Profinder: una aplicación móvil para gestionar servicios de expertos profesionales.
Profinder: a mobile app to manage professional expert's services

Trabajo de Fin de Grado
Curso 2023–2024

Autor
Jaime Pablo Vázquez Martín

Director
Antonio Sarasa Cabezuelo

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Profinder: una aplicación móvil para gestionar servicios de expertos profesionales.
Profinder: a mobile app to manage professional expert's services

Trabajo de Fin de Grado en Ingeniería Informática

Autor
Jaime Pablo Vázquez Martín

Director
Antonio Sarasa Cabezuelo

Convocatoria: *Junio 2024*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

27 de Mayo de 2024

Dedicatoria

A mis padres Pablo y Elena, por siempre apoyarme y ayudarme, sin ellos nada de esto habría sido posible.

Agradecimientos

A Antonio Sarasa, mi tutor por estar siempre disponible. A Jorge Roselló, que me ayudó a iniciarme en lo que ha sido y será mi viaje en el mundo del desarrollo Android. A Aristides Guimerá (Aristidevs), creador de contenido online sin cuyos vídeos y cursos no habría avanzado tan comodamente en mi proceso de aprendizaje. Y a Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín por facilitar esta plantilla tan útil.

Resumen

Profinder: una aplicación móvil para gestionar servicios de expertos profesionales.

Profinder es una aplicación que ha sido desarrollada para dispositivos con sistema operativo Android. Busca ser un sistema que ayude a profesionales de distintos campos a ponerse en contacto con nuevos clientes, mantener conversaciones con ellos y crear una reputación mediante la calificación que los clientes creen al terminar los servicios realizados por el profesional, a su vez los clientes serán calificados por los profesionales. Los usuarios podrán utilizar la funcionalidad de mapa para comprobar qué profesionales están trabajando por la zona, agilizando así los tiempos de espera. Tanto profesionales como clientes podrán ser agregados a una lista de favoritos para tener fácil acceso a estos si fuera necesario. Tanto profesionales como servicios estarán clasificados dentro de categorías que permitirán una mejor gestión y organización.

La aplicación ha sido desarrollada siguiendo las mejores prácticas posibles, tanto a nivel de código como de arquitectura. Se ha utilizado kotlin como lenguaje de programación de la aplicación (más moderno, seguro, y eficiente que java). Para las interfaces se ha utilizado el kit de herramientas de Jetpack Compose diseñado para simplificar el desarrollo de IU.

Palabras clave

profesional, usuario, servicio, mapa, chat, favoritos, categoría, android.

Abstract

Profinder: a mobile app to manage professional expert's services

Profinder is an application developed for devices running the Android operating system. It aims to serve as an app that assists professionals from various fields in connecting with new clients, engaging in conversations with them, and building a reputation through client ratings upon completion of services. Simultaneously, clients will be rated by professionals. Users can utilize the map functionality to check which professionals are working in their area, thus streamlining waiting times. Both professionals and clients can be added to a favorites list for easy access if needed. Professionals and services will be categorized for better management and organization.

The application has been developed following the best possible practices, both in terms of code and architecture. Kotlin has been used as the programming language for the application (which is more modern, secure, and efficient than Java). Jetpack Compose toolkit has been used for the interfaces, designed to simplify UI development.

Keywords

professional, user, service, map, chat, favourites, category, android.

Índice

1. Introducción **1**
1.1. Motivación 1
1.2. Objetivos 2
1.3. Plan de trabajo 3

2. Estado de la Cuestión **5**
2.1. Uber 5
2.2. Habitissimo 6
2.3. Booksy 6
2.4. Upwork 7

3. Tecnologías empleadas **9**
3.1. Aplicaciones y herramientas generales 9
3.1.1. Android Studio 9
3.1.2. Obsidian 9
3.1.3. Figma 10
3.1.4. Drawio 10
3.1.5. Git, Github y Lazygit 10
3.2. Tecnologías específicas de Android 11
3.2.1. Kotlin 11
3.2.2. Jetpack Compose 12
3.2.3. Testing 14
3.2.4. Dagger Hilt 14
3.2.5. Maps 14
3.2.6. Permisos y localización 14
3.2.7. Compose Destinations 14
3.2.8. Coil 14
3.2.9. Material design 14

4. Justificación del diseño y consideraciones técnicas fundamentales **15**
4.1. Código abierto 15
4.2. Idiomas 15

4.3. Arquitectura y patrones 15
4.3.1. CLEAN architecture 15
4.3.2. MVI 15
4.4. Gestión de errores 15

5. Modelo de datos **17**
5.1. Firebase 17
5.1.1. Authentication 17
5.1.2. Firestore 17
5.1.3. Realtime Database 17
5.1.4. Storage 17
5.2. Datastore 17
5.3. Uso del patrón Singleton como modelo de persistencia 17

6. Conclusiones y Trabajo Futuro **19**

Introduction **21**

Conclusions and Future Work **23**

Bibliografía **25**

A. Especificación de requisitos **27**
A.1. Actores 27
A.2. Casos de Uso 27
A.2.1. Casos de uso generales 28
A.2.2. Casos de uso de usuarios 35
A.2.3. Casos de uso de profesionales 39
A.2.4. Casos de uso de administrador 46

B. Interfaces de Profinder diseñadas en figma **51**

C. Diseño de las bases de datos **53**

Índice de figuras

1.1. Diagrama de Gantt del proyecto 3

2.1. Logotipo Uber 5
2.2. Logotipo Habitissimo 6
2.3. Logotipo Booksy 6
2.4. Logotipo Upwork 7

3.1. Función en Java 12
3.2. Función en Kotlin 12
3.3. Recycler View utilizando Android views (Khan, 2023) 13
3.4. LazyColumn utilizando Jetpack Compose 14

Índice de tablas

A.1. Registro/baja 28
A.2. Modificar datos 29
A.3. Login/Logout 30
A.4. Configurar app 31
A.5. Ver lista de favoritos 32
A.6. Valorar cliente/profesional 33
A.7. Chatear con profesional 34
A.8. Configurar búsqueda 35
A.9. Buscar servicio 35
A.10.Chatear con profesional 36
A.11.Contratar servicio 37
A.12.Contratar servicio 38
A.13.Cambiar estado de profesional 39
A.14.Dar de alta/baja servicio 40
A.15.Modificar servicio 41
A.16.Listar servicios dados de alta 42
A.17.Contestar solicitud de contratación. 43
A.18.Consultar cliente. 44
A.19.Añadir/Modificar/Quitar cliente de lista de favoritos. 45
A.20.Añadir/eliminar/Modificar categorías de servicios ofrecidos. 46
A.21.Consultar datos/estadísticas de profesional/clientes. 47
A.22.Buscar clientes/profesionales. 48
A.23.Modificar datos de clientes/profesionales/servicios ofrecidos. 49
A.24.Dar de baja usuarios. 50

Capítulo 1

Introducción

"We are stubborn on vision. We are flexible on details..."
— Jeff Bezos

1.1. Motivación

Profinder nace del concepto de agilizar y simplificar las relaciones entre profesionales que ofrecen un servicio y usuarios que están dispuestos a consumirlo.

Se trata de una aplicación móvil, inicialmente desarrollada para el sistema operativo android (pese a esto no se descarta la posibilidad de poder hacerla multiplataforma en un futuro, véase el capítulo 6) en la que tanto usuarios como profesionales se podrán registrar, creando una cuenta e interactuar entre ellos. Las funcionalidades para cada tipo de actor varían en distintos aspectos, manteniendo algunas funcionalidades comunes.

