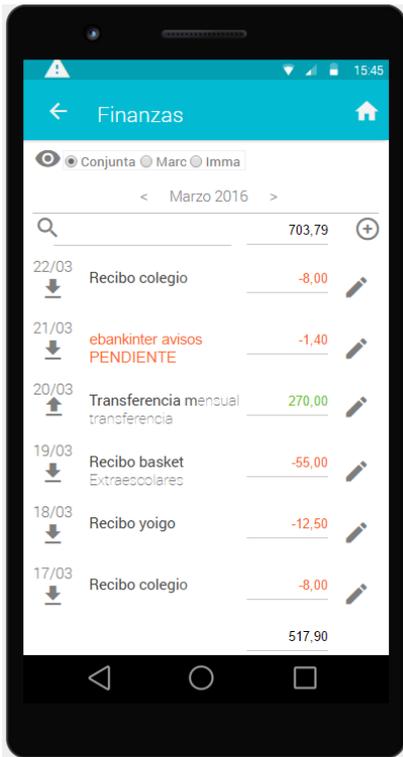
Calendario.



Se sustituye la forma de indicar la pernoctación, pasando de formato cita en el calendario a un icono correspondiente al cónyuge en la barra de título de cada día.

Se sustituye la barra de navegación vertical izquierda de días por navegación por semanas. De este modo, tendremos navegación diaria haciendo desplazamiento vertical sobre los mismos días del calendario, navegación semanal haciendo desplazamiento vertical sobre las semanas de la barra de la izquierda y navegación mensual mediante los iconos de navegación de la barra superior donde se indica el mes.

Finanzas.

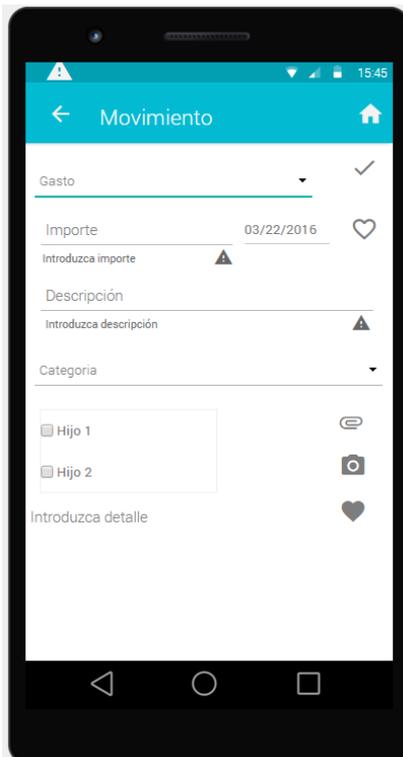


Se pasa a la parte superior y en horizontal la selección de vista conjunta o cada cónyuge.

Se añade en la parte superior un control de navegación mensual, indicando el mes seleccionado, para facilitar la búsqueda de movimientos.

Se identifican en rojo los movimientos no categorizados.

Para mejorar la identificación de gastos o aportaciones, se indican los importes de los gastos en rojo y de las aportaciones en verde.

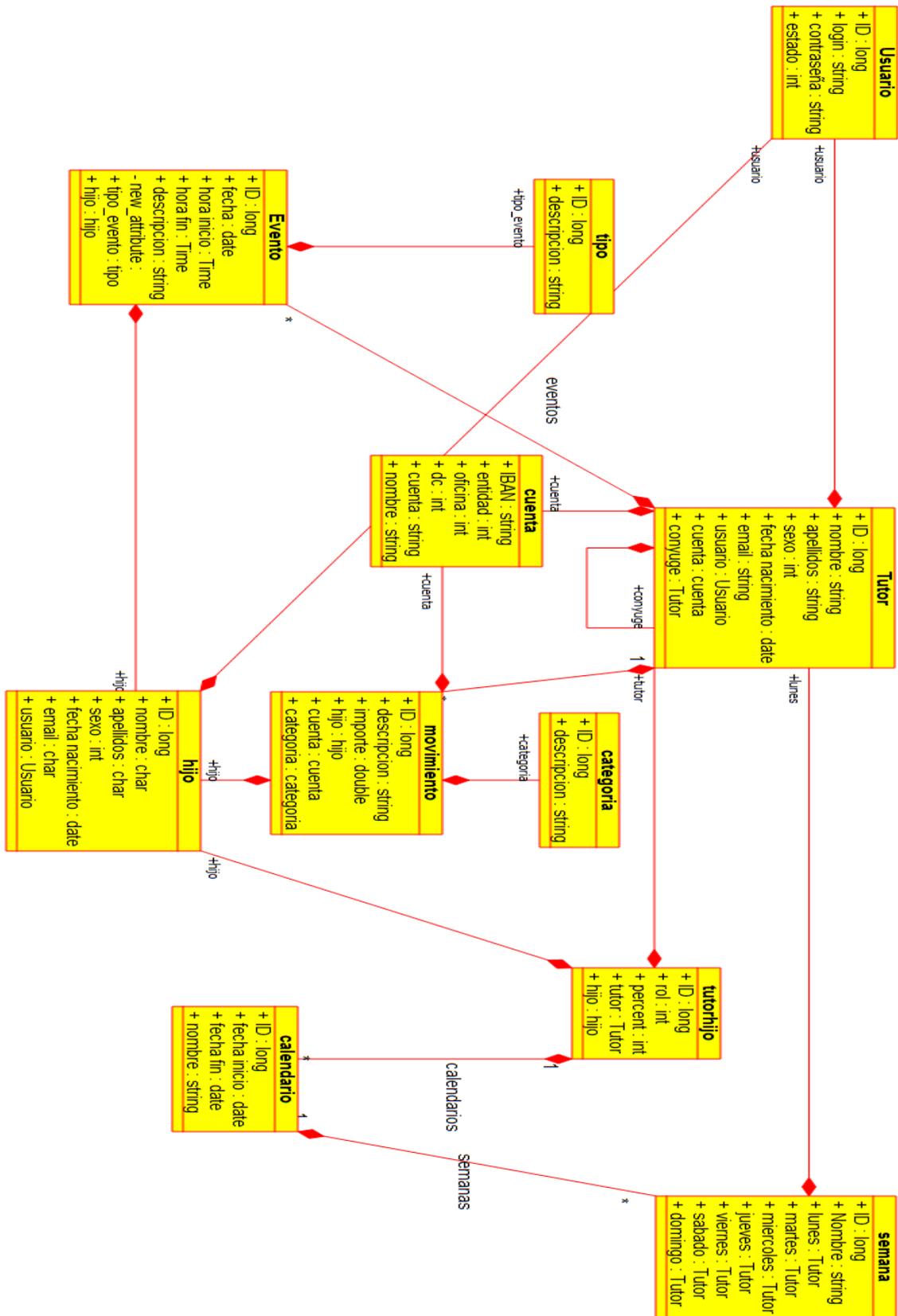


En el detalle de movimientos se añade en la parte superior la posibilidad de recuperar movimientos favoritos.

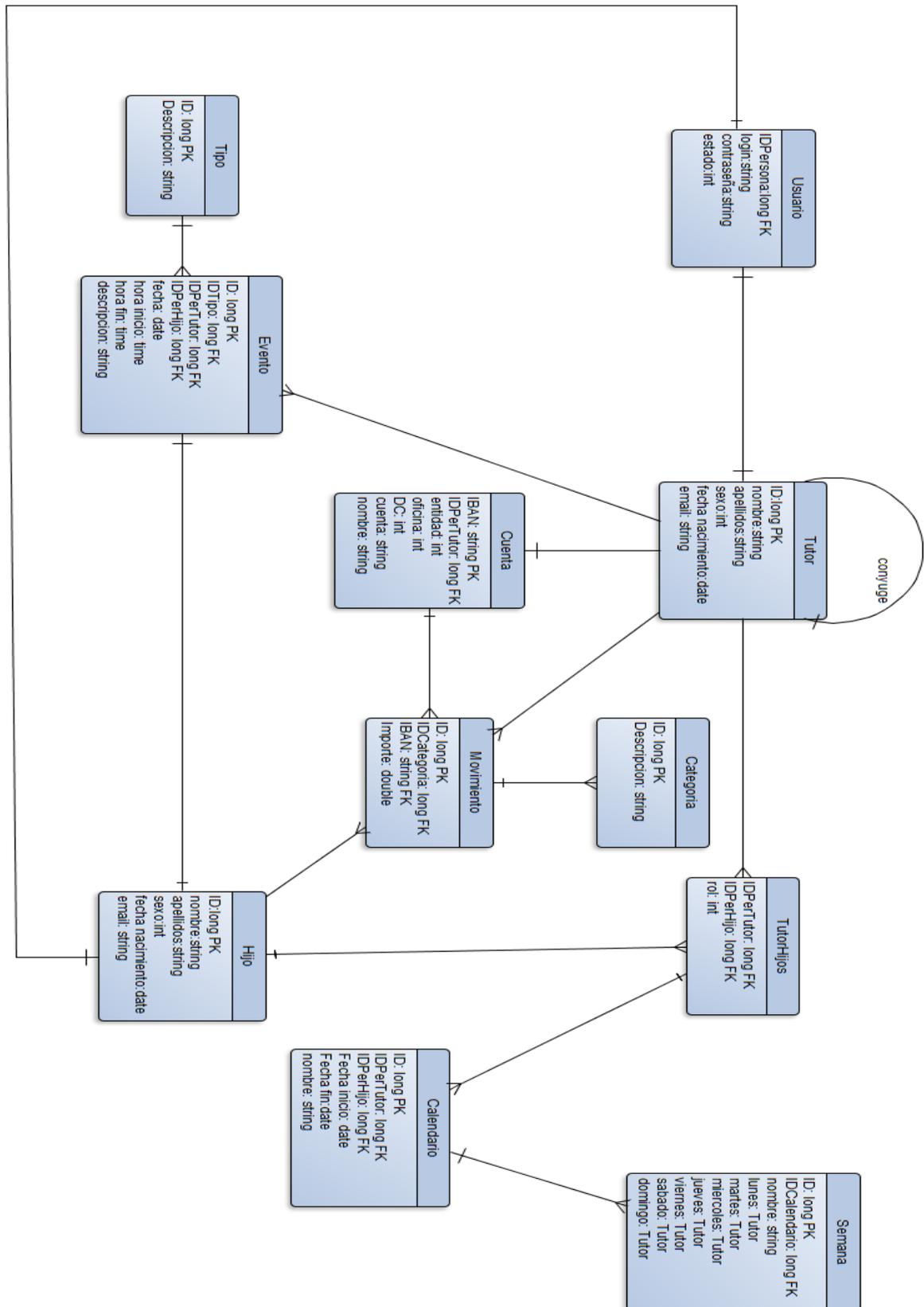
En la parte inferior se añade la posibilidad de guardar el movimiento como favorito.

