

**Queen Consolidated S.L**

Calle María de Luna S/N

50018 Zaragoza



**Plan de gestión, análisis, diseño y memoria del proyecto.**

**ALLPASS**

Septiembre 2021

**GitHub Organization:** <https://github.com/Queen-Consolidated-S-L>

Sergio Hernández, 712345

José Navarro, 712344

Alejandro Terrón, 712343

Paula Iglesias, 712342

Diego Marco, 712341

Felipe Nonay, 734941

# ÍNDICE

1. Introducción	2
2. Organización del proyecto	3
3. Plan de gestión del proyecto	5
3.1 Procesos	5
3.2 Planes	8
4. Análisis y diseño del sistema	13
4.1 Análisis de requisitos	13
4.2 Diagramas de casos de uso.	14
4.3 Prototipo de GUI	14
5. Funcionamiento GUI	15
6. Conclusiones	31

# 1. Introducción

La aplicación *Allpass* desarrollada por Queen Consolidated S.L, permite a los usuarios gestionar sus contraseñas y datos relativos al inicio de sesión, así como la generación de contraseñas acorde al nivel de seguridad que el usuario desee, haciendo uso de distintas alternativas de cifrado robusto para su implementación. Dicha aplicación se encuentra disponible tanto desde el ordenador mediante una aplicación web, como desde cualquier dispositivo móvil.

La motivación de este proyecto viene provocada por la necesidad de evitar la preocupación de tener que estar recordando todas y cada una de las contraseñas que un simple usuario posee en los distintos portales o servicios que utiliza en su día a día. O, por el contrario, el uso de la misma contraseña en todos los portales y sitios web. Todo ello, unido a la necesidad de garantizar nuestra seguridad digital, hace que mantener los datos relativos a las contraseñas e inicios de sesión bajo cifrado sea una buena apuesta.

Para ello, la aplicación desarrollada permite el almacenado de contraseñas bajo cifrado, así como asegurar un nivel de seguridad utilizando un protocolo de verificación en dos pasos como *Two-Factor Authentication (2FA)*.

Desde la aplicación multiplataforma, el usuario podrá acceder a su cuenta de usuario, haciendo uso de su contraseña personal y su contraseña maestra, garantizando así un mayor nivel de seguridad de acceso en la sincronización de dispositivos. Una vez se ha confirmado el acceso, el usuario podrá almacenar, modificar y eliminar la información relativa a un determinado inicio de sesión y sus contraseñas, así como organizar en categorías las contraseñas de los distintos portales web.

A parte de las funcionalidades anteriormente comentadas, el sistema ofrece la posibilidad de visualizar las distintas contraseñas ordenadas por nombre, fecha de modificación y fecha de expiración. Permite también buscar una determinada contraseña introduciendo el nombre por el que ha sido almacenada.

En el momento que el usuario almacene una nueva contraseña, el sistema permite añadir información adicional como una pequeña descripción y algún archivo en formato de imagen para tener así mayor retroalimentación. Entre los datos necesarios para añadir una contraseña, el usuario deberá introducir una fecha de expiración para el inicio de sesión que se está añadiendo. Una vez alcanzada esa fecha de expiración, el sistema notificará al usuario la necesidad de modificar la contraseña.

Respecto al nivel de seguridad de la aplicación, *Allpass* implementa varios protocolos de verificación y seguridad para el acceso a los datos de carácter sensible como son los datos relativos a un inicio de sesión concreto. Para ello, una vez se ha accedido a la cuenta de usuario mediante la contraseña personal y la contraseña maestra, a la hora de visualizar una contraseña o un inicio de sesión para un determinado sitio web o aplicación, el sistema envía un código de verificación al email asociado a la cuenta de usuario. Tras introducirlo correctamente, el usuario puede acceder a esa información.

En la gestión de la cuenta de usuario, el sistema permite modificar los datos personales y las contraseñas, tanto la contraseña personal como la contraseña maestra.

El coste total esperado del proyecto se ha estimado en 15.600€ y la duración, un plazo de cuatro meses, en los que se ha planteado una entrega en distintas versiones y varias reuniones para comentar y discutir la propuesta y/o estado de desarrollo en el que se encuentra el proyecto, así como si una versión entregada está lista para ser una versión estable y continuar con su desarrollo.

La organización del equipo que pertenece a *Queen Consolidated S.L* queda reflejada en el siguiente apartado de esta memoria.

La versión final entregada al cliente constará de una aplicación web alojada en un servidor, y una aplicación móvil con extensión *.apk* para Android y otra con extensión *.ipa* para iOS.

En el siguiente documento se describen los siguientes aspectos relativos al proyecto:

- Organización del proyecto, integrantes y roles dentro del equipo.
- Procesos de inicio, ejecución, procesos técnicos y planes de ejecución del proyecto. Descripción de las tecnologías y recursos empleados.
- Requisitos del sistema, descripción de la GUI desarrollada.

## 2. Organización del proyecto

Para abordar este proyecto, la empresa *Queen Consolidated S.L* ha contratado un equipo compuesto por seis estudiantes y graduados en la titulación de ingeniería informática, que se han dividido en dos grandes grupos de trabajo, repartiéndose las responsabilidades y tareas de la siguiente forma:

### *GRUPO TRABAJO FRONT-END*

**Sergio Hernández:** Desarrollador encargado de la aplicación móvil, dentro de esta, sus tareas son asignar las tareas de los programadores de la aplicación móvil. Responsable del correcto control de versiones en *GitHub* en su área, así como del cumplimiento de plazos de programación respecto a lo pactado con el otro grupo.

**Paula Iglesias:** Integrante del grupo que más domina las tecnologías que se van a emplear para el desarrollo en ambos grupos de trabajo. Sus funciones aportan experiencia y programación dentro del grupo de trabajo asignado, así como enlace y apoyo en el grupo de back-end. Ayuda a Diego con la implementación back-end de la API a través de la cual el front-end interactúa con la base de datos. Se encarga de establecer y coordinar las reuniones con el personal del otro subgrupo, así como del desarrollo front-end de la web.

**Felipe Nonay:** Jefe y líder del proyecto, encargado del reparto de tareas y supervisar el desarrollo de la aplicación. Responsable del correcto funcionamiento de ambos grupos de trabajo, así como del despliegue de la aplicación en las distintas versiones. Encargado del desarrollo de los mockups interactivos previos a la implementación y desarrollo del sistema. Junto a Paula, son los encargados del desarrollo front-end de la aplicación web.

#### *GRUPO TRABAJO BACK-END*

**Diego Marco:** Encargado de la gestión y desarrollo back-end, así como del control de cada una de las entregas al cliente. Responsable del equipo de back-end, encargado de realizar el seguimiento de las tareas, así como su asignación a los desarrolladores de esta área. Debe de comunicarse con Paula para gestionar las reuniones entre los dos grupos. Ambos se encargan de la implementación back-end de la API a través de la cual el front-end interactúa con la base de datos.

**José Navarro:** Desarrollador encargado de supervisar y controlar cada una de las actualizaciones realizadas en el repositorio de la organización. Se encarga de la puesta a punto de la base de datos. En el caso de que alguna de la aplicación web necesite una mayor carga de trabajo para su desarrollo, se incorporará como desarrollador al equipo front-end.

**Alejandro Terron:** Debido a sus conocimientos y habilidades en desarrollo back-end y aplicaciones móviles, no tiene un grupo de trabajo establecido. Aporta sus conocimientos cuando sea necesario y soluciona aquellos posibles atascos surgidos a la hora de cumplir cada uno de los requisitos necesarios para el funcionamiento de la aplicación.

## 3. Plan de gestión del proyecto

### 3.1 Procesos

#### 3.1.1 Procesos de inicio

Se han establecido unos códigos que permiten identificar el proceso de inicio realizado (**P\_INI\_X**).

Los recursos utilizados para llevar a cabo este proyecto no superarán lo estimado en el presupuesto, ya que, en caso contrario, no tendría sentido que *Queen Consolidated S.L* aceptase un proyecto de estas dimensiones.

**P\_INI\_00:** Presentación y primera toma de contacto del equipo, así como de cada uno de los subgrupos con su responsable.

**P\_INI\_01:** Creación del proyecto y subida al repositorio de la organización en *GitHub*, desde el cual, cada integrante del equipo deberá clonarlo en su ordenador personal antes de comenzar a trabajar.

**P\_INI\_02:** Cada integrante del grupo utilizará su computador personal para realizar el trabajo asignado. En él, será necesario instalar y configurar el software indicado por la organización para desarrollar el proyecto. En este caso, se va a utilizar el IDE propiedad de Microsoft, *Visual Studio Code*. Cabe destacar que cada uno de los integrantes que forman el equipo, tienen experiencia con el software anteriormente mencionado. Además, permite el uso de *GitHub* mediante sincronización con el repositorio desde el propio IDE.

**P\_INI\_03:** Para realizar pruebas sobre la aplicación móvil desarrollada, no será necesario adquirir ningún servidor ni dispositivo. Las pruebas se realizarán en la simulación de un dispositivo mediante el uso de emuladores. Para las pruebas previas a una entrega y pruebas finales, se utilizarán los dispositivos móviles personales de los integrantes del equipo.