A continuación se explica la idea de flujo de publicación, contratación y calificación de servicios de la aplicación:

1. Al registrarse, los profesionales seleccionaran la categoría de profesional a la que pertenecen -esta categoría podrá ser modificada en cualquier momento desde la pantalla de Éditar perfil'-.

2. Una vez registrados, tendrán la posibilidad de crear servicios que a su vez estarán clasificados en categorías y podrán ser públicos o privados (activos o inactivos).

3. Los servicios activos aparecerán a los usuarios que podrán solicitarlos ya sea desde la pantalla de listado de servicios o a través del perfil del profesional.

4. Al crear una solicitud un usuario, esta le aparece al profesional que podrá aceptarla o rechazarla. En caso de aceptarla, el trabajo se pone en marcha y en adelante hasta que termine se mostrará como trabajo activo.

5. Una vez terminado el trabajo, el profesional lo marcará como tal y calificará con estrellas (del 1 al 5) al usuario. Para el profesional el trabajo habrá terminado.

6. Por último al usuario le aparecerá la opción de calificar al profesional de la misma forma. Una vez hecho esto, el trabajo se marca como completado y se da por terminado el flujo de servicos.

A parte de este flujo en el que cada actor tiene un papel marcado. Hay cierta funcionalidad común, descrita a continuación. Usuarios y profesionales podrán chatear entre sí usando la pantalla de chat, donde podrán concretar fechas y horarios concretos y detalles adicionales. Todos los usuarios y profesionales podrán ser añadidos a listas de favoritos para un fácil acceso a sus perfiles. El perfil de cada actor en la aplicación podrá ser completado con atributos como una descripción o una foto de perfil.

1.2. Objetivos

El mundo del desarrollo android es inmenso, la forma de diseñar interfaces respecto a la programación web muy distinta, y se trata de un mundo que en los últimos años ha estado en constante cambio, cada año salen nuevas tecnologías que dejan obsoleta la que ya había e incluso hace unos años cambió el lenguaje de programación usado para crear aplicaciones android.

El objetivo principal de este trabajo es aprender muchas de las cosas necesarias para ser un desarrollador android competente, partiendo de cero hasta llegar a ser capaz de crear una aplicación robusta, bonita y escalable en el tiempo.

Los objetivos relativos a la aplicación marcados desde el comienzo se muestran a continuación:

- definición de una especificación de requisitos inicial, sujeta a cambios a lo largo del proceso de desarollo, en la que se definen los actores y casos de uso de la aplicación (vease el apéndice A).
- Una vez definida la especificación de requisitos, se marcarán una serie de hitos para la entrega de las funcionalidades.

También se han definido una serie de objetivos técnicos que serán llevados a cabo en paralelo a la consecución de los objetivos relativos a la aplicación:

- Aprender kotlin hasta alcanzar un nivel de competencia en el lenguaje apto para el desarrollo.
- Empezar en el desarrollo android utilizando el sistema de vistas [1], forma clásica de desarrollar interfaces en android, que sirve como primer acercamiento pese a que luego se sustituya por Jetpack Compose[2].
- Realizar un acercamiento a las arquitecturas, patrones de diseño y buenas prácticas más utillizadas con el objetivo de empezar a hacer proyectos más robustos que se acerquen a la estructura de proyecto final.

[1] Artículo que hace una introducción al sistema de vistas: `https://www.studytonight.com/android/introduction-to-views`

[2] `https://developer.android.com/develop/ui/compose`

- Aprender los fundamentos de Jetpack Compose e ir adquiriendo nuevos conocimientos progresivamente haciendo distintos proyectos de prueba.
- Familiarizarse con las distintos servicios de Firebase [3] para aplicaciones android que serán usados como backend en la aplicación final.
- Al final del proceso de desarrollo de la aplicación se espera haber obtenido una base de conocimientos sólidos que sirvan como punto de partida de una posible carrera profesional futura.

1.3. Plan de trabajo

Para la consecución de los objetivos descritos en el apartado anterior, se ha establecido al inicio del proyecto un plan de trabajo representado con un diagrama de Gantt, los objetivos se han dividido -igual que en el apartado anterior- en objetivos técnicos y de aplicación ya que la idea inicial es ir cumpliendo los dos tipos de objetivos en paralelo en el tiempo. Respecto a los objetivos hitos del proyecto, a

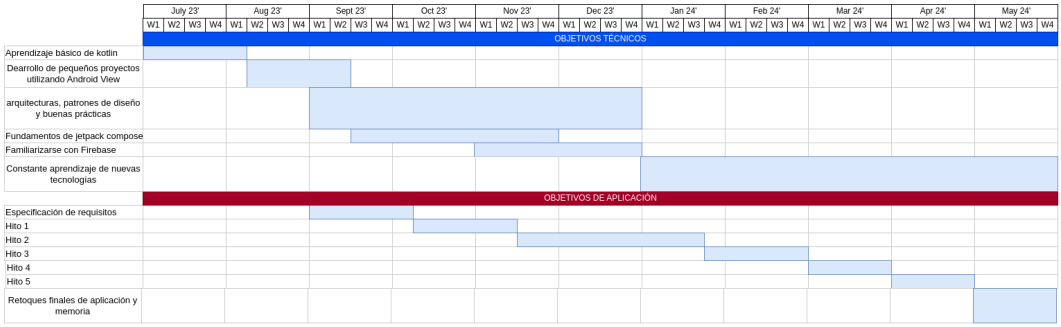

Figura 1.1: Diagrama de Gantt del proyecto

continuación se muestra una lista con los casos de uso a desarrollar en cada hito:

- Hito 1
  - Registro/baja de usuario.
  - Modificar datos usuario.
  - Login/Logout de usuario.
  - Configurar app.
- Hito 2
  - Cambiar estado de profesional.
  - Dar de alta/baja servicio.
  - Modificar servicio.
  - Listar servicios dados de alta.

[3] `https://firebase.google.com/`

    - Añadir/eliminar/Modificar categorías de servicios ofrecidos.

- Hito 3
    - Buscar servicio.
    - Configurar búsqueda.
    - Consultar profesional.
    - Consultar cliente.
    - Añadir/Quitar Profesional de lista de favoritos.
    - Añadir/Modificar/Quitar cliente de lista de favoritos.
    - Ver lista de favoritos.
- Hito 4
    - Contratar servicio.
    - Chatear con profesional/cliente.
    - Valorar cliente/profesional.
    - Contestar solicitud de contratación.
- Hito 5
    - Consultar datos/estadísticas de profesional/clientes.
    - Buscar clientes/profesionales.
    - Modificar datos de clientes/profesionales/servicios ofrecidos.
    - Dar de baja usuarios.

Debido a que los casos de uso eran una aproximación inicial, según ha ido avanzando el proceso de desarrollo, algunos casos de uso se han implementado antes, después, o se han descartado por no encajar bien en la aplicación. A pesar de esto, la planificación inicial ha sido de gran utilidad como guía y ha servido para marcar correctamente los plazos de entrega, asimilando el proceso a como sería en un entorno real.

Capítulo 2

Estado de la Cuestión

En este capítulo se ha realizado un analísis de otras aplicaciones en el mercado, viendo sus características generales y comprobando si se considerarían o no competencia de Profinder en el caso de que se tuviera la idea de comercializar en un futuro.

