



Desarrollo de la aplicación *Lived*

Nombre Estudiante: Marc Fernández Veiga

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Eduard Martin Lineros

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

01/2017



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:	<i>Desarrollo de la aplicación Lived</i>
Nombre del autor:	<i>Marc Fernández Veiga</i>
Nombre del consultor/a:	<i>Eduard Martín Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2017
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Recopilación, Recuerdos, Vida</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>Este trabajo tiene como finalidad el desarrollo de una aplicación para dispositivos Android: <i>Lived</i>.</p> <p>A medida que pasan los años, las personas van olvidando ciertos sucesos de sus vidas que, quizás no provocaron cambios en ellas, pero que sí desearían no olvidar. Es por ello por lo que se ha decidido desarrollar esta aplicación que permite al usuario recopilar todos sus momentos vividos en forma de fotografía, vídeo o texto. Con esto, las personas podrán revivir sus recuerdos como celebraciones, quedadas con las personas que les importan, nacimientos, fallecimientos...</p> <p>Para la construcción de dicha aplicación, se ha estudiado el público objetivo y se ha realizado un diseño de producto interactivo completamente nuevo y cómodo para el mismo. Seguidamente se ha creado un prototipo el cual ha sido probado con algunos usuarios potenciales. Se ha programado la aplicación en Android Studio durante la mayor parte de la duración del proyecto, así como la realización de pruebas y correjimientos necesarios, hasta conseguir una aplicación de alta calidad completamente funcional.</p>	

Abstract (in English, 250 words or less):

This work is aimed at developing an application for Android devices: *Lived*.

As the years pass, people forget certain events in their lives that perhaps did not cause changes in them, but do not wish to forget. This is why it has been decided to develop this app that allows the user to collect all the experienced moments as a photography, video or text. With this, people can relive their memories as celebrations, hangouts with people who they care about, births, deaths...

For the construction of this application, the target audience has been studied to make a completely new and comfortable interactive design. Then it has been created a prototype which has been tested with some potential users. The application has been programmed with Android Studio for the wide majority of the duration of the project, as well as testing and necessary changes, to get a fully functional application of high quality.

Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguido	2
1.4 Planificación del Trabajo	3
1.5 Breve sumario de productos obtenidos	5
1.6 Breve descripción de los otros capítulos de la memoria	6
2. Diseño	7
2.1 Definición del usuario, sus necesidades y contextos de uso	7
2.2 Definición de las funciones	9
2.3 Diseño conceptual	12
2.4 Prototipado	19
2.5 Evaluación	22
3. Implementación	25
3.1 LoginActivity	25
3.2 ConditionsActivity	26
3.3 EmailActivity	27
3.4 NameActivity	28
3.5 PasswordActivity	28
3.6 MainActivity	29
3.6.1 MemoriesFragment	30
3.6.2 CameraFragment	31
3.6.3 ProfileFragment	33
3.7 AddMemoryActivity	34
3.8 ViewMemoryActivity	35
3.9 SettingsActivity	35
4. Librerías de terceros	36
5. Pruebas	37
6. Conclusiones	38
7. Glosario	40
8. Bibliografía	41
9. Anexos	42
9.1 Instrucciones de compilación	42
9.2 Manual de usuario	43

Lista de figuras

Figura 1.4.1: Diagrama de Gantt.....	4
Figura 2.2.1 Diagrama de caso de uso Jose.....	12
Figura 2.2.2 Diagrama de caso de uso Sandra.....	12
Figura 2.2.3 Diagrama de caso de uso Francisca.....	12
Figura 2.3.1 Árbol de navegación de la app.....	13
Figura 2.3.2 Esbozo del primer camino de la app.....	14
Figura 2.3.3 Esbozo del segundo camino de la app	14
Figura 2.3.4 Esbozo del segundo camino de la app	15
Figura 2.3.5 Mockup del primer camino de la app	16
Figura 2.3.6 Mockup del segundo camino de la app.....	16
Figura 2.3.7 Mockup del segundo camino de la app.....	17
Figura 2.3.8 Diseño alternativo para la pantalla 6.1	18
Figuras 2.4.1 a 2.4.14 Pantallas del prototipo de alta fidelidad.....	22
Figura 2.5.1 Diagrama UML del diseño de la base de datos	23
Figura 2.5.2 Diagrama explicativo de la arquitectura del sistema	24
Figura 3.1.1 LoginActivity	25
Figura 3.1.2 Solicitar cambio de contraseña	26
Figura 3.2.1 ConditionsActivity	26
Figura 3.3.1 EmailActivity 1	27
Figura 3.3.2 EmailActivity 2.....	27
Figura 3.4.1 NameActivity	28
Figura 3.5.1 PasswordActivity 1	28
Figura 3.5.2 PasswordActivity 2	29
Figura 3.6.1 MainActivity	29
Figura 3.6.1.1 MemoriesFragment	30
Figura 3.6.2.1 CameraFragment	31
Figura 3.6.3.1 ProfileFragment.....	33
Figura 3.7.1 AddMemoryActivity	34
Figura 3.8.1 ViewMemoryActivity	35
Figura 3.9.1 SettingsActivity	35

Lista de tablas

Tabla 1.4.1: Planificación temporal	5
---	---

1. Introducción

1.1 Contexto y justificación del Trabajo

La vida es una sucesión de recuerdos, pero cuando se mira hacia atrás, el usuario sólo recuerda eventos muy importantes y que han supuesto un cambio en su vida. No obstante, se viven momentos importantes (vacaciones, encuentros con amigos, recuerdos con familiares, etc.) que, si no se tiene una forma de guardarlos, la memoria los va perdiendo. Esta herramienta pretende ser una especie de álbum de recuerdos del paso del tiempo en las vidas de los usuarios. Una forma para guardar hasta los recuerdos más personales, sin tener que compartirlos con el resto de la sociedad. La principal pregunta que formó la base de la idea de la aplicación es la siguiente: ¿Qué recuerda usted del año XXXX? La vida es muy corta y las personas no son capaces de acordarse de todos los momentos que han vivido. Por ese motivo, a la hora de responder la pregunta planteada, algunos pocos responderán algo como “Nació ...”, “Viajé a ...” o “Murió ...”; pero nada más. Así pues, la aplicación que se pretende desarrollar es una que permita al usuario acceder de forma sencilla a todos los recuerdos de su vida. Un calendario al que se le puede asignar cualquier recuerdo en forma de fotografía, vídeo o texto a cualquier día, semana, mes o año. De esta manera, cualquiera que lo desee podrá recopilar todos los detalles de su vida y observar en cualquier momento el recorrido de la misma. Algo que se pretende añadir y que hará destacar este producto sobre los demás es la incorporación de la realidad virtual. Simplemente publicando fotografías o vídeos en 360º y con un dispositivo de gafas VR, el usuario no sólo observará el recorrido de su vida, sino que podrá revivir cualquier momento.

Se ha buscado a ver si existía algo parecido en el mercado de las aplicaciones para *smartphones* y no se ha encontrado nada por el estilo, menos aun incorporando la realidad virtual. La única referencia de la que se parte es la red social Instagram. Esta aplicación permite publicar fotografías y vídeos, pero se hacen públicas. Sin embargo, se desea crear un producto que recopile todos los momentos que desee el usuario y es muy probable que muchos de esos momentos no se quieran hacer públicos.

Así pues, se pretende hacer la aplicación de manera fácil e intuitiva para que su uso sea satisfactorio para las personas que la utilizarán. Para ello, se disminuirá al máximo el trabajo cognitivo que la aplicación demanda del usuario. Para poner en práctica algunos conocimientos adquiridos en el máster, se realizará un diseño de producto multidispositivo, permitiendo así su uso en distintos dispositivos.

Resumiendo, con la aplicación que se pretende desarrollar durante este trabajo de final de máster, los usuarios podrán recopilar y revivir todos sus recuerdos de manera sencilla y desde distintos dispositivos. De esta manera, cuando se les pregunten ¿qué recuerda usted del año XXXX?, podrán contar todo lo que deseen.

1.2 Objetivos del Trabajo

El principal objetivo del trabajo final de máster es llevar a cabo un proyecto relacionado con el desarrollo de una aplicación para dispositivos móviles. Sin embargo, hay más objetivos ya que el TFM es el resultado final y natural del proceso de aprendizaje realizado durante el estudio del programa de máster:

- Poner en práctica los conocimientos adquiridos a lo largo de todo el máster.
- Saber definir una planificación realista para un proyecto, teniendo en cuenta su alcance y recursos disponibles.
- Ser capaz de documentar y justificar las decisiones tomadas en el desarrollo y los resultados conseguidos.
- Adquirir experiencia en afrontar los retos que supone sacar adelante un proyecto completo.
- Hacer converger los conocimientos adquiridos en un producto de alta calidad.

Además, a nivel personal se pretende evaluar la capacidad para saber utilizar de manera adecuada y conveniente las siguientes competencias:

- Capacidad de iniciativa y espíritu emprendedor.
- Capacidad para la expresión escrita para la vida académica y profesional.
- Capacidad para analizar proyectos de tecnología móvil considerando las características de las comunicaciones inalámbricas y su alcance, los diferentes tipos de dispositivos móviles y sus características, y las alternativas para el desarrollo de aplicaciones móviles.
- Capacidad para valorar de forma crítica las ventajas y desventajas del desarrollo para cada una de las plataformas móviles existentes.
- Capacidad para conocer y saber elegir los modelos de negocio más adecuados para el éxito de un proyecto de tecnología móvil.
- Capacidad para planificar y dirigir proyectos tecnológicos relacionados con tecnologías móviles.

1.3 Enfoque y método seguido

Para llevar a cabo el trabajo, se intentará desarrollar un producto nuevo. No se ha encontrado ninguna aplicación que ofrezca lo mismo, por lo tanto, no hay ningún producto a adaptar. Además, uno de los principales objetivos del trabajo de final de máster es el poner en práctica los conocimientos adquiridos a lo largo del máster. Por lo tanto, crear un producto desde su inicio permitirá trabajar con conceptos estudiados en asignaturas de “Diseño de productos interactivos multidispositivo” y su asignatura avanzada, cosa que no sería posible si se partiera de un diseño ya creado.

Tecnologías

Se utilizará un MacBook Air y la tecnología utilizada para el desarrollo de la aplicación es el entorno de desarrollo integrado (IDE) Android Studio, que permite desarrollar aplicaciones nativas para dispositivos Android.

Se ha descartado el desarrollo web por las necesidades tecnológicas de la aplicación, ya que no permite el acceso a herramientas del dispositivo como la cámara, ubicación o almacenamiento.

También se ha descartado el desarrollo con tecnologías híbridas ya que, a pesar de poder realizar un desarrollo para múltiples plataformas, el resultado obtenido no sería tan bueno como utilizando tecnologías del propio dispositivo.

Así pues, finalmente se ha decidido programar la aplicación con el SDK oficial debido a la optimización en usos de herramientas del dispositivo. Esto permite disponer de todas las herramientas para el desarrollador y también ofrece una experiencia cómoda al usuario ya que se utilizan los mismos elementos que el sistema.

Como se ha comentado anteriormente, la aplicación se desarrollará para Android y se ejecutará en un dispositivo Samsung Galaxy S6 Edge y Samsung Tab 2 para comprobar el funcionamiento de la misma. La app se pretende desarrollar con tecnologías nativas para poder utilizarla en dispositivos Android aprovechando al máximo las herramientas del propio dispositivo. Esto supondrá el posterior desarrollo para cada plataforma.

1.4 Planificación del Trabajo

Para la realización del trabajo, se necesitarán como recursos materiales un ordenador para la programación de la aplicación y distintos dispositivos de tamaños diferentes para poder probar la aplicación. A continuación, se presenta un diagrama de Gantt con la planificación temporal de cada tarea a realizar (en días). Los colores distinguen las tareas que pertenecen a cada PEC, de la 1 a la 5.

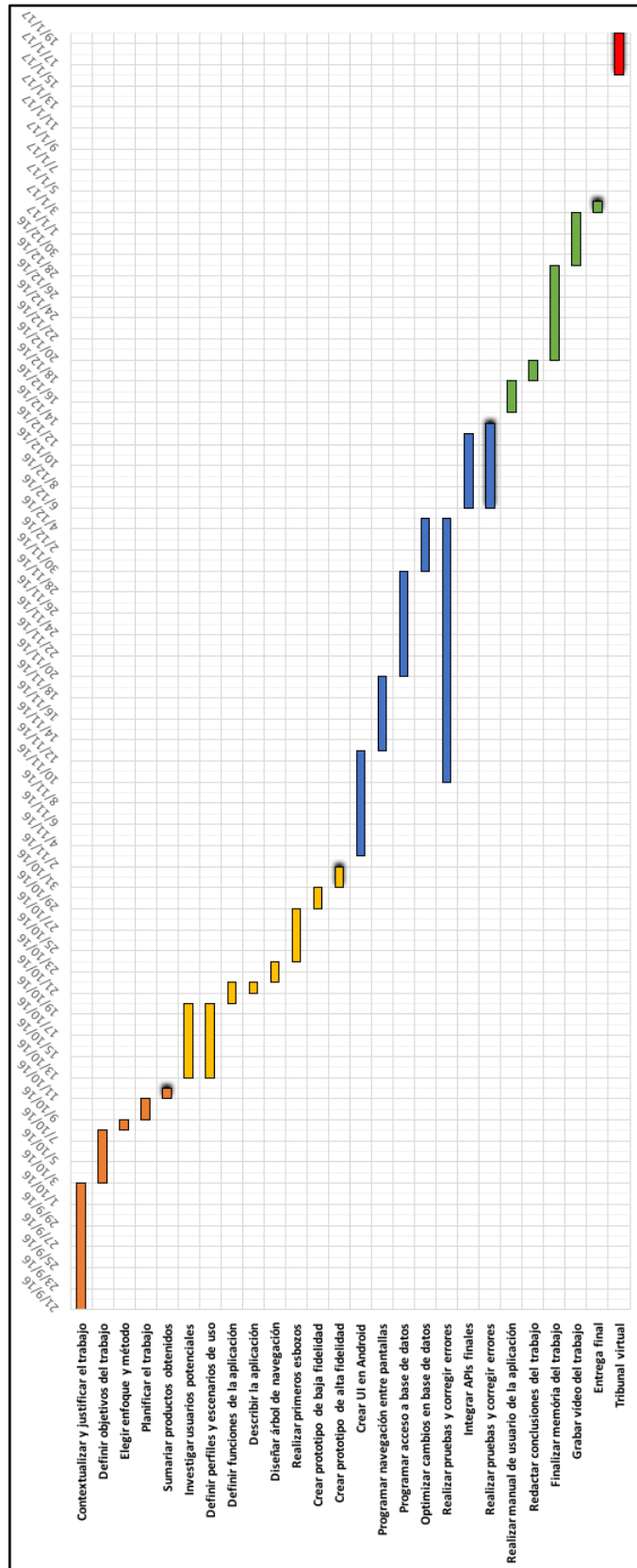


Figura 1.4.1: Diagrama de Gantt

Tarea	Inicio	Horas laborables	Horas festivos	Final
Contextualizar y justificar el trabajo	21/9/16	32	16	3/10/16
Definir objetivos del trabajo	3/10/16	20	0	8/10/16
Elegir enfoque y método	8/10/16	0	4	9/10/16
Planificar el trabajo	9/10/16	8	0	11/10/16
Sumariar productos obtenidos	11/10/16	4	0	12/10/16
Investigar usuarios potenciales	13/10/16	20	8	20/10/16
Definir perfiles y escenarios de uso	13/10/16	20	8	20/10/16
Definir funciones de la aplicación	20/10/16	8	0	22/10/16
Describir la aplicación	21/10/16	4	0	22/10/16
Diseñar árbol de navegación	22/10/16	0	8	24/10/16
Realizar primeros esbozos	24/10/16	20	0	29/10/16
Crear prototipo de baja fidelidad	29/10/16	0	8	31/10/16
Crear prototipo de alta fidelidad	31/10/16	8	0	2/11/16
Crear UI en Android	3/11/16	32	8	13/11/16
Programar navegación entre pantallas	13/11/16	20	8	20/11/16
Programar acceso a base de datos	20/11/16	28	12	30/11/16
Optimizar cambios en base de datos	30/11/16	12	8	4/12/16
Realizar pruebas y corregir errores*	10/11/16	4	0	5/12/16
Integrar APIs finales	6/12/16	20	8	13/12/16
Realizar pruebas y corregir errores*	6/12/16	4	0	14/12/16
Realizar manual de usuario de la aplicación	15/12/16	8	4	18/12/16
Redactar conclusiones del trabajo	18/12/16	4	4	20/12/16
Finalizar memoria del trabajo	20/12/16	28	8	29/12/16
Grabar vídeo del trabajo	29/12/16	12	8	3/1/17
Entrega final	3/1/17	4	0	4/1/17
Tribunal virtual	16/1/17	16	0	20/1/17

Tabla 1.4.1: Planificación temporal

*Las tareas de “Realizar pruebas y corregir errores” se realizan a medida que se va programando la aplicación, por lo tanto, no suponen horas extras, sino que se contabilizan dentro de las tareas de la propia programación.

Así pues, teniendo en cuenta una inversión diaria de 4h al proyecto, éste supondrá una duración de 336h en días laborables y 120h en días festivos. Cabe destacar que es una aproximación ya que pueden surgir imprevistos que alarguen o acorten la duración de algunas tareas.