2.4 Desarrollo de la aplicación.

2.4.1 Diagrama de clases.



2.4.2 Diagrama de Base de Datos.



2.4.3 Implementación.

2.4.3.1 BBDD.

La base de datos se implementa mediante Backendless[4], desarrollando las siguientes tablas

Users

Nombre de la columna	Tipo de dato	Valor predeterm.	Identidad	Req.	Validator
password	STRING		<input type="radio"/>	<input checked="" type="checkbox"/>	custom <input type="text"/>
email	STRING		<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	Email address

Parent

Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
Children	Data Relation (1:N) ParentChild		Select a constraint	
Spouse	Data Relation (1:1) Parent		Select a constraint	
Events	Data Relation (1:N) Event		Select a constraint	
User	Data Relation (1:1) Users		Select a constraint	
Account	Data Relation (1:1) Account		Select a constraint	
surname	STRING		Select a constraint	custom <input type="text"/>
name	STRING		Select a constraint	custom <input type="text"/>
gender	INT		Select a constraint	custom <input type="text"/>
email	STRING		UQ IDX	Email address
borndate	DATETIME		Select a constraint	custom <input type="text"/>

ParentChild

Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
Calendars	Data Relation (1:N) Calendar		Select a constraint	
Parent	Data Relation (1:1) Parent		Select a constraint	
Child	Data Relation (1:1) Child		Select a constraint	
rol	INT		Select a constraint	custom <input type="text"/>
percent	INT		Select a constraint	custom <input type="text"/>

Child

Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
User	Data Relation (1:1) Users		Select a constraint	
surname	STRING		Select a constraint	custom <input type="text"/>
name	STRING		Select a constraint	custom <input type="text"/>
gender	INT		Select a constraint	custom <input type="text"/>
email	STRING		Select a constraint	custom <input type="text"/>
borndate	DATETIME		Select a constraint	custom <input type="text"/>

Event

Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
Type	Data Relation (1:1) Type		Select a constraint	
Child	Data Relation (1:1) Child		Select a constraint	
startdate	DATETIME		Select a constraint	Date/Time
enddate	DATETIME		Select a constraint	Date/Time
description	STRING		Select a constraint	custom <input type="text"/>
allday	BOOLEAN	Null	Select a constraint	disabled

Type

Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
description	STRING		Select a constraint	custom <input type="text"/>
color	STRING		Select a constraint	custom <input type="text"/>

Account

<input type="checkbox"/> Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
<input type="checkbox"/> Movements	Data Relation (1:N): movement		Select a constraint	
<input type="checkbox"/> office	STRING		NN X	Number 1-1000000
<input type="checkbox"/> name	STRING		NN X	custom
<input type="checkbox"/> iban	STRING		UQ X IDX X	custom
<input type="checkbox"/> cd	STRING		NN X	Number 1-1000
<input type="checkbox"/> bank	STRING		NN X	Number 1-1000000
<input type="checkbox"/> account	STRING		NN X	custom

Movement

<input type="checkbox"/> Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
<input type="checkbox"/> Category	Data Relation (1:1): Category		Select a constraint	
<input type="checkbox"/> Parent	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> Child	Data Relation (1:1): Child		Select a constraint	
<input type="checkbox"/> movementdate	DATETIME		Select a constraint	custom
<input type="checkbox"/> description	STRING		Select a constraint	custom
<input type="checkbox"/> amount	DOUBLE		Select a constraint	custom

Category

<input type="checkbox"/> Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
<input type="checkbox"/> description	STRING		Select a constraint	custom

Calendar

<input type="checkbox"/> Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
<input type="checkbox"/> Weeks	Data Relation (1:N): week		Select a constraint	
<input type="checkbox"/> name	STRING		Select a constraint	custom
<input type="checkbox"/> indate	DATETIME		NN X	Date/Time
<input type="checkbox"/> enddate	DATETIME		NN X	Date/Time

Week

<input type="checkbox"/> Nombre de la columna	Tipo de dato	Valor predeterm.	Constraint	Validator
<input type="checkbox"/> thursday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> saturday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> sunday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> wednesday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> friday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> tuesday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> monday	Data Relation (1:1): Parent		Select a constraint	
<input type="checkbox"/> name	STRING		Select a constraint	custom

2.4.3.2 Entorno desarrollo.

La aplicación se implementa para dispositivos Android [\[5\]](#), utilizando el entorno de desarrollo nativo de Android Studio [\[6\]](#) 1.5.1.

Se implementa para smartphones Android Api 23: Android 6.0 Marshmallow, pero habilitada para dispositivos desde la Api 15.

Se utilizan las API:

com.google.android.gms:play-services-ads

com.google.android.gms:play-services-analytics

com.google.android.gms:play-services-identity

com.google.android.gms:play-services-gmc

com.backendless:backendless para la persistencia de datos en la nube.

2.4.3.3 Modelo de clases.

Para implementar las entidades de BBDD se implementa el siguiente modelo

```

1  * Created by emgarcia on 28/05/2016.
2  */
3  public class Parent implements Serializable {
4      private String objectId;
5      private String name;
6      private String surname;
7      private int gender;
8      private String email;
9      private Date borndate;
10     private Account Account;
11     private BackendlessUser User;
12     private Parent Spouse;
13     private ArrayList<Event> Events;
14     private ArrayList<ParentChild> Children;
15
16     public String getObjectId() { return objectId; }
17     public void setObjectId(String objectId) { this.objectId = objectId; }
18     public String getName() { return name; }
19     public void setName(String name) { this.name = name; }
20     public String getSurname() { return surname; }
21     public void setSurname(String surname) { this.surname = surname; }
22     public int getGender() { return gender; }
23     public void setGender(int gender) { this.gender = gender; }
24     public String getEmail() { return email; }
25     public void setEmail(String email) { this.email = email; }
26     public Date getBorndate() { return borndate; }
27     public void setBorndate(Date borndate) { this.borndate = borndate; }
28     public edu.uoc.pec2.android.sharedfamily.model.Account getAccount() { return Account; }
29     public void setAccount(edu.uoc.pec2.android.sharedfamily.model.Account account) {...}
30     public BackendlessUser getUser() { return User; }
31     public void setUser(BackendlessUser user) { User = user; }
32     public Parent getSpouse() { return Spouse; }
33     public void setSpouse(Parent spouse) { Spouse = spouse; }
34     public ArrayList<ParentChild> getChildren() { return Children; }
35     public void setChildren(ArrayList<ParentChild> children) { Children = children; }
36     public ArrayList<Event> getEvents() { return Events; }
37     public void setEvents(ArrayList<Event> events) { Events = events; }
38 }
39
40 /**
41  * Created by emgarcia on 28/05/2016.
42  */
43 public class ParentChild implements Serializable {
44     private String objectId;
45     private Integer rol;
46     private Integer percent;
47     private ArrayList<Calendar> Calendars;
48     private Parent Parent;
49     private Child Child;
50
51     public String getObjectId() { return objectId; }
52     public void setObjectId(String objectId) { this.objectId = objectId; }
53     public Integer getRol() { return rol; }
54     public void setRol(Integer rol) { this.rol = rol; }
55     public Integer getPercent() { return percent; }
56     public void setPercent(Integer percent) { this.percent = percent; }
57     public edu.uoc.pec2.android.sharedfamily.model.Parent getParent() { return Parent; }
58     public void setParent(edu.uoc.pec2.android.sharedfamily.model.Parent parent) {...}
59     public edu.uoc.pec2.android.sharedfamily.model.Child getChild() { return Child; }
60     public void setChild(edu.uoc.pec2.android.sharedfamily.model.Child child) { Child = child; }
61     public ArrayList<Calendar> getCalendars() { return Calendars; }
62     public void setCalendars(ArrayList<Calendar> calendars) { Calendars = calendars; }
63 }

```

```

/**
 * Created by emgarcia on 03/06/2016.
 */
public class Child implements Serializable {
    private String objectId;
    private String name;
    private String surname;
    private int gender;
    private String email;
    private Date borndate;
    private BackendlessUser User;

    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getSurname() { return surname; }
    public void setSurname(String surname) { this.surname = surname; }
    public int getGender() { return gender; }
    public void setGender(int gender) { this.gender = gender; }
    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }
    public Date getBorndate() { return borndate; }
    public void setBorndate(Date borndate) { this.borndate = borndate; }
    public BackendlessUser getUser() { return User; }
    public void setUser(BackendlessUser user) { User = user; }
}

/**
 * Created by emgarcia on 28/05/2016.
 */
public class Event implements Serializable {
    private String objectId;
    private edu.uoc.pec2.android.sharedfamily.model.Type Type;
    private Date inidate;
    private Date enddate;
    private String description;
    private boolean allday;
    private Child Child;

    public boolean isAllday() { return allday; }
    public void setAllday(boolean allday) { this.allday = allday; }
    public edu.uoc.pec2.android.sharedfamily.model.Child getChild() { return Child; }
    public void setChild(edu.uoc.pec2.android.sharedfamily.model.Child child) { Child = child; }
    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public edu.uoc.pec2.android.sharedfamily.model.Type getType() { return Type; }
    public void setType(edu.uoc.pec2.android.sharedfamily.model.Type type) { Type = type; }
    public Date getInidate() { return inidate; }
    public void setInidate(Date inidate) { this.inidate = inidate; }
    public Date getEnddate() { return enddate; }
    public void setEnddate(Date enddate) { this.enddate = enddate; }
    public String getDescription() { return description; }
    public void setDescription(String description) { this.description = description; }
}

/**
 * Created by emgarcia on 28/05/2016.
 */
public class Type implements Serializable {
    private String objectId;
    private String description;
    private String color;

    public String getColor() { return color; }
    public void setColor(String color) { this.color = color; }
    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getDescription() { return description; }
    public void setDescription(String description) { this.description = description; }
}