En todos los casos analizados se ha encontrado una diferencia significativa respecto a Profinder: todas son aplicaciones de código cerrado mientras que Prodinder ha sido concebida desde sus inicios como una aplicación de código abierto, con todo su código fuente y diseños abiertos al público general, en el que cualquiera pueda, aportar (como se explica más en detalle en el capítulo 4).

2.1. Uber

Figura 2.1: Logotipo Uber

Uber [1] es una aplicación muy conocida que sirve para contratar servicios de desplazamiento en V.T.C. (Vehículo de Transporte con Conductor), el motivo por el que se considera una aplicación relacionada con Profinder es que los servicios se contratan por proximidad, se utiliza la ubicación del usuario -al igual que se hace en Profinder- para encontrar los conductores más próximos, reduciendo así tiempos de espera y costes de desplazamiento innecesario. Uber se ha utilizado como una de las aplicaciones de referencia ya que representa muy bien el concepto que se ha buscado desde el principio con Profinder.

[1] `https://www.uber.com`

2.2. Habitissimo

Figura 2.2: Logotipo Habitissimo

Habitissimo [2] es una aplicación que opera principalmente en España e Italia.En la que distintos profesionales pueden publicar ofertas de servicios, que los usuarios pueden contratar pidiendo un presupuesto. Es la posible competencia más directa que se ha encontrado.

Esta aplicación está epsecificamente enfocada al sector de la reforma y reparación, a diferencia de Profinder, que no está enfocada a ningún sector particular, sino que cuenta con distintas categorías que pueden ir ampliándose y ajustándose con el tiempo. Profinder también busca diferenciarse por ser una aplicación global, al ser una aplicación que no depende de una infraestructura por países ya que no ofrece un servicio final sino una ayuda al contacto de usuarios y profesionales en su ámbito local. Asimismo, Profinder tiene un proceso más ágil que no depende de ningún tipo de tercero y en el que se puede tener un contacto más directo entre usuarios y profesionales, disminuyendo así posibles comisiones.

2.3. Booksy

Figura 2.3: Logotipo Booksy

Booksy[3] es un aplicación centrada en el sector de la estética, en la que se pueden encontrar distintos profesionales como peluqueros, maquilladores o masajistas. Operan principalmente en Estados Unidos y cuentan con una batería específica de profesionales.

Las principales diferencias con Profinder es que están enfocados solo en belleza, ofreciendo un rango de categorías menor, a su vez, Booksy tiene una gestión centralizada de profesionales y servicios, no teniendo libertad los primeros para gestionar su trabajo de la forma en que deseen. Esto es también una desventaja repecto a Profinder ya que al tener una gestión descentralizada es una aplicación escalable a nivel internacional.

[2] https://www.habitissimo.es
[3] https://booksy.com/en-us

2.4. Upwork

Figura 2.4: Logotipo Upwork

Upwork [4] es una plataforma que comenzó hace más de 20 años a operar, destinada a poner en contacto empresas con nuevo talento que busque trabajo, se situan como uno de los líderes en su sector, siendo este volumen alto de usuarios una de sus principales competencias frente a sus competidores.

El concepto de la aplicación tiene algunas cosas en común con Profinder en el sentido de crear un contacto directo entre contratante y contratado. Sin embargo, Upwork se mueve en el contexto empresarial, dejando de lado el particular, por ello no sería considerada competencia directa y pasaría más a un segundo plano en relación con este proyecto, sirviendo algunas de sus funcionalidades como referencia e inspiración.

[4] `https://www.upwork.com`

Capítulo 3

Tecnologías empleadas

En este capítulo se explican todas la tecnologías, aplicaciones y herramientas que se han utilizado durante el proceso de desarrollo. El capítulo se ha dividido en dos apartados principales, uno dedicado a tecnologías generales y otro dedicado a tecnologías específicas del desarollo Android.

3.1. Aplicaciones y herramientas generales

3.1.1. Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial utilizado para el desarrollo de aplicaciones en Android. Esta basado en Intellij Idea, un editor multiplataforma con soporte para lenguajes como Kotlin, Java y Scala entre otros.

Android Studio ha sido utilizado a lo largo del proceso completo de desarrollo, siendo de gran utilidad sus modernas funcionalidades como ayuda al correcto flujo de trabajo. Este entorno te permite utilizar emuladores para probar las aplicaciones, sin embargo estos emuladores consumen gran cantidad de recursos del ordenador, esto ha sido un poco problemático en el caso de este proyecto ya que el ordenador con el que se contaba para el desarrollo no contaba con recursos suficientes para ejecutar el emulador correctamente, provocando en algunos casos problemas de *lag* y *crashes* ocasionales. A pesar de esto, ha sido posible desarrollar la aplicación usando estos emuladores.

3.1.2. Obsidian

Obsidian es una aplicación de código abierto que sirve como sistema de notas, te permite organizar las mismas con gran cantidad de posibilidades. Cuenta con funcionalidades que han sido de gran utilidad como la referenciación entre notas, vista de grafo y multitud de extensiones desarrolladas por la comunidad.

En este proyecto, Obsidian ha servido como complemento al desarrollo. Se ha utilizado, entre otras cosas, para la planificación de hitos (apuntes personales sobre los casos de uso, fechas de entrega, dudas para tutoría...), reunión de ideas sobre las funcionalidades y documentación del ciclo de vida del proyecto.

3.1.3. Figma

Figma es una plataforma de diseño basada en la web que permite crear y prototipar diseños de interfaces de usuario, con ella ha sido posible crear diseños de una forma rápida y eficaz, que luego han sido materializados en Jetpack Compose.

En este proyecto en particular, figma ha sido utilizada para plasmar las ideas de diseño, sirviendo estas como una referencia no estricta de las interfaces finales. En el apéndice B se han adjuntado capturas de pantalla de todas las interfaces diseñadas. Como la aplicación se ha desarrollado utilizando Material 3 y Dynamic Colors (véase la subseccion 3.2.9 para más detalles acerca de Material design) los colores varían dinámicamente en función del tema de cada dispositivo por lo que los colores de los diseños de figma no tienen por qué corresponderse con los colores que luego se han materializado en cada dispositivo.

3.1.4. Drawio

Drawio es una herramienta de código abierto que permite a los usuarios crear una amplia variedad de diagramas (flujo, Gantt...), seleccionada para el proyecto por ser cómoda, personalizable y con gran capacidad de portabilidad, utilizando ficheros con extensión .io y dando la capacidad al usuario de cambiar de herramienta con facilidad.

Esta aplicación ha sido utilizada para la realización de todos los diagramas que han sido necesarios durante el proyecto. Ha sido especialmente útil para el diseño de las bases de datos del proyecto (véase el apéndice C con los diseños) ya que, al haberse utilizado bases de datos no SQL, ha sido necesario depurar el diseño en varias ocasiones para evitar la duplicación lo máximo posible.

3.1.5. Git, Github y Lazygit

Para el control de versiones de la aplicación se ha utilizado la conocida herramienta utilizada por la gran mayoría de los desarrolladores de software a nivel mundial: Git. Pese a haber sido un proyecto desarrollado por una sola persona, Git ha sido una herramienta esencial para el proceso de desarrollo, sirviendo como backup en el desarrollo de nuevas funcionalidades y como control general del estado de la aplicación.