**P\_INI\_04:** Para realizar pruebas sobre la aplicación web, las pruebas se realizarán mediante el uso de *localhost* y *Visual Studio Code*, para poder ver los cambios en tiempo real. Para las pruebas previas a una entrega y pruebas finales, se realizarán con la versión web desplegada en *Microsoft Azure*.

**P\_INI\_05:** Se garantizará el correcto funcionamiento de la API una vez esté en funcionamiento.

**P\_INI\_06:** Será necesaria la autoformación para aquellos miembros que no dominen las tecnologías con las que se va a trabajar, lenguaje *Dart* para dispositivos móviles mediante el uso de *Flutter* y programación web usando *React*. También será necesaria la formación en *Node JS* con su framework *Nest JS* para back-end. Para ello, tendrán que hacer uso de tutoriales, cursos online acreditados y documentación oficial.

**P\_INI\_07:** Respecto a la gestión y edición de documentos técnicos y memorias, se va a hacer uso de la plataforma *Google Docs*, ya que, permite la edición y el trabajo en línea de múltiples usuarios simultáneamente.

### **3.1.2 Procesos de ejecución y control**

Se han establecido unos códigos que permiten identificar el proceso ejecución y control que se ha realizado durante el proyecto. (**P\_EJE\_C\_X**).

**P\_EJE\_C\_00:** Va a ser necesario establecer una propuesta técnica y económica para entregar al cliente. Sin incluir el anexo de estimación de costes, ya que es un documento de gestión interna.

**P\_EJE\_C\_01:** Sobre la gestión de la comunicación a nivel interno de organización, se van a establecer canales de comunicación entre los integrantes de BACK-END y los integrantes de FRONT-END. Para ello, se van a crear dos grupos en la plataforma WhatsApp, uno para cada grupo.

Para asegurar la correcta comunicación entre ambas partes, se creará un nuevo grupo de WhatsApp en el que solamente podrán enviar mensajes los responsables de cada uno de los subgrupos. De esta manera, todas las cuestiones pasan por ellos antes de ser enviadas, evitando así discusiones en horas punta de comunicación. En caso de que hubiese mucha discrepancia respecto a un tema concreto, se apuntará para ser tratado en la siguiente reunión grupal.

Mediante este canal, se comentan dudas sencillas, problemas de sincronización entre back-end y front-end, recordatorios de reuniones y, sobre todo, puntos a tratar en las próximas reuniones.

**P\_EJE\_C\_02:** Se establecen reuniones periódicas semanales, avisando de la fecha y hora concretas mediante la plataforma de comunicación mencionada anteriormente. A dichas reuniones deberán acudir todos los integrantes del equipo o en su defecto, y bajo ausencia justificada, únicamente los representantes de cada subgrupo (back-end y front-end). En dichas reuniones se tratan los aspectos en los que hay discrepancia en el equipo, las dudas surgidas y se revisa el trabajo realizado hasta la fecha. Las reuniones se realizarán de manera presencial en la oficina de la organización.

**P\_EJE\_C\_03:** Es necesario especificar el tamaño de tareas a repartir, así como aclarar su reparto a cada uno de los integrantes. En caso de que un miembro no asista a una reunión concreta, se le comunicará la parte que le toca realizar, siempre que la no asistencia sea justificada, una de las normativas internas de la organización como se ha comentado anteriormente. En dichas reuniones semanales, se establecen las tareas a realizar de cara a la próxima reunión, y se recogen en un acta interna los aspectos más relevantes.

**P\_EJE\_C\_04:** Se planifican reuniones con el cliente, en las que se valorará y discutirá la versión entregada con anterioridad. También se comentarán aquellos puntos en los que el cliente no esté de acuerdo. Para ello, se nombrará un encargado de recoger las actas correspondientes a cada reunión

con el cliente, ya que los apuntes recogidos son muy relevantes de cara a próximas correcciones y mejoras.

**P\_EJE\_C\_05:** Se establecen unos plazos de entrega internos, así como realizar un seguimiento semanal del estado del proyecto. Para ello, se revisarán las funcionalidades nuevas completadas. En caso de que alguna funcionalidad no se haya completado, será necesario analizar correctamente el motivo y ponerles solución (dificultades justificadas en la realización implicarán la colaboración de algún otro miembro del grupo para conseguir implementar la funcionalidad.) En caso de conflictos personales, el jefe de proyecto será el encargado de mediar y tomar cartas en el asunto para resolverlo de la mejor manera. En el caso de discrepancia respecto a alguna decisión que el grupo debe acometer, se resolverá mediante votación por parte de los integrantes del equipo.

**P\_EJE\_C\_06:** En las reuniones previas a cada una de las entregas al cliente, se analiza el producto a entregar y se justifica el estado del proyecto, de esta manera, si hay alguna funcionalidad planteada que no se ha conseguido implementar en esa fecha, será necesario justificar su motivo al cliente.

**P\_EJE\_C\_07:** De cara a la última entrega, se realiza una última reunión con el objetivo de revisar si el software producido es de calidad (cumple con los todos aquellos requisitos acordados con el cliente en el momento de la firma del contrato). En caso contrario, será necesario solucionar aquellos requisitos que no se han alcanzado, todo ello antes de la fecha límite de entrega al cliente.

**P\_EJE\_C\_08:** Se realizarán pruebas al sistema desarrollado, y se prepararán los archivos finales que se van a entregar al cliente (entregables), así como desplegar la última versión de la aplicación web en el servidor correspondiente.

### 3.1.3 Procesos técnicos

Respecto a los métodos de trabajo, herramientas y técnicas, el equipo ha planteado una entrega en dos iteraciones, sin incluir en ellas, la entrega final propuesta. Para ello, todos los miembros del equipo deben actualizar localmente la última versión del código antes de empezar a trabajar. Esto se va a realizar gracias al uso de *GitHub*, un repositorio online gratuito que permite gestionar proyectos y versiones de código.

Una vez se haya terminado de realizar las modificaciones pertinentes en el código, los integrantes del equipo deberán subirlas de nuevo al repositorio de la organización. Permitiendo así, que en todo momento el código esté actualizado a la última versión (integración continua). En caso de conflictos, se solucionarán mediante las herramientas *pull request* y *merge* disponibles en *Git*. Para la gestión de *bugs*, se utilizará *Issues*.

La aplicación móvil se va a desarrollar empleando *Visual Studio Code*. Para ello, se hace uso de *Flutter*, un kit de desarrollo de software (*SDK*) que permite mediante el lenguaje *Dart*, el desarrollo de software y aplicaciones multiplataforma, en nuestro caso para *iOS* y *Android*. Posteriormente, se utilizará para su comunicación con back-end la API desarrollada con *Node JS* para su respectivo servidor.



Se realizan pruebas en tiempo real mediante un emulador Android con sistema operativo Android 8.0 o superior. Para las pruebas realizadas con sistema operativo iOS se utilizará Xcode, un IDE que permite migrar el proyecto y poder lanzar el emulador. No obstante, para las pruebas previas a una entrega al cliente y las pruebas finales, se utilizarán los dispositivos móviles personales de los distintos integrantes del grupo.

Respecto a la aplicación web desarrollada, se va a utilizar una biblioteca de *JavaScript* que facilita el desarrollo de interfaces de usuario interactivas, denominada *React*. Para el desarrollo de software con *React*, se hace uso del IDE *Visual Studio Code* junto con *localhost*, permitiendo así la visualización de cambios en tiempo real sin tener que volver a inicial el IDE. Para dar forma a la aplicación y portal web se hace uso de *Bootstrap*, un kit de herramientas de código abierto para desarrollo web responsive, que incluye *HTML*, *CSS* y *JavaScript*.

Por último, respecto a la implementación y gestión del back-end se realizará el despliegue en cloud mediante *Microsoft Azure*. Gestionando así, la base de datos MySQL implementada.

## 3.2 Planes

### 3.2.1 Plan de gestión de configuraciones

Para las convenciones de nombres de documentos y estándares de código se va a hacer uso de los últimos estándares y guías de estilo publicadas para cada tecnología empleada:

- Flutter: <https://github.com/flutter/flutter/wiki/Style-guide-for-Flutter-repo>
- HTML/CSS: <https://google.github.io/styleguide/htmlcssguide.html>
- JavaScript: <https://google.github.io/styleguide/jsguide.html>
- Dart: <https://dart.dev/guides/language/effective-dart/style>
- Node: <https://github.com/google/nodejs/wiki>
- Nest: <https://docs.nestjs.com/>
- React: <https://es.reactjs.org/docs/getting-started.html>

Con la intención de mantener la consistencia entre la aplicación web y la aplicación móvil, se ha seguido la normativa *Material Design*, desarrollada por Google para Android. Dicha librería sirve de apoyo en la construcción de software Android y web. Para sistemas iOS se utilizará *Cupertino*.

- Material Design: <https://material.io/design>
- Cupertino: <https://flutter.dev/docs/development/ui/widgets/cupertino>

Respecto a la organización del código fuente, se utiliza *GitHub* como repositorio y herramienta para gestionar ambos proyectos (aplicación web y aplicación móvil). La organización se encuentra alojado en la siguiente dirección web:

- GitHub Allpass: [https://github.com/Queen-Consolidated-S-L/PSOFT\\_SEPT](https://github.com/Queen-Consolidated-S-L/PSOFT_SEPT)

El encargado de comprobar y gestionar cada una de las actualizaciones realizadas en la rama principal del repositorio de la organización es José Navarro. Todos los integrantes del grupo tienen permisos de acceso y modificación sobre el repositorio de la organización.