```

```

/**
 * Created by emgarcia on 28/05/2016.
 */
public class Account implements Serializable {
    private String objectId;
    private String name;
    private String iban;
    private String bank;
    private String office;
    private String cd;
    private String account;
    private ArrayList<movement> Movements;

    public ArrayList<movement> getMovements() { return Movements; }
    public void setMovements(ArrayList<movement> movements) { Movements = movements; }
    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getIban() { return iban; }
    public void setIban(String iban) { this.iban = iban; }
    public String getBank() { return bank; }
    public void setBank(String bank) { this.bank = bank; }
    public String getOffice() { return office; }
    public void setOffice(String office) { this.office = office; }
    public String getCd() { return cd; }
    public void setCd(String cd) { this.cd = cd; }
    public String getAccount() { return account; }
    public void setAccount(String account) { this.account = account; }
}

/**
 * Created by emgarcia on 28/05/2016.
 */
public class movement implements Serializable {
    private String objectId;
    private String description;
    private Date movementdate;
    private Category Category;
    private Double amount;
    private Child Child;
    private Parent Parent;

    public Date getMovementdate() { return movementdate; }
    public void setMovementdate(Date movementdate) { this.movementdate = movementdate; }
    public String getDescription() { return description; }
    public void setDescription(String description) { this.description = description; }
    public edu.uoc.pec2.android.sharedfamily.model.Parent getParent() { return Parent; }
    public void setParent(edu.uoc.pec2.android.sharedfamily.model.Parent parent) {...}
    public edu.uoc.pec2.android.sharedfamily.model.Child getChild() { return Child; }
    public void setChild(edu.uoc.pec2.android.sharedfamily.model.Child child) { Child = child; }
    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public edu.uoc.pec2.android.sharedfamily.model.Category getCategory() { return Category; }
    public void setCategory(edu.uoc.pec2.android.sharedfamily.model.Category category) {...}
    public Double getAmount() { return amount; }
    public void setAmount(Double amount) { this.amount = amount; }
}

/**
 * Created by emgarcia on 28/05/2016.
 */
public class Category implements Serializable {
    private String objectId;
    private String description;

    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getDescription() { return description; }
    public void setDescription(String description) { this.description = description; }
}

```

```

/**
 * Created by emgarcia on 28/05/2016.
 */
public class Calendar implements Serializable {
    private String objectId;
    private String name;
    private Date inidate;
    private Date enddate;
    private ArrayList<week> Weeks;

    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public Date getInidate() { return inidate; }
    public void setInidate(Date inidate) { this.inidate = inidate; }
    public Date getEnddate() { return enddate; }
    public void setEnddate(Date enddate) { this.enddate = enddate; }
    public ArrayList<week> getWeeks() { return Weeks; }
    public void setWeeks(ArrayList<week> weeks) { Weeks = weeks; }
}

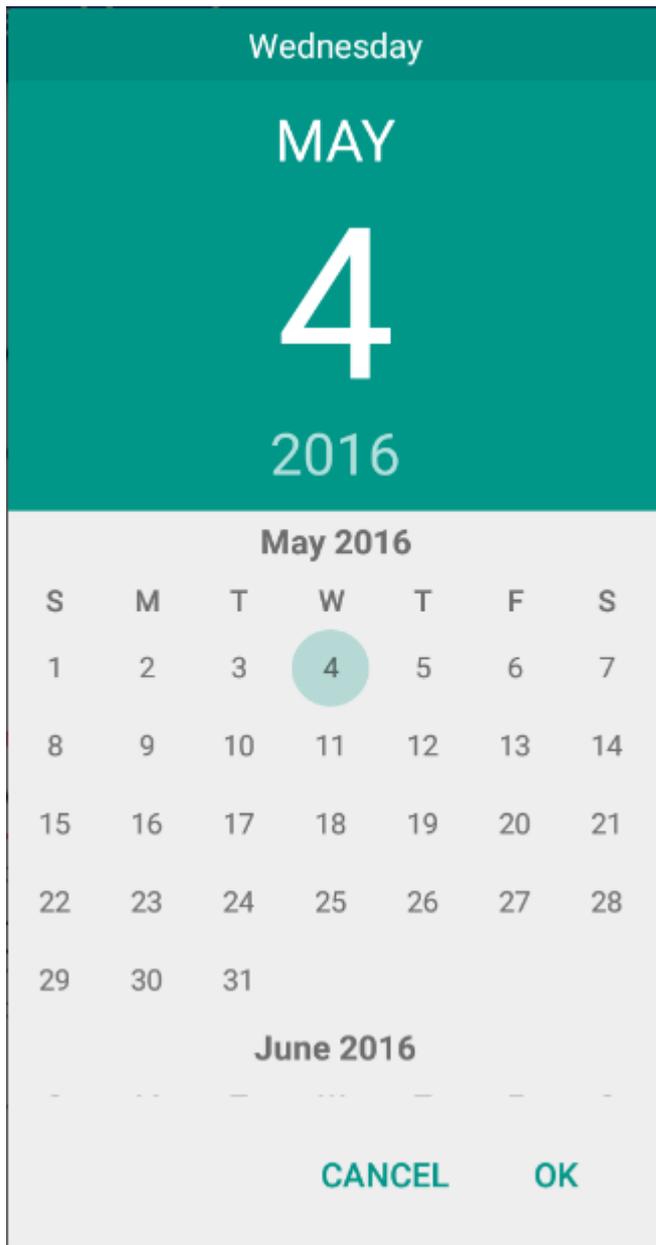
/**
 * Created by emgarcia on 28/05/2016.
 */
public class week implements Serializable {
    private String objectId;
    private String name;
    private Parent monday;
    private Parent tuesday;
    private Parent wednesday;
    private Parent thursday;
    private Parent friday;
    private Parent saturday;
    private Parent sunday;

    public String getObjectId() { return objectId; }
    public void setObjectId(String objectId) { this.objectId = objectId; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public Parent getMonday() { return monday; }
    public void setMonday(Parent monday) { this.monday = monday; }
    public Parent getTuesday() { return tuesday; }
    public void setTuesday(Parent tuesday) { this.tuesday = tuesday; }
    public Parent getWednesday() { return wednesday; }
    public void setWednesday(Parent wednesday) { this.wednesday = wednesday; }
    public Parent getThursday() { return thursday; }
    public void setThursday(Parent thursday) { this.thursday = thursday; }
    public Parent getFriday() { return friday; }
    public void setFriday(Parent friday) { this.friday = friday; }
    public Parent getSaturday() { return saturday; }
    public void setSaturday(Parent saturday) { this.saturday = saturday; }
    public Parent getSunday() { return sunday; }
    public void setSunday(Parent sunday) { this.sunday = sunday; }
}

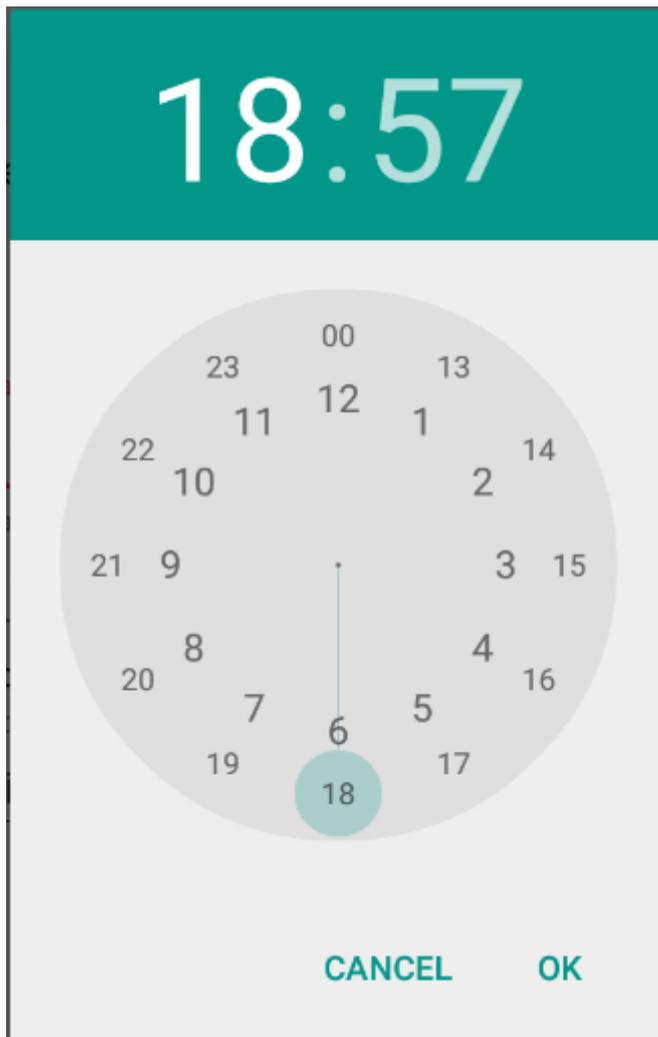
```