El repositorio con el código de la aplicación y el código latex utilizado para la redacción de esta memoria se han almacenado de forma pública en Github, plataforma de desarrollo colaborativo basada en la web que utiliza Git como sistema de control de versiones.

También merece mención en este apartado la herramienta de código abierto Lazygit, interfaz gráfico de terminal para linux que proporciona una forma fácil y visual de interactuar con los repositorios Git. Está diseñado para simplificar el flujo de trabajo al proporcionar una interfaz gráfica intuitiva dentro de la terminal. Ha sido de gran ayuda para la agilización del uso de Git a lo largo del desarrollo del proyecto en conjunción con la extension de Android Studio para Git.

3.2. Tecnologías específicas de Android

3.2.1. Kotlin

Kotlin es un lenguaje de programación de código abierto creado por JetBrains, es un lenguaje de tipado estático, lo que significa que se puede desarrollar sobre la JVM (Java Virtual Machine) y es totalmente compatible e interoperable con Java lo que facilita la migración desde el primero [1].

Este lenguaje de programación ha sido seleccionado para el proyecto ya que, a parte de ser el recomendado por Google y ser muy cómodo y útil de usar, con el paso de los años se está consolidando como el lenguaje de referencia para el desarrollo Android, dejando obsoleto el anterior lenguaje de referencia: Java.

3.2.1.1. Kotlin vs. Java

Kotlin empezó a considerarse como una opción viable para el desarrollo Android en 2017, cuando Google le empezó a dar soporte. En ese momento, Java era el lenguaje universal utilizado para el desarrollo de aplicaciones en Android. A partir de este punto, en los siguientes años comenzó el debate de si Kotlin era mejor alternativa. En estos últimos años este debate ha quedado prácticamente resuelto en favor de Kotlin, que ha demostrado ser un lenguaje mucho más moderno y útil dejando obsoleto a Java.

Algunas de las características -entre muchas otras- que hacen a kotlin destacar por encima de java son:

- Una sintaxis más moderna más moderna, que permite hacer más con menos código.
- Al ser el lenguaje principal utilizado para Android hoy en día ofrece un mayor -y más actualizado- número de bibliotecas, más comunidad y recursos diponibles.
- Inmutabilidad de variables: una variable no puede cambiar a no ser que se indique explicitamente que sí. Esto ahorra mucho posibles errores.
- Kotlin está diseñado ser un lenguaje *null-safe*, es decir ofrece seguridad sobre valores nulos (¡No más NullPointerException!). Esto quiere decir que los valores no pueden ser nulos por defecto, para serlo se debe indicar expresamente con el operador '?'.
- Relativo a Android, ofrece soporte nativo para corrutinas y flows, muy útiles en el desarrollo de aplicaciones.

A continuación se muestra una breve comparación del mismo código -que declara una variable, sumándola a un número entero, guardándola en otra variable y mostrándola por consola- en los dos lenguajes para intentar ilustrar algunas de las diferencias entre ambos:

[1] Artículo que explica en detalle qué es kotlin: Plain Concepts

```java
void javaFunction() {
    Integer exampleVariable = null;

    int exmpleSum = 3 + exampleVariable; //<- NullPointerException

    System.out.println(exmpleSum); //Crash
}
```

Figura 3.1: Función en Java

```kotlin
fun kotlinFunction(){
    val exampleVariable: Int = null // ← No puede ser nulo (error de compilación)
                                    //    inmutabilidad;  no más punto y comas;
    val exampleSum :Int = 3 + exampleVariable //← inferencia del tipo (Int);
                                              //← Sabemos que nunca saltará NullPointerException
                                              //   porque exampleVariable no puede ser nulo y es
                                              //   inmutable

    println(exampleSum) //Sintaxis más sencilla
}
```

Figura 3.2: Función en Kotlin

3.2.2. Jetpack Compose

Jetpack Compose es un moderno kit de herramientas de desarrollo de interfaces de usuario (UI) para Android, desarrollado por Google. Permite a los desarrolladores construir interfaces de usuario de aplicaciones de forma más fácil y eficiente, utilizando un enfoque declarativo. Jetpack Compose fue elegida como tecnología principal para el desarrollo de las interfaces porque suponía un reto apasionante, es una tecnología puntera y muy demandada actualmente en el mundo profesional. Por estos motivos, desde el principio del desarrollo de Profinder se decidió utilizar.

Jetpack Composes es una forma más cómoda, sencilla y mejor de desarrollar aplicaciones frente al sistema tradicional de vistas con XML. A continuación se intenta explicar el por qué de esta afirmación.

3.2.2.1. Jetpack Compose vs. Android views

Android Views son el enfoque tradicional para construir interfaces de usuario en aplicaciones Android. Se definen principalmente utilizando archivos XML y se gestionan programáticamente en Java o Kotlin. Esta era la forma de desarrollar interfaces en Android hasta que a partir de 2019, cuando salió Jetpack Compose en modo *preview*, ambas fueron poniendose a la par. Fue en 2021 cuando Google estableció Jetpack Compose como la forma recomendada de aprender a desarrollar aplicaciones en Android. Hoy en día, Jetpack Compose está totalmente asentado y es un hecho que cada vez va dejando más atrás el sistema de vistas y es algo lógico puesto qeu presenta numerosas ventajas:

- Todo en Kotlin: mientras que usando Views las interfaces se definen en XML y se manipulan programáticamente en kotlin -o Java, pero no lo tendremos en

cuenta para esta comparativa debido a que ya ha sido realizada en la seccion 3.2.1.1- en Jetpack Compose se unifica todo en kotlin. Esto reduce enormemente la cantidad de código necesaria y hace el código mucho más entendible

- Jetpack Compose ofrece un estilo de programación declarativa y reactiva que se adapta mucho mejor al paradigma de programación funcional de Kotlin, apoyándose mucho por ejemplo, en el uso de lambdas para añadir robustez al código.

- Debido a que Jetpack Compose es hoy en día la principal forma de programar interfaces, constanmente van saliendo actualizaciones y nuevas utilidades como el *live edit*, que permite ver los cambios en diseño de tu aplicación en tiempo real mientras que esta está ejecutandose; o las *previews* que permiten comprobar el diseño en distintos dispositivos y tamaños.

- Jetpack Compose permite programar para distintos tipos de dispositivos, esto significa que se podría por ejemplo, crear en el mismo proyecto una aplición de móvil su versión de wearable y su versión de escritorio sin más complicación que la que tuviero adaptar las interfaces a cada tamaño.