1.5 Breve resumen de productos obtenidos

Una vez finalizado el trabajo final de máster se obtendrá una aplicación completamente funcional en distintos dispositivos, así como una memoria recopilando todo el trabajo realizado para llegar al producto final.

Además, se proporcionará un manual de usuario y un manual técnico de la aplicación.

1.6 Breve descripción de los otros capítulos de la memoria

El resto de la memoria consta principalmente de los dos capítulos correspondientes a las entregas de este TFM: “diseño” e “implementación”.

En el capítulo de “diseño” se empezará por describir los usuarios potenciales del producto final, sus necesidades y objetivos, así como los contextos de uso y, consecuentemente, las funcionalidades del producto. Una vez definidos algunos posibles escenarios de uso y las funcionalidades a incorporar, se realizarán prototipos de baja y alta fidelidad.

En el capítulo de “implementación” se describirán las pantallas (en Android: *activities*) que forman la aplicación, así como las partes más importantes del código. También se mostrarán algunas pruebas, representando los escenarios de uso descritos en el capítulo anterior, para definir el correcto funcionamiento de la app. Cabe destacar que, tal y como se indica en el anexo, está disponible todo el código fuente en un fichero Zip adjuntado con esta misma memoria.

Finalmente, se incorporarán unas conclusiones donde se hablará sobre lo aprendido en este trabajo y las diferencias entre los objetivos inicialmente marcados y los obtenidos. También se mencionarán algunas líneas de trabajo futuro que han quedado pendientes.

2. Diseño

Una vez decidido el enfoque de este producto, se procederá a definir las personas que van a utilizarlo: ¿Quiénes son? ¿Qué hacen? ¿Cómo lo hacen? ¿Qué utilizan para llevar a cabo lo que hacen? ¿Dónde lo hacen y bajo qué circunstancias espaciotemporales? ¿Qué historia tiene la actividad que están llevando a cabo? Siendo las personas el foco de atención se podrá diseñar un producto mucho más completo, con unas funcionalidades y en unos dispositivos que satisfagan el mayor número de necesidades posibles de los usuarios.

2.1 Definición del usuario, sus necesidades y contextos de uso

Tal y como se ha mencionado anteriormente, es necesario estudiar a los usuarios potenciales del producto ya que permitirá determinar las necesidades de los mismos: cuáles son los requisitos de los usuarios que deben quedar plasmados y qué funcionalidades debe contemplar. Para recoger información de los posibles usuarios potenciales de la aplicación, se realizó una entrevista totalmente abierta (*focus group*) con un grupo de personas de edades de 15 a 82 años. En las entrevistas abiertas no se dispone de un guion de preguntas, sino de puntos que deben ser abordados por el usuario a lo largo de la entrevista. Después de introducir la entrevista, se realizó una primera pregunta como punto de partida: ¿Qué recuerdas del año 2000? Ninguno se acordaba de lo que hicieron en el año 2000 y, tras una discusión acerca de los temas que se pretendía indagar, se llegó a la conclusión de que a todos les interesaría dejar constancia de sus recuerdos. Puesto que la audiencia prevista es muy heterogénea, se trabajará con unos arquetipos llamados personas y sus *scenarios*.

José, médico:



Descripción de la persona

José es un médico de 61 años residente en Barcelona. Vive en un piso cerca de la playa con su mujer y sus dos hijos. Entre sus aficiones se encuentra la literatura, viajar y dar largos paseos. Le gusta estar a la moda disponiendo del último modelo de iPhone e iPad disponibles. Utiliza su *smartphone* como principal fuente de información y le gusta sacar provecho de todas sus funcionalidades. También lo utiliza para estar en contacto y para sacar fotografías de sus viajes, agrupándolas luego en carpetas con el nombre del destino. A menudo utiliza su iPad para leer noticias antes de irse a dormir. Le encantaría poder revivir eventos importantes como el nacimiento de sus hijos, su boda o alguno de los viajes que ha hecho. Es una persona reservada y no le gusta compartir sus experiencias en las redes sociales.

Descripción del escenario

Es un jueves a las 19:00h y, después del trabajo y de asistir a una función que hacía su hijo menor en el colegio, José está tumbado en el sofá y no le apetece

ver la televisión. Tiene un rato antes de que la cena esté lista así que decide revisar las fotografías que ha tomado de la función con su iPhone. Una vez revisadas, elige la que más le gusta y, mediante la opción de compartir, la sube a la aplicación asignándola a este mismo jueves. No utiliza filtros porque le parece bien tal y como está la foto ni la comparte en las redes sociales porque no le interesa que sus contactos sepan mucho de su vida privada. Añade una pequeña descripción para saber que esa fotografía corresponde a la función de su hijo.

Crédito foto: <http://www.iese-es.org/blog/blog/manfred-benzing-ha-pasado-a-la-presencia-del-senor/>

Sandra, estudiante:



Descripción de la persona

Sandra es una joven estudiante de 22 años, nacida en Madrid y ha decidido pasar su último cuatrimestre de la universidad en Londres de erasmus. Está estudiando ingeniería y, a pesar de dedicar muchas horas de estudio, le gusta salir con sus amigas e ir al gimnasio para mantenerse en forma. Es una persona abierta que ha hecho rápidamente muchas amigas, compartiendo piso con 3 de ellas. A Sandra le encanta disfrutar de los pequeños momentos y siempre prefiere las experiencias a lo material. Dispone de un *smartphone*, una *tablet*, un Mac y recientemente se ha comprado unas gafas VR ya que le fascina la idea de la realidad virtual y considera que es el futuro. Sandra es una chica presumida y le gusta compartir un montón de experiencias en las redes sociales.

Descripción del escenario

Es un lunes soleado a las 16:30h y Sandra ha salido con sus compañeras de piso a tomar un café en la terraza de su cafetería favorita, cerca de Picadilly Circus. Sandra ha pedido un café con leche, que se lo sirven con un corazón dibujado en la espuma del mismo, así que decide immortalizar el momento. Utiliza la aplicación para asignar la fotografía a este lunes de marzo al igual que hace con muchas experiencias que vive, ya que considera que estos meses de erasmus no los volverá a vivir. Le pone un filtro a la fotografía para realzar el día soleado que está haciendo y utiliza la opción de compartir en Instagram y Facebook para que sus contactos puedan verla. Ha añadido una pequeña descripción con una frase en inglés y varios tags.

Crédito foto: http://duvapropiedades.com/uploads/slider/20150615125555_estudiante.png

Francisca, jubilada:



Descripción de la persona

Francisca es una mujer de 82 años y vive con su marido en Ripollet. Francisca está muy mayor para hacer según qué actividades así que salir de paseo, ver la televisión y limpiar la casa es lo que más ocupa su tiempo. Francisca tiene tres hijos y una hija, todos casados y con sus respectivos hijos. Una de sus aficiones es ver los álbumes de fotos que tiene Francisca de sus hijos y nietos para recordar momentos del pasado. Francisca acostumbra a seguir la misma rutina así que le produce una enorme alegría cuando sus hijos o nietos la llaman por teléfono y cuando se reúnen todos juntos los domingos para la comida familiar. No dispone de internet en casa ni tiene ningún dispositivo ya que no sabe cómo utilizarlos y considera que no les sacaría provecho.

Descripción del escenario

Es un domingo a las 16:00h y Francisca se encuentra en su casa, junto con su marido, sus hijos y sus nietos. Han acabado la comida familiar y ya han tomado el café, así que están todos juntos en el salón viendo la televisión y hablando entre ellos. El nieto más mayor ha estado recientemente de vacaciones con su pareja y se dispone a mostrar las fotografías y videos que ha tomado del viaje. El nieto se sienta al lado de sus abuelos con su *tablet* y los demás familiares se sitúan alrededor. Abre la aplicación y Francisca ve un calendario con fotografías de su nieto con su pareja en cada uno de los días que ha estado fuera. Francisca contempla, seleccionando en el día deseado, lo que visitó su nieto en ese día con descripciones para entender de qué monumento o palacio se trata.

Crédito foto: <http://i.mkt.lu/cont/232553/280/240/mujer.jpg>

Estas han sido las fichas de persona y *escenario* de tres diferentes tipos de personas. Como se puede observar, los posibles usuarios de la aplicación son muchos así que el público objetivo es universal. Se trata de un producto de difusión general ya que los usuarios tienen un rango de intereses variables y tienen diferentes niveles de conocimiento del entorno informático. Como la aplicación puede ser utilizada por personas de edad avanzada, como por ejemplo Francisca, se tendrá que priorizar los factores de accesibilidad. Debe ser una aplicación que satisfaga todas las necesidades de los usuarios y que sea a la vez fácil e intuitiva.

2.2 Definición de las funciones

Una vez definidos los usuarios potenciales del producto y sus necesidades, se procederá a hacer un listado con todas las funciones del mismo. Para empezar, pueden definirse tres tipos de personas diferentes viendo las fichas del apartado anterior. Sandra sería una persona focal, el principal tipo de usuario al que está dirigido el producto. José sería una persona secundaria, un usuario que también utiliza el producto de manera

satisfactoria pero que, en caso de conflicto de intereses, se priorizan las necesidades de la persona focal. Y finalmente, Francisca sería una persona involucrada, que no utiliza el producto pero que se ve afectada por sus resultados.

Así pues, según las necesidades de los arquetipos descritos anteriormente, se intentará sintetizar al máximo todas las funciones de la aplicación.

Funciones principales:

- Publicar imágenes, fotografías, videos o notas asignándolos a un día, semana, mes o año.
- Ver un calendario con todas las publicaciones hechas a lo largo de la vida.

Nota: También se pueden publicar imágenes, fotografías y videos hechos en 360°.

Funciones secundarias:

- Recibir instrucciones de uso de la aplicación y de la opción realidad virtual.
- Ver las imágenes, fotografías y videos que se hayan publicado en 360° en realidad virtual.
- Opción de editar las imágenes y fotografías a la hora de publicarlas (aplicar filtros, brillo, contraste, recortes, etc.)
- Opción de añadir una descripción a la imagen, fotografía y video.
- Opción de añadir ubicación para tener constancia de a qué lugar corresponde lo que se publica.
- Opción de compartir lo que se publica en alguna de las redes sociales más utilizadas: *Facebook*, *Instagram* y *Twitter*.
- Opción de ver en modo presentación un resumen de lo ocurrido a lo largo del año seleccionado.
- Opción de descargar los resúmenes del año seleccionado.
- Opción de cambiar el formato de la fuente utilizada en las publicaciones modo texto.

A continuación, se comprobará que las funciones de la aplicación cumplan los requisitos de los arquetipos de usuarios definidos en el apartado anterior, comenzando por la persona focal: Sandra.

Sandra disponía de un *smartphone*, una *tablet*, un PC y unas gafas de realidad virtual. Al igual que Sandra, hoy en día millones de usuarios disponen de un *smartphone*, una *tablet* y/o un PC. En cambio, las gafas VR es un dispositivo muy novedoso que cada vez va ganando más popularidad entre los usuarios. En la elección de los dispositivos que se utilizarán para el producto, se ha optado focalizar el diseño para *smartphone* de momento. Esto no significa que no se puedan añadir otros dispositivos como la *tablet*, las gafas VR, el ordenador portátil o la televisión al ecosistema de múltiples dispositivos conectados. Sin

embargo, para simplificar el proyecto se tendrá como foco el desarrollo en los dispositivos mencionado anteriormente. No obstante, se pretende realizar un diseño que permita relacionar los distintos dispositivos y crear cambios dinámicos para poder ofrecer al usuario una mejor experiencia, en vez de ofrecer accesos independientes en cada dispositivo.

Volviendo a la persona: Sandra ya dispone de unas gafas de realidad virtual ya que considera que impactará de manera notable en el futuro de la tecnología. Es probable que hoy en día no se saque mucho provecho a la realidad virtual ya que de momento tiene un mercado muy pequeño. Sin embargo, se utilizará este dispositivo para poder ver en el *smartphone* las fotos y videos en 360° ya que es algo muy novedoso y que destaca en comparación a posibles productos similares. La aplicación le permitirá sacar provecho de su nuevo dispositivo visualizando las fotos y videos que tome en 360° y revivir así el momento una y otra vez. También podrá inmortalizar muchos más momentos de su vida como hace ahora que está de erasmus, añadirles descripciones y ubicaciones. Así, en el futuro podrá ver que un lunes de marzo de 2016 se tomó un café con sus amigas en un café cerca de Picadilly Circus. La aplicación también le permite editar sus fotografías y compartirlas en las redes sociales para que sus amigos las vean.

En cuanto a José, él también podrá publicar sus fotografías y videos tomadas en sus viajes y ponerles descripciones y ubicaciones. Podrá captar cualquier momento de su vida e irlos publicando en alguna fecha del calendario. De esta manera, viendo los resúmenes del año que proporcionan la aplicación, podrá disfrutar de todo el recorrido de su vida. Podrá disfrutarlo tanto en su *smartphone* como en su *tablet* si le resulta más cómodo, aunque por el momento este trabajo consiste en el desarrollo para *smartphones*. Quizás el tema de la realidad virtual no le pueda hacer revivir eventos ya pasados como su boda o el nacimiento de sus hijos. Sin embargo, le permitirá revivir eventos del futuro. En un futuro, José podrá grabar en 360° el nacimiento de sus nietos por ejemplo y visualizarlos mediante realidad virtual para tener la sensación de estar ahí una y otra vez. Si tiene cualquier duda sobre su funcionamiento, la aplicación le dará instrucciones de uso.

Francisca en cambio es una persona involucrada y el provecho que le pueda sacar a la aplicación es lo que le muestren los clientes de la aplicación de su alrededor. Los familiares que utilicen la aplicación podrán mostrarle el recorrido de sus vidas incluso en realidad virtual, cosa que seguro fascinará a Francisca. Probablemente, Francisca se anime a comprar un *smartphone* sencillo para ella y utilice la aplicación para recopilar las fotos de sus álbumes, visualizando así el recorrido de su propia vida.

En definitiva, la aplicación puede ser utilizada por una gran variedad de usuarios y ésta cumplirá sus requisitos. Es probable que las funciones de la realidad virtual no sean tan utilizadas de momento, pero pensando en el futuro, quizás merezca la pena su desarrollo.

Antes de describir el diseño de la aplicación, así como su estructura e interacción, se deben tener muy presentes los *scenarios* vistos anteriormente. Por lo tanto, se muestran a continuación unos diagramas de los casos de uso de cada una de las personas:

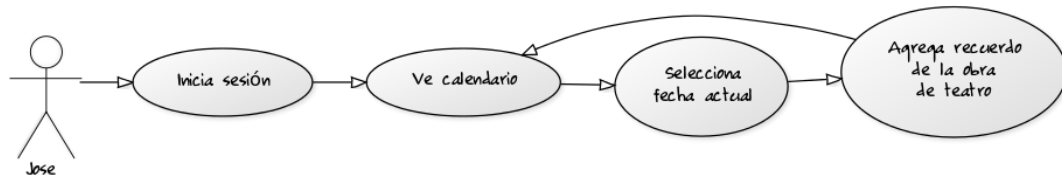


Figura 2.2.1 Diagrama de caso de uso Jose



Figura 2.2.2 Diagrama de caso de uso Sandra

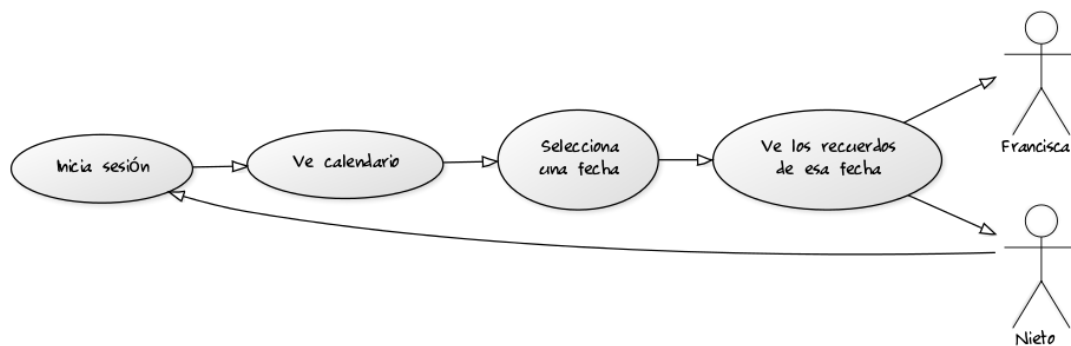


Figura 2.2.3 Diagrama de caso de uso Francisca

2.3 Diseño conceptual

Con las necesidades de los usuarios y sus *scenarios* descritos en el apartado anterior, se han definido unas ciertas funciones que debía tener la aplicación. En este apartado se avanzará un paso más y se procurará empezar a conceptualizar la estructura de la aplicación y los flujos de interacción basándolos en las funcionalidades que formarán la app.

Árbol de navegación:

En un inicio, se ha realizado un árbol de navegación para poder tener en cuenta las pantallas que se iban a necesitar y establecer una jerarquía de pantallas. A continuación, se muestra en formato pequeño, el árbol de navegación de la app con TODAS las pantallas que tiene actualmente. Cabe destacar que este es un diseño inicial y que, a lo largo del desarrollo de nuevas versiones, puedan aparecer nuevas pantallas.