2.4.3.4 Controles.

Para la selección de fechas se utiliza DatePickerDialog

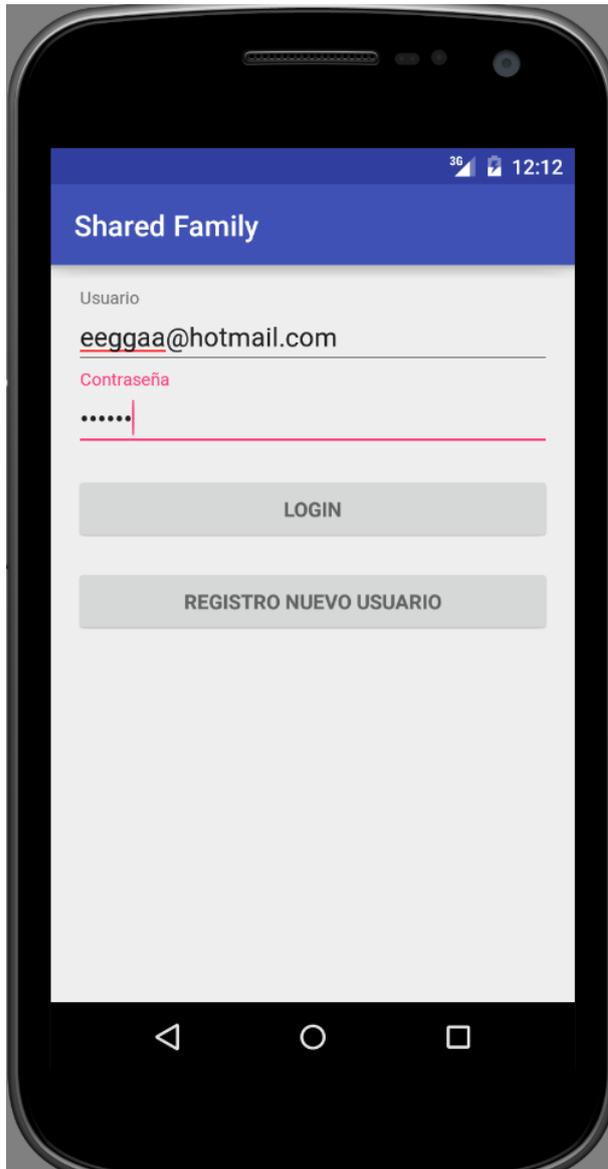


Para la selección de horas se utiliza TimePickerDialog



2.4.3.5 Login.

En la pantalla de login se implementa que el usuario sea el email del mismo.

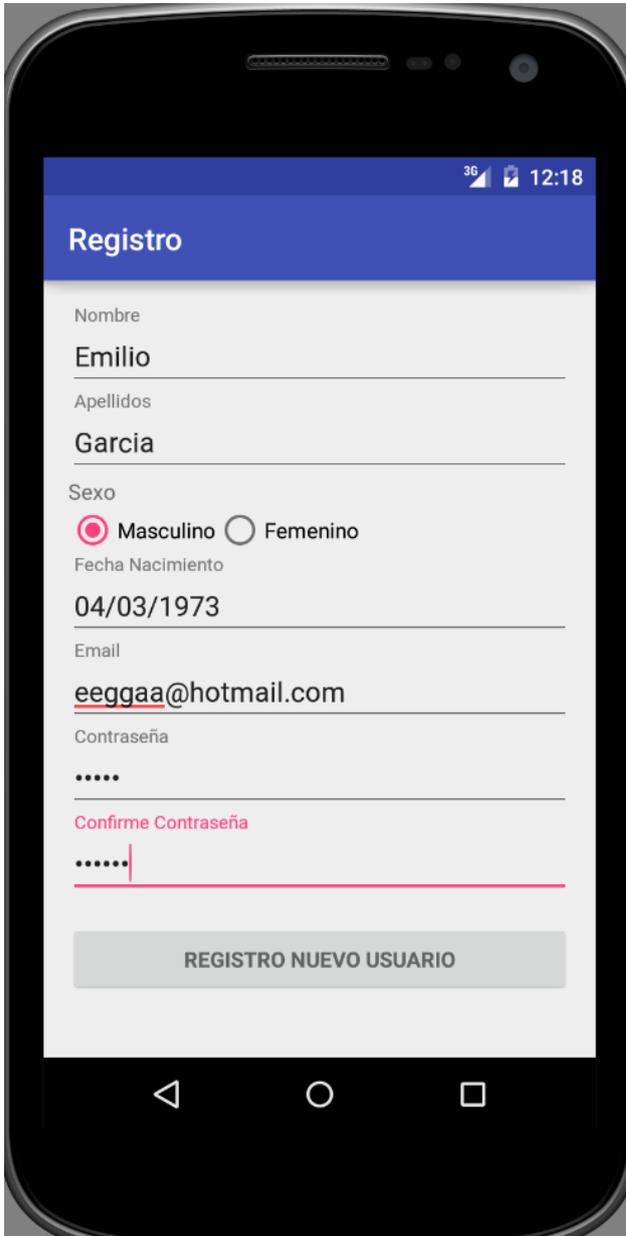


Se valida usuario y contraseña contra BackEndless[4] y se es correcto se da acceso a la aplicación, mostrando mensaje de error en caso contrario.

En esta pantalla se da acceso al registro de nuevo usuario.

2.4.3.6 Registro de nuevo usuario.

Se pide al usuario nombre, apellidos, sexo, fecha nacimiento, email, contraseña y confirmación de contraseña, validándose que todos ellos estén informados y que el email tenga un formato correcto.



The image shows a mobile application registration screen. At the top, there is a blue header with the word "Registro" in white. Below the header, the form fields are as follows: "Nombre" with the value "Emilio"; "Apellidos" with the value "Garcia"; "Sexo" with radio buttons for "Masculino" (selected) and "Femenino"; "Fecha Nacimiento" with the value "04/03/1973"; "Email" with the value "eeggaa@hotmail.com"; "Contraseña" with a masked input (dots); and "Confirme Contraseña" with a masked input (dots). A red underline is visible under the "Confirme Contraseña" field. At the bottom of the form, there is a grey button labeled "REGISTRO NUEVO USUARIO". The status bar at the top of the phone shows "3G" and the time "12:18".

El sistema verifica si ya existe un usuario registrado con dicha cuenta de correo.

Si no existe, se crea el usuario enviándose correo de confirmación al email indicado.

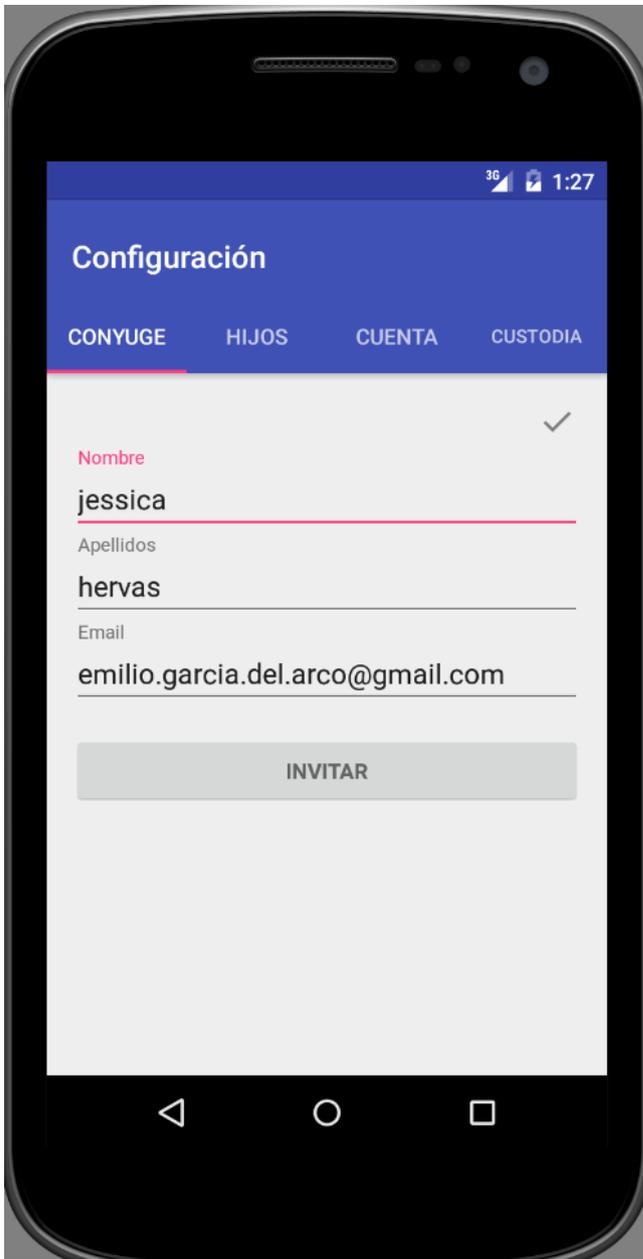
El usuario no podrá entrar en la aplicación con las credenciales suministradas hasta que confirme el email enviado.

2.4.3.7 Configuración.

La configuración se compone de cuatro pestañas, la primera para configurar el cónyuge, la segunda para configurar los hijos, la tercera para configurar la cuenta y finalmente la configuración del calendario de custodia compartida.

2.4.3.7.1 Configuración cónyuge.

En esta pantalla se indica el cónyuge, informando de su nombre, apellidos y email.



Se valida que los campos estén informados y la corrección del email.

Para que los cambios surjan efecto, es necesario guardar pulsando en el 'visto' de la parte superior derecha.

Se puede lanzar la invitación de uso de la aplicación al cónyuge mediante el botón 'invitar'.

Dicha acción, envía un email al email indicado, instando a la persona a que se descargue e instale la aplicación.