Para terminar esta comparativa, a continuación se muestran dos formas de hacer exactamente lo mismo (una columna que renderice y muestre los elementos en pantalla) utilizando las dos formas analizadas en este apartado. Como se puede comprobar lo que se hace en 6 líneas de código en Jetpack Compose necesita de tres clases en el sistema de vistas (Adapter, Viewholder y Activity):

Figura 3.3: Recycler View utilizando Android views (Khan, 2023)

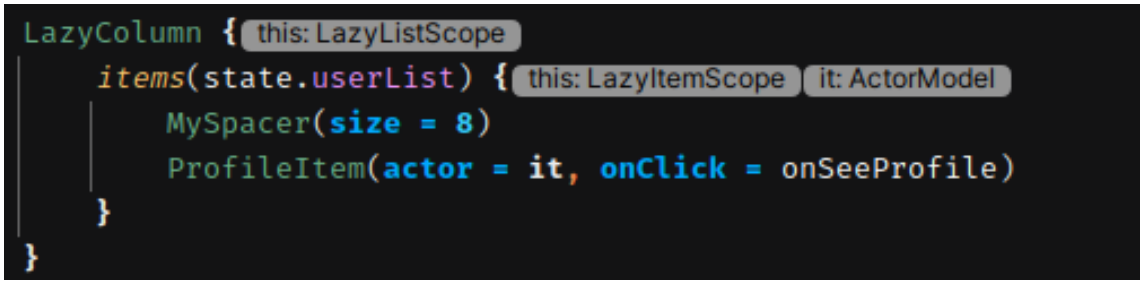

Figura 3.4: LazyColumn utilizando Jetpack Compose

3.2.3. Testing

3.2.4. Dagger Hilt

3.2.5. Maps

3.2.6. Permisos y localización

3.2.7. Compose Destinations

3.2.8. Coil

3.2.9. Material design

Capítulo 4

Justificación del diseño y consideraciones técnicas fundamentales

hablar entre otros de: error handling, mvi, clean, código abierto, idiomas

4.1. Código abierto

4.2. Idiomas

4.3. Arquitectura y patrones

4.3.1. CLEAN architecture

4.3.2. MVI

4.4. Gestión de errores

Capítulo 5

Modelo de datos

5.1. Firebase

5.1.1. Authentication

5.1.2. Firestore

5.1.3. Realtime Database

5.1.4. Storage

5.2. Datastore

5.3. Uso del patrón Singleton como modelo de persistencia

Capítulo 6

Conclusiones y Trabajo Futuro

Patashnik (1988)

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.

trabajo futuro: refactor mvi, notificaciones, animaciones, rol de administrador

Introduction

Introduction to the subject area. This chapter contains the translation of Chapter 1.

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter 6.

URLs referenciadas

- https://developer.android.com/studio/intro
- https://www.jetbrains.com/idea
- https://www.java.com
- https://www.scala-lang.org
- https://obsidian.md
- https://www.figma.com
- https://m3.material.io
- https://m3.material.io/styles/color/system/overview
- https://www.drawio.com/
- https://git-scm.com/
- https://github.com/
- https://github.com/jesseduffield/lazygit
- https://kotlinlang.org/
- https://www.plainconcepts.com/es/kotlin-android/
-
-
-
-
-
-
-
-
-
-
-
-
-
-

Bibliografía

Y así, del mucho leer y del poco dormir, se le secó el celebro de manera que vino a perder el juicio.
(modificar en Cascaras\bibliografia.tex)

Miguel de Cervantes Saavedra

Khan, M. H. Recyclerview vs. lazyrow/lazycolumn: Choosing the right layout for dynamic lists in android. *Medium*, 2023. Disponible en `https://medium.com/@humzakhalid94/recyclerview-vs-lazyrow-lazycolumn-choosing-the-right-layout-for-dynamic-lists-i` (último acceso, Mayo, 2024).

Patashnik, O. BibTeXing. 1988. Disponible en `ftp://ftp.ctan.org/tex-archive/biblio/bibtex/contrib/doc/btxdoc.pdf` (último acceso, Octubre, 2009).

Apéndice A

Especificación de requisitos

En este apendice se especifican los actores y casos de uso de la aplicación Profinder. Se trata de una especificación inicial por lo que tanto actores como casos de uso han sido cambiados, ampliados o recortados a lo largo del proceso de desarrollo.

A.1. Actores

A continuación se describen los distintos actores de la aplicación:

- **Usuario**: Un usuario es cualquier persona que se dé de alta como tal, debe haberse creado una cuenta e iniciado sesión para ser identificado, estos son los que contratan los servicios que los profesionales ofrecen, entre otras funciones los usuarios pueden ver el mapa de profesionales en tiempo real y filtrar los datos que le aparecen según la categoría de trabajo seleccionada.
- **Profesional**: Un profesional es cualquier persona que se dé de alta como tal, su proceso de registro es ligeramente diferente al de un usuario normal ya que deben especificar la categoría en la que son profesionales, así como otros detalles que sean importantes para su profesión.
- **Administrador**: El administrador de la aplicación es el que se encarga de la organización y el correcto funcionamiento de la misma, podrá añadir o eliminar categorías, bloquear usuarios y/o profesionales, así como responder a mensajes dirigidos a soporte, los administradores deben ser asignados una vez creada la cuenta desde fuera de la aplicación.

A.2. Casos de Uso

Los casos de uso se han clasificado en cuatro categorías según el actor principal del mismo.

A.2.1. Casos de uso generales

Requisito	**Registro/baja**	
Identificador	1.1	
Prioridad	Alta	
Precondición	En caso de baja: tener una cuenta creada.	
Descripción	Los actores de la aplicación tendrán que crear una cuenta para poder interactuar con la misma, así mismo tendrán la capacidad de dar de baja su cuenta cuando lo deseen.	
Entrada	Nombre de usuario, correo electrónico, contraseña.	
Salida	Actor registrado/dado de baja	
Secuencia normal	Paso	Acción
	1	El actor abre la aplicación y se le muestra la pantalla para iniciar sesión con un botón específico para registrarse.
	2	se selecciona la opción 'Registrarse'.
	3	El actor rellena el formulario con los datos de entrada y pulsa 'Enviar'.
	4	El sistema procesa el formulario y crea la cuenta.
Postcondición	Se ha creado la cuenta.	
Excepciones	Paso	Acción
	4	Los datos introducidos son incorrectos o no cumplen los requisitos. No se crea la cuenta y se avisa al usuario.
Comentarios	Este caso de uso permite registrar o dar de baja de la aplicación a actores de la misma.	
Actores	Usuario, Profesional	

Tabla A.1: Registro/baja

Requisito	**Modificar datos**	
Identificador	1.2	
Prioridad	Media	
Precondición	Haber iniciado sesión.	
Descripción	Una vez creada una cuenta, tanto profesionales como usuarios podrán modificar sus datos de perfil.	
Entrada	Datos a cambiar.	
Salida	Nuevos datos.	
Secuencia normal	Paso	Acción
	1	El actor se dirigirá al apartado de 'Mi perfil'y seleccionará la opción 'Modificar perfil'.
	2	Dentro de esta pantalla se modificarán todos los datos deseados.
Postcondición	Se han modificado los datos de perfil deseados.	
Excepciones	Paso	Acción
	2	Los nuevos datos no son válidos. No se guarda la modificación.
Comentarios	Este caso de uso permite modificar datos de su perfil a los actores.	
Actores	Usuario, Profesional	

Tabla A.2: Modificar datos

Requisito	**Login/Logout**	
Identificador	1.3	
Prioridad	Alta	
Precondición	Haber creado una cuenta previamente, en caso de logout tener sesión iniciada.	
Descripción	Para poder interactuar con la aplicación, tanto usuarios como profesionales tendrán que iniciar sesión en la aplicación. Los usuarios que ya hayan iniciado sesión y quieran cerrarla tendrán la opción de hacerlo.	
Entrada	En caso de login: correo electrónico/nombre de usuario, contraseña.	
Salida	N.A.	
Secuencia normal de login	Paso	Acción
	1	El actor abre la aplicación y se le muestra la pantalla de 'Iniciar sesión'.
	2	Se muestra una pantalla con el formulario de inicio de sesión. El usuario introduce los datos de entrada y pulsa el botón 'Iniciar sesión'.
Secuencia normal de logout	Paso	Acción
	1	El actor se dirige a la sección 'Mi perfil' donde se le mostrarán múltiples opciones de gestión de su cuenta.
	2	Dentro de estas opciones se encontrará la opción 'Cerrar sesión'. El actor pulsará el botón.
Postcondición	Se ha iniciado sesión/se ha cerrado sesión.	
Excepciones	Paso	Acción
	2 (login)	Los campos rellenados no concuerdan con ningún actor del sistema. No se completa el login.
Comentarios	Este caso de uso permite iniciar sesión en la aplicación, así como cerrar la misma.	
Actores	Usuario, Profesional, Administrador	