Respecto a la organización del repositorio del equipo en GitHub, se plantea una división en varios directorios. Un directorio para la aplicación móvil, un directorio para la aplicación web y un directorio para el back-end. En cada uno de ellos, los cambios que se revisen por el encargado de comprobar las actualizaciones y nuevas versiones del repositorio, José Navarro, serán establecidos en la rama principal. Los cambios y progresos en desarrollo se realizarán en otra rama distinta, hasta que se complete la funcionalidad a desarrollar y el encargado la revise.

La gestión del repositorio se realiza bajo licencia MIT, licencia que permite reutilizar software siempre que se sea propietario de dicho software.

Para la gestión y edición de documentos técnicos y memorias, se va a hacer uso de la plataforma Google Docs, ya que, permite la edición y el trabajo en línea de múltiples usuarios simultáneamente. Para ello, basta con acceder mediante la cuenta universitaria proporcionada por la Universidad de Zaragoza, a la cual todos los integrantes del equipo de trabajo tienen acceso, debido a su condición de estudiantes o investigadores. El responsable de la revisión de dichos documentos técnicos será el líder del equipo, Felipe Nonay.

El resto de actividades no les ha sido asignado un encargado/responsable concreto dentro del grupo, ya que la división del trabajo a realizar en subgrupos (front-end y back-end), implica una responsabilidad de trabajo y cumplimiento implícita. Para ello, cada subgrupo se encargará de gestionar las distintas copias de seguridad antes de realizar un *commit* y actualizarlo en su repositorio, así como de cumplir los objetivos marcados en cada una de las reuniones.

### **3.2.2 Plan de construcción y despliegue del software**

Respecto a la construcción y despliegue del software, se va a garantizar el idéntico funcionamiento en todos los dispositivos de los desarrolladores.

Para ello, se va a crear un proyecto para web y otro para móvil desde un único dispositivo incluyendo las librerías que se consideren necesarias en ese momento. Una vez creados, se realizará un *commit* a su directorio correspondiente en el repositorio de *GitHub*. De esta manera, el resto de integrantes del grupo únicamente clonarán dicho directorio en su computador personal y podrán comenzar a trabajar.

Para la gestión del repositorio es aconsejable la instalación del software *GitHub Desktop*, que incorpora todas las funciones disponibles mediante el uso de terminal en una interfaz gráfica. No obstante, no es una normativa dentro del equipo.

*Visual Studio Code* permite una integración sencilla con la tecnología *Node* y *Flutter*, elegidas para el desarrollo de este proyecto. Además, incluye una herramienta que permite utilizar *GitHub* desde el propio IDE, permitiendo así actualizar el repositorio.

Es necesario garantizar el uso de las mismas versiones de las extensiones, paquetes y dependencias de la aplicación. Por ello, se plantea el uso de ficheros que incluyen los paquetes necesarios para lanzar el proyecto.

La frecuencia con la que se prueba todo el sistema se realiza una vez cada semana. De esta manera, aprovechando la reunión de todos los integrantes del equipo, se muestra el estado del sistema en su versión más actual, y se comprueba que los cambios implementados superan o no las pruebas realizadas.

Se escoge *Microsoft Azure* como SaaS donde alojar el software desarrollado, ya que permite desplegar el sistema de manera automática conectando el repositorio de la organización en *GitHub*. Para la gestión del modelo de datos, es necesario utilizar *Azure Database for MySQL*. El despliegue se realiza utilizando *GitHub Actions*. También se hará uso de *Azure App Services*, un servicio que permite gestionar aplicaciones web, servicios y API REST.

### **3.2.3 Plan de aseguramiento de la calidad**

La calidad del proyecto se basa en el seguimiento de documentación y guías de estilo, que permiten asegurar la calidad gráfica del diseño de la aplicación, siendo así necesario la toma de decisiones para distintas áreas del desarrollo, como la paleta de colores corporativa e iconos de la misma familia. Todo ello, común en ambas aplicaciones, aplicación web y aplicación móvil.

Respecto al diseño de la GUI a través de código para la aplicación móvil se hace uso de la librería incorporada en *Flutter* como *Material Design* para Android y *Cupertino* para iOS.

Además, para la aplicación web se usa la librería de *React* y la de *Bootstrap* para mantener unos estándares y calidades apropiados para un proyecto de esta envergadura, haciendo uso de CSS para la gestión y control de estilos.

Las pruebas realizadas para controlar la calidad del código desarrollado se dividen en tres tipos; pruebas de caja blanca, pruebas de caja negra (mediante tablas de equivalencia) y pruebas del sistema (validando los requisitos implementados). Dichas pruebas permiten detectar errores, problemas y fallos que se han pasado por alto anteriormente. Para la realización de pruebas se ha hecho uso del software y servicio de integración continua *Travis CI*.

Por último, antes de cada entrega al cliente, se revisará la navegabilidad de la aplicación, así como una pequeña preparación de una presentación con las nuevas funcionalidades implementadas.

### 3.2.4 Calendario del proyecto y división del trabajo

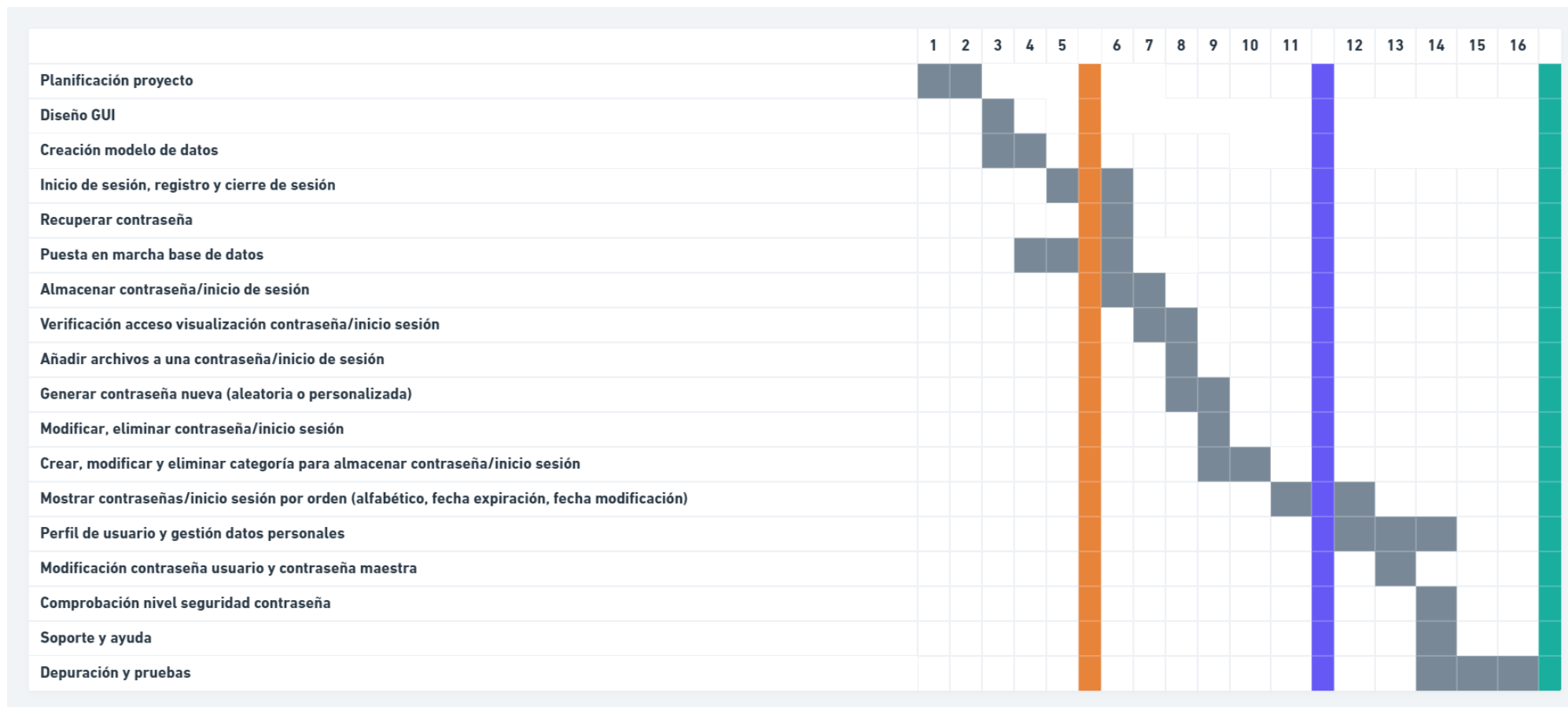
Se presenta a continuación un diagrama de Gantt en el cual se muestra la evolución del desarrollo del software planteado. La implementación y desarrollo de la aplicación web y de la aplicación móvil se realizarán en paralelo, permitiendo así resolver aquellas dudas comunes que puedan ir surgiendo.

El grupo se ha dividido en dos subgrupos de tres personas cada uno, uno específico en desarrollo back-end y otro en desarrollo front-end. Estos grupos no son definitivos, ya que como se ha comentado anteriormente, la distribución del trabajo puede variar dependiendo de las necesidades y dificultades encontradas.