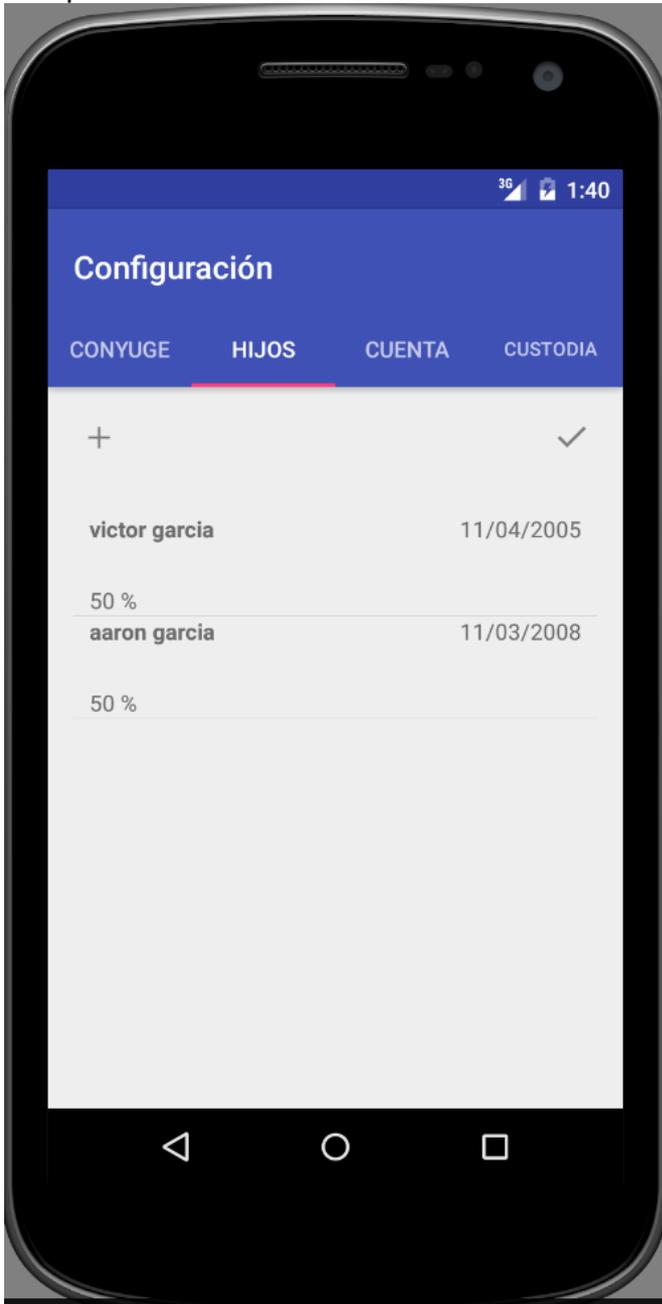
Una vez instalada, la persona deberá realizar el proceso de registro de nuevo usuario con el mismo email con el que ha recibido la invitación.

De este modo, ambos usuarios quedan relacionados cada uno como cónyuge del otro.

- En esta versión de la aplicación no funciona correctamente el email de invitación. Será solucionado en la FASEII.

2.4.3.7.2 Configuración hijos.

De entrada se muestra la lista de hijos configurados, mostrando para cada uno de ellos, el nombre y apellidos, la fecha de nacimiento y el porcentaje de compartición.

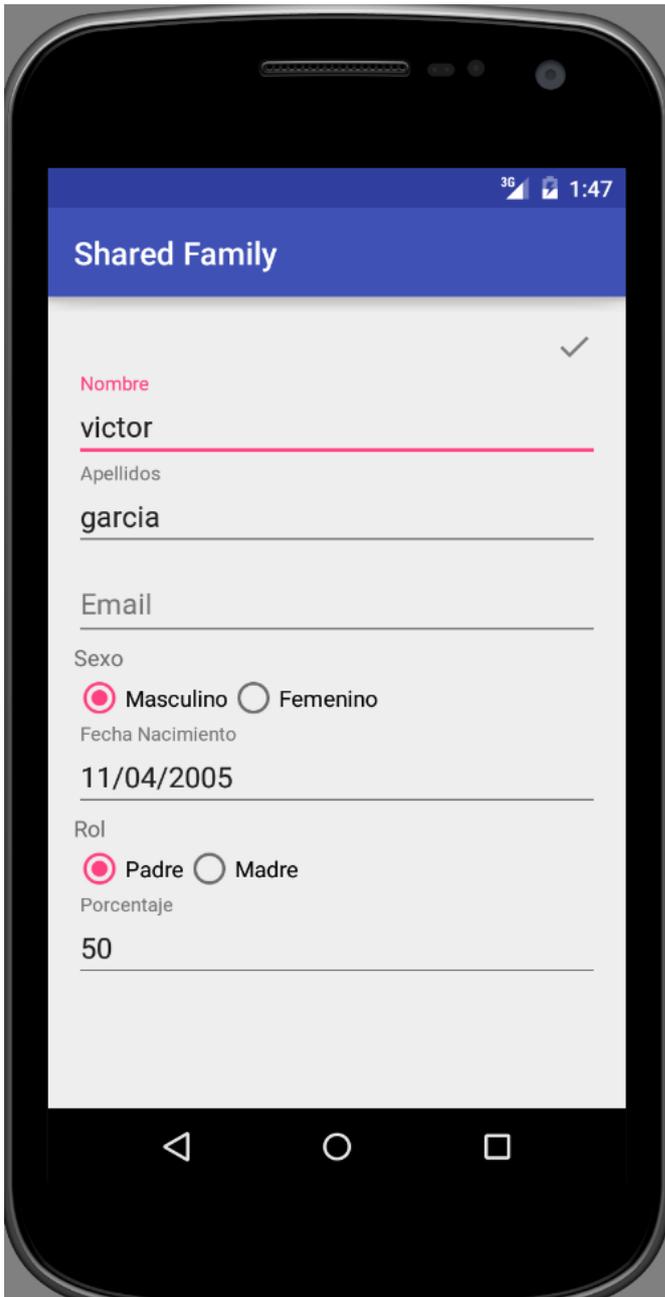


Pulsando sobre el icono + se abrirá la pantalla de detalle de hijo para añadir un hijo nuevo y pulsando sobre uno de los hijos se abrirá la pantalla de detalle de hijo para editar el hijo seleccionado.

Una vez realizados los cambios en la pantalla de detalle, será necesario pulsar el botón 'visto' de la parte superior derecha para guardar los cambios en base de datos, de manera que si no se realiza dicha acción se descartarán los cambios.

Al guardar los cambios en base de datos, se guardan tanto para el usuario actual como para el cónyuge.

En la pantalla de detalle de hijo, se podrá editar o crear hijos, indicando nombre, apellidos, email opcional, sexo, fecha nacimiento rol sobre el que actual la persona y porcentaje de compartición asignado a la misma.



Shared Family

3G 1:47

Nombre ✓
victor

Apellidos
garcia

Email

Sexo
 Masculino Femenino

Fecha Nacimiento
11/04/2005

Rol
 Padre Madre

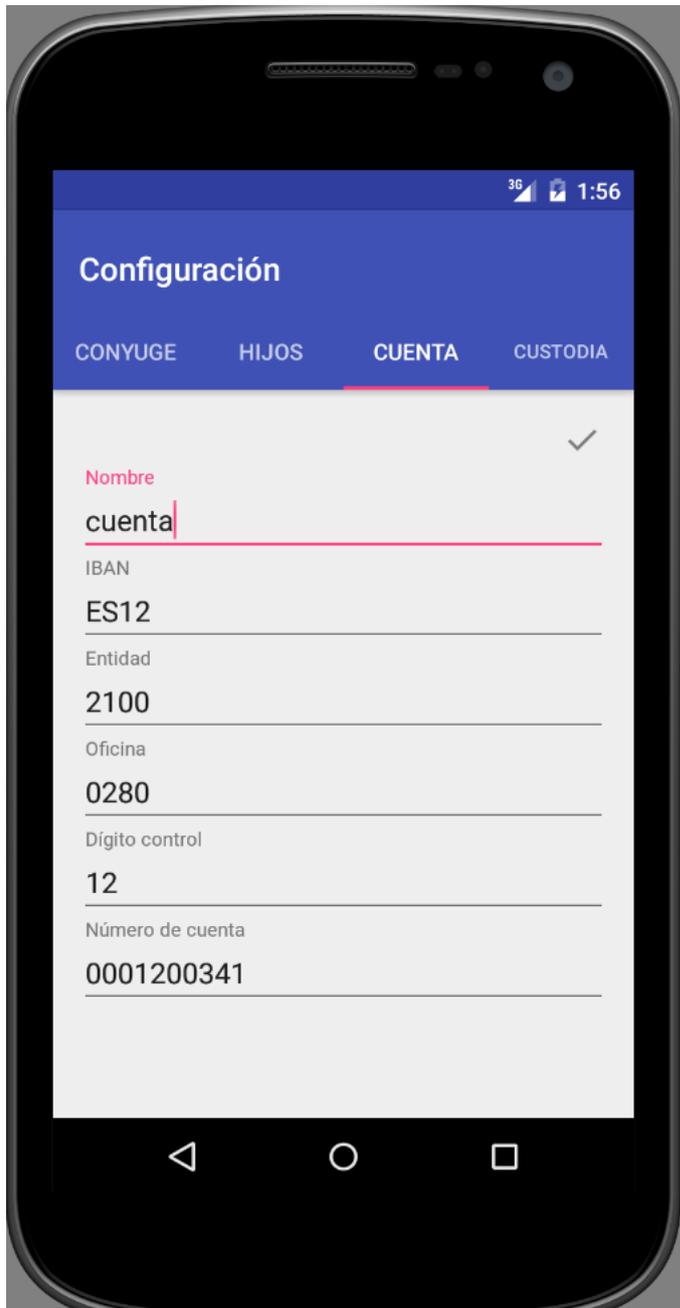
Porcentaje
50

Tras pulsar el 'visto' de la parte superior derecha, se valida la información introducida y se actualiza la lista de hijos en memoria, a la espera de que el usuario desee guardarlo en base de datos.

- En esta versión de la aplicación no se ha implementado la invitación de uso de la aplicación a los hijos, la cual queda para la FASEII.

2.4.3.7.3 Configuración cuenta.

Aquí se podrá indicar la cuenta compartida por ambos cónyuges. Se deberá informar de un nombre para la cuenta, así como el iban, entidad, oficina, dígito de control y número de cuenta.



The screenshot shows a mobile application interface for configuring a shared account. The screen is titled 'Configuración' and has a blue header. Below the header, there are four tabs: 'CONYUGE', 'HIJOS', 'CUENTA', and 'CUSTODIA'. The 'CUENTA' tab is selected. The form contains the following fields:

- Nombre: cuenta
- IBAN: ES12
- Entidad: 2100
- Oficina: 0280
- Dígito control: 12
- Número de cuenta: 0001200341

A checkmark is visible in the top right corner of the form area.