Tabla A.3: Login/Logout

Requisito	**Configurar app**	
Identificador	1.4	
Prioridad	Baja	
Precondición	Haber iniciado sesión.	
Descripción	Dentro de la app se podrán configurar aspectos como el tema de la aplicación (claro/oscuro).	
Entrada	Nuevos datos de configuración.	
Salida	Configuración modificada.	
Secuencia normal	Paso	Acción
	1	El actor se dirigirá a la sección de 'Mi Perfil' y entre las opciones seleccionará 'Configuración de la aplicación'.
	2	Dentro de esta pantalla el usuario cambiará los valores deseados.
Postcondición	Se han cambiado los parámetros de la aplicación deseados.	
Excepciones	Paso	Acción
	2	Los nuevos parámetros de configuración no son correctos. Se mantiene la configuración anterior.
Comentarios	Este caso de uso permite modificar la configuración de la aplicación.	
Actores	Usuario, Profesional	

Tabla A.4: Configurar app

Requisito	**Ver lista de favoritos**	
Identificador	1.5	
Prioridad	Media	
Precondición	El Usuario o Profesional debe estar autenticado en la aplicación y tener al menos un elemento en su lista de favoritos.	
Descripción	Permite al Usuario o Profesional ver la lista de elementos marcados como favoritos (por ejemplo, Clientes o Profesionales) para un acceso rápido y conveniente.	
Entrada	Selección de la opción para ver la lista de favoritos.	
Salida	Lista de elementos marcados como favoritos con detalles relevantes.	
Secuencia normal	Paso	Acción
	1	El Usuario o Profesional accede a la sección de lista de favoritos.
	2	Selecciona la opción para ver la lista de favoritos.
	3	Visualiza la lista de elementos marcados como favoritos con sus detalles.
Postcondición	El Usuario o Profesional ve la lista de elementos marcados como favoritos.	
Excepciones	Paso	Acción
	2	Si la lista de favoritos está vacía, se muestra un mensaje indicando que no hay elementos en la lista.
Comentarios	N.A.	
Actores	Usuario, profesional	

Tabla A.5: Ver lista de favoritos

Requisito	**Valorar cliente/profesional**	
Identificador	1.6	
Prioridad	Alta	
Precondición	Haber iniciado sesión, haber recibido el trabajo de un profesional o haber realizado un trabajo a un usuario.	
Descripción	Una vez realizado/recibido un trabajo, los usuarios/profesionales podrán valorar a ese profesional/usuario así como denunciarlo si lo vieran necesario.	
Entrada	Datos de valoración, mensaje con más detalles (opcional)	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	Una vez recibido/realizado el trabajo, se mostrará la opción de valorar al profesional/usuario.
	2	El usuario/profesional rellena el formulario de valoración.
Postcondición	Se ha valorado al profesional/usuario.	
Excepciones	Paso	Acción
	2	Los datos de entrada no son correctos. No se completa la valoración.
Comentarios	N.A.	
Actores	Usuario, profesional	

Tabla A.6: Valorar cliente/profesional

Requisito	**Chatear con cliente/profesional**	
Identificador	1.7	
Prioridad	Media	
Precondición	Haber iniciado sesión, estar en el perfil de un profesional/-cliente.	
Descripción	En el perfil de los profesionales/clientes habrá un botón para iniciar un chat privado que servirá para aclarar dudas, pedir/dar presupuestos para cosas concretas, etc...	
Entrada	botón de chat, mensaje(s) a enviar.	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	Una vez en el perfil del profesional/usuario se pulsa el botón 'Chat privado'.
	2	Al hacerlo, se abre un chat entre usuario y profesional donde se podrán enviar mensajes.
	3	El usuario/profesional envía los mensajes deseados.
Postcondición	Se ha creado un chat privado entre usuario y profesional.	
Excepciones	N.A.	
Comentarios	Este caso de uso permite establecer una conversación entre usuario y profesional.	
Actores	Usuario, profesional	

Tabla A.7: Chatear con profesional

A.2.2. Casos de uso de usuarios

Requisito	**Configurar búsqueda**	
Identificador	2.1	
Prioridad	Media	
Precondición	Haber iniciado sesión.	
Descripción	A la hora de buscar el servicio de un profesional, los usuarios tendrán la opción de configurar los parámetros de la búsqueda como por ejemplo, seleccionar una categoría u ordenar por valoración.	
Entrada	Datos de configuración de búsqueda.	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	Dentro de la aplicación, el usuario seleccionará la opción 'Buscar profesional'.
	2	Se configurarán todos los parámetros de la búsqueda y se pulsará el botón 'Buscar'.
Postcondición	Se han configurado los parámetros de la búsqueda.	
Excepciones	N.A.	
Comentarios	Este caso de uso permite establecer unos parámetros determinados de búsqueda.	
Actores	Usuario	

Tabla A.8: Configurar búsqueda

Requisito	**Buscar servicio**	
Identificador	2.2	
Prioridad	Media	
Precondición	Haber iniciado sesión, haber configurado la búsqueda.	
Descripción	Una vez configurados los parámetros de búsqueda, se mostrarán los servicios que cumplen los parámetros de la búsqueda para que el usuario pueda realizar la búsqueda.	
Entrada	Servicio a buscar.	
Salida	Resultados de servicios que coinciden con la búsqueda.	
Secuencia normal	Paso	Acción
	1	El usuario podrá recorrer una lista con todos los servicios ofrecidos que cumplen los filtros establecidos.
Postcondición	Se ha realizado una búsqueda de servicio.	
Excepciones	N.A.	
Comentarios	Este caso de uso sirve para que los usuarios puedan buscar servicios de acuerdo con unos parámetros	
Actores	Usuario	

Tabla A.9: Buscar servicio

Requisito	**Consultar profesional**	
Identificador	2.3	
Prioridad	Media	
Precondición	Haber iniciado sesión, Hay profesionales disponibles.	
Descripción	Los usuarios tendrán la posibilidad de consultar el perfil de los distintos profesionales: sus datos (categoría, honorarios, etc...), sus valoraciones y otras estadísticas.	
Entrada	Profesional a consultar.	
Salida	Detalles del profesional consultado	
Secuencia normal	Paso	Acción
	1	El usuario realiza una búsqueda de profesional o escoge uno de los que se muestra en el mapa.
	2	Al pulsar en el profesional se abrirá su perfil con todos los datos de ese profesional.
Postcondición	Se ha consultado el profesional seleccionado.	
Excepciones	N.A.	
Comentarios	Este caso de uso permite obtener más detalles acerca de un profesional determinado.	
Actores	Usuario	