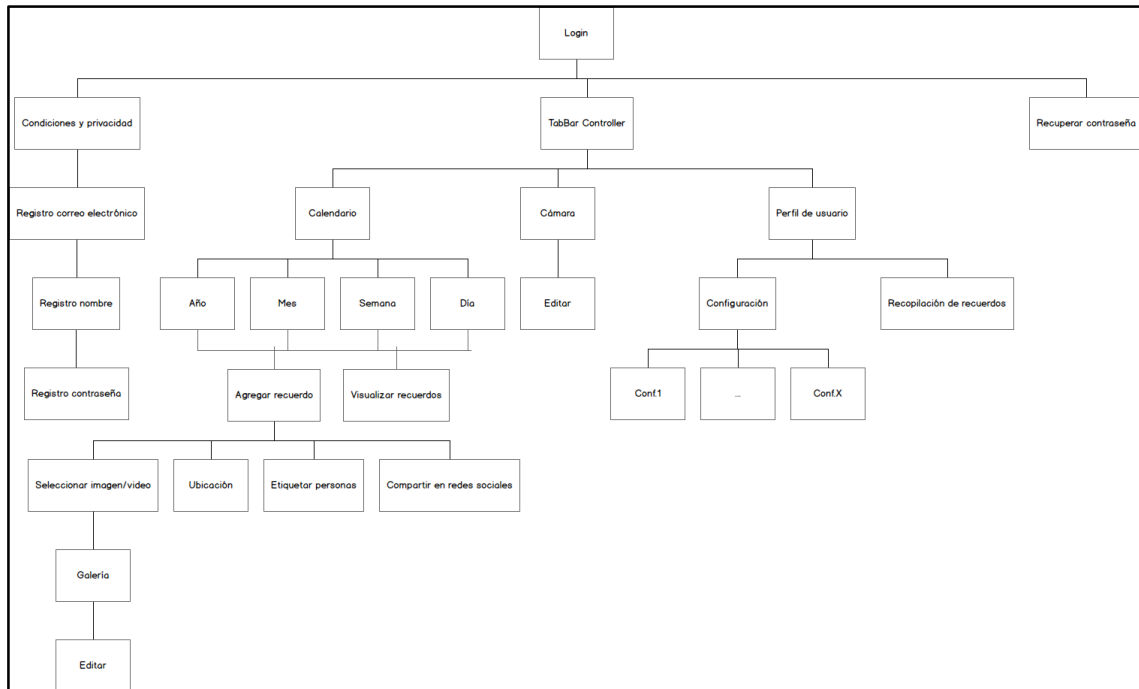


Figura 2.3.1 Árbol de navegación de la app

Con este árbol, se pueden definir los caminos a recorrer por la app y establecer una jerarquía. Una vez definida la jerarquía de las pantallas y la navegación entre ellas, se ha procedido a la creación de un primer diseño de las pantallas más importantes.

Primeros esbozos:

A continuación, se muestra en formato pequeño, los primeros esbozos de la app a papel y lápiz con algunas notas. Se muestran los primeros diseños para las pantallas más importantes de la aplicación. En el árbol de navegación se pueden ver tres caminos principales a seguir a partir de la primera pantalla de *Log In*. Pues bien, en los esbozos se muestran los dos primeros caminos principales que son los que tienen más relevancia para el usuario.

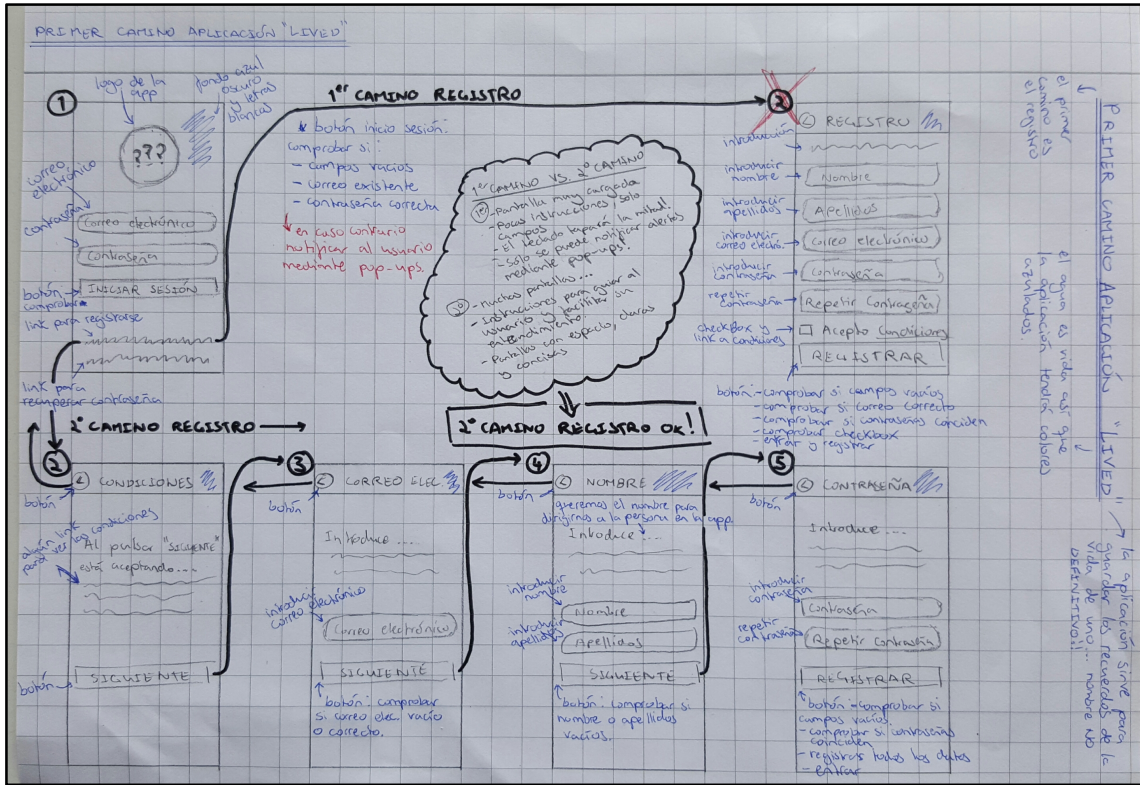


Figura 2.3.2 Esbozo del primer camino de la app

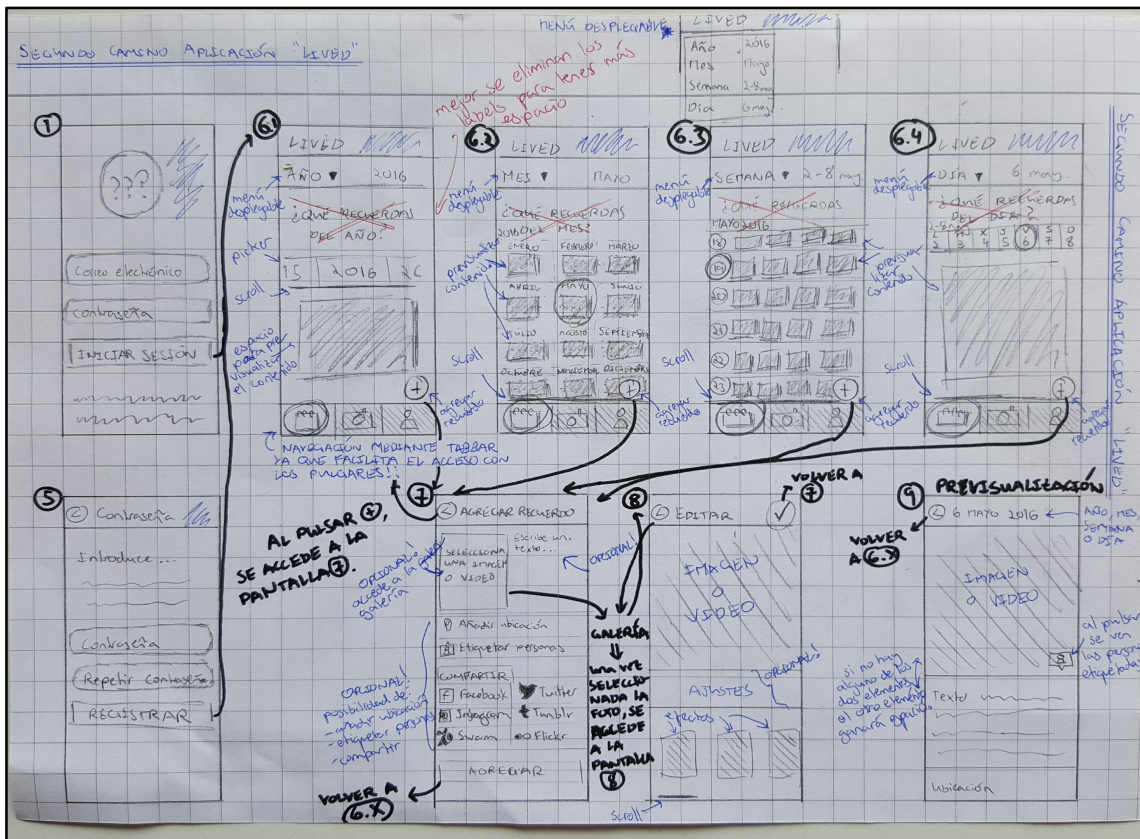


Figura 2.3.3 Esbozo del segundo camino de la app

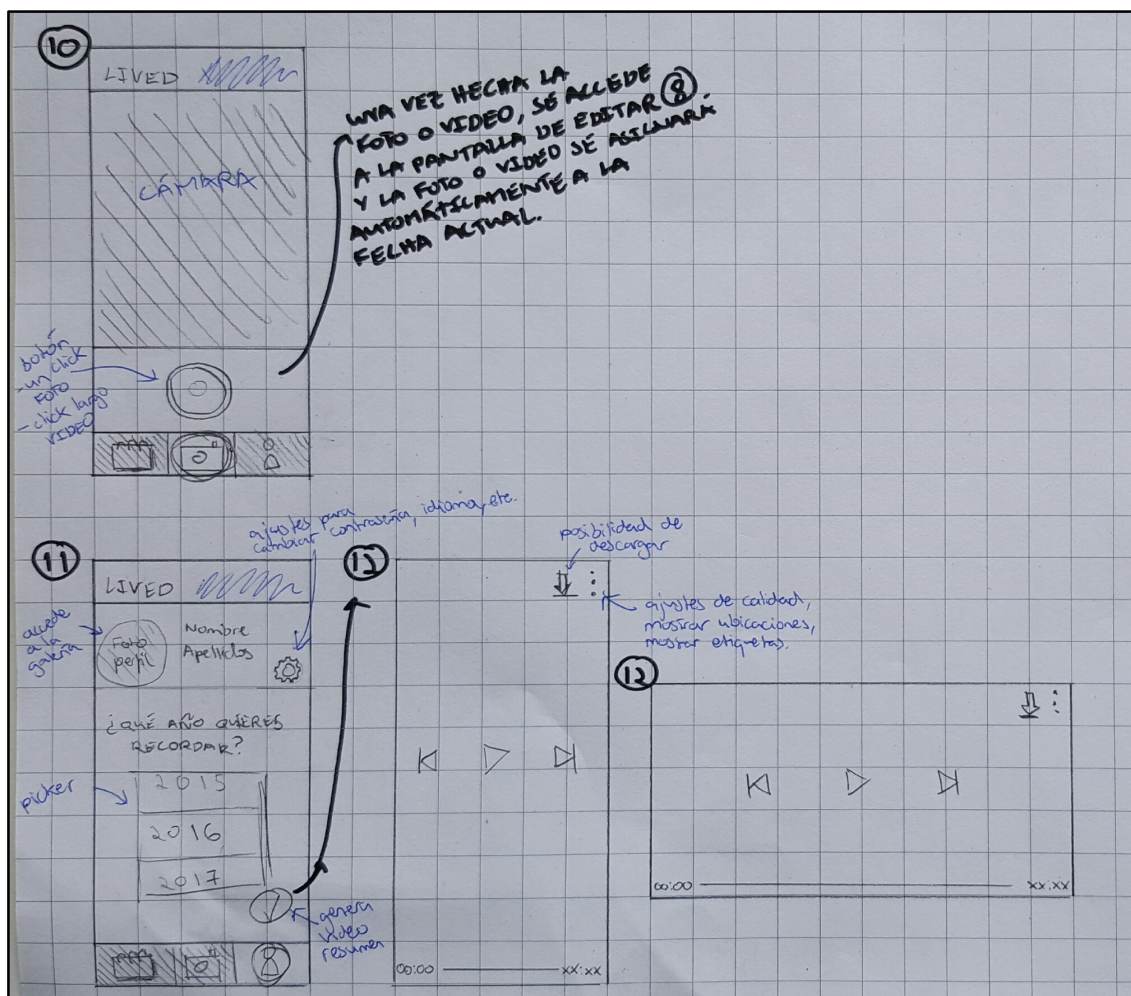


Figura 2.3.4 Esbozo del segundo camino de la app

A la hora de pensar en un primer diseño para la aplicación, se ha tomado como referencia el diseño de aplicaciones mundialmente conocidas y de un éxito incomparable como son las aplicaciones de las redes sociales Instagram y Facebook. Si bien es cierto que los tonos azulados también están presentes en estas dos aplicaciones, para esta aplicación se ha decidido utilizar por su referencia al agua, principal fuente de vida.

Esta aplicación trata sobre la vida de las personas, sus recuerdos. Por lo tanto, el nombre debe representar algo por el estilo y sus colores también. El nombre aún no está definido del todo, pero de momento se utilizará el nombre *Lived* que significa “vivido” en inglés.

Estos fueron los primeros diseños realizados para el producto. Sin embargo, a la hora de realizar un primer prototipo se cambió ligeramente alguno de los diseños como se mostrará a continuación.

Prototipo de baja fidelidad:

Una vez definido el diseño de las pantallas, se realizó un primer prototipo (de baja fidelidad) con la herramienta *balsamiq*^[1]. Para cada pantalla se ha adjuntado una nota con los elementos que contendrá la misma.



Figura 2.3.5 Mockup del primer camino de la app

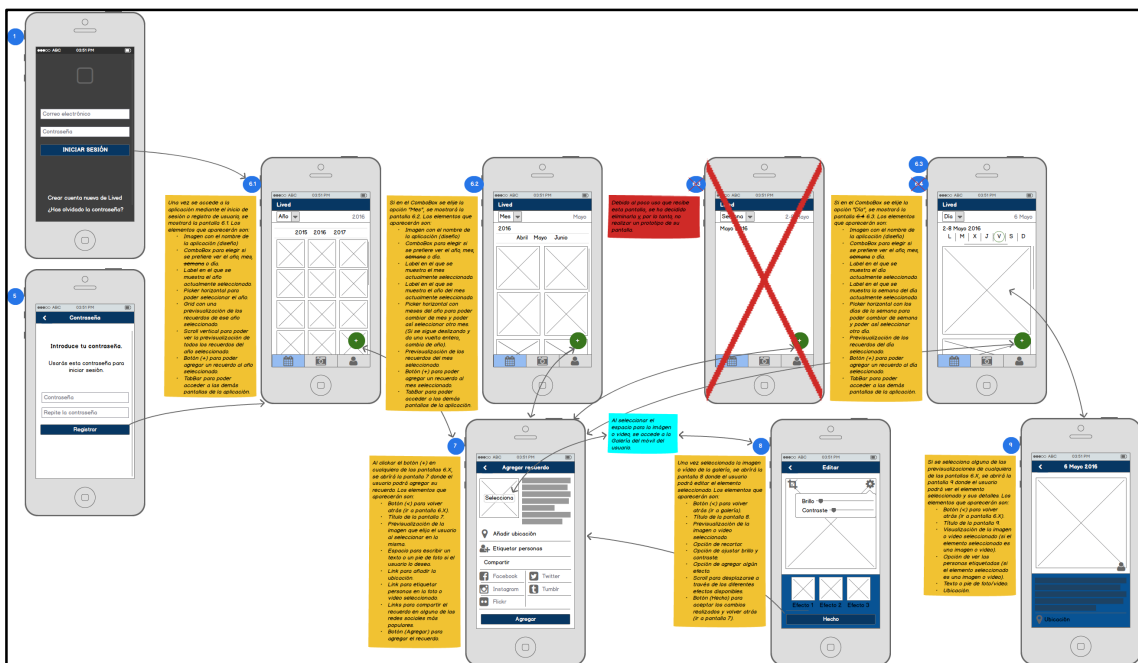


Figura 2.3.6 Mockup del segundo camino de la app

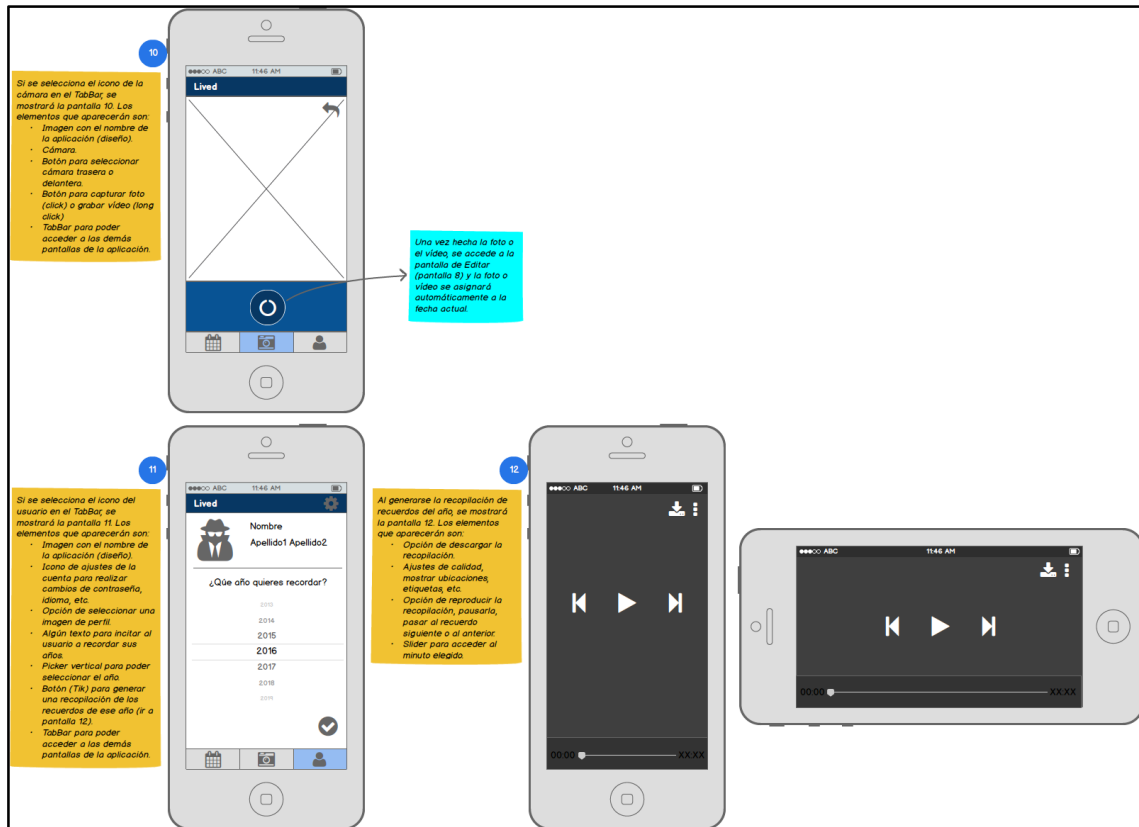


Figura 2.3.7 Mockup del segundo camino de la app

Como se puede observar, algunas de las pantallas han sufrido ligeras modificaciones respecto a los primeros esbozos en papel y lápiz. También mencionar que es un prototipo de baja fidelidad, ha sufrido cambios respecto a los esbozos y, por lo tanto, no se puede descartar que sufra más cambios en futuros rediseños.

Diseños alternativos:

En la figura 2.3.2, se puede observar un diseño alternativo para el registro de usuarios. En un primer momento se planteó hacer todo el registro del usuario en una sola pantalla. Sin embargo, se consideró que tener una pantalla llena de *TextFields* ocupaba mucho espacio y, con la aparición de un teclado, no sería cómoda la navegación entre los mismos. Hay que tener en cuenta que los usuarios de esta aplicación están comprendidos en un rango de edades muy amplios y, por lo tanto, el entendimiento de la aplicación debe ser lo más perfecto posible. Así pues, se decidió eliminar la pantalla 2 inicialmente definida y se diseñaron las pantallas 2, 3, 4 y 5 para realizar un registro de usuario paso a paso, con instrucciones en cada campo para guiar al usuario.

Otro diseño alternativo para una pantalla es la de la 6.1, la del calendario anual en la que se pueden agregar recuerdos a cada año seleccionado y visualizar los mismos.

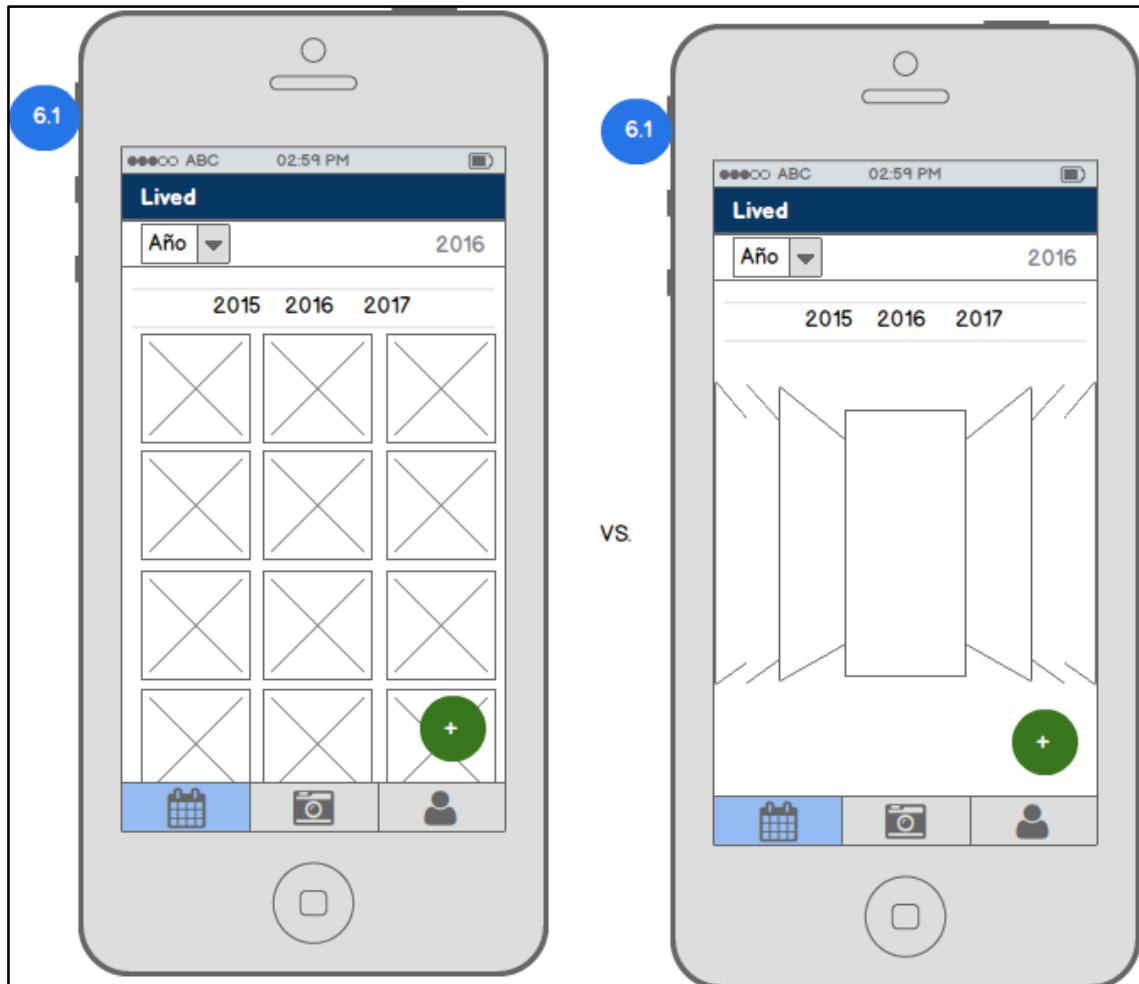


Figura 2.3.8 Diseño alternativo para la pantalla 6.1

Como se puede observar, la diferencia está en la previsualización de los recuerdos del año. En uno se ven todos los recuerdos juntos en forma de *Grid* mientras que en otro se ven mediante un *CoverFlow*. La opción elegida finalmente fue la de la visualización en forma de *Grid* ya que el *CoverFlow* podría provocar una frustración al usuario al deslizar los elementos y estar saltándose los que le interesan sin querer. También supone una confusión ya que la selección del año se realiza mediante un *Swipe* horizontal en el *Picker* que se encuentra en la parte superior. Si se realiza también un *Swipe* horizontal en el *CoverFlow* para cambiar de recuerdo, puede confundir al usuario y no saber si está cambiando de año o de recuerdo.

Finalmente, ver los elementos en forma de *Grid* permiten al usuario una visión más global de los recuerdos y puede seleccionar el que desee para verlo de forma ampliada.

Siguiendo el mismo criterio, se han diseñado de forma parecida las pantallas 6.2 y 6.3.

Finalmente, para este diseño de la aplicación se tuvo en cuenta que cumpliera las funcionalidades descritas en el apartado anterior. A parte, en todo momento se pedía la opinión de algunos clientes potenciales.

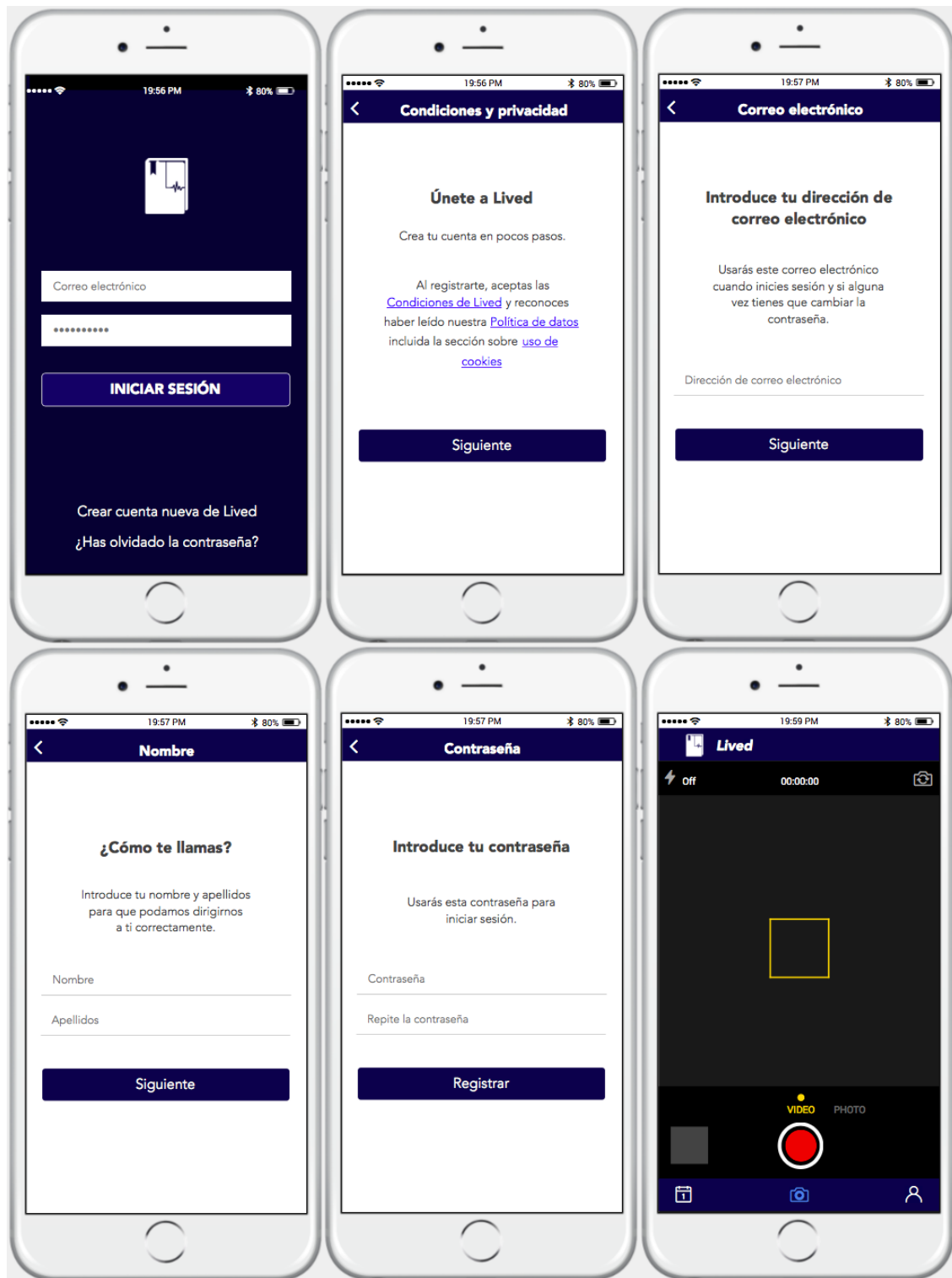
Buscaban un diseño sencillo, fácilmente accesible con los pulgares (p. ej. navegación mediante *TabBar*) y estiloso. Gracias a ellos se tomaron decisiones como el diseño alternativo a la hora de registrarse descrito en el apartado anterior. También se decidió eliminar la pantalla 6.3 inicial que mostraba el calendario semanal. El motivo de ello fue que se pidió a una serie de usuarios que asignaran una serie de recuerdos sencillos. Todos ellos asignaban los recuerdos de los que conocían la fecha a días en concretos (nacimientos entre otros). También asignaban otros recuerdos a años (como por ejemplo eventos importantes). Incluso asignaban recuerdos a algunos meses (como por ejemplo festivos o comienzos en un nuevo empleo). Sin embargo, ninguno fue capaz de asignar un recuerdo a la semana XX del año XXXX. Si bien es cierto que algunos podían definir si el recuerdo había sucedido a principios de mes o a finales, ninguno podía definir en qué semana de la 1 a la 52 que tiene un año. Así pues, se decidió eliminar esta pantalla que podría ser inútil para los usuarios.

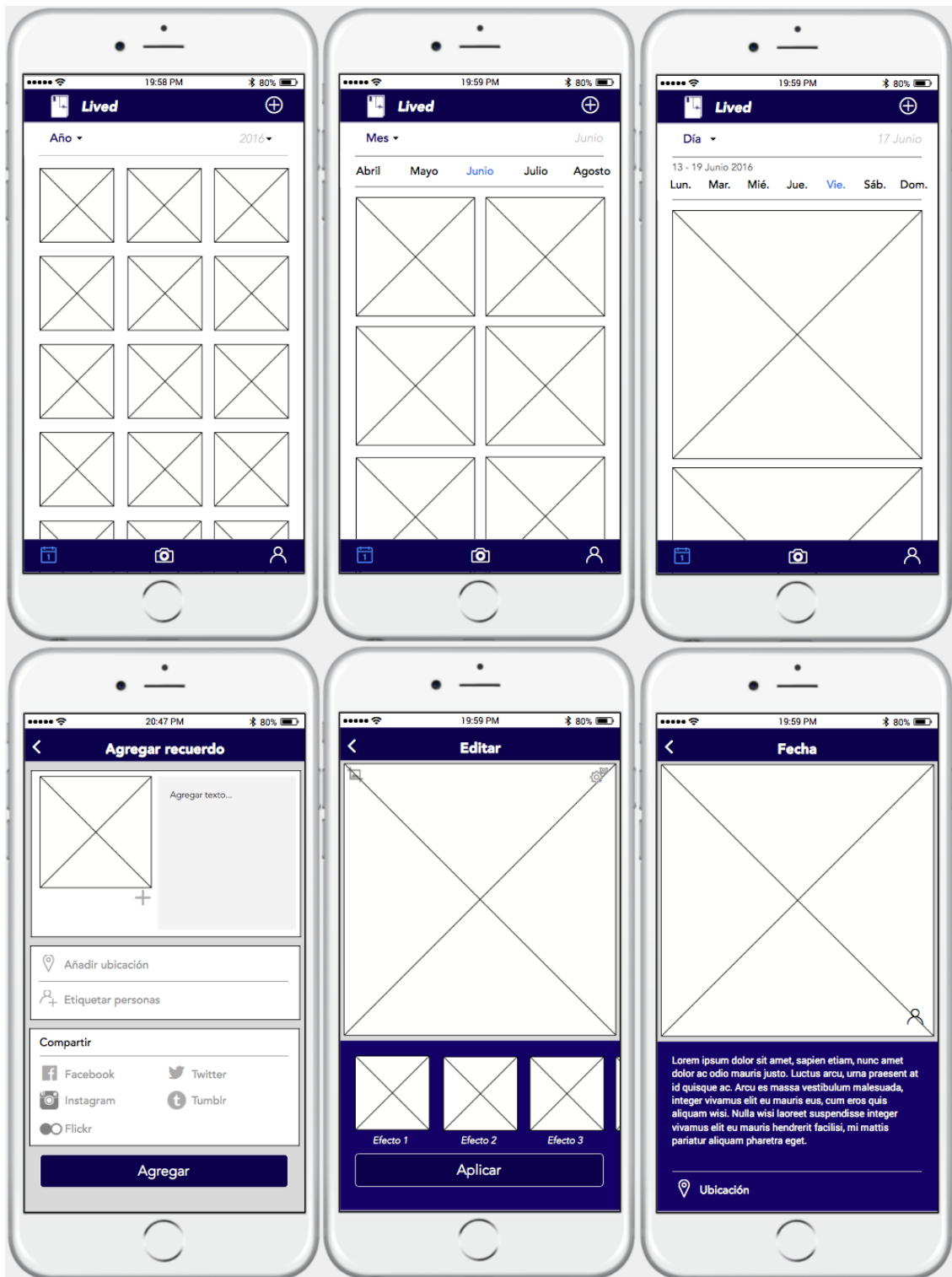
Con todo esto, se realizó un primer prototipo de las pantallas más importantes de la aplicación que seguramente sufrirán más cambios en un futuro.

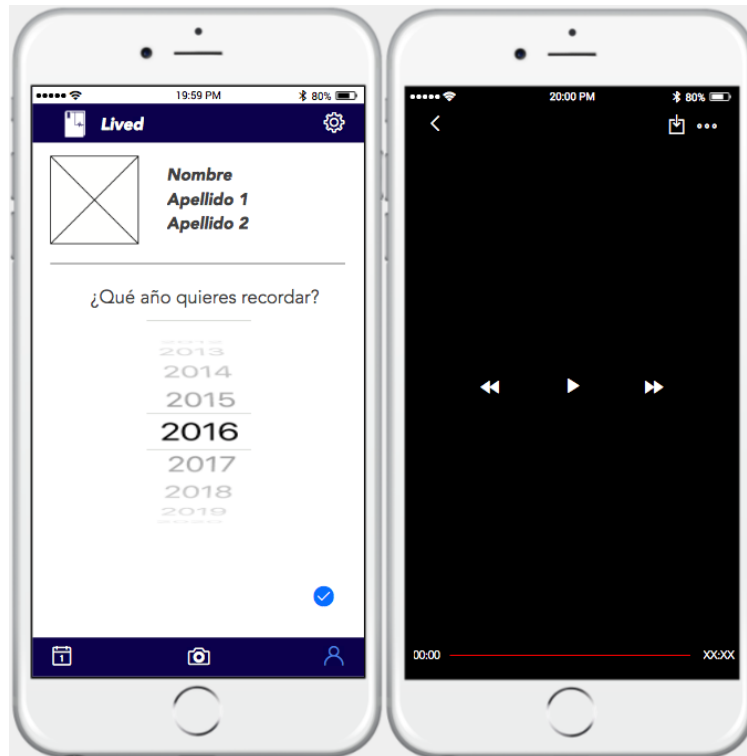
2.4 Prototipado

Para el prototipo de alta fidelidad, se han creado las pantallas descritas en el apartado anterior para ver cómo será la aplicación y detectar posibles errores de concepto o diseño. Cabe destacar que el prototipo sirve para comprobar que las ideas inicialmente definidas tienen validez, pero no se trata de construir una réplica exacta de la aplicación final. Por lo tanto, hay algunas funcionalidades que no funcionan y algunas pantallas que no se han diseñado en el prototipo, como por ejemplo la pantalla de ajustes de usuario.

A continuación, se presentan las capturas de pantalla de las pantallas creadas, cuya interacción entre ellas se puede observar mediante las flechas visibles en el diseño visto anteriormente.







Figuras 2.4.1 a 2.4.14 Pantallas del prototipo de alta fidelidad

Para realizar el prototipo en alta definición se ha utilizado la herramienta de prototipado **Justinmind**^[21]. Siendo ésta la primera vez que la utilizaba, se invirtió un tiempo en aprender cómo funcionaba la plataforma mediante videos tutoriales de la misma web.

Una vez conocidos los distintos elementos de *Justinmind* y sus posibilidades, se estudió el diseño creado y se intentó elaborar el mismo diseño.

Prácticamente todo el diseño era bastante correcto así que no ha sufrido grandes cambios respecto al primero planteado. Sin embargo, a la hora de realizar el prototipo se cambiaron ciertos elementos:

El primer cambio fue, en la pantalla del calendario “Año”, se sustituyó un *picker* con todos los años por un *ComboBox*. El motivo de este cambio fue que con el *picker* se ve el año actual en el centro, a la derecha los años futuros y a la izquierda los años pasados. Pues bien, se concluyó que no tenía mucho sentido definir años del futuro para agregar recuerdos, así que el *picker* quedaba desequilibrado. Los demás cambios fueron simplemente en cuanto a tipo de letra, color de letra, etc.

Así pues, después de ir retocando el diseño mediante la pantalla de prototipado y probando su funcionamiento, el prototipo definitivo es el que se puede ver anteriormente.

2.5 Evaluación

Una vez definido el diseño del prototipo, se puede comprobar que éste permite cubrir todas las funcionalidades que se han descrito

anteriormente, dando así por finalizada la fase de DCU (Diseño Centrado en el Usuario).

No obstante, también se debe definir la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para gestionar los diferentes procesos y la estructura de la API que servirá para realizar las peticiones enviados al servidor desde un cliente.

Así pues, se muestra un diagrama UML^[3] correspondiente al diseño de la base de datos:

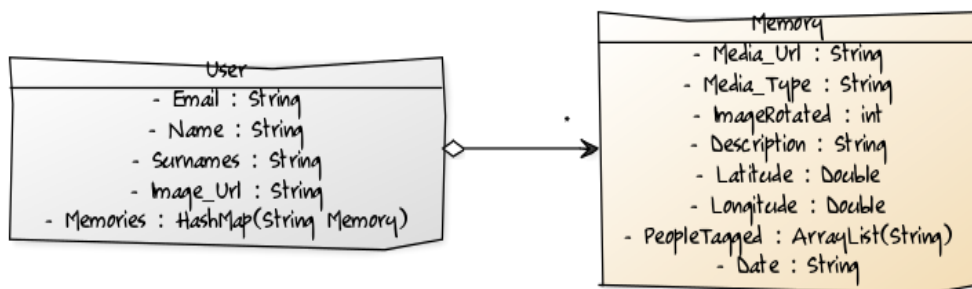


Figura 2.5.1 Diagrama UML del diseño de la base de datos

Como se puede observar, en la base de datos quedarán registrados todos los usuarios y éstos tendrán una serie de recuerdos. Así mismo, todos los recuerdos registrados en la base de datos pertenecen a un único usuario.

A parte de una base de datos, se programarán unos modelos de “Usuario” y “Recuerdo” para poder ser invocados por los métodos necesarios para completar todas las funcionalidades de la aplicación. Éstos métodos pueden ser la gestión de inicio de sesión, recibir los recuerdos de la base de datos, poder modificarlos, guardar recuerdos, compartirlos, etc.

Finalmente, se ha creado un diagrama explicativo de la arquitectura del sistema siguiendo el paradigma MVC (*Model-View-Controller*).

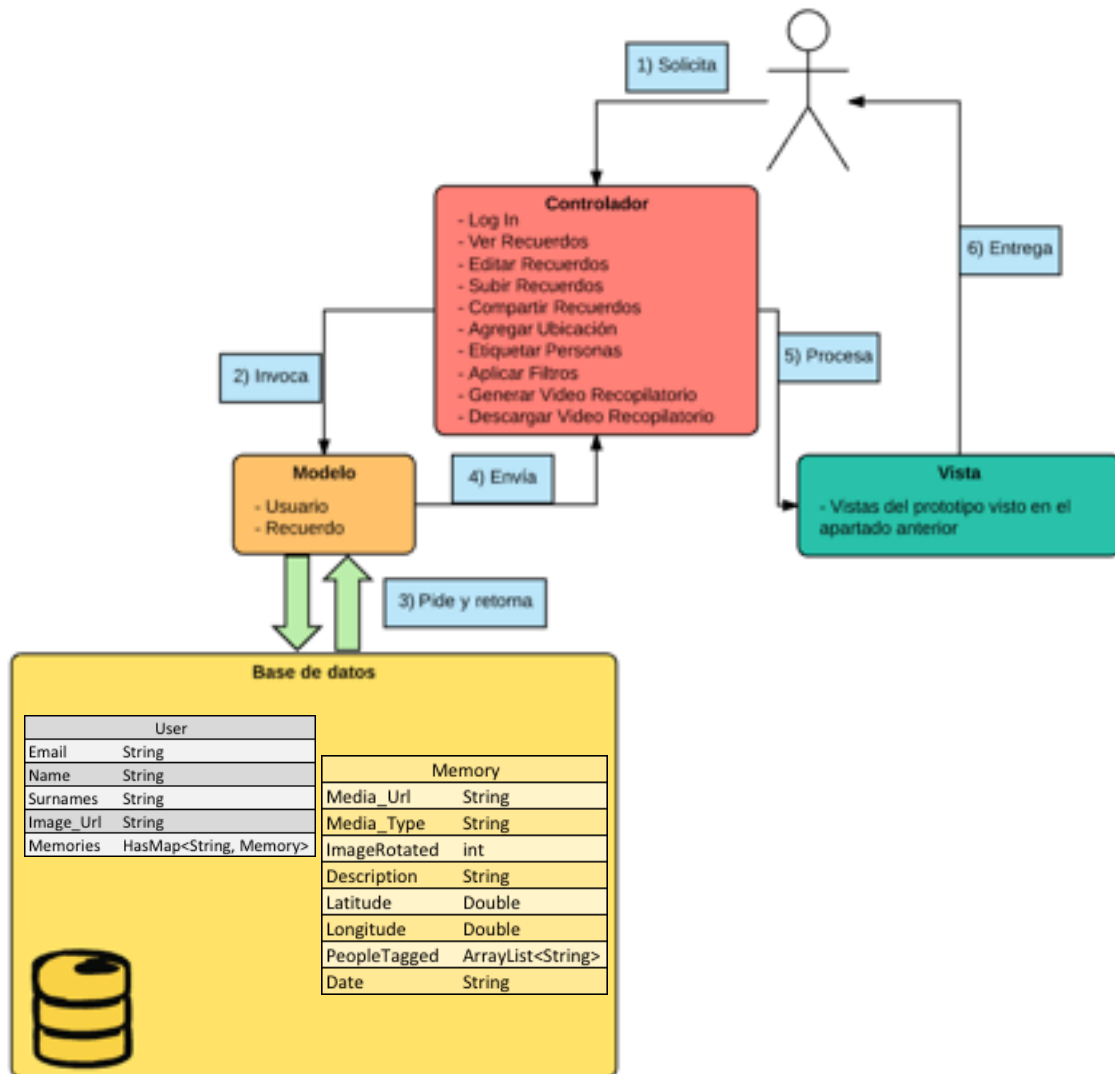


Figura 2.5.2 Diagrama explicativo de la arquitectura del sistema

El usuario realizará una serie de solicitudes que invocarán a unos modelos. Éstos interactúan con la base de datos para enviar las respuestas a las solicitudes del usuario. Los resultados de las mismas, se presentarán en la vista para que el usuario pueda ver toda la información.

En definitiva, así concluye la fase de diseño técnico de la aplicación y, aunque no están definidos exactamente los métodos y clases que se utilizarán, se dispone ya de una base para poder empezar la implementación del producto.

3. Implementación

Partiendo del árbol de navegación visto en el capítulo 2, el prototipo de alta fidelidad y la arquitectura del sistema; se ha programado el producto final con Android Studio y con ayuda de algunas librerías a las que se harán referencia más adelante.

Para explicar de manera más entendible el proceso de la implementación del trabajo, se recorrerá cada una de las pantallas (desde ahora *activities*) y se discutirán las decisiones tomadas en su desarrollo. También se justificarán los cambios respecto al prototipo de alta fidelidad y se comentarán los resultados logrados.

3.1 LoginActivity

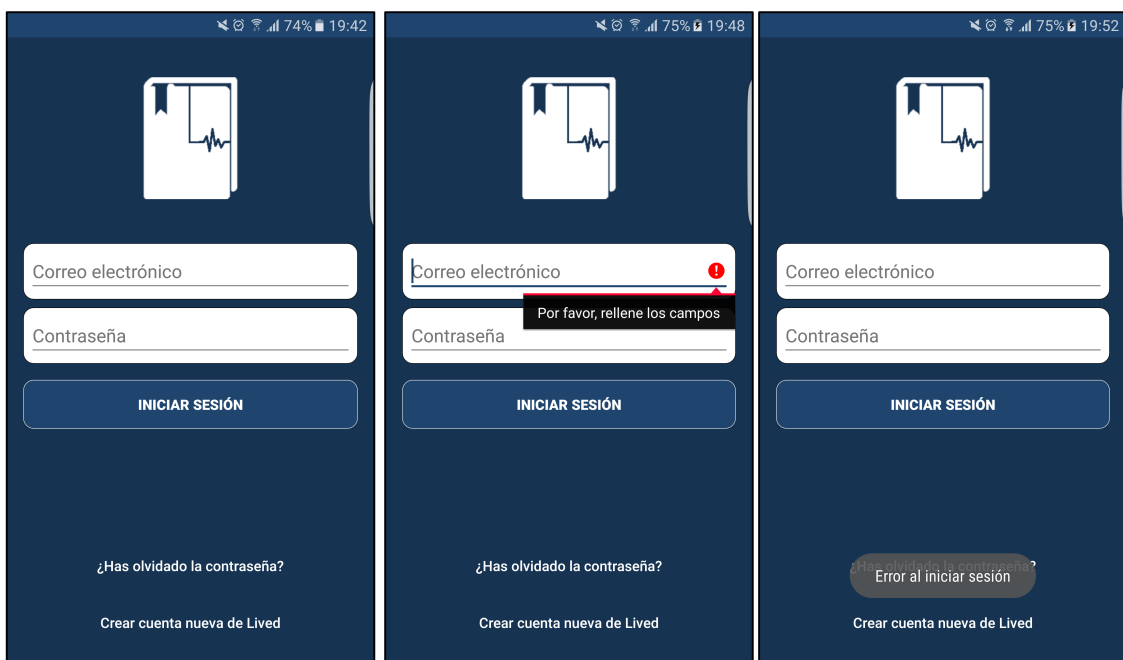


Figura 3.1.1 LoginActivity

Esta es la primera pantalla que se ve al iniciar la aplicación. Como se puede observar, el diseño es prácticamente idéntico al del prototipo a excepción de que el icono es ligeramente mayor y los bordes de los *EditTexts* y del botón son redondeados.

Desde esta pantalla, al hacer click en el botón, el usuario simplemente llama al método `signInWithEmailAndPassword` de `FirebaseAuth`^[4] y, si los datos son correctos, accede a la `MainActivity`. Si no son correctos aparecerá un mensaje, así como si no se ha introducido ningún correo electrónico y/o contraseña.

Además de iniciar sesión, el usuario puede solicitar cambiar la contraseña si no la recuerda. Se llama al método `sendPasswordResetEmail`^[5] de `FirebaseAuth` y desde `firebase` se enviará un correo electrónico con un enlace para cambiar la contraseña.

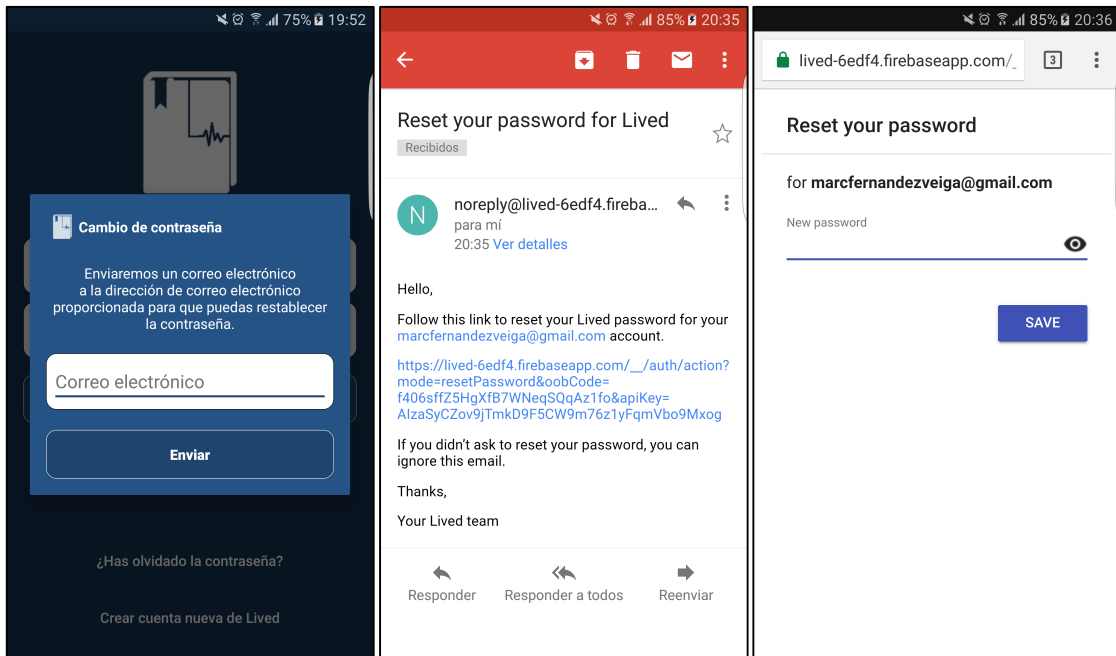


Figura 3.1.2 Solicitar cambio de contraseña

Si en cambio se trata de un usuario nuevo, puede seleccionar “Crear cuenta nueva de Lived” e iniciará su registro a la app, comenzando por la pantalla *ConditionsActivity*.

3.2 ConditionsActivity

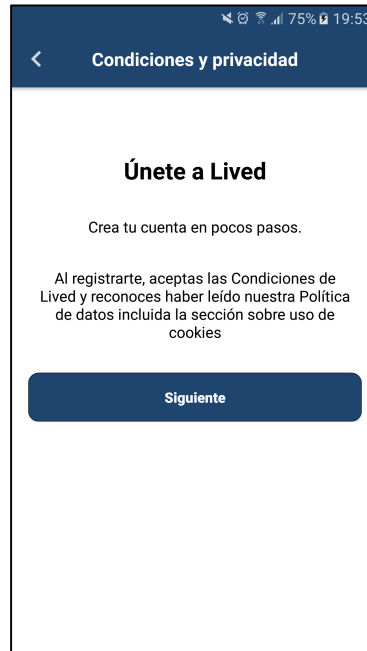


Figura 3.2.1 *ConditionsActivity*

En esta pantalla se muestra un mensaje para advertir al usuario de que acepta las condiciones de la aplicación. Si está dispuesto, al seleccionar el botón “Siguiente”, navegará a *EmailActivity*.

3.3 EmailActivity

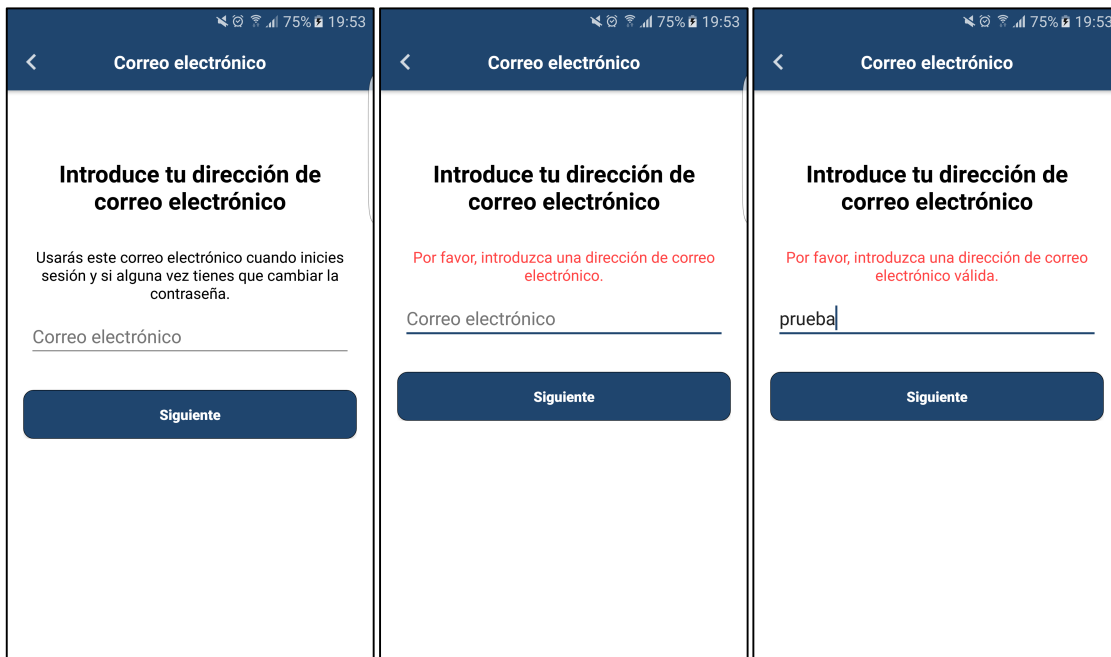


Figura 3.3.1 EmailActivity 1

En esta *activity*, el usuario debe definir la cuenta de correo electrónico que desea utilizar para su cuenta de Lived. Este correo será el que utilizará para iniciar sesión o cambiar la contraseña de su cuenta.

Como se puede observar, aquí se gestiona que la cuenta sea una cuenta válida, comprobando que el valor introducido sea correcto. Si la cuenta no es correcta, se muestran algunos mensajes de error para indicar al usuario qué está haciendo mal.

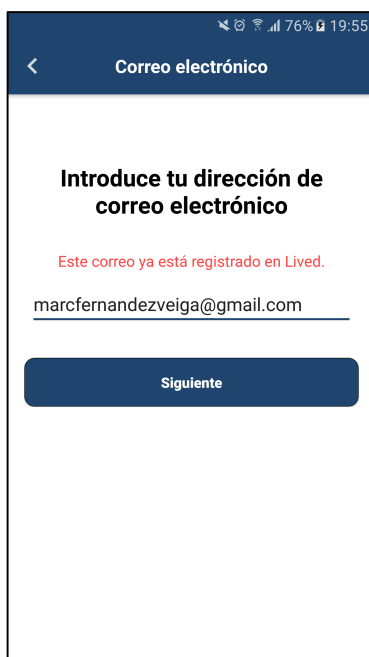


Figura 3.3.2 EmailActivity 2

Además, otro aspecto clave a la hora de elegir una cuenta de correo es detectar si la cuenta elegida está ya registrada en la aplicación o no. Esto se logra gracias al método `createUserWithEmailAndPassword` de `FirebaseAuth`^[4]. En definitiva, lo que se está haciendo es registrar una cuenta con el correo introducido y una contraseña cualquiera. Si el registro no es correcto es debido a que la cuenta ya existe, así que se notifica al usuario.

Si, por el contrario, el registro es correcto; se elimina la cuenta que se acaba de registrar y se deja al usuario navegar a `NameActivity`, pasando con `intent.putExtra` el correo deseado.

3.4 NameActivity

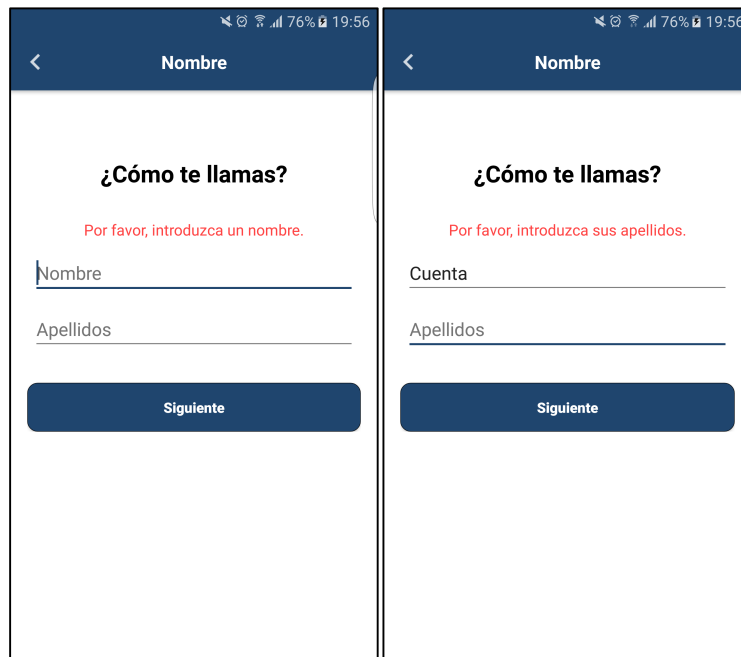


Figura 3.4.1 NameActivity

Esta pantalla recoge el email introducido en la pantalla anterior y define lo que serán nombre y apellidos del usuario. Al igual que en *EmailActivity*, se comprueba que el usuario introduzca algún nombre y apellidos. Si es así, al hacer click en “Siguiete” se navegará a la última pantalla del registro: *PasswordActivity*.

3.5 PasswordActivity

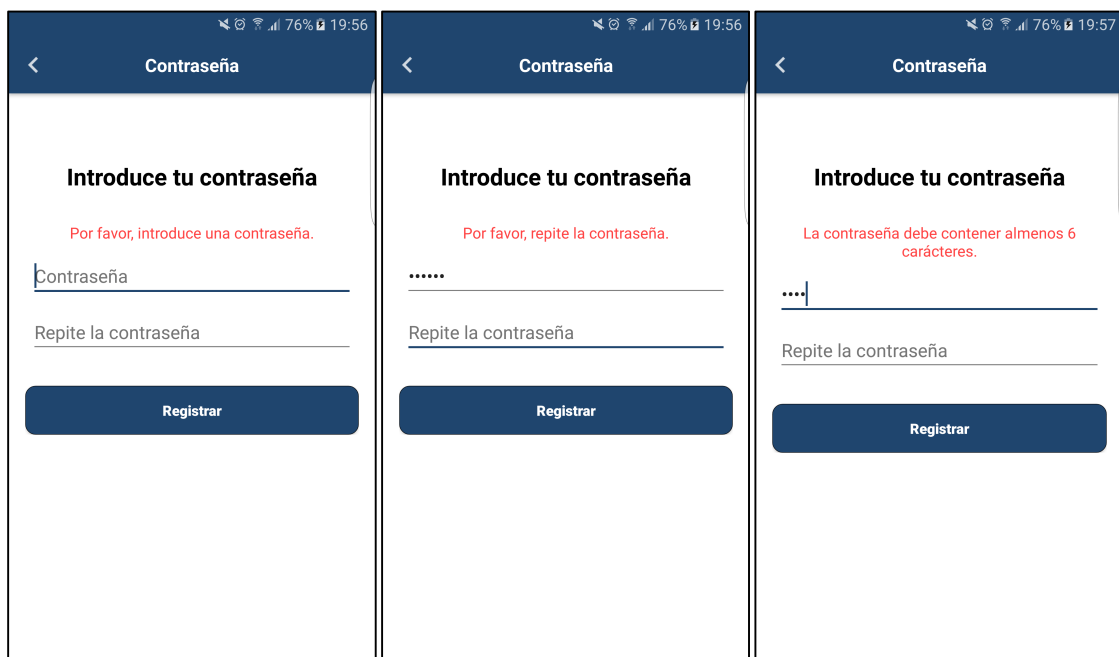
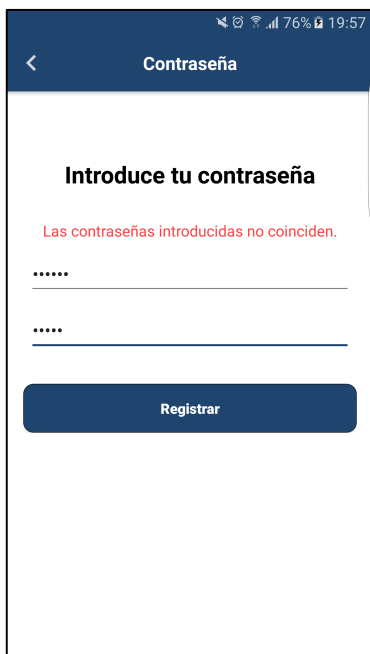


Figura 3.5.1 PasswordActivity 1

Figura 3.5.2 *PasswordActivity*

Esta es la pantalla más importante del registro de un usuario. Aquí se reciben los valores introducidos en las otras *activities* (correo electrónico, nombre y apellidos) con `getIntent().getStringExtra`.

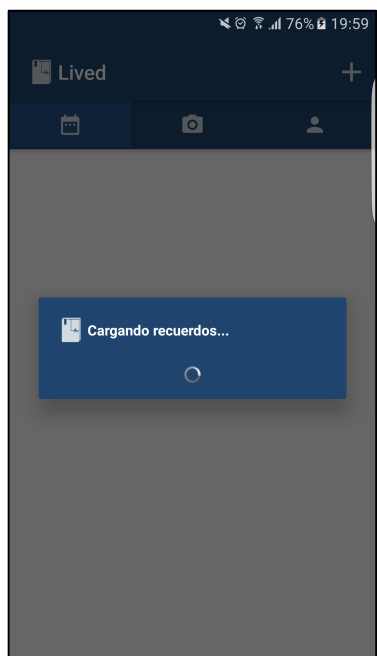
Al igual que en las pantallas anteriores, se comprueban posibles errores del usuario como no introducir una contraseña, no repetirla, que la contraseña sea lo suficientemente larga y que las contraseñas coincidan. Si no se da el caso, se muestra un mensaje de error.

Si la contraseña se introduce correctamente, se registrará al usuario en *firebase* con el método `createUserWithEmailAndPassword`^[4] utilizado en la pantalla *EmailActivity* también. Sin embargo, además de registrar el usuario, se añadirá un nodo a una base de datos de *firebase*^[6] llamada "users".

Si se observa la figura 2.5.2 del apartado 2.5, se puede ver que el modelo *User* contiene propiedades que no se han definido aún en el registro, como sus "*Memories*" o el "*Image_UrI*". De momento se guarda en la base de datos con esos valores vacíos y más adelante el usuario podrá actualizarlos.

Además de registrar al usuario en *firebase* y de guardar sus datos en una base de datos, se navega a *MainActivity*, la pantalla principal de la aplicación.

3.6 *MainActivity*

Figura 3.6.1 *MainActivity*

MainActivity es la pantalla principal y consta de un *ToolBar*^[7] con un logo, un *TabLayout*^[8] con tres pestañas y un *CustomViewPager*. Esto significa que, en sí, no tiene un layout propio, sino que dispone de un *CustomViewPager* en el que se mostrará uno de los tres distintos *Fragments*^[9] según la pestaña seleccionada.

Nótese que se ha creado una clase *CustomViewPager* que extiende del *ViewPager* de Android. Esto se ha realizado para evitar que el usuario pueda hacer *swipe* para cambiar de pestaña. Más adelante se explicará el porqué de esta decisión.

Ya que se ha explicado un poco sobre su UI, ahora se mencionarán algunas partes del código:

Al iniciar la *activity*, se prepara el *CustomViewPager* con un *ViewFragmentAdapter*, que extiende de *FragmentPagerAdapter*, para añadir los tres *Fragments*. A continuación, se muestra un *ProgressDialog*^[10] personalizado para indicar al usuario de que se están cargando sus datos y evitar así que la aplicación quede congelada sin notificar que se están obteniendo datos. Mientras, se busca en la base de datos de *firebase* cuál es el usuario que está conectado y se obtiene toda su información; incluido sus recuerdos.

3.6.1 MemoriesFragment

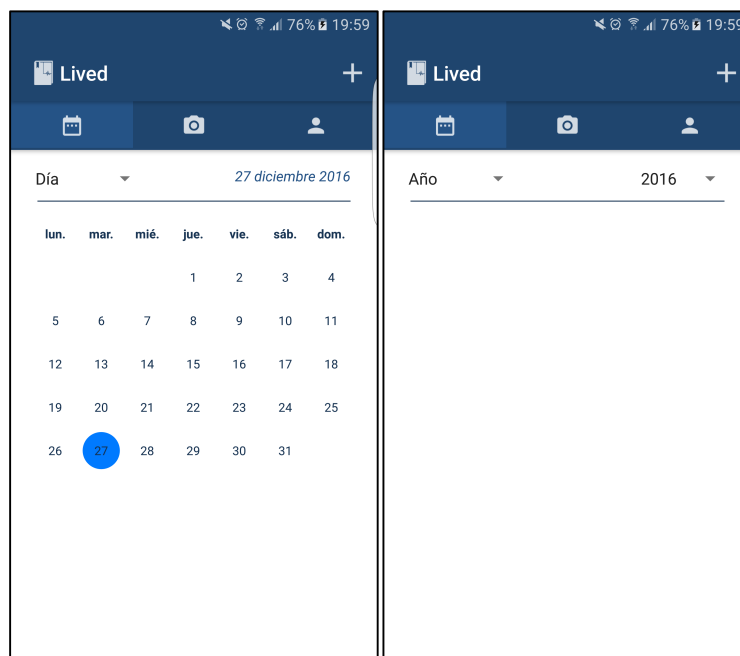


Figura 3.6.1.1 MemoriesFragment

Este fragment es el primero que compone el *CustomViewPager* de *MainActivity*. En él, el usuario puede consultar y añadir sus recuerdos, así que quizás sea donde se produzca la mayor parte de la interacción con el cliente. Como se puede observar en la figura 3.6.1.1, hay dos opciones, “Día” o “Año”.

Si el usuario selecciona “Día”, se muestra un calendario con la fecha actual o, si se ha seleccionado un año anteriormente, el 1 de enero del año seleccionado. El calendario es mensual, así que si se hace un *swipe* hacia la derecha se muestra el mes anterior, y si se hace un *swipe* hacia la izquierda se muestra el posterior. Debido a que se realiza un *swipe* para cambiar de meses, se decidió eliminar el *swipe* del *ViewPager* de Android creando un *CustomViewPager*. De esta manera se evitarían confusiones al intentar cambiar de mes y cambiar de pestaña o viceversa. Se puede seleccionar cualquier fecha del calendario para añadir un recuerdo a ese día, o para ver los recuerdos ya asignados.

Si se selecciona “Año”, se muestra un *RecyclerView* en el que se ven todos los recuerdos del año seleccionado. Esto es posible gracias a una

clase *GridAdapter*. También se incluye un *SwipeRefreshLayout* para poder refrescar la lista de recuerdos.

En *MemoriesFragment* también se añade al menú la opción de añadir un recuerdo, y se puede observar en la *Toolbar* de *MainActivity*. Al hacer click en esa opción, se abre *AddMemoryActivity* para añadir un recuerdo a la fecha (en caso de “Día”) o año (en caso de “Año”) que está actualmente seleccionado.

Nótese que el diseño ha cambiado en relación al planteado en el prototipo. Esto se debe a que la pantalla se veía muy vacía al seleccionar un día y ver simplemente una foto, además de un poco cansado tener que ir navegando entre semanas para poder seleccionar una fecha en concreto. Se encontró una librería de código abierto a la que se hará mención más adelante, que permitía mostrar un calendario de diseño atractivo y añadir eventos al mismo de forma sencilla. Evidentemente, al mostrar un calendario mensual, se eliminó la opción de “Mes” planteada en el prototipo.

3.6.2 CameraFragment

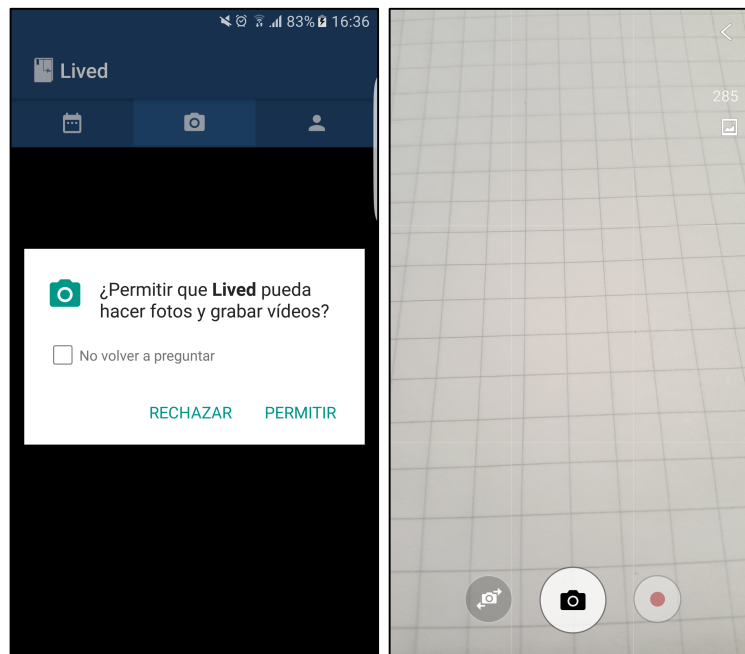


Figura 3.6.2.1 CameraFragment

En este fragment lo único que se hace es llamar a la cámara mediante *Intent(MediaStore.ACTION_IMAGE_CAPTURE)*. No obstante, a partir de Android 6.0 (nivel de API 23), los usuarios conceden permisos a las apps mientras se ejecutan, no cuando instalan la app. Así pues, primero hay que asegurarse de que se ha concedido el permiso y si no se ha concedido, pedirlo.

Una vez tomada la foto, es importante detectar si la foto ha sido tomada con la pantalla vertical u horizontal. Por defecto, las imágenes se guardan en horizontal y se guarda una pequeña información sobre la orientación

de la imagen. Sin embargo, si guardamos la imagen en firebase y luego la recogemos, no tenemos esa información almacenada. Por ese motivo se añadió al modelo "Memory" la propiedad "ImageRotated", para almacenar la orientación de la imagen. El siguiente bloque de código muestra cómo hacerlo una vez obtenido el *path* de la imagen.

```
try {
    ExifInterface exifInterface = new ExifInterface(mediaPath);
    int orientation = exifInterface.getAttributeInt(ExifInterface.TAG_ORIENTATION,
        ExifInterface.ORIENTATION_UNDEFINED);
    switch(orientation) {
        case ExifInterface.ORIENTATION_ROTATE_90:
            Log.d("ROTATED IMAGE", "IMAGE ROTATED 90 DEGREES");
            imageRotation = 90;
            break;

        case ExifInterface.ORIENTATION_ROTATE_180:
            Log.d("ROTATED IMAGE", "IMAGE ROTATED 180 DEGREES");
            imageRotation = 180;
            break;

        case ExifInterface.ORIENTATION_ROTATE_270:
            Log.d("ROTATED IMAGE", "IMAGE ROTATED 270 DEGREES");
            imageRotation = 270;
            break;

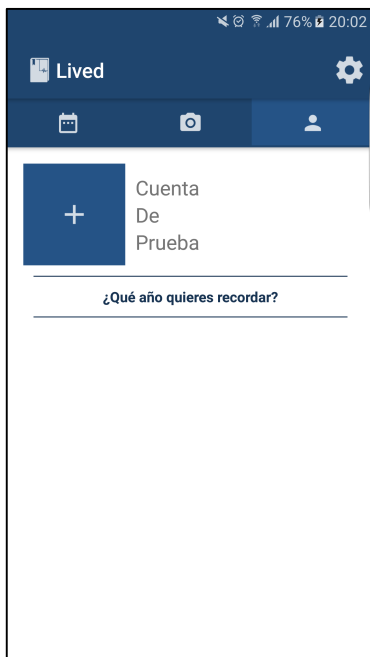
        case ExifInterface.ORIENTATION_NORMAL:
        default:
            Log.d("ROTATED IMAGE", "IMAGE ROTATED NORMAL");
            imageRotation = 0;
            break;
    }
}
catch (IOException e) {
    e.printStackTrace();
}
```

Utilizando la clase *ExifInterface* se pueden leer la información de la que se hablaba anteriormente.

Con la foto y toda su información obtenida, se abre *CropImageActivity* que proviene de una librería de terceros. Esta *activity* permite recortar la imagen que se ha tomado.

Una vez obtenida la foto recortada, se redirige al usuario a *AddMemoryActivity* para que acabe de editar el recuerdo si lo desea y guardarlo.

3.6.3 ProfileFragment



Este es el último de los tres *Fragments* que componen la pantalla principal de la app. Aquí se muestra el nombre y apellidos del usuario, así como su imagen de perfil (si dispone de una). Si el usuario no dispone de imagen de perfil, puede seleccionar el símbolo + para elegir una de la galería. Al igual que en *CameraFragment*, se debe pedir permiso al usuario antes de acceder a su galería de imágenes.

También dispone este *Fragment* de un *RecyclerView*, pero en este caso será una lista de todos los años que tienen algún recuerdo almacenado. La idea es que al seleccionar un elemento del *RecyclerView*, se genere un video con todos los recuerdos de ese año. Sin embargo, se decidió no invertir tiempo en desarrollar esta funcionalidad debido a que y más adelante se explicará el porqué.

Figura 3.6.3.1 ProfileFragment

Nótese que en el *Toolbar* ahora aparece la opción de “Ajustes”, y al seleccionarla, se abre *SettingsActivity*.

En cuanto al diseño, cabe mencionar que en el prototipo se diseñó esta pantalla con un *DatePicker* en vez de un *RecyclerView*. El cambio se debe a que el *DatePicker* en Android se muestra como un *popup* o *dialog* y quedaba absurdo mostrar un único botón que lo abriera en una gran pantalla blanca. Además, con el *DatePicker* se mostraban todas las fechas y quizás molestaba al usuario ir seleccionando años y que la aplicación mostrara cada vez algo como “No tiene recuerdos de este año”. Así pues, se decidió diseñar un *ítem* atractivo para el *RecyclerView* y mostrar únicamente los años que tienen recuerdos almacenados.

3.7 AddMemoryActivity

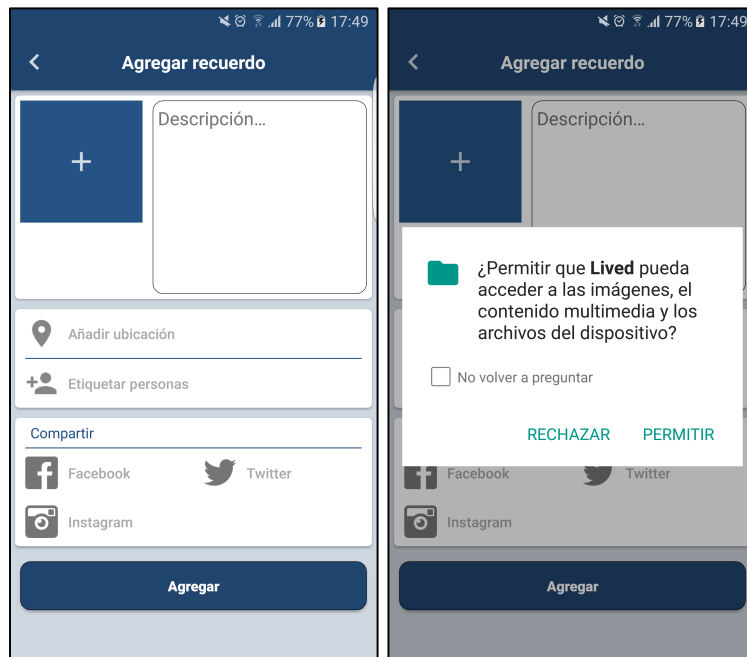


Figura 3.7.1 AddMemoryActivity

Esta pantalla se abre desde la opción + del menú de *MainActivity* cuando está puesto el *MemoriesFragment* y sirve para añadir un recuerdo a *firebase* asignándolo a un año o una fecha concreta. El recuerdo puede ser una imagen o un video con o sin descripción; o un simple texto.

Al seleccionar el + para añadir un elemento de la galería, se debe comprobar que se tiene permiso para acceder al almacenamiento del dispositivo. Es el mismo permiso que se pide en *ProfileFragment* a la hora de seleccionar una imagen de perfil.

Actualmente no se permite añadir ubicación, etiquetar personas o compartir en otras redes el recuerdo. Estas son algunas de las funcionalidades que no se han logrado terminar y se discutirá sobre ello en el apartado de conclusiones.

3.8 ViewMemoryActivity



Al seleccionar una fecha del calendario de *MemoriesFragment* en la que hay recuerdos asignados o un elemento del *RecyclerView* en la opción “Año”, se abre esta pantalla. En ella podemos observar otro *RecyclerView* en el que se muestran los recuerdos de la fecha seleccionada o el recuerdo seleccionado del *GridAdapter*.

En la opción “Día” de *MemoriesFragment*, se seleccionaba una fecha del calendario y la opción + del menú para agregar un recuerdo. Sin embargo, si ya hay un recuerdo asignado y se selecciona esa fecha en el calendario, se abre esta pantalla directamente y no hay posibilidad de agregar otro recuerdo. Así pues, se decidió añadir un *FloatingActionButton* para poder añadir más recuerdos a la misma fecha.

Figura 3.8.1 *ViewMemoryActivity*

3.9 SettingsActivity

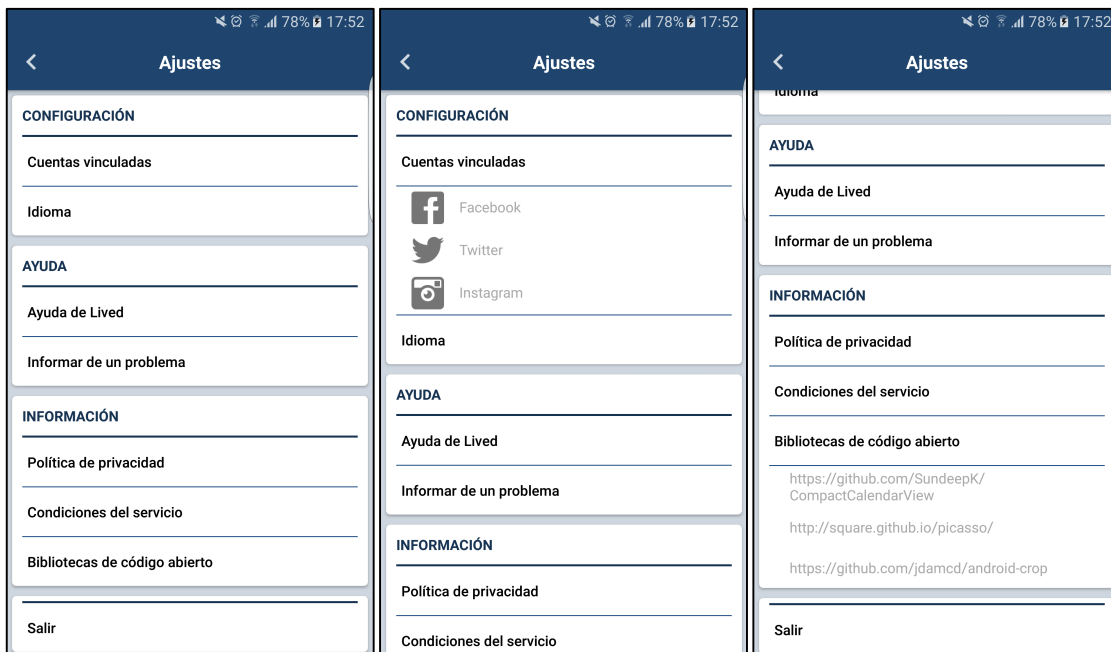


Figura 3.9.1 *SettingsActivity*

Esta es la última pantalla de la app y muestra opciones triviales como las cuentas vinculadas, cambiar de idioma, ayuda, política de privacidad, etc. Estas opciones abrirán un *WebView* vinculado a la página web oficial de Lived cuando se disponga de ella. Además de lo mencionado, está la opción de hacer *log out* de la aplicación.

4. Librerías de terceros

A lo largo del desarrollo de esta aplicación, se ha centrado mucho en el concepto de reutilización de código. Es por eso que se han definido múltiples *styles* para los distintos elementos de la UI, para poder aplicar el mismo código. También se han utilizado tres librerías de terceros, una con licencia libre del MIT, y las otras dos con licencia Apache. De esta manera se evita el tener que desarrollar funcionalidades que ya han desarrollado otras personas. Las librerías, y sus funcionalidades, son las siguientes:

CompactCalendarView

Con esta librería se muestra un calendario en el que se puede navegar por meses y permite la asignación de eventos al mismo. Es el calendario que se utiliza en *MemoriesFragment*. El creador es Sundeepek y la ofrece desde el portal Github^[11].

Android-crop

Esta librería proporciona una *activity* para recortar una imagen. Se utiliza al tomar una foto desde *CameraFragment* o al seleccionar una imagen de la galería en *AddMemoryActivity* y *ProfileFragment*. El creador es SoundCloud y la ofrece desde el portal Github^[12].

Picasso

Esta librería permite cargar imágenes en *ImageViews* de manera rápida, sin mucho consumo de memoria y en una sola línea de código. Se utiliza para cargar imágenes tanto en los *RecyclerViews*, como la foto de perfil o la imagen seleccionada en *AddMemoryActivity*. El creador es Square, Inc. y la ofrece desde el portal Github^[13].

5. Pruebas

A medida que se iba desarrollando la aplicación, ésta se iba probando para comprobar que las funcionalidades programadas funcionaran correctamente. Éstas pruebas se conocen como unitarias, que el desarrollador va realizando sobre la marcha. Se realizan teniendo en cuenta el código, insertando *breakpoints* para detener el hilo de la aplicación y comprobando que las variables contienen los datos correctos.

En Android es posible establecer pruebas de unidad local que se ejecuten en la máquina virtual Java o pruebas instrumentadas que se ejecuten en un dispositivo o emulador de hardware. Sin embargo, para este proyecto no se han realizado *tests* de este tipo.

Al tener el desarrollo casi listo, se realizaron pruebas de carácter más funcional, que se utilizan para garantizar el correcto funcionamiento de la aplicación. Estas pruebas no requieren de los conocimientos de un desarrollador, así que pueden ser realizadas por clientes potenciales de la aplicación. Se definen una serie de acciones sobre la aplicación, que se deben ejecutar del mismo modo que lo haría un usuario. Quién realiza cada prueba, ha de comprobar que el resultado obtenido es el esperado.

Así pues, se entregó el mismo dispositivo a tres usuarios distintos: un chico de 16 años, un joven de 24 años y una mujer de 55 años. Se les asignaron unas pruebas funcionales como crear un nuevo usuario, almacenar un recuerdo en una fecha concreta, visualizar un recuerdo de una fecha concreta y añadir un recuerdo desde la cámara.

Con estas tareas se detectaron errores cuando quizás alguno apretaba el botón *back* sin querer o seleccionaba alguna fecha sin dejar que se cargaran los recuerdos del todo.

6. Conclusiones

En este apartado se describirán las conclusiones del trabajo realizado, lo que se ha aprendido y una reflexión sobre el logro de los objetivos planteados inicialmente. También se realiza un análisis crítico de la planificación y se comentarán las líneas de trabajo futuro que no se han podido explorar en este trabajo y han quedado pendientes.

Descripción de las conclusiones

El propósito de un trabajo final de máster es aplicar los conocimientos adquiridos en el mismo. Pues bien, esta es la primera aplicación que creo yo sólo desde cero y puedo decir que he aplicado prácticamente todo lo que se me ha enseñado. Teniendo en cuenta de que provengo de la rama de la ingeniería industrial y no de informática, programar toda una aplicación ha supuesto un reto para mí y el resultado ha sido una app funcional de la que me puedo sentir orgulloso. Nunca antes del máster había visto la programación orientada a objetos y todo lo que he realizado para crear el producto final ha sido gracias a lo que he aprendido en él.

Con este trabajo se han reforzado mis conocimientos sobre la programación en Android. Las tareas de analizar y definir los potenciales usuarios, llegar a las funcionalidades y a un diseño fueron complicadas pero necesarias. He aprendido mucho también sobre utilización de bases de datos y reutilización de código.

Si bien es cierto que me hubiera gustado incluir más funcionalidades; considero que, dada mi poca experiencia en el campo, he conseguido entregar una aplicación que cumple el principal objetivo de la misma: almacenar los recuerdos de la vida del usuario.

Resumiendo, y teniendo en cuenta los objetivos planteados:

- Se han puesto en práctica los conocimientos adquiridos a lo largo de todo el máster.
- Se ha documentado y justificado las decisiones tomadas en el desarrollo y los resultados conseguidos.
- Se ha adquirido experiencia en afrontar los retos que supone sacar adelante un proyecto completo.
- Se ha hecho converger los conocimientos adquiridos en un producto de alta calidad.

Sin embargo, hay un objetivo que no se ha podido alcanzar y es el de definir una planificación realista para un proyecto, teniendo en cuenta su alcance y recursos disponibles. A continuación, se realiza una reflexión crítica sobre el logro de los objetivos planteados inicialmente.

Objetivos alcanzados

En cuanto a los objetivos planteados en un principio, se comentaba que no se ha realizado una planificación realista del proyecto. El problema se debe al desconocimiento del ámbito y la falta de experiencia en planificaciones de este estilo. A continuación, se expondrán ciertas funcionalidades de la aplicación que no han sido alcanzadas y todas se deben al mismo motivo. Un experto en Android quizás haga una aplicación de este estilo en semanas, pero alguien como yo que no ha programado antes hasta su inicio en el máster, necesitará más tiempo. Además, mi falta de experiencia me hacía atascarme en fallos triviales que me consumían mucho tiempo.

- Esta aplicación estaba pensada para ser algo completamente innovador incluyendo la realidad virtual. No sólo poder ver los recuerdos almacenados sino poder verlos en 360° y tener la sensación de estar reviviéndolo era el elemento diferenciador de la aplicación.
- Las opciones de agregar una ubicación, etiquetar personas o compartir en las redes sociales.
- Ver el video recopilatorio de los recuerdos del año.

Seguimiento de la planificación

En un principio, la planificación era la correcta para las PECs 1 y 2. Cuando empezó la tarea de programar e implementar la aplicación, se complicó un poco. Aparecían errores de todos lados y tenía que consultar continuamente soluciones para arreglarlos. Todo ello me consumió mucho tiempo, haciendo inevitable el tener que descartar ciertas funcionalidades inicialmente planteadas.

Líneas de trabajo futuro

Finalmente, y para terminar el apartado de conclusiones, cabe destacar que este proyecto surge de una gran motivación personal y no se ha realizado como un simple trabajo que entregar. Así pues, las funcionalidades que no se han alcanzado en el tiempo que ha durado el TFM, se intentarán integrar en un futuro no muy lejano.

7. Glosario

- App: término en inglés para referirse a aplicaciones para teléfonos móviles inteligentes.
- Android: sistema operativo, destinado principalmente a dispositivos móviles con pantalla táctil, desarrollado por Google.
- VR: Virtual Reality, Realidad Virtual.
- Smartphones: teléfonos móviles inteligentes.
- TFM: Trabajo Final de Máster.
- Macbook Air: línea de ordenadores portátiles de alta gama, de Apple.
- IDE: Integrated Development Environment, entorno de desarrollo integrado.
- SDK: Software Development Kit, kit de desarrollo de software
- PEC: Prueba de Evaluación Continuada
- H: horas
- Activity: pantalla en Android.
- Scenario: contexto de uso.
- iPhone: línea de teléfonos móviles inteligentes, de Apple.
- iPad: línea de tabletas, de Apple.
- Tablet: tableta.
- Tag: etiqueta.
- PC: Personal Computer, ordenador personal.
- Mockup: modelo de un diseño.
- TextField: elemento de Android para introducir un texto.
- DatePicker: elemento de Android para seleccionar una fecha.
- TabLayout: elemento de Android para disponer de pestañas.
- ComboBox: elemento de iOS para desplegar una serie de opciones.
- DCU: Diseño Centrado en el Usuario.
- API: Application Programming Interface, interfaz de programación de aplicaciones.
- UML: Unified Modeling Language, lenguaje unificado de modelado.
- MVC: Modelo-Vista-Controlador.
- EditText: elemento de Android para introducir un texto.
- Firebase: plataforma móvil.
- ViewPager: clase de Android.
- Fragment: clase de Android.
- UI: User Interface, interfaz de usuario.
- FragmentPagerAdapter: clase de Android.
- MIT: Massachusetts Institute of Technology, instituto tecnológico de Massachusetts.
- Apache: licencia de software libre.
- ImageView: elemento de Android para ver imágenes.
- Java: lenguaje de programación.
- Test: prueba.
- Back: atrás.

8. Bibliografía

- [1]<https://balsamiq.com/>, Fecha de visita: 10/16
- [2]<https://www.justinmind.com/>, Fecha de visita: 10/16
- [3]<https://yuml.me/>, Fecha de visita: 10/16
- [4]<https://firebase.google.com/docs/auth/android/manage-users?hl=es-419>, Fecha de visita: 11/16
- [5]<http://www.androidhive.info/2016/06/android-getting-started-firebase-simple-login-registration-auth/>, Fecha de visita: 11/16
- [6]<https://firebase.google.com/docs/database/android/start/?hl=es-419>, Fecha de visita: 11/16
- [7][https://mzgreen.github.io/2015/06/23/How-to-hideshow-Toolbar-when-list-is-scrolling\(part3\)/](https://mzgreen.github.io/2015/06/23/How-to-hideshow-Toolbar-when-list-is-scrolling(part3)/) , Fecha de visita: 11/16
- [8]<https://www.simplifiedcoding.net/android-tablayout-example-using-viewpager-fragments/>, Fecha de visita: 11/16
- [9]<https://developer.android.com/guide/components/fragments.html?hl=es> , Fecha de visita: 11/16
- [10]<http://www.pcsalt.com/android/create-alertdialog-with-custom-layout-using-xml-layout/>, Fecha de visita: 12/16
- [11]<https://github.com/SundeepK/CompactCalendarView>, Fecha de visita: 11/16
- [12]<https://github.com/jdamcd/android-crop>, Fecha de visita: 12/16
- [13]<http://square.github.io/picasso/>, Fecha de visita: 12/16

9. Anexos

9.1 Instrucciones de compilación

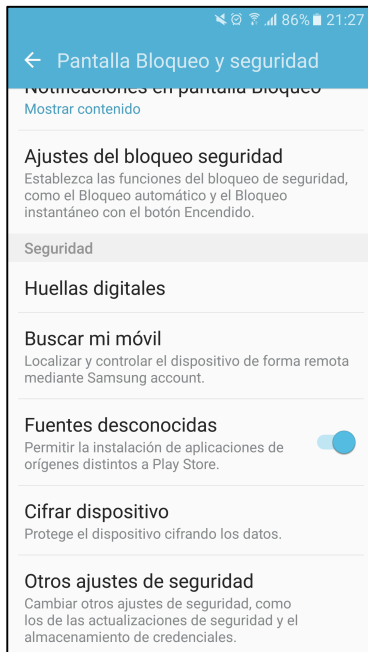
Descripción y requerimientos

Este documento sirve para poder instalar y probar la aplicación que se ha desarrollado a lo largo de este Trabajo Final de Máster. *Lived* es una aplicación para *smartphones* **Android con versión 4.0 o posterior**. El dispositivo debe estar dotado de acceso a internet, cámara y señal GPS para poder utilizar las funciones de la aplicación.

Método 1 de compilación: con el *apk*

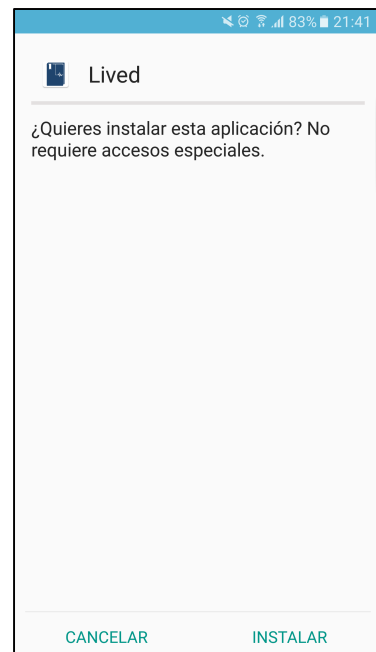
El *apk* de una aplicación es su versión empaquetada, el “binario” en un formato apto para instalarse en Android.

1. El primer paso descargar el archivo *apk* en el dispositivo.
2. Una vez descargado el *apk*, lo siguiente que hay que hacer es instalarlo en el sistema. Para ello, es necesario habilitar el sistema para que acepte la instalación de aplicaciones que no vengan desde Google Play.



En a Ajustes > Seguridad hay una opción llamada “Fuentes desconocidas” la cual hay que activar para poder instalar la aplicación.

En el momento en el que se active la opción, en la pantalla aparecerá una advertencia informando sobre los riesgos de instalar aplicaciones que no provengan de Google Play. Es necesario aceptarlos para poder continuar.

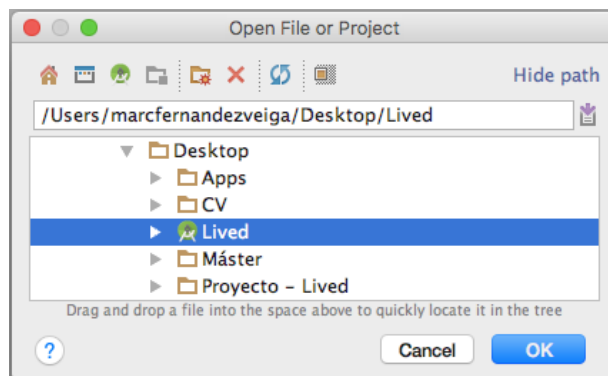


3. Una vez descargada la aplicación, hay buscar el archivo *apk* en la carpeta de “Descargas”. Al seleccionar el archivo, se permitirá su instalación y la aplicación ya estará lista para usarse.

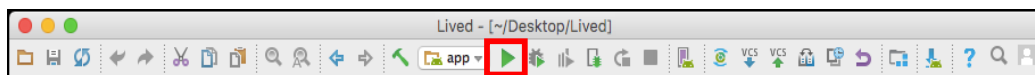
Método 2 de compilación: desde Android Studio

Desde Android Studio y con el código fuente, disponible en la misma carpeta en la que se ha encontrado esta memoria, es posible instalar la aplicación.

1. En primer lugar, se abrirá Android Studio y se seleccionará File > Open para abrir un nuevo proyecto. Una vez encontrada la ruta del proyecto *Lived*, lo seleccionaremos y pulsaremos “OK”.



2. Con el proyecto abierto, en la barra de herramientas seleccionaremos el botón “Run ‘app’” marcado en la siguiente imagen. Esto hará que se ejecute la aplicación y se puede hacer tanto en el simulador de Android o con un dispositivo real si está conectado mediante USB.



Nota: para poder compilar la aplicación desde Android Studio en un dispositivo real, éste debe tener activada la opción “Depuración de USB”. Ésta opción se encontrará en la ruta Ajustes > Opciones de desarrollador.

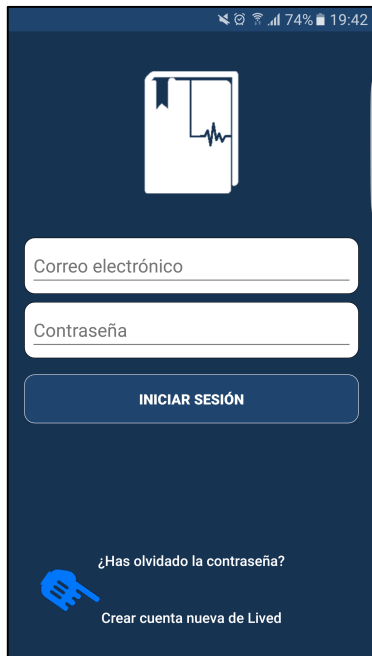
9.2 Manual de usuario

Introducción

Por favor lea este manual antes de utilizar la aplicación *Lived*. Este documento le servirá como guía a través de las actuales funciones principales de la aplicación. Para poder utilizarla, es necesario disponer un dispositivo de un *smartphone* **Android con versión 4.0 o posterior**. El dispositivo debe estar dotado de acceso a internet, cámara y señal GPS. Las descripciones de este manual están basadas en los ajustes por defecto de un Samsung Galaxy S6 Edge.

Registro

Al iniciar la aplicación, aparecerá la pantalla de inicio.



Si es un usuario nuevo, deberá seleccionar **“Crear cuenta nueva de Lived”** en la parte inferior de la pantalla.

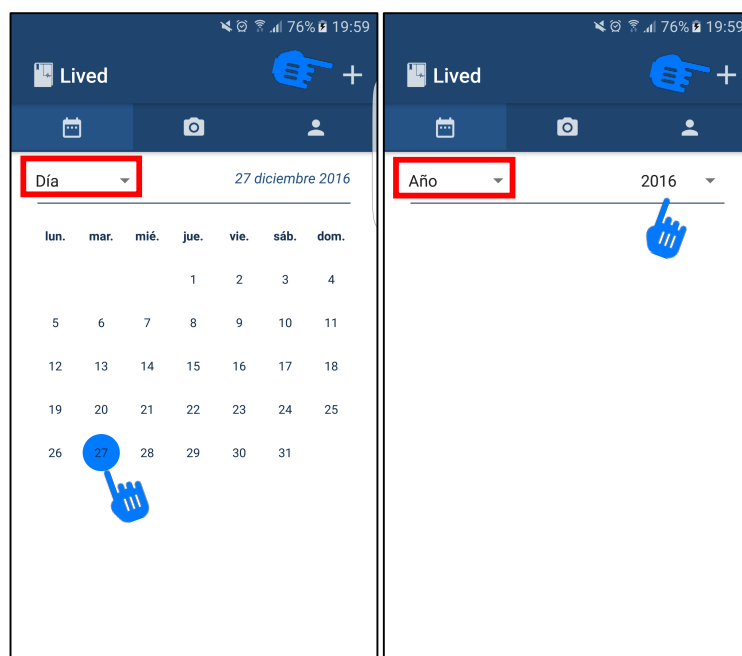
De esta manera se iniciará el proceso de registro a través de 4 pantallas en las cuales se deben definir:

- Si se aceptan las condiciones y términos de privacidad.
- La dirección de correo electrónico que se desee utilizar.
- Nombre y apellidos.
- Una contraseña para la cuenta.

Agregar un recuerdo desde una fecha

En la pestaña del calendario, vista al abrir la aplicación, se pueden agregar recuerdos de dos maneras: asignándolos a una fecha en concreto o a un año. Para alternar entre estas dos opciones deberemos seleccionar en el *spinner* superior izquierdo **“Día”** o **“Año”**.

- Opción **“Día”**: hay que seleccionar una fecha del calendario y pulsar el botón **+**.
- Opción **“Año”**: hay que seleccionar un año del *spinner* superior derecho y pulsar el botón **+**.





Tras pulsar el botón **+**, se abrirá la pantalla de agregar recuerdo. En esta pantalla, se puede ver otro símbolo **+** que, si es pulsado, abrirá todos los archivos del dispositivo para poder seleccionar una imagen o video.

Nota: la primera vez que se decida agregar una imagen o video, *Lived* pedirá permiso para poder acceder al almacenamiento del dispositivo. Es necesario conceder este permiso para poder agregar recuerdos en forma de imagen o video.

También se puede añadir un texto o una descripción al archivo seleccionado pulsando sobre el campo “**Descripción...**”.

Con todo listo, pulsar el botón “**Agregar**”.

Agregar un recuerdo desde la cámara

Anteriormente se ha explicado cómo agregar un recuerdo desde la pestaña del calendario, pero hay otra manera de agregar recuerdos. Si se selecciona la pestaña de la cámara, se puede tomar una fotografía y posteriormente recortarla.

Nota: la primera vez que se abra esta pestaña, *Lived* pedirá permiso para poder acceder a la cámara del dispositivo. Es necesario conceder este permiso para poder tomar fotografías desde la misma aplicación.

Tras recortar la imagen del tamaño deseado, se abrirá la pantalla de “**Agregar recuerdo**” con la fotografía tomada apareciendo al lado del campo de la descripción. Para continuar agregando el recuerdo, siga los mismos pasos vistos en el apartado anterior.

Ver un recuerdo

Desde la pestaña del calendario, se pueden observar los recuerdos almacenados también desde las dos distintas opciones “**Día**” y “**Año**”:

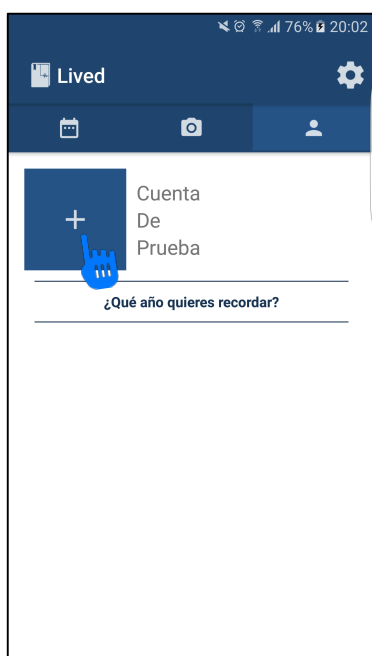
- Opción “**Año**”: los recuerdos almacenados y asignados al año actualmente seleccionado, se verán en una cuadrícula.
- Opción “**Día**”: en el calendario, las fechas que tengan recuerdos asignados se verán con un pequeño indicador. Se puede seleccionar esa fecha con el indicador para poder ver el recuerdo asignado.

En cualquiera de los dos casos; al seleccionar un elemento de la cuadrícula o seleccionar una fecha del calendario, se abrirá una pantalla con el recuerdo en cuestión.



Desde esta pantalla en la que se muestra el recuerdo, se puede agregar un nuevo recuerdo a la misma fecha o año seleccionando el botón +. Si se pulsa este botón, se abrirá la pantalla de “**Agregar recuerdo**” y los pasos a seguir son los mismos que los del apartado “**Agregar un recuerdo desde una fecha**”.

Agregar una imagen de perfil



En la pestaña de usuario, está la opción de agregar una imagen de perfil para el usuario actualmente conectado.

Simplemente seleccionando el botón + se podrá acceder al almacenamiento del dispositivo y seleccionar una imagen de la galería para utilizar como imagen de perfil.

Nota: la primera vez que se decida agregar una imagen de perfil, *Lived* pedirá permiso para poder acceder al almacenamiento del dispositivo. Es necesario conceder este permiso para poder agregar una imagen de perfil.

Otros

Para un mejor entendimiento sobre el funcionamiento de la aplicación *Lived*, está disponible un video llamado “**DEMO Lived.mp4**” en el que se hace una guía sobre sus funciones. Este video se puede encontrar adjunto en la misma carpeta que esta memoria.