Tras pulsar el icono 'visto' en la parte superior derecha se valida que se hayan introducido todos los datos y se guarda la cuenta en la base de datos, asignando dicha cuenta también al cónyuge respectivo.

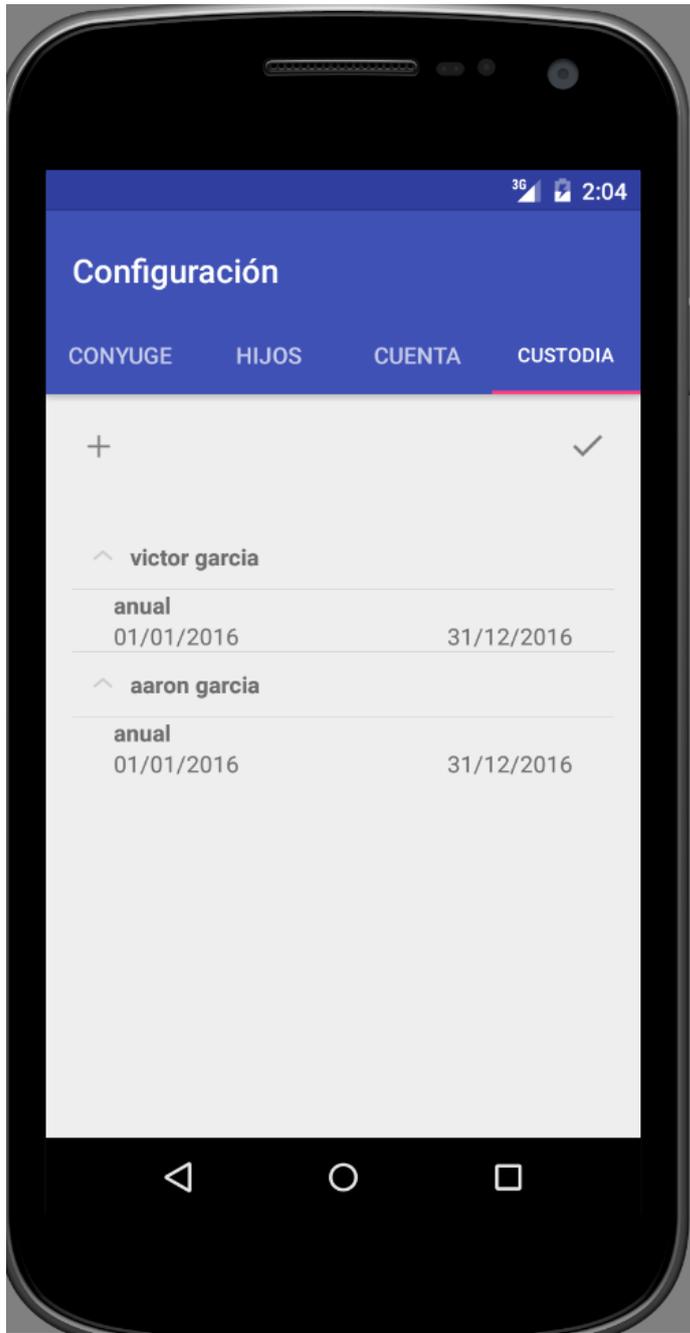
Por el momento no se valida la coherencia de los datos introducidos.

Se deja para futuras versiones en que se integre la aplicación con la información bancaria.

2.4.3.7.4 Configuración calendarios de custodia.

Se muestran los calendarios de custodia agrupados por hijo.

Para cada calendario de custodia se muestra el nombre y la fecha inicial y final.



Pulsando sobre el icono + se abrirá la pantalla de detalle de calendario para añadir uno nuevo y pulsando sobre uno de los calendarios se abrirá la pantalla de detalle para editar el calendario seleccionado.

Una vez realizados los cambios en la pantalla de detalle, será necesario pulsar el botón 'visto' de la parte superior derecha para guardar los cambios en base de datos, de manera que si no se realiza dicha acción se descartarán los cambios.

Al guardar los cambios en base de datos, se guardan tanto para el usuario actual como para el cónyuge.

En la pantalla de detalle de calendario de custodia se define el nombre, hijo afectado, fechas iniciales y finales, nombre de la semana y detalle del tutor que tiene la custodia cada uno de los días de la semana.



Tras pulsar el 'visto' de la parte superior derecha, se valida la información introducida y se actualiza la lista de calendarios de custodia en memoria, a la espera de que el usuario desee guardarlo en base de datos.

- En esta versión de la aplicación solo se ha implementado la posibilidad de introducir una semana por calendario de custodia.
 - No se ha implementado la funcionalidad de validación de calendarios de custodia por parte de los cónyuges.
- Queda para la FASEII el desarrollo de dichas funcionalidades.

2.4.3.8 Pantalla inicial.

La pantalla de inicio se compone de tres pestañas, la primera muestra el 'dashboard', la segunda para la gestión financiera y la cuarta para gestionar el calendario de eventos.



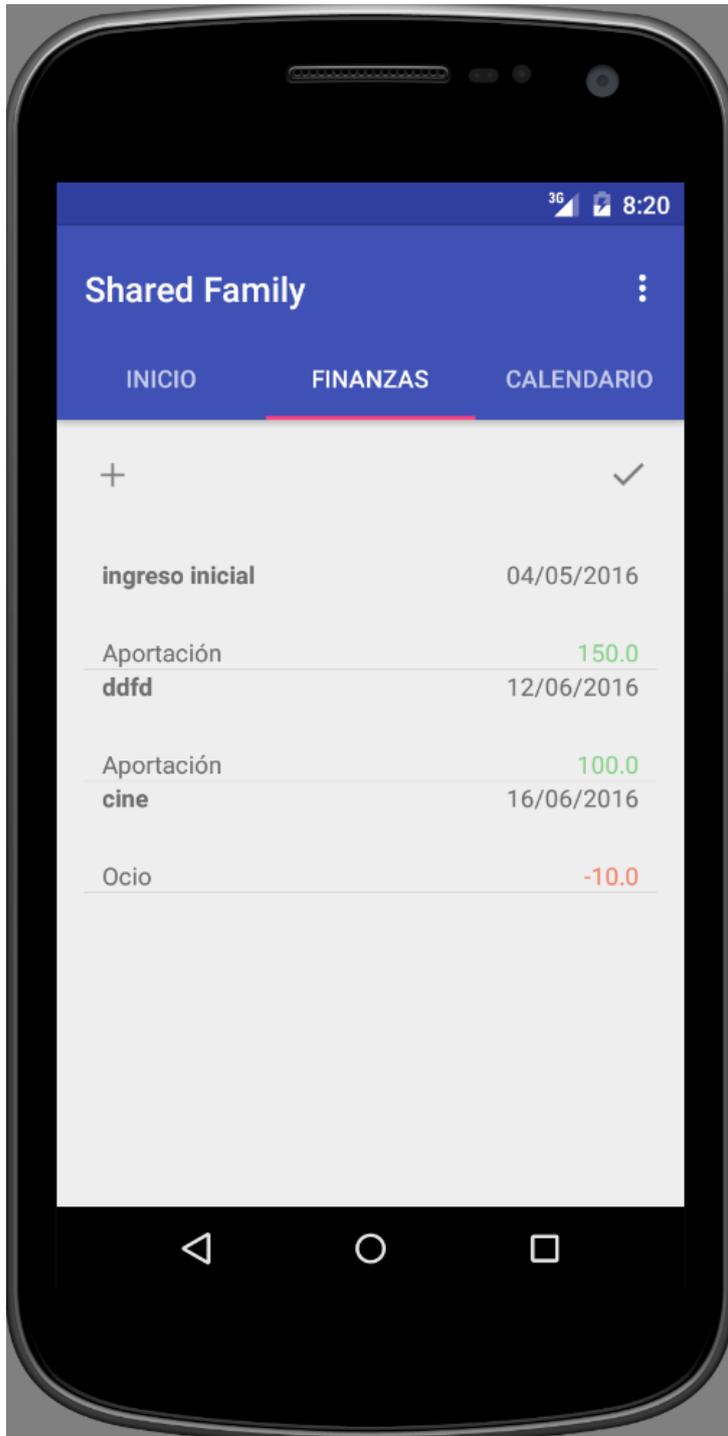
En el 'dashboard' se cargan los eventos del calendario del día de hoy, ordenados por hora de inicio, mostrando los primeros caracteres de la descripción, la hora inicio y fin, el hijo implicado y el tipo de evento.

En la parte inferior se muestra el resumen financiero del mes en curso desglosado por cónyuge, mostrando el saldo al inicio del mes total y de cada cónyuge, los ingresos totales del mes y de cada cónyuge, los gastos totales del mes y de cada cónyuge y el saldo final total y de cada cónyuge.

- Se pospone para la FASEII de desarrollo dar la posibilidad de navegar por meses en el resumen financiero.

2.4.3.9 Finanzas.

Se muestran todos los movimientos financieros de la cuenta compartida por los cónyuges, mostrando la descripción, fecha, categoría e importe de cada uno de ellos.



En función de si el movimiento es de ingreso o gasto se muestra el importe el verde o rojo respectivamente.

Pulsando sobre el icono + se abrirá la pantalla de detalle de movimiento para añadir un nuevo movimiento y pulsando sobre uno de ellos se abrirá la pantalla de detalle para editarlo.

Una vez realizados los cambios en la pantalla de detalle, será necesario pulsar el botón 'visto' de la parte superior derecha para guardar los cambios en base de datos, de manera que si no se realiza dicha acción se descartarán los cambios.

- Queda pendiente para la FASEII de desarrollo dar la posibilidad de navegar por meses y la posibilidad de visualizar los movimientos del cónyuge, los míos o todos.

En la pantalla de detalle de movimiento se debe seleccionar si el movimiento es de ingreso o gasto, la categoría del mismo, el hijo afectado, la fecha del mismo, el importe y la descripción.