Tabla A.10: Chatear con profesional

Requisito	**Contratar servicio**	
Identificador	2.4	
Prioridad	Alta	
Precondición	Haber iniciado sesión.	
Descripción	Cuando un usuario encuentre el servicio que necesita tendrá la posibilidad de contratarlo y el profesional decidirá si tomar o no el trabajo.	
Entrada	Servicio elegido, respuesta del profesional.	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	Una vez seleccionado el profesional, en la pantalla del perfil del mismo habrá una opción llamada 'Contratar servicio', el usuario pulsará el botón.
	2	El sistema procesa y valida la contratación.
	3	La propuesta de trabajo llega al profesional que decide si la acepta o rechaza.
Postcondición	Se ha contratado el servicio.	
Excepciones	Paso	Acción
	2	Fallo en el proceso y validación de la petición. Se avisa al usuario y no se envía la propuesta.
Comentarios	N.A.	
Actores	Usuario	

Tabla A.11: Contratar servicio

Requisito	**Añadir/Quitar Profesional de lista de favoritos**	
Identificador	2.5	
Prioridad	Media	
Precondición	Haber iniciado sesión, estar en el perfil de un profesional.	
Descripción	Los usuarios tendrán la posibilidad de añadir o quitar a profesionales de su lista de favoritos, lista en la que podrán tener un acceso rápido a los perfiles de dichos profesionales.	
Entrada	Botón de añadir/quitar de favoritos.	
Salida	Lista de favoritos modificada.	
Secuencia normal	Paso	Acción
	1	Una vez dentro del perfil de un profesional, el usuario dispondrá de un botón de favoritos.
	2	Dependiendo de si quiere añadir o quitar al profesional de favorito lo activará o desactivará.
Postcondición	Se ha añadido/quitado un profesional en la lista de favoritos.	
Excepciones	N.A.	
Comentarios	N.A.	
Actores	Usuario	

Tabla A.12: Contratar servicio

A.2.3. Casos de uso de profesionales

Requisito	**Cambiar estado de profesional**	
Identificador	3.1	
Prioridad	Alta	
Precondición	Haber iniciado sesión como profesional.	
Descripción	El profesional puede cambiar su estado entre activo/inactivo/trabajando para indicar a los usuarios su disponibilidad actual.	
Entrada	Nuevo estado.	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	El profesional se dirige al apartado de 'Mi Perfil'.
	2	Dentro del mismo selecciona la opción 'Cambiar estado'.
	3	El profesional selecciona el nuevo estado que figura en su perfil.
Postcondición	Se ha cambiado el estado de profesional.	
Excepciones	N.A.	
Comentarios	N.A.	
Profesional		

Tabla A.13: Cambiar estado de profesional

Requisito	**Dar de alta/baja servicio**	
Identificador	3.2	
Prioridad	Alta	
Precondición	Haber iniciado sesión como profesional.	
Descripción	Los profesionales tendrán la opción de dar de alta y baja los servicios, en el primer caso esto significa que sube un servicio y en el segundo que retira de la oferta de servicios.	
Entrada	Servicio a dar de baja/alta.	
Salida	Confirmación de la baja/alta del servicio.	
Secuencia normal: alta de servicio	Paso	Acción
	1	Cuando un usuario contrata el servicio de un profesional, a este le llega una solicitud de servicio.
	2	El profesional abre la oferta, donde verá los detalles del servicio.
	3	Pulsa el botón 'Aceptar servicio'.
Secuencia normal: baja de servicio	Paso	Acción
	1	El profesional abre el servicio que había aceptado con anterioridad.
	2	Entre los detalles del servicio se muestra la opción 'Dar de baja servicio' el profesional pulsa el botón.
	3	Se muestra un mensaje de confirmación de baja de servicio.
Postcondición	Se ha dado de alta/baja un servicio.	
Excepciones	N.A.	
Comentarios	N.A.	
Actores	Profesional	

Tabla A.14: Dar de alta/baja servicio

Requisito	**Modificar servicio**	
Identificador	3.3	
Prioridad	Alta	
Precondición	Haber iniciado sesión como profesional, tener al menos un servicio dado de alta.	
Descripción	Permite al Profesional modificar la información de un servicio que ofrece.	
Entrada	Detalles actualizados del servicio.	
Salida	Confirmación de la modificación del servicio.	
Secuencia normal	Paso	Acción
	1	El profesional navega hasta la sección de gestión de servicios.
	2	Selecciona el servicio que desea modificar.
	3	Realiza las modificaciones necesarias en los detalles del servicio.
	4	Confirma la acción de modificación.
Postcondición	El servicio se actualiza con la nueva información en el perfil del Profesional.	
Excepciones	Paso	Acción
	3	Si no se proporciona la información necesaria, se muestra un mensaje de error.
	4	Si la operación falla por algún motivo, se notifica al Profesional.
Comentarios	Este caso de uso permite a los Profesionales mantener actualizada la información de sus servicios.	
Actores	Profesional	

Tabla A.15: Modificar servicio

Requisito	**Listar servicios dados de alta**	
Identificador	3.4	
Prioridad	Baja	
Precondición	Haber iniciado sesión como profesional, tener al menos un servicio dado de alta.	
Descripción	Permite al Profesional ver una lista de todos los servicios que ha dado de alta.	
Entrada	Selección de la opción para listar servicios.	
Salida	Lista de servicios con detalles.	
Secuencia normal	Paso	Acción
	1	El profesional navega hasta la sección de gestión de servicios.
	2	Selecciona la opción para listar sus servicios.
Postcondición	El Profesional puede ver una lista de los servicios que ha dado de alta.	
Excepciones	Paso	Acción
	2	Si no tiene servicios dados de alta, se muestra un mensaje indicando que no tiene servicios registrados.
Comentarios	N.A.	
Actores	Profesional	

Tabla A.16: Listar servicios dados de alta

Requisito	**Contestar solicitud de contratación.**	
Identificador	3.5	
Prioridad	Alta	
Precondición	El Profesional debe estar autenticado en la aplicación y haber recibido una solicitud de contratación.	
Descripción	Permite al Profesional aceptar o rechazar una solicitud de contratación de un Usuario.	
Entrada	Solicitud de contratación.	
Salida	Confirmación de la respuesta a la solicitud.	
Secuencia normal	Paso	Acción
	1	El profesional recibe una notificación de solicitud de contratación.
	2	Accede a la solicitud y selecciona aceptar o rechazar.
	3	Confirma la respuesta.
Postcondición	La solicitud de contratación se responde y se notifica al Usuario.	
Excepciones	Paso	Acción
	3	Si la solicitud ha caducado, se informa al Profesional.
Comentarios	N.A.	
Actores	Usuario, Profesional	

Tabla A.17: Contestar solicitud de contratación.

Requisito	**Consultar cliente**	
Identificador	3.6	
Prioridad	Media	
Precondición	Haber iniciado sesión como profesional y estar trabajando o haber trabajado con el cliente.	
Descripción	Permite al Profesional consultar información sobre el cliente, incluyendo sus datos, las valoraciones que ha recibido y otras estadísticas relevantes.	
Entrada	Selección del Cliente a consultar.	
Salida	Información detallada del Cliente, incluyendo sus datos personales, valoraciones recibidas y estadísticas.	
Secuencia normal	Paso	Acción
	1	El profesional accede a la sección de consulta de Clientes.
	2	Selecciona el Cliente cuya información desea consultar.
	3	Visualiza los datos y estadísticas del cliente.
Postcondición	El Profesional obtiene información sobre el cliente.	
Excepciones	N.A.	
Comentarios	Este caso de uso brinda al Profesional acceso a información relevante sobre los Clientes con los que ha interactuado.	
Actores	Usuario, Profesional	

Tabla A.18: Consultar cliente.