Cada persona tiene asignadas una serie de tareas a realizar de cara a la próxima reunión o fecha de entrega, dichas tareas se registran a través de la herramienta que proporciona *GitHub* para asignación de tareas. Dichas tareas, deberán ser completadas a tiempo para evitar problemas y conflictos.

Cada uno de los subgrupos podrá acotar el tamaño de las tareas asignadas, por ejemplo, si para una tarea concreta se cree necesario incorporar a otro miembro del subgrupo debido a su dificultad, el subgrupo tiene libertad para hacerlo.

El diagrama de Gantt se ha realizado asumiendo una duración de dieciséis semanas, o lo que es lo mismo, un proyecto con duración estimada de cuatro meses. Teniendo como fecha límite de entrega de producto al cliente el día 3 de septiembre de 2021. Se marcan en dicho diagrama las tres entregas planteadas al cliente, dos iteraciones en entregas intermedias y la entrega final, en naranja, morado y verde respectivamente.



## 4. Análisis y diseño del sistema

### 4.1 Análisis de requisitos

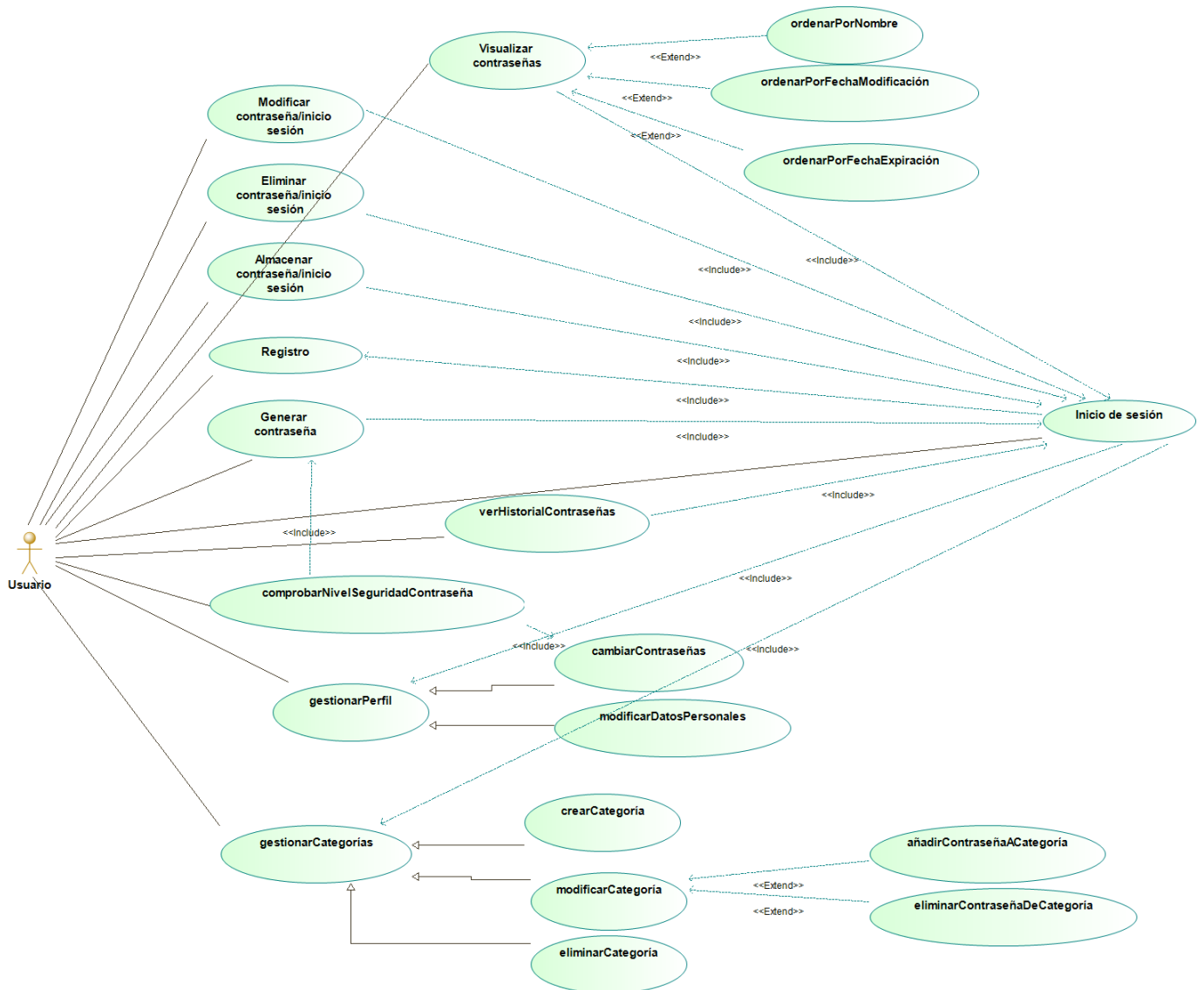
#### Requisitos funcionales

1. El sistema almacenará contraseñas e inicios de sesión.
2. El sistema almacenará contraseñas online mediante el uso de una única contraseña maestra.
3. El sistema permitirá el uso de la aplicación mediante registro e inicio de sesión (autenticación) en ella, para múltiples usuarios.
4. El sistema permitirá al usuario añadir los datos relativos a una contraseña en su cuenta de usuario.
5. El sistema permitirá al usuario modificar los datos relativos a una contraseña almacenada en su cuenta de usuario.
6. El sistema permitirá al usuario eliminar los datos relativos a una contraseña ya almacenada en su cuenta de usuario.
7. El sistema permitirá al usuario recuperar la información relativa a las contraseñas/inicios de sesión almacenados en su cuenta desde cualquier dispositivo mediante autenticación de usuario en la aplicación/portal web de *Allpass*.
8. El sistema permite generar y modificar una contraseña almacenada, tantas veces como el usuario desee.
9. El usuario podrá buscar y visualizar las contraseñas almacenadas por orden alfabético, fecha de modificación y fecha de expiración.
10. El usuario será capaz de filtrar las contraseñas almacenadas por categorías, así como crear, modificar o eliminar una determinada categoría.
11. El sistema permite clasificar los datos relativos a un inicio de sesión o contraseña haciendo uso de las categorías existentes.
12. El usuario podrá almacenar información adicional relativa a una contraseña, sitio web, descripción, imágenes...
13. El usuario podrá recuperar el historial de datos de inicio de sesión almacenados para un servicio o portal concreto.
14. El sistema deberá exigir el cambio de contraseña una vez alcanzada su fecha de expiración.
15. El usuario podrá generar contraseñas aleatoriamente o de manera personalizada seleccionando los parámetros que crea conveniente longitud y contenido (números, mayúsculas o caracteres especiales).
16. El usuario tendrá un perfil de usuario con datos personales que podrá modificar, así como una imagen de perfil.
17. El usuario podrá modificar la contraseña de acceso a la aplicación y la contraseña maestra desde su perfil de usuario.
18. El sistema informará al usuario del nivel de seguridad de sus contraseñas a la hora de generarlas y modificarlas.

#### Requisitos no funcionales

1. El sistema funcionará tanto en plataformas móviles (iOS y Android), como en web.
2. El usuario debe autenticarse para poder acceder.
3. El sistema utiliza sistemas de cifrado robustos para garantizar la seguridad de las contraseñas.
4. El sistema debe permitir la sincronización entre dispositivos.
5. El sistema implementa un sistema de seguridad para el acceso a datos de carácter sensible basado en el protocolo *Two Factor Authentication* (2FA).

## 4.2 Diagramas de casos de uso.



Se muestra en la imagen anterior el diagrama de casos de uso basado en el análisis de requisitos realizado para el sistema *Allpass* con la herramienta UML de código abierto, *Modelio*.

## 4.3 Prototipo de GUI

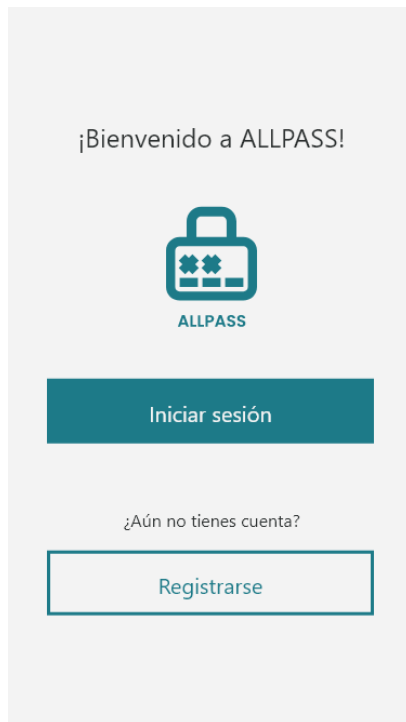
Se incorpora a continuación un enlace al prototipo interactivo del sistema, cuyo diseño ha sido realizado cuidadosamente, intentando plasmar en él todas las funcionalidades descritas anteriormente y dotando a la interfaz del máximo detalle.

<https://xd.adobe.com/view/8bf4cb2c-eae3-4929-9f53-ab941fa697cb-dea7/?fullscreen>


## 5. Funcionamiento GUI

En este capítulo se va a describir cómo funciona la GUI desarrollada, así como las funciones que el sistema es capaz de realizar.

### PANTALLA DE INICIO



¡Bienvenido a ALLPASS!

  
ALLPASS

Iniciar sesión

¿Aún no tienes cuenta?

Registrarse

Pantalla *inicial* del sistema, en ella el usuario debe decidir si inicia sesión en *Allpass* (ya posee una cuenta de usuario), o, por el contrario, decide registrarse en el sistema.

### INICIO DE SESIÓN



Inicio de sesión

  
ALLPASS

734941@unizar.es

\*\*\*\*\*

[¿Has olvidado tu contraseña?](#)

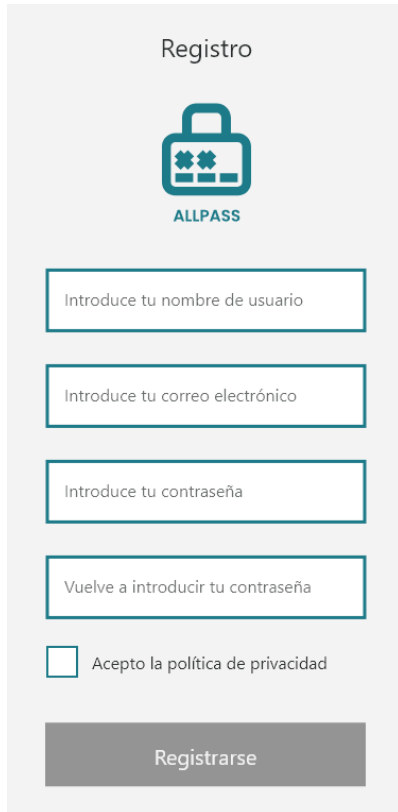
Iniciar sesión

Pantalla de *inicio de sesión*, en ella el usuario debe introducir sus credenciales en el sistema (email asociado a la cuenta y contraseña).

El requisito funcional alcanzado es RF-3 con la autenticación de usuarios.



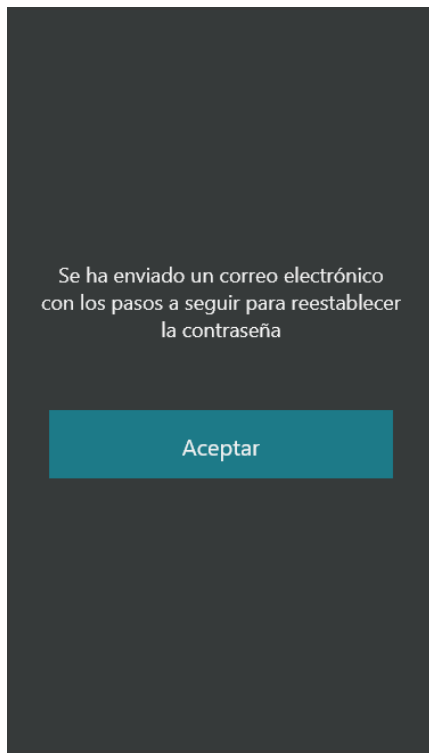
## REGISTRO



Pantalla de *registro de usuario*, en ella el usuario debe introducir sus los datos solicitados para registrarse en el sistema. También deberá aceptar la política de privacidad.

El requisito funcional completado junto con la pantalla de inicio de sesión es RF-3.

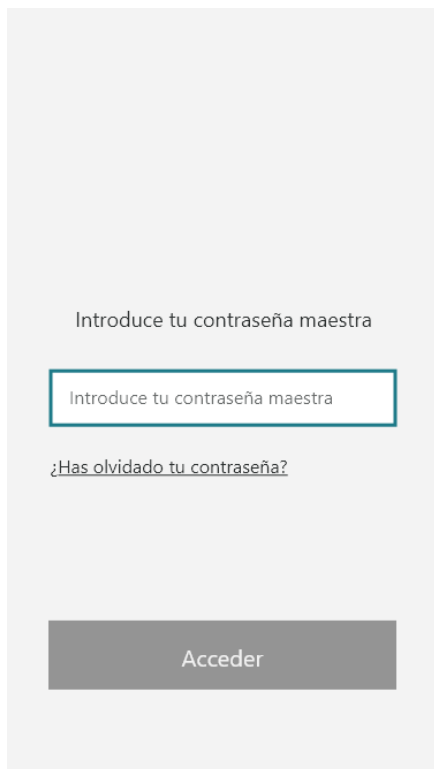
## RECUPERACIÓN DE CONTRASEÑA DE USUARIO



Pantalla de *recuperación de contraseña de usuario*, en ella el sistema envía una nueva contraseña al email asociado a la cuenta. El usuario deberá cambiarla en el siguiente *login*.

No se ha considerado necesario simular el comportamiento de recuperación de contraseña de usuario en la GUI interactiva. Respecto a la recuperación de la contraseña maestra, se muestra el comportamiento más adelante.

## VERIFICACIÓN ACCESO MEDIANTE CONTRASEÑA MAESTRA



Introduce tu contraseña maestra

[¿Has olvidado tu contraseña?](#)

Acceder

Pantalla de *acceso a la sesión de usuario*, en ella el sistema solicita la introducción de la contraseña maestra. El usuario deberá insertarla correctamente para acceder.

El requisito funcional completado es RF-2.

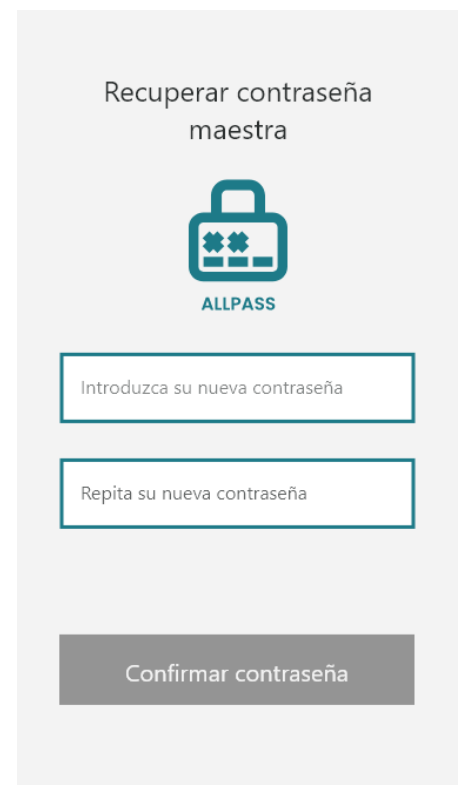
## RECUPERACIÓN CONTRASEÑA MAESTRA




Se ha enviado un correo electrónico con los pasos a seguir para reestablecer la contraseña maestra

Aceptar

Pantalla de *recuperación de contraseña maestra*, en ella el sistema envía una nueva contraseña al email asociado a la cuenta. El usuario deberá cambiarla en la pantalla que aparece a continuación.



Recuperar contraseña maestra



ALLPASS

Confirmar contraseña

## PANTALLA GESTIÓN CONTRASEÑAS

Gestiona tus contraseñas

Filtrar contraseñas

- Instagram  
Válido hasta 07/08/2025
- Facebook  
Válido hasta 01/02/2024
- Gmail UNIZAR  
Válido hasta 04/06/2022
- Gmail FNONAY  
Válido hasta 15/02/2024
- Moodle UNIZAR  
Válido hasta 04/12/2022
- BBVA Banco BBVA  
Válido hasta 21/06/2023
- LinkedIn  
Válido hasta 29/08/2021

Pantalla de *gestión de contraseñas*, en ella el usuario puede realizar las distintas funciones implementadas; filtrar contraseñas, añadir una nueva contraseña (mediante el uso del botón “+”) y visualización de los datos relativos a una contraseña o inicio de sesión concreto.

Se aprecia el aviso mostrado para cambiar la contraseña del portal *Linkedin*, ya que se ha alcanzado su fecha de expiración.

## AÑADIR UNA CONTRASEÑA

Añade una contraseña

Rellena los siguientes campos para almacenar los datos de inicio de sesión

Introduce nombre

Introduce sitio web

Introduce nombre de usuario

Introduce contraseña

Introduce fecha de expiración

Introduce descripción

Añade archivos

Confirmar

Pantalla para *añadir contraseñas*, en ella el usuario debe introducir los datos en los campos de texto que aparecen. También podrá asociar archivos en formato imagen al un inicio de sesión o contraseña concretos.

Los requisitos funcionales completados son RF-1, RF-2, RF-4 y RF-12.

## MODIFICAR UNA CONTRASEÑA



Tus contraseñas

Instagram

Válido hasta 07/08/2025

Descripción

Instagram es una red social y una aplicación móvil al mismo tiempo, que permite a sus usuarios subir imágenes y videos con múltiples efectos fotográficos como filtros, marcos, colores retro, etc., para

Archivos

Categorías

☒ Personal ☒ RRSS

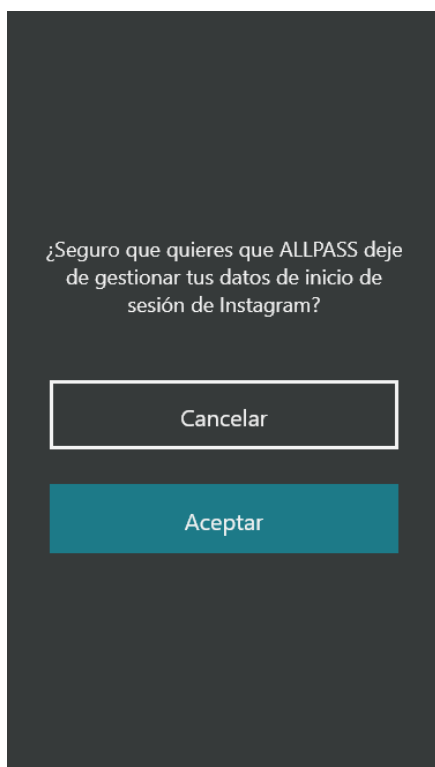
☒ Profesional ☐ Finanzas

Aplicar cambios

Pantalla para *modificar contraseñas*, en ella el usuario debe introducir los datos en los campos de texto que aparecen. Así como, seleccionar los archivos nuevos que desee añadir y las categorías en las que clasificar un determinado inicio de sesión o contraseña.

Los requisitos funcionales completados son RF-5, RF-11 y RF-12.

## ELIMINAR UNA CONTRASEÑA



¿Seguro que quieres que ALLPASS deje de gestionar tus datos de inicio de sesión de Instagram?

Cancelar

Aceptar

Pantalla para *eliminar contraseñas*, en ella el usuario debe confirmar si desea que la aplicación deje de recordar los datos relativos a un inicio de sesión o contraseña para un portal o sitio web concreto.

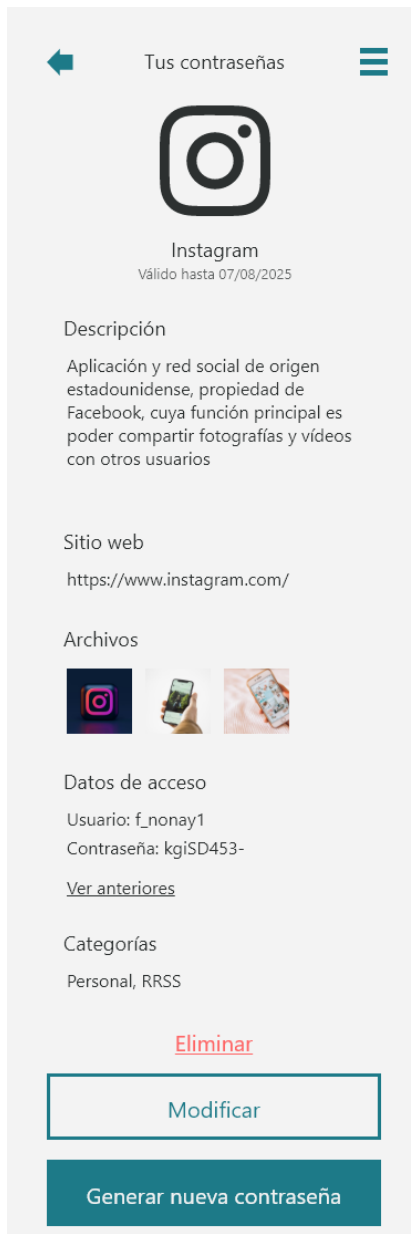
El requisito funcional completado es RF-6.

## VERIFICACIÓN ACCESO DATOS INICIO SESIÓN/CONTRASEÑA



Pantalla para confirmar el acceso mediante *verificación*, en ella el usuario debe confirmar el código enviado por el sistema al correo electrónico asociado. De esta manera, se implementa una autenticación de tipo Two Factor Authentication (2FA), que permite asegurar que la persona que accede a los datos de inicio de sesión o contraseña almacenados, es realmente el propio usuario.

## VISUALIZACIÓN DATOS INICIO SESIÓN/CONTRASEÑA



Pantalla que *muestra* la información de los datos relativos a un inicio de sesión o contraseña concretos. En ella, se puede observar la información introducida por el usuario (descripción, sitio web, archivos...), la fecha de expiración de la contraseña, los datos de acceso y las categorías en las que se ha clasificado dicha contraseña/inicio de sesión.

Desde esta pantalla, se pueden realizar las acciones de eliminar y modificar contraseña, generar una nueva contraseña, y por último, acceder al historial de datos de acceso almacenados.

El requisito funcional completado es RF-7.

## VISUALIZACIÓN HISTORIAL DE CONTRASEÑAS ALMACENADO



Pantalla que *muestra el histórico de contraseñas almacenadas* en el sistema para un determinado portal web o aplicación.

El requisito funcional completado es RF-13.

## GENERAR NUEVA CONTRASEÑA ALEATORIA



Pantalla que *genera una nueva contraseña aleatoria*. Tras seleccionar generar contraseña nueva, en la pantalla de visualización de datos relativos a una contraseña, el usuario debe escoger si dicha contraseña va a ser personalizada o *aleatoria*.

En este último caso, debe seleccionar el nivel de seguridad deseado y pulsar el botón de generar contraseña. En caso de que desee generar una contraseña de nuevo, el usuario podrá realizar el proceso tantas veces como desee.

Los requisitos funcionales completados son RF-8, RF-15 y RF-18.

## GENERAR NUEVA CONTRASEÑA PERSONALIZADA



Tus contraseñas

Instagram  
Válido hasta 07/08/2025

Selecciona nivel de seguridad

Números

Caracteres especiales

Mayúsculas

Introduce número de caracteres

Generar contraseña

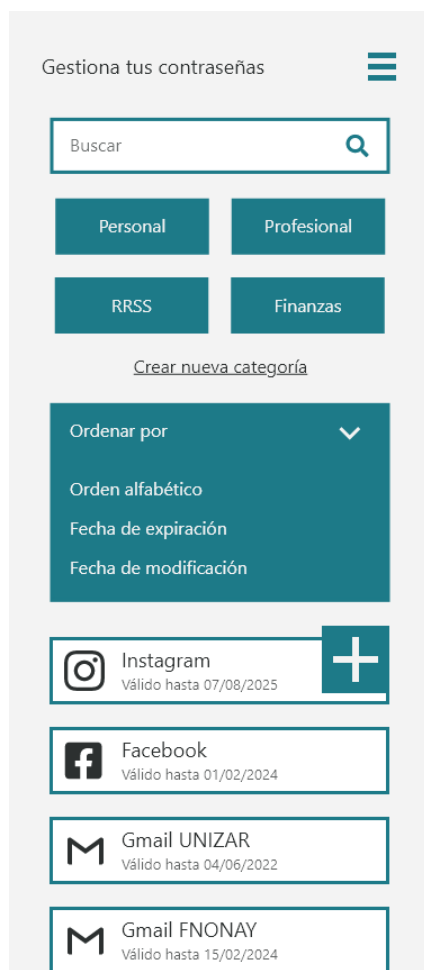
Pantalla que *genera una nueva contraseña personalizada*.

Tras seleccionar generar contraseña nueva, en la pantalla de visualización de datos relativos a una contraseña, el usuario debe escoger si dicha contraseña va a ser aleatoria o *personalizada*.

En este último caso, debe marcar los parámetros que desee que incluya dicha contraseña, así como la longitud de la misma y confirmar la acción pulsando el botón de generar contraseña. En caso de que desee generar una contraseña de nuevo, el usuario podrá realizar el proceso tantas veces como desee.

Los requisitos funcionales completados son RF-8 y RF-15.

## FILTRAR CONTRASEÑAS



Gestiona tus contraseñas

Buscar

Personal Profesional

RRSS Finanzas

[Crear nueva categoría](#)

Ordenar por

Orden alfabético

Fecha de expiración

Fecha de modificación

Instagram  
Válido hasta 07/08/2025

Facebook  
Válido hasta 01/02/2024

Gmail UNIZAR  
Válido hasta 04/06/2022

Gmail FNONAY  
Válido hasta 15/02/2024

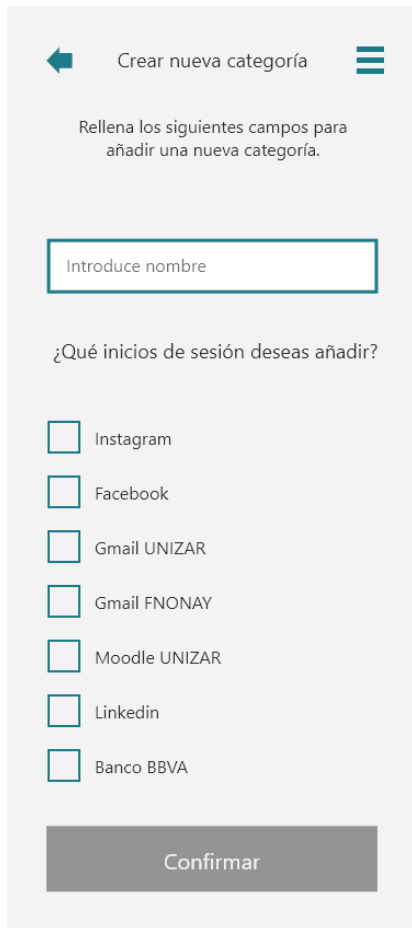
Pantalla que *permite filtrar las contraseñas e inicios de sesión almacenados*.

Permite introducir el nombre por el que se ha almacenado una contraseña para acceder a sus datos de manera más rápida.

Se muestran las distintas categorías existentes en el sistema, así como los inicios de sesión almacenados en ellas. Estos se pueden ver *clickando* sobre el nombre de la categoría.

Por otro lado, aparece la opción de crear una categoría. También incorpora una visualización por orden alfabético, fecha de expiración o fecha de modificación, según el usuario desee.

## CREAR CATEGORÍA

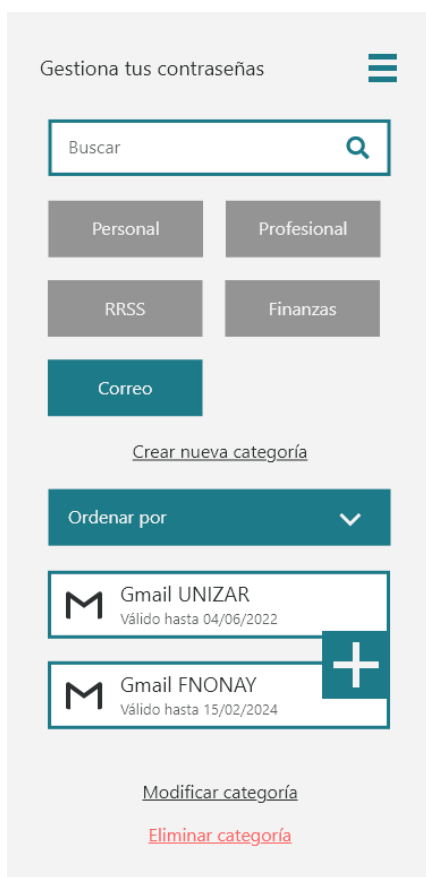


Pantalla que permite *crear una nueva categoría*.

Es necesario introducir el nombre de la categoría a añadir y los inicios de sesión/contraseñas que se desean clasificar en ella.

El requisito funcional completado es RF-10 y RF-11.

## VISUALIZAR CATEGORÍA



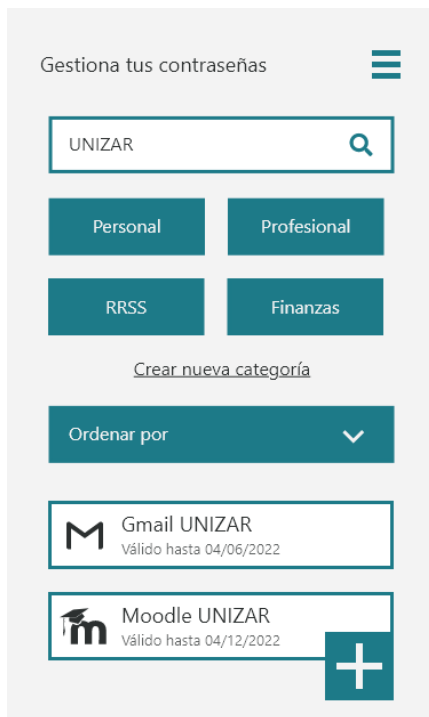
Pantalla que permite *visualizar una categoría*.

Desde esta pantalla se tiene acceso a la visualización ordenada según el criterio que desee el usuario, así como, las opciones de modificar y eliminar dicha categoría.

Los requisitos funcionales completados son RF-10 y RF-11.



## BUSCAR CONTRASEÑA/INICIO DE SESIÓN

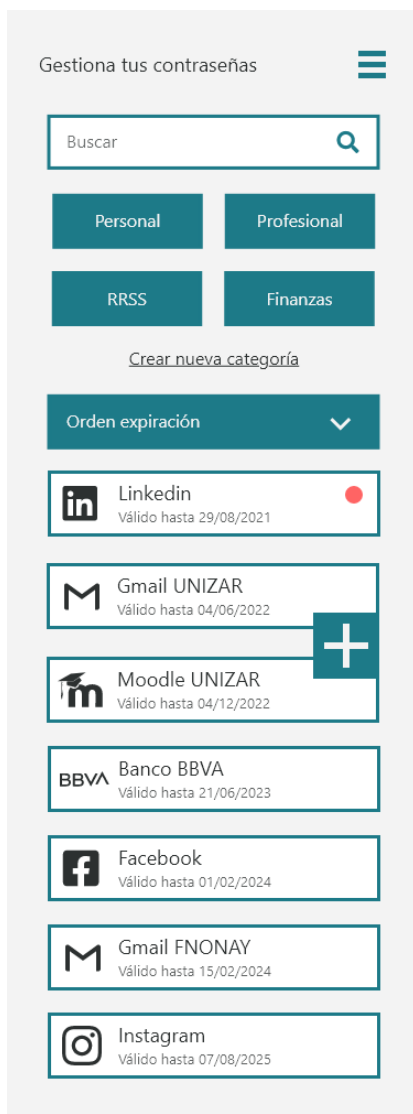


Pantalla que permite *buscar una contraseña*.

Desde esta pantalla se puede visualizar la búsqueda de un inicio de sesión o contraseña concreto. Una vez encontrado, se pueden realizar las funciones de creación y clasificación en categorías, así como ordenar según el criterio deseado.

El requisito funcional completado es RF-9.

## ORDENAR CONTRASEÑA/INICIO DE SESIÓN



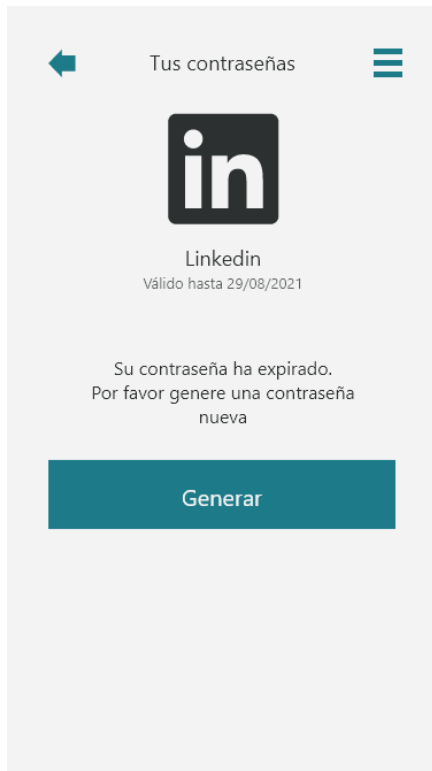
Pantalla que permite *ordenar una contraseña según el criterio seleccionado*.

En esta pantalla, se ha realizado el orden según la fecha de expiración, mostrando en primer lugar *LinkedIn*, con fecha de expiración más próxima. Se puede seleccionar también el orden por nombre y orden por fecha de modificación.

En la pantalla también se puede apreciar el aviso que nos muestra el sistema en el inicio de sesión/contraseña correspondiente a *LinkedIn*, ya que la validez de su contraseña ha caducado. Será necesario cambiar la contraseña.

Los requisitos funcionales completados son RF-9 y RF-14.

## CAMBIO CONTRASEÑA QUE HA EXPIRADO

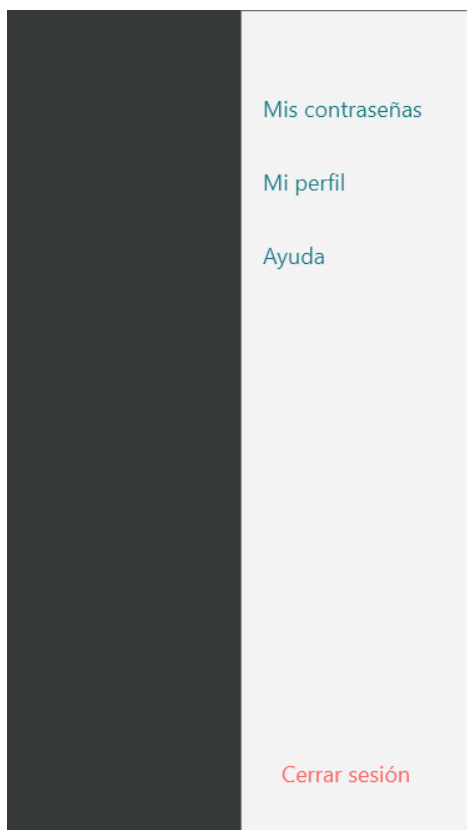


Pantalla que indica la necesidad de *cambiar una contraseña ya que su validez ha expirado.*

En esta pantalla, se puede apreciar el aviso que nos muestra el sistema, ya que la validez de su contraseña ha caducado. Será necesario generar una nueva contraseña.

Los requisitos funcionales completados son RF-9 y RF-14.

## NAVEGACION POR LA APP



Pantalla que muestra las distintas *opciones y menús* en la app *Allpass*.

En esta pantalla, se puede navegar hacia la pantalla de gestión de contraseñas, el perfil de usuario dentro del sistema, y la pantalla de ayuda y soporte.

También permite hacer *log out* pulsando *Cerrar sesión*.

## PERFIL DE USUARIO



Pantalla que permite visualizar el *perfil de usuario* en la app *Allpass*.

A través de él, se permite la edición de datos personales y modificación de contraseñas.

El requisito funcional completado es RF-16.

## EDITAR DATOS PERSONALES



Pantalla que permite la *edición de datos personales* de un usuario concreto.

Permite introducir el nombre y el correo electrónico y de esta manera realizar el cambio. También permite cambiar el avatar o imagen de perfil del usuario.

El requisito funcional completado es RF-16.

## MODIFICAR CONTRASEÑAS

← Mi perfil ☰

Personal Maestra

Contraseña de la cuenta

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

[Comprobar nivel de seguridad](#)

Aceptar cambios

Pantalla que permite la *modificación de las contraseñas de usuario*.

Permite distinguir la contraseña que se desea modificar, seleccionando *personal* en caso de querer modificar la contraseña de autenticación de usuario. O en caso contrario, seleccionar *maestra*, que permite modificar la contraseña maestra de acceso.

Una vez se han introducido los campos necesarios, permite comprobar el nivel de seguridad de la contraseña introducida.

El requisito funcional completado es RF-17.

← Mi perfil ☰

Personal Maestra

Contraseña maestra

\*\*\*\*\*

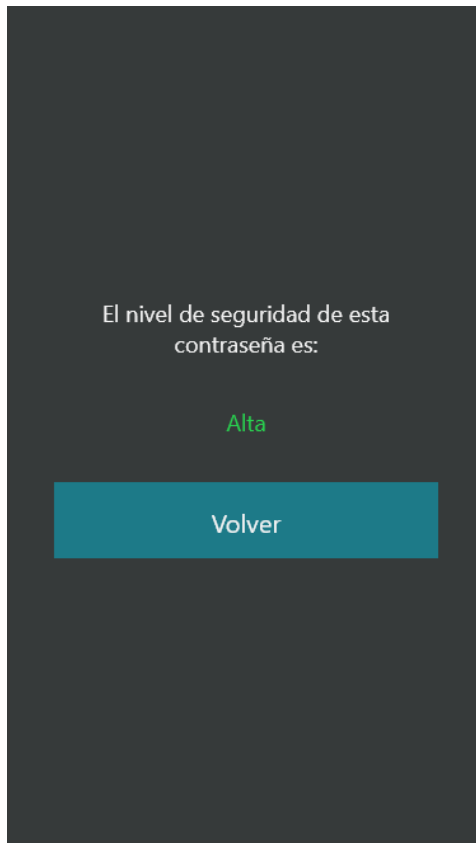
\*\*\*\*\*

\*\*\*\*\*

[Comprobar nivel de seguridad](#)

Aceptar cambios

## COMPROBAR NIVEL SEGURIDAD CONTRASEÑA



Pantalla que permite la *comprobación del nivel de seguridad*.

Informa del nivel de seguridad de la contraseña introducida.

El requisito funcional completado es RF-18.

## SOPORTE Y AYUDA



Pantalla que muestra las distintas opciones de *soporte y ayuda*.

El desarrollo del prototipo interactivo se ha realizado utilizando la aplicación *Adobe XD*. Para ello, ha sido necesario registrarse con la cuenta de estudiante de la Universidad de Zaragoza, haciendo uso de la versión gratuita de Adobe.

El prototipo se ha desarrollado intentando cuidar los detalles y el diseño al máximo, dotando así a la propia interfaz de un comportamiento lo más similar al esperado con el desarrollo planteado. Siguiendo la línea de trabajo planteada en este documento, la realización del diseño del prototipo se ha llevado a cabo de manera incremental, incorporando pantallas nuevas a la GUI hasta alcanzar y completar los requisitos planteados.

Se ha incorporado un perfil de usuario en el que se pueden visualizar y modificar los datos personales y gestionar datos de carácter sensible como son las contraseñas. El sistema también incorpora una pantalla de ayuda con información de contacto con el soporte de *Allpass*.

La comprobación del nivel de seguridad de una contraseña se considera un requisito importante ya que el sistema desarrollado pretende facilitar la gestión de contraseñas personales de un usuario, así como garantizar su seguridad digital.

Se ha hecho uso de una contraseña maestra como método de validación de usuario para el acceso a las distintas contraseñas e inicios de sesión almacenados desde los distintos dispositivos. La validez de la sesión de usuario tiene una duración de cinco minutos.

El prototipo de interfaz interactiva desarrollado ha intentado simular al máximo el comportamiento del sistema que se implementará, un ejemplo de ello es la comprobación de campos no nulos al introducir datos por parte del usuario. Aun así, se han incorporado pantallas en las que el sistema no ejecuta alguna de las opciones u operaciones que se indica, ya que esto implicaría ejecutar un bucle de comportamiento que no tiene ningún sentido realizar.

También tras simular el comportamiento de una acción concreta, hay pantallas que te remiten a una pantalla vista anteriormente en la que ese comportamiento simulado da la impresión de que ha dejado de realizarse. Esto no es así, el motivo es el mismo que se acaba de indicar, evitar un bucle infinito de comportamiento que no tendría ningún sentido en un prototipo.

Un ejemplo concreto de lo descrito anteriormente es el siguiente; tras intentar eliminar la contraseña/inicio de sesión relativo a *Instagram*, si se procede a visualizar cualquier otra contraseña/inicio de sesión, no se permite su eliminación, modificación o generación de nueva contraseña, evitando así un bucle de comportamiento. Por el contrario, será necesario retroceder haciendo uso de “atrás”, o accediendo de nuevo a “Mis contraseñas” en el menú de opciones, lo que mostrará la pantalla inicial en la que vuelve a aparecer *Instagram*.

Cabe mencionar, que tras eliminar la contraseña/inicio de sesión relativo a *Instagram*, al acceder a cualquier otra contraseña almacenada, se muestra la información relativa a *Facebook*. Esto se ha realizado para evitar desarrollar una pantalla distinta para cada una de las contraseñas guardadas en el sistema, ya que se entiende el funcionamiento del prototipo sin distinguir cada una de las distintas contraseñas que *Allpass* es capaz de almacenar.

El diseño del sistema final entregado al cliente será idéntico al funcionamiento mostrado en la GUI interactiva, evitando las pantallas sin navegación y las simulaciones que se acaban de comentar, ya que en la versión desarrollada esto se evitará mediante la implementación del código fuente.

## 6. Conclusiones

*Allpass* se ha desarrollado siguiendo un proceso preestablecido, siendo la base de éste un plan de análisis y gestión inicial. Tras su desarrollo y aprobación por todos los integrantes del grupo, se ha necesitado de la aprobación de la plantilla técnica y presupuesto por el cliente.

A partir de ahí, se ha continuado con un desarrollo e implementación incremental en la que un buen análisis de requisitos y un desarrollo de una buena GUI interactiva han sido elementos clave.

Se ha considerado bastante relevante la necesidad de tener un jefe de proyecto que se encargue de todos los asuntos internos de la organización, así como la planificación y distribución inicial de tareas para cada uno de los integrantes que forman el grupo de *Queen Consolidated S.L.* Todos estos aspectos se han interiorizado gracias a la realización de este proyecto, ya que no se había realizado anteriormente ningún proyecto de desarrollo de software de estas características.

El producto final entregado al cliente apenas difiere de lo planteado en la versión final de la GUI interactiva realizada con *Adobe XD*. Para su realización fue necesario un análisis exhaustivo del alcance y requisitos necesarios para el desarrollo de la aplicación, algo que permite, tras un buen diseño de los elementos y pantallas involucradas en la interfaz, se proporcione una GUI sencilla e intuitiva.

El software desarrollado es un software de calidad, ya que cumple todos los requisitos planteados y aceptados por el cliente en el momento de la firma del contrato.

Respecto a las habilidades y aprendizaje obtenido por los miembros del equipo, todos coinciden en la necesidad de utilizar una herramienta como Travis CI para la realización de pruebas de sistema a proyectos de integración continua como es el caso que se está detallando. También coinciden en la facilidad mostrada por la tecnología *React* para el desarrollo de software tanto para *iOS* como para *Android*. Respecto al despliegue del software en *cloud* mediante *Microsoft Azure*, los miembros del equipo aseguran haberse visto sorprendidos por la rapidez encontrada en este proceso, ya que al haber planteado una entrega en distintas versiones y realizar un despliegue incremental tras revisar una versión estable, el software desplegado en la nube ha sido un software de calidad acorde a la versión en la que se encontraba el proyecto en *GitHub*.

La gestión y planificación del proyecto, se ha realizado para una duración de cuatro meses y una estimación horaria de 600-700 horas de trabajo. Se puede afirmar que el diagrama de Gantt planteado ha sido de gran ayuda a la hora de gestionar las distintas tareas a realizar. Cabe mencionar que, de cara a próximos proyectos, los miembros de *Queen Consolidated S.L* intentarán anticipar la fase de realización de pruebas.

Los integrantes del grupo afirman estar satisfechos con el trabajo realizado, creyendo sinceramente que el producto entregado al cliente es un trabajo correcto, de calidad y acorde a los requisitos planteados en el marco de la asignatura Proyecto Software.



Para concluir, respecto a la realización de este trabajo en la convocatoria de septiembre, se han seguido las bases dictadas por el personal docente de la asignatura Proyecto Software. Cabe mencionar que, respecto a los aspectos y directrices no especificadas o que no quedaban muy claras en los distintos guiones y documentos, se ha optado por la propia interpretación del alumno con el mejor criterio y la mejor de las intenciones.