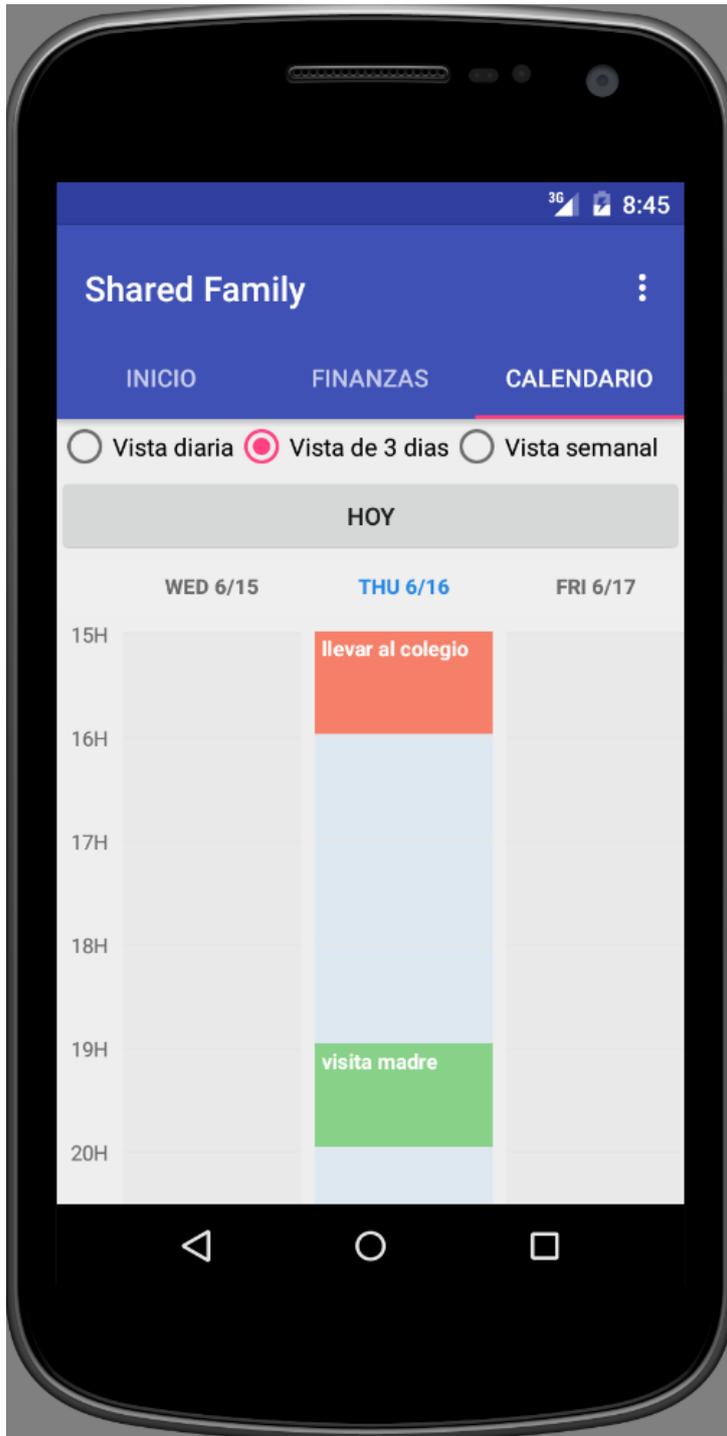


En el caso de estar editando un movimiento del cónyuge, no podrá ser modificado. Tras pulsar el 'visto' de la parte superior derecha, se valida la información introducida y se actualiza la lista de movimientos en memoria, a la espera de que el usuario desee guardarlo en base de datos.

- Queda pendiente para la FASEII de desarrollo dar la posibilidad de gestionar movimientos favoritos y adjuntar archivos y fotos a los mismos.

2.4.3.10 Calendario.

La implementación del calendario se ha realizado distinta a la diseñada, utilizando un formato de calendario más estándar en dispositivos móviles.



Se muestra por defecto en formato de vista de tres días, dando la opción de vista diaria y vista semanal, presentando los días como columna y las horas del día como filas, siempre mostrando como primer día el día de hoy.

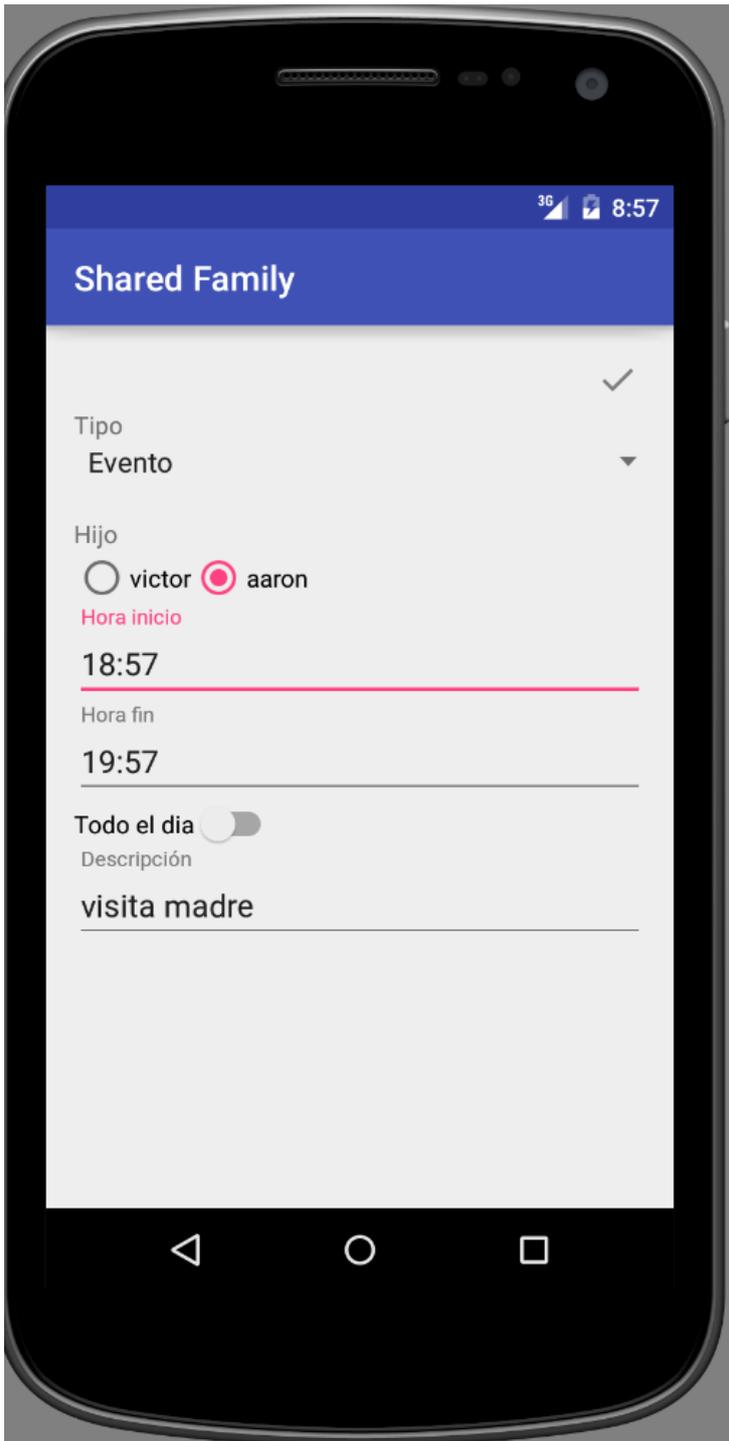
Para visualizar las horas del día anteriores o posteriores a las visibles se utiliza la navegación por desplazamiento vertical, mientras que para mostrar los días anteriores o posteriores se utiliza la navegación por desplazamiento horizontal. Los eventos se muestran en el calendario ocupando el espacio necesario entre su hora de inicio y hora de finalización y mostrándose de un color distinto en función del tipo de evento. Se muestra para cada evento su descripción.

Se incluye el botón HOY, siempre visible para acceder al día de hoy si se ha navegado adelante o atrás mediante desplazamiento horizontal.

Pulsando sobre un evento del calendario accedemos a la pantalla de detalle del evento.

Accederemos a la pantalla de detalle de eventos en modo creación de nuevo evento, haciendo un toque largo en un día y hora del calendario.

En la pantalla de detalle de eventos de calendario se seleccionará el tipo de evento, el hijo afectado, la hora inicial y final y la descripción del mismo.



Se dispone de la opción 'Todo el día', activando la cual se selecciona automáticamente el evento para el día completo.

Tras pulsar el 'visto' de la parte superior derecha, se valida la información introducida, se guarda el evento en la base de datos y se actualiza el calendario.

2.4.3.11 Plan de pruebas.

El plan de pruebas consiste en las pruebas unitarias y pruebas de usuario tras la entrega de cada funcionalidad.

Pruebas unitarias.

Han sido realizadas por los desarrolladores tras la implementación de cada funcionalidad. Están integradas en las propias herramientas de desarrollo y consisten en el test aislado de cada funcionalidad para asegurar que cumple con los requisitos y resultados esperados.

Pruebas de usuario.

Tras la entrega de cada funcionalidad, los usuarios finales han realizado un test de cada una de ellas en dispositivos reales.

Detalle de las pruebas de usuario realizadas.

Módulo de acceso y registro.

Pantalla	Acciones a testear	Resultados esperados
Acceso usuario	Introducir usuario y contraseña.	Acceso a la aplicación si usuario y contraseña correcto. Mensaje error si usuario o contraseña incorrecto.
Nuevo usuario	Introducir nombre, apellidos, sexo, fecha nacimiento, email, contraseña y confirmación contraseña.	Creación de nuevo usuario, creación de nuevo tutor y acceso a la aplicación si email correcto y contraseña segura. Mensaje de error si usuario ya existe, contraseña insegura, email incorrecto o datos obligatorios no introducidos.
Nuevo usuario tras invitación	Introducir nombre, apellidos, sexo, fecha nacimiento, email, contraseña y confirmación contraseña.	Creación de nuevo usuario, asignación al tutor coincidente por email si no tiene usuario creado previamente y acceso a la aplicación si email correcto y contraseña segura. Mensaje de error si usuario ya existe, contraseña insegura, email incorrecto o datos obligatorios no introducidos.

Módulo de configuración

Pantalla	Acciones a testear	Resultados esperados
Configuración cónyuge	<p>Introducir nombre, apellidos, email y pulsar el icono </p> <p>Invitar al cónyuge pulsando el botón </p>	<p>Se guarda el cónyuge en BBDD si email correcto y nombre y apellidos introducidos. Mensaje de error en caso contrario.</p> <p>Se envía email con las instrucciones de descarga de la aplicación y registro de nuevo usuario si email correcto y nombre y apellidos introducidos. Mensaje de error en caso contrario.</p>
Configuración hijos	<p>En la pestaña configuración hijos</p>	<p>Se visualizan todos los hijos configurados en modo lista.</p>
Detalle hijo	<p>En la pestaña configuración hijos, pulsar el botón  para añadir un nuevo hijo.</p> <p>En la pantalla de detalle de hijo, introducir nombre, apellidos, email (opcional), sexo, fecha nacimiento, rol, porcentaje de custodia del tutor y pulsar el botón .</p>	<p>Se abre la pantalla de detalle de hijo en modo creación.</p> <p>Se guarda el nuevo hijo en memoria volviendo a la lista de hijos añadiendo el nuevo hijo a la lista, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.</p>
Configuración hijos	<p>En la pestaña configuración hijos, pulsar sobre un hijo de la lista para editarlo.</p>	<p>Se abre la pantalla de detalle de hijo en modo edición, mostrando los datos del hijo seleccionado.</p>
Detalle hijo	<p>En la pantalla de detalle de hijo, modificar la información deseada y pulsar el botón .</p>	<p>Se guarda el hijo editado en memoria volviendo a la lista de hijos modificando las propiedades modificadas del hijo en la lista, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.</p>
Configuración hijos	<p>En la pestaña configuración hijos, pulsar el icono </p>	<p>Se guardan los hijos en BBDD</p>

Configuración cuenta	Introducir nombre, IBAN, entidad, oficina, dígito control, número de cuenta y pulsar el botón 	Se guarda la información de la cuenta en BBDD si los datos introducidos son correctos. Mensaje de error en caso contrario.
Configuración custodia	En la pestaña configuración de custodia	Se muestran en modo lista de dos niveles los calendarios de custodia configurados agrupados por hijo.
Detalle custodia	En la pestaña configuración custodia, pulsar el botón  para añadir un nuevo calendario de custodia. En la pantalla de detalle de custodia, introducir nombre, hijo afectado, fecha inicio y fin, nombre de la semana, a quien de los dos cónyuges le corresponde la custodia cada uno de los días de la semana y pulsar el botón  .	Se abre la pantalla de detalle de calendario de custodia en modo creación. Se guarda el nuevo calendario de custodia en memoria volviendo a la lista de calendarios de custodia añadiendo el nuevo calendario a la lista bajo el hijo correspondiente, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.
Configuración custodia	En la pestaña configuración custodia, pulsar sobre uno de los calendarios de la lista para editarlo.	Se abre la pantalla de detalle de calendario de custodia en modo edición, mostrando los datos del calendario seleccionado.
Detalle custodia	En la pantalla de detalle de custodia, modificar la información deseada y pulsar el botón  .	Se guarda el calendario editado en memoria volviendo a la lista de calendarios modificando las propiedades modificadas del calendario en la lista, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.
Configuración custodia	En la pestaña configuración custodia, pulsar el icono 	Se guardan los calendarios de custodia en BBDD

Pantalla inicio

Pantalla	Acciones a testear	Resultados esperados
Inicio	<p>Aparece la lista de eventos para el día de hoy ordenados por hora.</p> <p>Aparece el resumen financiero del mes en curso.</p>	<p>Se muestran todos los eventos del día de hoy ordenados por hora</p> <p>Se muestra el resumen financiero del mes en curso, mostrando la suma de movimientos de meses anteriores como saldo inicial de cada cónyuge y total, la suma de movimientos de ingreso del mes en curso de cada cónyuge y total, la suma de movimientos de gasto del mes en curso de cada cónyuge y total y el saldo final de cada cónyuge y total.</p>

Módulo de finanzas

Pantalla	Acciones a testear	Resultados esperados
Finanzas	En la pestaña finanzas	Se visualizan todos los movimientos financieros introducidos en modo lista, ordenados por fecha, mostrando el importe en positivo y de color verde para los ingresos y en negativo y de color rojo para los gastos.
Detalle movimiento financiero	En la pestaña finanzas, pulsar el botón para añadir un nuevo movimiento Financiero.	Se abre la pantalla de detalle de movimiento financiero en modo creación.
Finanzas	En la pantalla de detalle de movimiento financiero, seleccionar ingreso o gasto, categoría, hijo afectado, introducir fecha movimiento, importe, descripción y pulsar el botón .	Se guarda el nuevo movimiento en memoria volviendo a la lista de finanzas añadiendo el nuevo movimiento a la lista, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.
Finanzas	En la pestaña finanzas, pulsar sobre un movimiento de la lista para editarlo.	Se abre la pantalla de detalle movimiento financiero en modo edición, mostrando los datos del movimiento seleccionado. Si el propietario del movimiento es el cónyuge, no aparece el botón .
Detalle movimiento financiero	En la pantalla de detalle de movimiento financiero, modificar la información deseada y pulsar el botón .	Se guarda el movimiento financiero editado en memoria volviendo a la lista de finanzas modificando las propiedades modificadas del movimiento en la lista, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.
Finanzas	En la pestaña finanzas, pulsar el icono .	Se guardan los movimientos financieros en BBDD

Módulo calendario eventos

Pantalla	Acciones a testear	Resultados esperados
Calendario	En la pestaña calendario	Se visualizan todos los eventos de hoy, mañana y pasado en modo calendario, ordenados por fecha, mostrando en los colores correspondientes, los eventos, las tareas y las visitas.
	Seleccionar vista diaria	Se visualizan todos los eventos de hoy en modo calendario, ordenados por fecha, mostrando en los colores correspondientes, los eventos, las tareas y las visitas. Se abre la pantalla de detalle de movimiento financiero en modo creación.
	Seleccionar vista semanal	Se visualizan todos los eventos de la semana actual en modo calendario, ordenados por fecha, mostrando en los colores correspondientes, los eventos, las tareas y las visitas.
	Realizar una pulsación larga sobre un día y hora en la que no existe ningún evento	Se abre la pantalla de detalle de evento en modo creación.
Detalle evento	En la pantalla de detalle de evento, seleccionar evento, tarea o visita, hijo afectado, introducir hora inicio y fin, descripción y pulsar el botón  .	Se guarda el nuevo evento en BBDD volviendo al calendario añadiendo el nuevo evento en el día y hora seleccionados, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.
Calendario	Pulsar sobre un evento para editarlo.	Se abre la pantalla de detalle de evento en modo edición, mostrando los datos del evento

Detalle evento	<p>En la pantalla de detalle de evento, modificar la información deseada y pulsar el botón </p> <p>En modo creación o en modo edición, activar el botón  Todo el día</p>	<p>Se guarda el evento editado en BBDD volviendo al calendario modificando las propiedades modificadas del evento en el mismo, si se ha introducido correctamente toda la información. Mensaje de error en caso contrario.</p> <p>Se selecciona automáticamente como hora inicio las 00:00 y como hora fin las 23:59</p>
----------------	--	--

3. Conclusiones

Se ha realizado una fase de investigación muy completa, mediante la cual se han obtenido unos requerimientos y funcionalidades muy claros.

El diseño de la interacción ha sido muy productivo, obteniendo un diseño de la aplicación en alta fidelidad, el cual ha servido para poder realizar una investigación con usuarios muy completa, la cual ha dado resultados muy positivos para acabar afinando el diseño de la aplicación.

Para el desarrollo de la aplicación, viendo el poco tiempo y recursos disponibles, se ha decidido dejar para siguientes fases algunas de las funcionalidades no primordiales y finalmente, durante el propio proceso de desarrollo, se ha optado por limitar ciertas funcionalidades y dejarlas también para siguientes fases.

Durante la fase de desarrollo, ha sido muy útil poder contar con usuarios finales para realizar pruebas de usuario de cada funcionalidad, lo que ha servido para pulir mucho la aplicación.

Como conclusión final, podemos afirmar que se ha conseguido diseñar y desarrollar una aplicación que cumple con los objetivos propuestos, eso sí, a mínimos, pero que constituye una base muy sólida para seguir ampliando funcionalidades en siguientes fases y finalmente poder obtener una aplicación muy completa.

4. Glosario

- **Custodia**
Responsabilidad que se tiene sobre la educación y el bienestar de una persona menor de edad.
- **Pernoctación**
Acción y efecto de pasar la noche en un lugar que no es la vivienda habitual.
- **Cónyuge**
Esposo o esposa de una persona.
- **App**
Aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo —profesional, de ocio, educativas, de acceso a servicios, etc.—, facilitando las gestiones o actividades a desarrollar.
- **DashBoard**
Cuadro de mando con la información más relevante o importante de un conjunto de datos o conjunto de sistemas.
- **IBAN**
Código alfanumérico que identifica una cuenta bancaria determinada en una entidad financiera en cualquier lugar del mundo (si el país está adherido al sistema IBAN).
- **Smartphone**
Tipo de teléfono móvil construido sobre una plataforma informática móvil, con mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional. El término «inteligente», que se utiliza con fines comerciales, hagan referencia a la capacidad de usarse como un computador de bolsillo, y llega incluso a reemplazar a una computadora personal en algunos casos.

5. Bibliografía

- [1] <https://www.2houses.com/en/>
- [2] <https://www.whatsapp.com/?l=es>
- [3] <http://www.justinmind.com/>
- [4] <https://backendless.com/>
- [5] <https://www.android.com/>
- [6] <https://developer.android.com/studio/index.html>

6. Anexos

- [Prototipo SharedFamily](#)
- [Diagrama de Gantt](#)