Requisito	**Añadir/Modificar/Quitar cliente de lista de favoritos.**	
Identificador	3.7	
Prioridad	Media	
Precondición	Haber iniciado sesión como profesional.	
Descripción	Permite al Profesional agregar, modificar o quitar Clientes de su lista de favoritos para un acceso más rápido y conveniente.	
Entrada	Selección de la acción (añadir, modificar o quitar) y Cliente seleccionado.	
Salida	N.A.	
Secuencia normal	Paso	Acción
	1	El profesional navega hasta la sección de gestión de favoritos.
	2	Selecciona la acción deseada (añadir, modificar o quitar) y el cliente correspondiente.
	3	Confirma la acción.
Postcondición	La lista de favoritos se actualiza según la acción realizada.	
Excepciones	Paso	Acción
	2	Si el Cliente ya está en la lista de favoritos y se selecciona .ªñadir", se muestra un mensaje informativo.
Comentarios	Este caso de uso permite al Profesional gestionar su lista de favoritos para un acceso más rápido a los Clientes preferidos.	
Actores	Usuario, Profesional	

Tabla A.19: Añadir/Modificar/Quitar cliente de lista de favoritos.

A.2.4. Casos de uso de administrador

Requisito	**Añadir/eliminar/Modificar categorías de servicios ofrecidos.**	
Identificador	4.1	
Prioridad	Alta	
Precondición	Estar registrado en la aplicación como administrador.	
Descripción	Permite al Administrador gestionar las categorías de servicios ofrecidos, incluyendo la adición, eliminación o modificación de categorías existentes.	
Entrada	Selección de la acción (añadir, eliminar o modificar) y detalles de la categoría.	
Salida	Confirmación de la acción realizada en las categorías.	
Secuencia normal	Paso	Acción
	1	El administrador accede a la sección de gestión de categorías de servicios.
	2	Selecciona la acción deseada (añadir, eliminar o modificar) y proporciona los detalles necesarios.
	3	Confirma la acción.
Postcondición	Las categorías se actualizan según la acción realizada.	
Excepciones	Paso	Acción
	3	Si la categoría ya existe y se selecciona ‘añadir’, se muestra un mensaje informativo.
	3	Si la categoría no existe y se selecciona ‘eliminar’ o ‘modificar’, se muestra un mensaje informativo.
Comentarios	Este caso de uso permite al Administrador gestionar las categorías de servicios para mantener la organización de la plataforma.	
Actores	Administrador	

Tabla A.20: Añadir/eliminar/Modificar categorías de servicios ofrecidos.

Requisito	**Consultar datos/estadísticas de profesional/clientes.**	
Identificador	4.2	
Prioridad	Media	
Precondición	Estar registrado en la aplicación como administrador.	
Descripción	Permite al Administrador acceder a datos y estadísticas relacionadas con Profesionales y Clientes, lo que le permite realizar análisis y tomar decisiones informadas.	
Entrada	Selección de Profesional o Cliente a consultar.	
Salida	Información detallada y estadísticas del Profesional o Cliente seleccionado.	
Secuencia normal	Paso	Acción
	1	El administrador accede a la sección de consulta de datos/estadísticas.
	2	Selecciona el Profesional o Cliente cuya información desea consultar.
	3	Visualiza los datos y estadísticas.
Postcondición	El Administrador obtiene información detallada sobre el Profesional o Cliente seleccionado.	
Excepciones	Paso	Acción
	2	Si no se encuentra información para el Profesional o Cliente seleccionado, se muestra un mensaje informativo.
Comentarios	Este caso de uso proporciona al Administrador acceso a datos relevantes para la toma de decisiones y la gestión de la plataforma.	
Actores	Administrador	

Tabla A.21: Consultar datos/estadísticas de profesional/clientes.

Requisito	**Buscar clientes/profesionales.**	
Identificador	4.3	
Prioridad	Media	
Precondición	Estar registrado en la aplicación como administrador.	
Descripción	Permite al Administrador buscar Clientes o Profesionales dentro de la aplicación según diversos criterios.	
Entrada	Criterios de búsqueda.	
Salida	Lista de Clientes o Profesionales que coinciden con los criterios de búsqueda.	
Secuencia normal	Paso	Acción
	1	El administrador accede a la sección de búsqueda de Clientes o Profesionales.
	2	Ingresa los criterios de búsqueda.
	3	Realiza la búsqueda.
	4	Visualiza la lista de resultados.
Postcondición	El Administrador obtiene una lista de Clientes o Profesionales que coinciden con los criterios de búsqueda.	
Excepciones	Paso	Acción
	3	Si no se encuentran resultados que coincidan con los criterios, se muestra un mensaje informativo.
Comentarios	Este caso de uso permite al Administrador buscar y acceder a perfiles de Clientes o Profesionales de manera eficiente.	
Actores	Administrador	

Tabla A.22: Buscar clientes/profesionales.

Requisito	**Modificar datos de clientes/profesionales/servicios ofrecidos.**	
Identificador	4.4	
Prioridad	Alta	
Precondición	Estar registrado en la aplicación como administrador.	
Descripción	Permite al Administrador realizar modificaciones en los datos de Clientes, Profesionales o Servicios Ofrecidos cuando sea necesario.	
Entrada	Selección del tipo de modificación (Cliente, Profesional o Servicio) y detalles de la modificación.	
Salida	Confirmación de la modificación realizada.	
Secuencia normal	Paso	Acción
	1	El administrador accede a la sección de modificación de datos.
	2	Selecciona el tipo de modificación deseada (Cliente, Profesional o Servicio) y proporciona los detalles necesarios.
	3	Confirma la modificación.
Postcondición	Los datos se actualizan según la modificación realizada.	
Excepciones	N.A.	
Comentarios	Este caso de uso permite al Administrador gestionar y mantener actualizados los datos de la plataforma.	
Actores	Administrador	

Tabla A.23: Modificar datos de clientes/profesionales/servicios ofrecidos.

Requisito	**Dar de baja usuarios.**	
Identificador	4.5	
Prioridad	Alta	
Precondición	Estar registrado en la aplicación como administrador.	
Descripción	Permite al Administrador dar de baja a Usuarios de la aplicación en casos de incumplimiento de términos y condiciones u otras razones legítimas.	
Entrada	Selección del Usuario a dar de baja y motivo de la baja.	
Salida	Confirmación de la baja del Usuario.	
Secuencia normal	Paso	Acción
	1	El administrador accede a la sección de gestión de bajas de Usuarios.
	2	Selecciona el Usuario a dar de baja y especifica el motivo.
	3	Confirma la baja del Usuario.
Postcondición	El Usuario queda dado de baja de la aplicación.	
Excepciones	N.A.	
Comentarios	Este caso de uso permite al Administrador mantener la integridad de la plataforma al dar de baja a Usuarios que incumplen las reglas o políticas.	
Actores	Administrador	

Tabla A.24: Dar de baja usuarios.

Apéndice B

Interfaces de Profinder diseñadas en figma

Se pueden añadir los apéndices que se consideren oportunos.

Apéndice C

Diseño de las bases de datos

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes

–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.

Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes