



INGENIERÍA TELEMÁTICA

Curso Académico 2017/2018

Trabajo Fin de Grado

APLICACIÓN MÓVIL ANDROID PARA RED
COLABORATIVA PARA OBTENER
INFORMACIÓN SOBRE TRABAJOS Y SUELDOS
EN EL MERCADO LABORAL

Autor : Rubén Sánchez Rivero

Tutor : Pedro de Las Heras Quirós

Trabajo Fin de Grado

Aplicación Móvil Android para Red Colaborativa para obtener Información
sobre Trabajos y Sueldos en el Mercado Laboral

Autor : Rubén Sánchez Rivero

Tutor : Pedro de Las Heras Quirós

La defensa del presente Proyecto Fin de Carrera se realizó el día de
de 2018, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 2018

Agradecimientos

Primero de todo, me gustaría agradecer a mi familia y especialmente a mis padres por su apoyo y comprensión.

Además, me gustaría darles las gracias a mis amigos por sus ánimos.

Por supuesto, también quiero agradecer a Pedro de Las Heras Quirós, por su tiempo, su paciencia y su esfuerzo para ayudarme a completar este proyecto.

AGRADECIMIENTOS

Resumen

En la actualidad la poca información y la falta de experiencia hace que resulte complicado la elección de un puesto de trabajo, esto se acentúa más si se disponen de pocas ofertas y sobre todo por el sistema actual de becarios como entrada en el mundo laboral. Además, una vez se está asentado en una empresa es fácil desconocer la actualidad del mercado que nos rodea, el cual ha podido mejorar y cuáles son las empresas que mejor valoran las aptitudes, los conocimientos y la experiencia de la que se disponga.

En este proyecto se pretende realizar una aplicación que permita a los usuarios compartir información que les ayude a elegir su futuro trabajo y a evaluar el actual. Debido al carácter delicado de los datos su manejo se realiza cifrado y los usuarios son los que deciden que información comparten de forma pública, o incluso mantenerse en el anonimato mostrando únicamente su sueldo bruto. Los usuarios contactan entre ellos y entablan una negociación mediante la que solicitan un intercambio de datos.

En este trabajo fin de grado se ha seguido una metodología basada en desarrollo ágil. En el desarrollo de la aplicación se ha ido incrementando la funcionalidad mediante ciclos iterativos hasta cumplir con los objetivos generales y los requisitos específicos que se iban planteando durante su elaboración.

Para el desarrollo se han aplicado tecnologías y herramientas como Pencil, Android Studio, Java, PostgreSQL, Git, GitHub y Eclipse. Además, para la parte del servidor se ha utilizado Travis CI y el sistema de hosting de Heroku, permitiendo una integración continua del código mediante la automatización de Travis CI para validar, compilar los cambios subidos en GitHub y posteriormente desplegar el servidor actualizado en Heroku.

Los resultados obtenidos ofrecen una valoración positiva de la aplicación, por su funcionalidad, su interfaz intuitiva y el cumplimiento de los objetivos.

Índice general

1. Introducción	1
1.1. Aplicaciones móviles	1
1.2. Tecnologías y Herramientas	1
1.2.1. Pencil	1
1.2.2. Lucidchart	2
1.2.3. Android Studio	2
1.2.4. Eclipse	2
1.2.5. Java	2
1.2.6. PostgreSQL	3
1.2.7. Git	3
1.2.8. GitHub	3
1.2.9. Travis CI	3
1.2.10. Heroku	3
1.3. Estructura de la memoria	4
2. Motivación y objetivos	5
2.1. Motivación	5
2.2. Objetivos generales	6
2.3. Especificación de requisitos	6
3. Aspectos del desarrollo	9
3.1. Metodología	9
3.2. Acceso al software del proyecto	11

4. Arquitectura y desarrollo de la aplicación	13
4.1. Arquitectura general	13
4.1.1. Diagrama relación cliente-servidor	13
4.1.2. Diagrama base de datos cliente	14
4.1.3. Diagrama base de datos servidor	15
4.2. Apartados de la aplicación	16
4.2.1. Pantalla de inicio de sesión	17
4.2.2. Pantalla de registro	18
4.2.3. Pantalla de búsqueda	19
4.2.4. Pantalla del listado de contactos	20
4.2.5. Pantalla del perfil del contacto	21
4.2.6. Pantalla de envío de negociación	22
4.2.7. Pantalla del perfil del usuario	23
4.2.8. Pantalla de edición del usuario	24
4.2.9. Pantalla de contactos guardados	25
4.2.10. Pantalla del perfil del contactos guardado	26
4.2.11. Pantalla del listado de negociaciones	27
4.2.12. Pantalla de revisión de negociación	28
4.2.13. Pantalla de recepción de negociación	29
5. Conclusiones	31
5.1. Consecución de objetivos	31
5.2. Trabajos futuros	32
Bibliografía	33

Índice de figuras

3.1. Estructura metodología ágil	11
4.1. Diagrama relación cliente-servidor	14
4.2. Diagrama base de datos cliente	15
4.3. Diagrama base de datos servidor	16
4.4. Pantalla de inicio de sesión	17
4.5. Pantalla de registro	18
4.6. Pantalla de búsqueda	19
4.7. Pantalla de listado de contactos	20
4.8. Pantalla del perfil del contacto	21
4.9. Pantalla de envío de negociación	22
4.10. Pantalla del perfil del usuario	23
4.11. Pantalla de edición del usuario	24
4.12. Pantalla de listado de contactos	25
4.13. Pantalla del perfil del contacto guardado	26
4.14. Pantalla del listado de negociaciones	27
4.15. Pantalla de revisión de negociación	28
4.16. Pantalla de recepción de negociación	29

ÍNDICE DE FIGURAS

Capítulo 1

Introducción

Este Trabajo Fin de Grado se desarrolla en el ámbito de las aplicaciones móviles, concretamente en Android. En este capítulo introduciremos los conceptos básicos, las tecnologías y las herramientas utilizadas.

1.1. Aplicaciones móviles

Las aplicaciones permiten al usuario realizar funciones muy concretas y de cualquier tipo, facilitando las gestiones o actividades a desarrollar.

Actualmente los dispositivos móviles se han convertido en uno de los principales medios de conexión a la red, siendo una alternativa a las formas tradicionales. Además debido a su fácil acceso, disponibilidad, rapidez y manejo hace que el potencial y el uso de las aplicaciones móviles sea muy alto.

1.2. Tecnologías y Herramientas

Entran en este ámbito todas aquellas herramientas y tecnologías más relevantes usadas en el Trabajo Fin de Grado.

1.2.1. Pencil

Herramienta de código abierto para la creación de prototipos de interfaz gráfica de usuario (GUI) [1], Pencil [2] permite crear maquetas de sitios web y aplicaciones sin hacer uso de

lenguajes de programación.

1.2.2. Lucidchart

Herramienta de diagramación web basada en la nube, Lucidchart [3] permite a los usuarios colaborar y trabajar en equipo en tiempo real. Sirve para crear diagramas, modelos, organigramas, esquemas, diseños UML [4], mapas y bocetos.

1.2.3. Android Studio

Desarrollado por Google [5], Android Studio [6] es el entorno de desarrollo integrado (IDE) oficial para la plataforma Android [7] y está basado en IntelliJ IDEA [8]. Permite crear y emular las apps en tiempo real para todos los dispositivos de Android, además se pueden ejecutar directamente desde el móvil. Tiene herramientas completas de edición, depuración, pruebas y perfilamiento de códigos.

1.2.4. Eclipse

Desarrollado originalmente por IBM [9] y actualmente por la Fundación Eclipse, Eclipse [10] es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Kit (JDK) y el compilador (ECJ) que son parte de Eclipse.

Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en Java.

1.2.5. Java

Desarrollado originalmente por James Gosling [11], Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y fuertemente tipado. Usado principalmente para aplicaciones de cliente-servidor.

1.2.6. PostgreSQL

PostgreSQL [12] es un sistema de gestión de bases de datos relacional orientado a objetos y libre.

1.2.7. Git

Diseñado por Linus Torvalds [13], Git [14] es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

1.2.8. GitHub

GitHub [15] es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en Ruby on Rails [16].

1.2.9. Travis CI

Travis CI [17] es un servicio de integración continua alojado y distribuido utilizado para crear y probar proyectos de software alojados en GitHub . Se configura agregando un archivo llamado `.travis.yml`, que es un archivo de texto en formato YAML [18], al directorio raíz del repositorio. Cuando se ha activado Travis CI para un repositorio dado, GitHub le notificará cada vez que se actualice el contenido de ese repositorio o se envíe una solicitud de extracción. Cuando Travis CI recibe las notificaciones de GitHub ejecutará los comandos especificados en `.travis.yml`, que generalmente crean el software y ejecutan cualquier prueba automatizada.

1.2.10. Heroku

Heroku [19] es una plataforma como servicio (PaaS) que ofrece servicios de computación a través de la red y que soporta distintos lenguajes de programación.

1.3. Estructura de la memoria

Los contenidos y la organización de la información en esta memoria es la siguiente:

- **Capítulo 2:** en este capítulo se presentan las motivaciones de realizar este proyecto, los objetivos y los requisitos del mismo.
- **Capítulo 3:** en este capítulo se describe la metodología utilizada y se muestra dónde se puede acceder a todo el software de la aplicación
- **Capítulo 4:** en este capítulo se detalla el proceso de diseño y desarrollo de la aplicación. Se ha organizado en función de las distintas pantallas.
- **Capítulo 5:** en este capítulo se analizan los resultados obtenidos una vez finalizado el proyecto.
- **Apéndices:** este apartado sirve de apoyo para el seguimiento de esta memoria.

Capítulo 2

Motivación y objetivos

En este capítulo se exponen los problemas que motivan la realización de este proyecto, los objetivos generales y los requisitos que han condicionado su desarrollo.

2.1. Motivación

En la actualidad existen distintas aplicaciones y páginas web que permiten conocer información del mercado laboral. En aplicaciones como Indeed [20], LinkedIn [21], Infojobs [22], CornerJob [23] y Jobtoday [24] puedes buscar ofertas de trabajo y en algunas también mostrar tus datos a forma de Currículum público. Por otro lado sitios web como TuSalario.es [25], OCCMundial.com [26] y Hays.es [27] puedes obtener información orientativa para tu trabajo según los datos insertados.

Sin embargo no existe una red que permita obtener datos sobre sueldos y trabajos de primera mano, y menos aún de forma anónima y segura, en la que únicamente se compartan la información que uno quiera y exclusivamente con la persona con quien quiera intercambiar dichos datos.

Dentro del departamento de Sistemas Telemáticos y Computación (GSyC) [28] de la Universidad Rey Juan Carlos [29] se pretendía dar una herramienta a los alumnos con la que puedan conocer mejor el mercado laboral que les espera al finalizar los estudios y que les ayude en la elección de un trabajo.

De esta necesidad surge la motivación de crear una aplicación móvil Android que cumpla dichos requisitos, FairWage [30].

2.2. Objetivos generales

Como resultado del análisis de la motivación descrita, se pueden extraer una serie de objetivos generales que deben de cumplirse en la implementación de este proyecto. Estos objetivos son:

1. Desarrollar una aplicación funcional que permita intercambiar información entre usuarios en tiempo real.
2. Garantizar la confidencialidad y la seguridad de los datos ingresados por los usuarios y su intercambio.
3. Desarrollar un servidor mínimo, que guarde la menor cantidad de datos el menor tiempo posible.
4. Proporcionar una interfaz sencilla e intuitiva para facilitar su manejo.
5. Asegurar fluidez y optimización en la aplicación.

2.3. Especificación de requisitos

1. Proporcionar una interfaz de registro con la que el usuario pueda darse de alta en la aplicación.
2. Autenticar a los usuarios para poder acceder a la funcionalidad de la aplicación.
3. Dar al usuario la posibilidad de rellenar y editar su perfil información, así como elegir cuáles de esos datos son de dominio público y cuáles son privados. El único campo que siempre será público es el sueldo bruto.
4. Mostrar al usuario los datos de su perfil y la posibilidad de borrar al mismo dándose de baja en la aplicación.
5. Poder enviar solicitudes a otros usuarios para intercambiar información, una vez enviadas se debe poder cancelar y eliminar dicha solicitud.

6. Poder elegir concretamente que datos son los que quiero obtener de otro usuario y cuales son los que le ofrezco.
7. Poder aceptar, rechazar o mandar una contraoferta de las solicitudes recibidas por otros usuarios.
8. Almacenar y mostrar una lista con todos los datos recibidos de otros usuarios, dando la posibilidad de ser borrados de manera individual.
9. Almacenar y mostrar una lista con todas las solicitudes, tanto las enviadas como las recibidas, así como el estado en el que se encuentran en ese momento.
10. Hacer uso de claves públicas y privadas para cifrar los datos enviados de forma que únicamente el destinatario pueda descifrar dicha información.
11. Garantizar la privacidad del usuario, para ello el servidor aparte del nombre del usuario no conocerá ningún dato que no sea público.
12. Otorgar la posibilidad de actualizar de manera manual los listados de solicitudes y de contactos.
13. Notificar al cliente si hay novedades en las solicitudes o los contactos.

Capítulo 3

Aspectos del desarrollo

En este capítulo se expone la metodología utilizada para realizar el proyecto y la ubicación del software del mismo.

3.1. Metodología

En este Trabajo Fin de Grado se ha realizado una fase inicial que consiste en:

- **Consolidación, búsqueda y aprendizaje:** se establecen los objetivos generales de la aplicación, la búsqueda de las herramientas necesarias para realizarla y su aprendizaje.
- **Prototipado:** se elabora un diseño básico de la aplicación móvil usando la herramienta Pencil.
- **Configuración:** se prepara la configuración del servidor para que los cambios publicados en el repositorio de Github sean automáticamente validados y compilados por Travis y enviados a Heroku, que es el encargado de ejecutar el servidor.

Posteriormente durante la elaboración del proyecto se realizó una metodología de desarrollo ágil en la cual mediante ciclos iterativos se incrementaba la funcionalidad del software.

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la

necesidad del proyecto. Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación.

El objetivo de cada iteración es agregar software sin errores, cada iteración en el ciclo de desarrollo ha incluido:

- **Planificación:** se establecen los requisitos específicos prioritarios los cuales pueden cambiar o incluso añadirse nuevos según se va desarrollando la aplicación.
- **Análisis de requisitos:** se recogen las mejoras y funcionalidades que se pueden añadir en los requisitos cumplidos y los que quedan pendientes. Se procede según las prioridades decididas en la fase de planificación.
- **Diseño:** se elabora una arquitectura similar a la propuesta en el prototipado para implementar los requisitos o las mejoras.
- **Codificación:** desarrollo de las funcionalidades pertinentes.
- **Pruebas:** comprobar que el diseño y la codificación cumplen con los requisitos acordados y se evalúa el resultado para posibles mejoras.
- **Documentación:** se anotan los requisitos implementados y el proceso realizado para conseguirlo, se anotan posibles mejoras que pueden ser introducidas posteriormente.
- **Integración:** se publican los cambios realizados en la aplicación en los repositorios de GitHub.

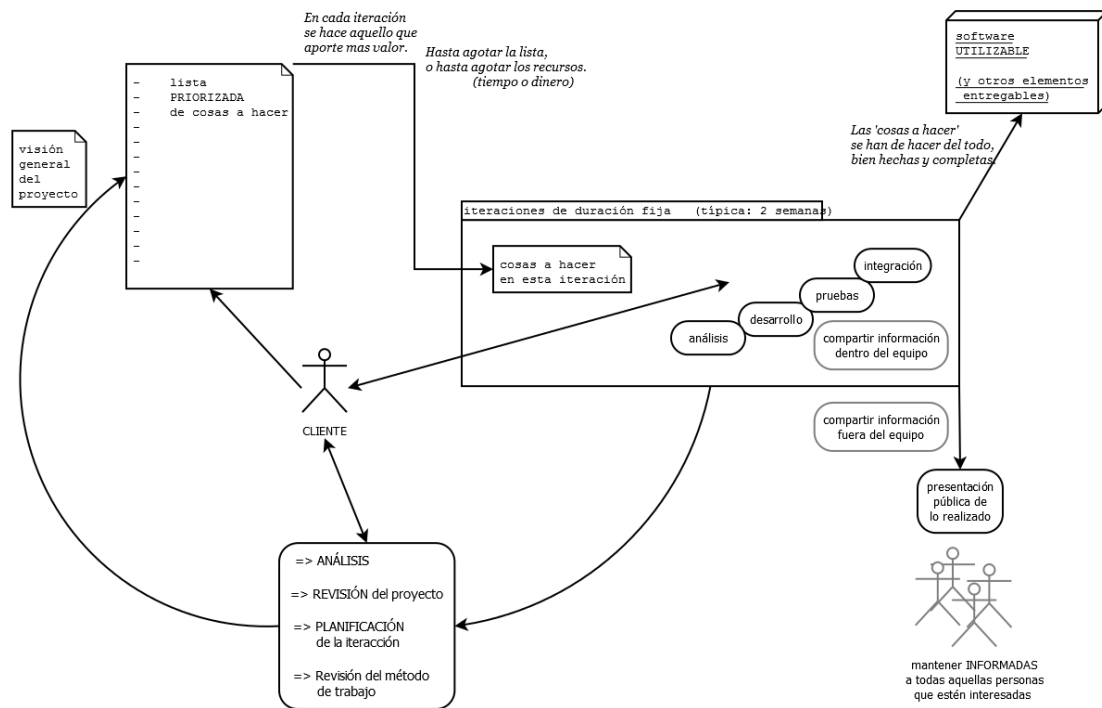


Figura 3.1: Estructura metodología ágil

3.2. Acceso al software del proyecto

El código fuente de la aplicación está disponible en un repositorio público de Github [30].

El código fuente de la memoria está disponible en un repositorio público de Github [31].

Capítulo 4

Arquitectura y desarrollo de la aplicación

En este capítulo se expone detalladamente la arquitectura y la implementación de la aplicación, esta se encuentra dividida en dos partes, el lado del cliente y el del servidor.

4.1. Arquitectura general

Para facilitar el seguimiento de la funcionalidad de la aplicación y dar una visión global de la misma se ha realizado un diagrama que explique las interacciones de los clientes con el servidor y otros diagramas mostrando la estructura de las bases de datos, tanto en el cliente como en el servidor.

4.1.1. Diagrama relación cliente-servidor

Las interacción entre cliente y servidor se puede dividir en cinco tipos, el más complejo que es el proceso de negociación a pesar de sus variaciones siempre sigue el mismo flujo.

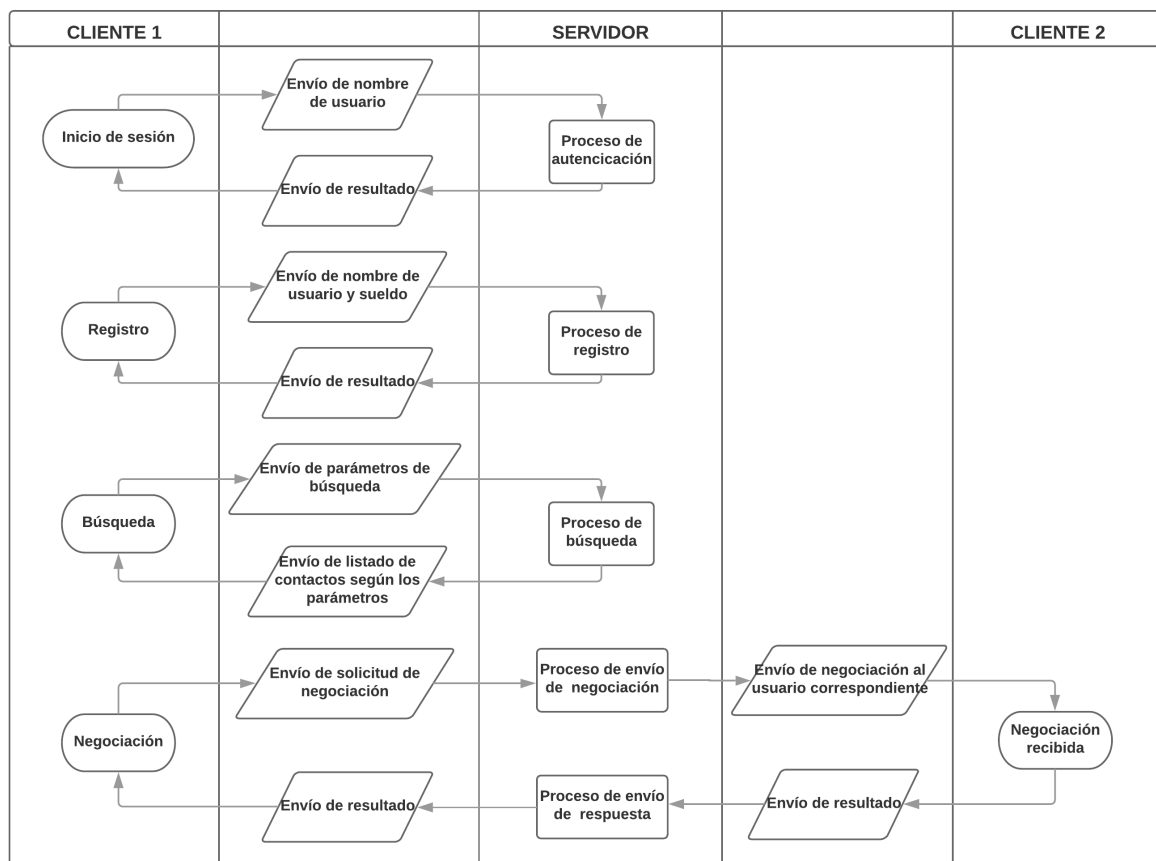


Figura 4.1: Diagrama relación cliente-servidor

4.1.2. Diagrama base de datos cliente

El cliente divide la estructura de su base de datos de la siguiente forma:

1. **Usuarios:** aquí se guardan los datos de cada usuario, permitiendo que los cambios realizados en el perfil sean persistentes y que se puedan crear varios perfiles en una misma aplicación.
2. **Contactos:** aquí se guardan los datos de los contactos guardados obtenidos mediante negociaciones, permitiendo poder acceder a su información en cualquier momento.
3. **Negociaciones:** aquí se guardan los datos de las negociaciones, permitiendo poder realizar un seguimiento del estado de las mismas.

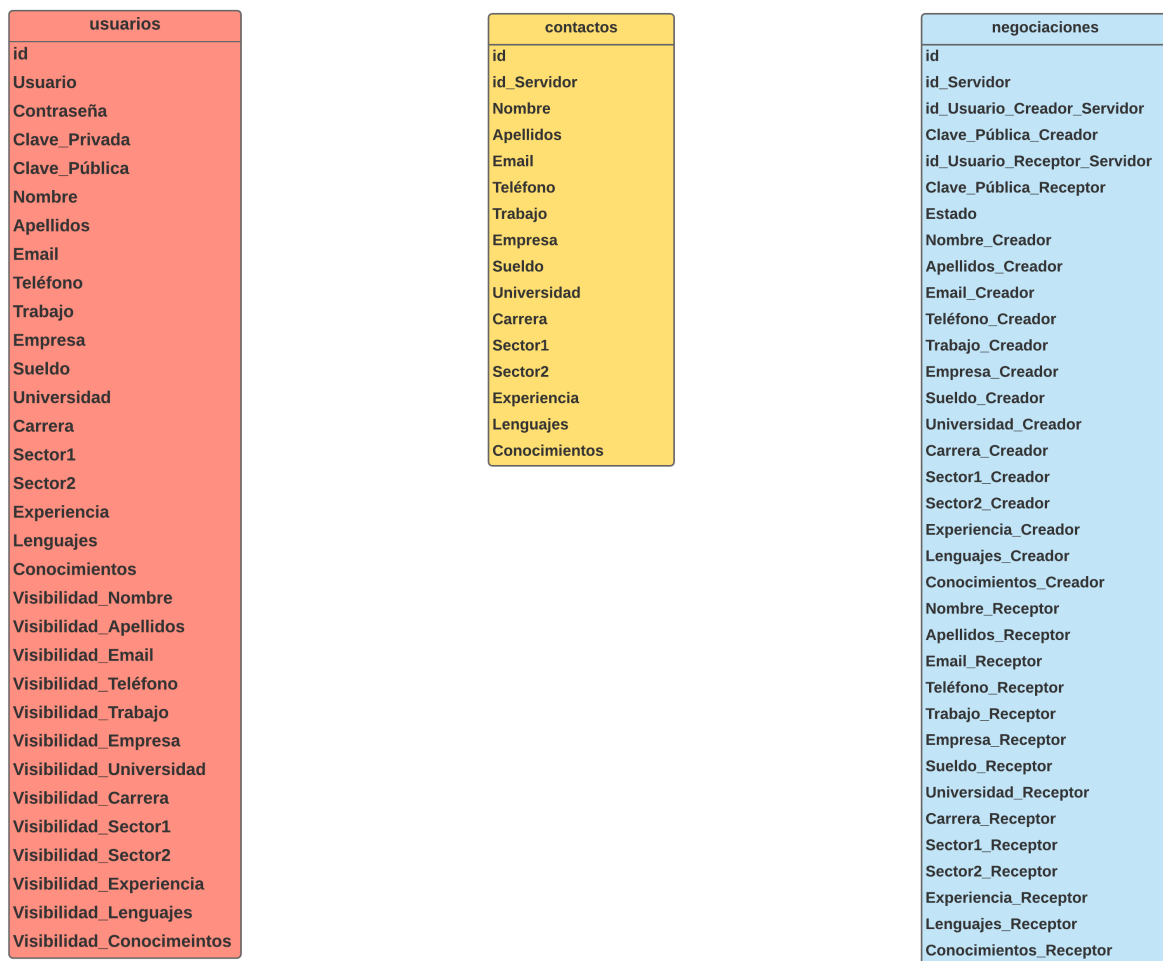


Figura 4.2: Diagrama base de datos cliente

4.1.3. Diagrama base de datos servidor

El servidor divide la estructura de su base de datos de la siguiente forma:

1. **Usuarios:** aquí se guardan los datos públicos de cada usuario, permitiendo dar una mayor visibilidad a la hora de que un usuario realice una búsqueda. Además se almacena su nombre de usuario para llevar un control del registro y su clave pública para poder recibir información de otros contactos de forma segura.
2. **Contactos:** aquí se guardan temporalmente los datos de un contacto cuando ha aceptado una negociación, esta información sera entregada al creador de la solicitud y posteriormente eliminado del servidor.

3. **Negociaciones:** aquí se guardan los datos de las negociaciones.

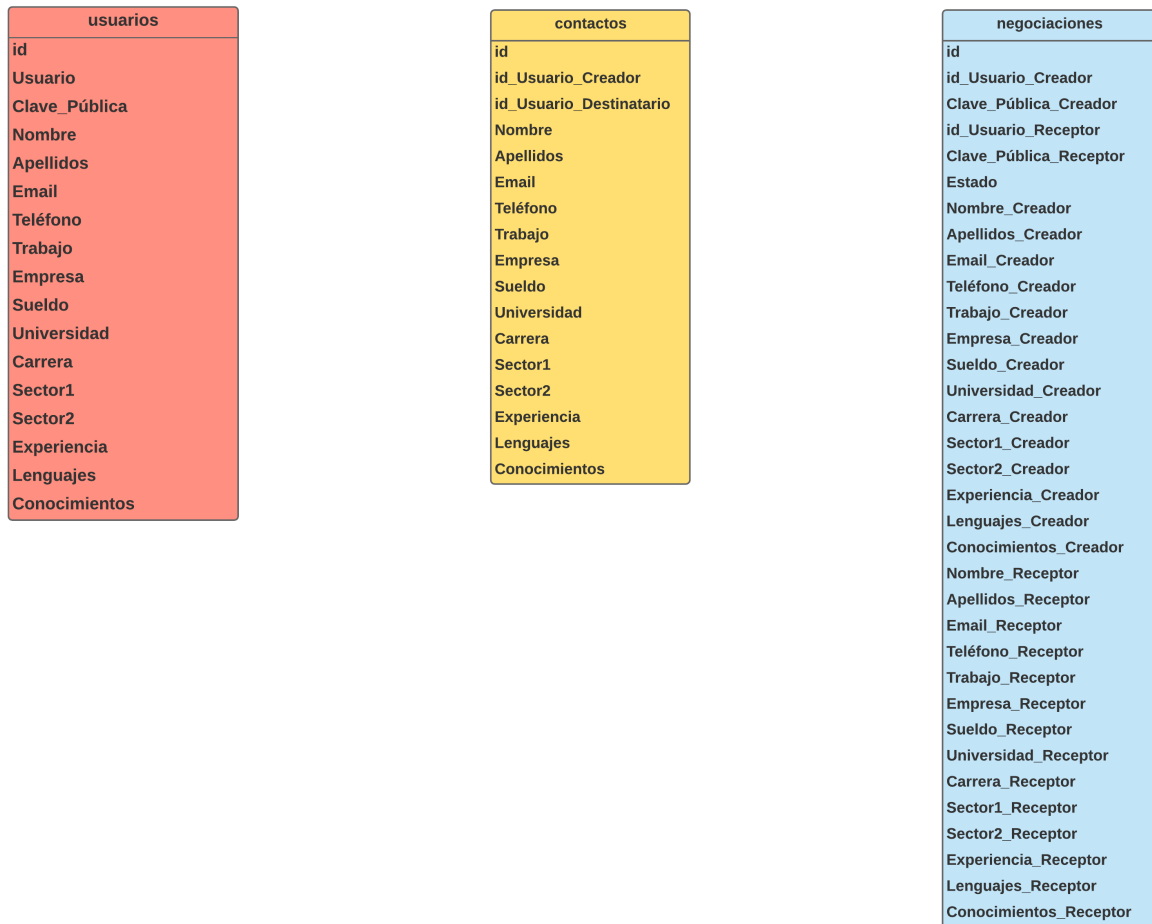


Figura 4.3: Diagrama base de datos servidor

4.2. Apartados de la aplicación

A continuación se detallarán las diferentes pantallas que tiene la aplicación y se mostrará una imagen del prototipo correspondiente para una mejor visualización.

4.2.1. Pantalla de inicio de sesión



Figura 4.4: Pantalla de inicio de sesión

El propósito de esta pantalla es la autenticación de los usuarios. Está compuesta por:

1. Un campo para introducir el nombre del usuario.
2. Un campo para introducir la contraseña del usuario.
3. Un apartado para ir a la pantalla de registro.
4. Un botón para comprobar que el usuario está registrado en el servidor y que los datos son correctos.
5. Un botón para salirse de la aplicación.

Cuando se pulsa el botón de enviar se comprueba que ningún campo esté vacío, si alguno lo está, será marcado y se informa al usuario que tiene que rellenarlo. Posteriormente se comprueba que el usuario está registrado en el servidor y de forma local que la autenticación es correcta, en caso de no serla, se avisa al usuario de que los datos son erróneos. Si la autenticación se realiza con éxito se inicia la pantalla de búsqueda.

4.2.2. Pantalla de registro



Figura 4.5: Pantalla de registro

El propósito de esta pantalla es dar de alta a los usuarios nuevos de la aplicación. Está compuesta por:

1. Un campo para introducir el nombre del usuario.
2. Un campo para introducir la contraseña del usuario.
3. Un campo para introducir el sueldo bruto.
4. Un apartado para ir a la pantalla de inicio de sesión.
5. Un botón para realizar el alta del usuario en el servidor en caso de que los datos introducidos sean correctos y el nombre de usuario esté disponible.
6. Un botón para salirse de la aplicación.

Cuando se pulsa el botón de enviar se comprueba que ningún campo esté vacío, si alguno lo está, será marcado y se informa al usuario que tiene que rellenarlo. Posteriormente se envía

el nombre de usuario al servidor el cual comprueba que esté disponible, en caso de que se esté utilizando en otro dispositivo se informa de que el nombre elegido no es válido. Si es válido el servidor guarda tanto el nombre de usuario como el sueldo bruto, que fueron previamente enviados juntos. Si el registro se realiza con éxito se guardará de forma local los datos del usuario y se inicia la pantalla de búsqueda.

4.2.3. Pantalla de búsqueda



Figura 4.6: Pantalla de búsqueda

El propósito de esta pantalla es ofrecer una interfaz mediante la cual puedan realizarse búsquedas de sueldos brutos de otros usuarios, además se pueden introducir ciertos parámetros para acotar los resultados obtenidos. Está compuesta por:

1. Un botón para ir a la pantalla del perfil del usuario.
2. Un botón para ir a la pantalla de edición del perfil del usuario.
3. Un botón para ir al listado de contactos guardados.

4. Un botón para ir al listado de solicitudes, tanto las enviadas como las recibidas.
5. Un campo para introducir el sueldo bruto mínimo buscado.
6. Un campo para introducir el trabajo buscado.
7. Dos campos para introducir los sectores del empleo buscado.
8. Un apartado para añadir conocimientos variados.
9. Un botón para realizar la búsqueda en el servidor de los perfiles que encajen con los parámetros que se hayan proporcionado.
10. Un botón para desconectar al usuario actual e ir a la pantalla de inicio de sesión.

Cuando se pulsa el botón de buscar se inicia la pantalla del listado de contactos.

4.2.4. Pantalla del listado de contactos



Figura 4.7: Pantalla de listado de contactos

El propósito de esta pantalla es mostrar un listado de los contactos según los parámetros introducidos en la pantalla de búsqueda. Cada contacto se divide en filas y cada fila está compuesta por:

1. El trabajo del contacto, no aparecerá nada en caso de que este campo no sea público.
2. El sueldo bruto del contacto.
3. Los sectores del contacto, si alguno no es público no aparecerá.

Inicialmente los parámetros introducidos en la pantalla de búsqueda son enviados al servidor que devuelve los contactos correspondientes. Si no se ha rellenado ningún campo se muestran todos los perfiles existentes, de lo contrario únicamente se muestran aquellos contactos que coinciden al menos con uno de los parámetros.

Cuando se pulsa en un contacto en concreto se inicia la pantalla del perfil del contacto.

4.2.5. Pantalla del perfil del contacto



Figura 4.8: Pantalla del perfil del contacto

El propósito de esta pantalla es mostrar la información pública de un contacto, así como dar la posibilidad de enviar una solicitud en la que se pedirán nuevos datos. Está compuesta por:

1. Los campos públicos de un contacto.
2. Un botón para iniciar una negociación con dicho contacto.

Cuando se pulsa el botón iniciar negociación se inicia la pantalla de negociación.

4.2.6. Pantalla de envío de negociación



Figura 4.9: Pantalla de envío de negociación

El propósito de esta pantalla es elegir los datos personales que queremos conseguir del contacto seleccionado y marcar que datos se ofrecen a cambio. Está compuesta por:

1. Una columna con los diferentes campos de un usuario y un switch para indicar que datos se ofrecen.
2. Una columna con los diferentes campos de un usuario y un switch para indicar que datos se quieren obtener.

3. Un botón para enviar la solicitud de negociación a dicho contacto.
4. Un botón para cancelar la negociación y volver a la pantalla del perfil del contacto.

Cuando se pulsa el botón de enviar negociación se comprueba que el usuario tenga rellenos los campos que ha ofrecido, en caso de estar alguno vacío se avisa de que debe rellenarlos para cumplimentar el envío. Posteriormente se envían al servidor los datos seleccionados para que este se encargue de su transmisión al otro usuario. Además, en local se añade a la lista de negociaciones la solicitud creada con sus parámetros. Finalmente se retorna al listado de contactos.

4.2.7. Pantalla del perfil del usuario



Figura 4.10: Pantalla del perfil del usuario

El propósito de esta pantalla es mostrar los datos personales que estén rellenos así como si actualmente son públicos o privados. Está compuesta por:

1. Un botón para ir a la pantalla de edición del perfil del usuario.

2. Un botón para ir a la pantalla de búsqueda.
3. Un botón para ir al listado de contactos guardados.
4. Un botón para ir al listado de negociaciones.
5. Una columna con los diferentes campos rellenos de un usuario.
6. Una columna con imágenes que diferencian el estado público o privado de cada campo.

4.2.8. Pantalla de edición del usuario



Figura 4.11: Pantalla de edición del usuario

El propósito de esta pantalla es ofrecer una interfaz mediante la cual un usuario pueda cambiar los campos de su perfil y su correspondiente privacidad. Está compuesta por:

1. Una columna con los diferentes campos con el valor actual en caso de haberse rellenado previamente.
2. Una columna con la configuración de privacidad actual asociada a cada campo.
3. Un botón para guardar los cambios.
4. Un botón para cancelar la edición y volver a la pantalla anterior.

Cuando se pulsa el botón de guardar cambios se envían al servidor aquellos campos que estén marcados como públicos y se borran del servidor aquellos que se hayan cambiado a privados. Además se guarda en local los datos actualizados y se vuelve a la pantalla anterior.

4.2.9. Pantalla de contactos guardados



Figura 4.12: Pantalla de listado de contactos

El propósito de esta pantalla es mostrar un listado de los contactos según guardados. Cada contacto se divide en filas y cada fila está compuesta por:

1. El trabajo del contacto, no aparecerá nada en caso de que no se haya obtenido mediante la previa negociación.
2. El sueldo bruto del contacto.
3. Los sectores del contacto, si alguno no se ha obtenido mediante la previa negociación no aparecerá.

Cuando se pulsa en un contacto en concreto se inicia la pantalla del perfil del contacto guardado.

4.2.10. Pantalla del perfil del contactos guardado



Figura 4.13: Pantalla del perfil del contacto guardado

El propósito de esta pantalla es mostrar la información de un contacto guardado, así como dar la posibilidad de borrar dicho contacto. Está compuesta por:

1. Los campos del contacto obtenidos en la negociación.
2. Un botón para borrar el contacto y volver a la pantalla de contactos guardados.
3. Un botón para volver a la pantalla de contactos guardados.

4.2.11. Pantalla del listado de negociaciones



Figura 4.14: Pantalla del listado de negociaciones

El propósito de esta pantalla es mostrar un listado de las negociaciones guardadas y su estado para tener un seguimiento de las mismas. Cada negociación se divide en filas y cada fila está compuesta por:

1. El trabajo del contacto, no aparecerá nada en caso de que este campo no sea público.
2. El sueldo bruto del contacto.
3. Los sectores del contacto, si alguno no es público no aparecerá.
4. El estado de la negociación que puede ser: enviada, aceptada, rechazada, recibida.

Cuando se pulsa en una negociación en concreto se iniciarán distintos procesos según su estado:

- Si la negociación se encuentra en estado enviada se iniciará la pantalla de revisión de negociación.

- Si la negociación se encuentra en estado aceptada se dará la posibilidad de iniciar la pantalla del perfil del contacto obtenido y de borrar el registro de dicha negociación del listado.
- Si la negociación se encuentra en estado rechazada se dará la posibilidad de borrar el registro de dicha negociación del listado.
- Si la negociación se encuentra en estado recibida se iniciará la pantalla de recepción de negociación.

4.2.12. Pantalla de revisión de negociación



Figura 4.15: Pantalla de revisión de negociación

El propósito de esta pantalla es revisar que datos personales se han solicitado y cuales se han ofrecido a cambio. Está compuesta por:

1. Una columna con los diferentes campos de un usuario y un switch para indicar que datos han sido solicitados.

2. Una columna con los diferentes campos de un usuario y un switch para indicar que datos se han ofrecido.
3. Un botón para mandar una contraoferta.
4. Un botón para eliminar la negociación, avisar al servidor para que también borre dicha solicitud y volver a la pantalla del listado de negociaciones.
5. Un botón para cancelar y volver a la pantalla del listado de negociaciones.

Cuando se pulsa el botón de contraoferta se envía al servidor que la solicitud actual ha sido rechazada y se inicia la pantalla envío de negociación. Finalmente se retorna al listado de negociaciones.

4.2.13. Pantalla de recepción de negociación



Figura 4.16: Pantalla de recepción de negociación

El propósito de esta pantalla es revisar que datos personales se han solicitado y cuales se han ofrecido a cambio. Está compuesta por:

1. Una columna con los diferentes campos de un usuario y un switch para indicar que datos han sido solicitados.
2. Una columna con los diferentes campos de un usuario y un switch para indicar que datos se han ofrecido.
3. Un botón para aceptar la negociación.
4. Un botón para mandar una contraoferta.
5. Un botón para rechazar la negociación, avisar al servidor de que dicha solicitud ha sido rechazada y volver a la pantalla del listado de negociaciones.
6. Un botón para cancelar y volver a la pantalla del listado de negociaciones.

Cuando se pulsa el botón de aceptar negociación se comprueba que el usuario tenga rellenos los campos que ha ofrecido, en caso de estar alguno vacío se avisa de que debe rellenarlos para cumplimentar el envío. Posteriormente se envían al servidor los datos seleccionados para que este se encargue de su transmisión al otro usuario. Además, en local se añade a la lista de contactos guardados el nuevo contacto obtenido. Cuando se pulsa el botón de contraoferta se envía al servidor que la solicitud actual ha sido rechazada y se inicia la pantalla envío de negociación. Finalmente se retorna al listado de negociaciones.

Capítulo 5

Conclusiones

En este capítulo se exponen las conclusiones a las que se han llegado tras haber realizado el proyecto y posibles mejoras que se pueden desarrollar en la aplicación en el futuro.

5.1. Consecución de objetivos

Se ha desarrollado una plataforma que permite a un usuario poder evaluar salarialmente el trabajo que se desempeña actualmente o de ofertas de trabajo que posea. Para ello se hace uso de un intercambio de información de primera mano con otros usuarios del mismo mercado laboral.

Se ha salvaguardado la integridad, la confidencialidad y la disponibilidad de los datos de los usuarios.

Sin embargo, dentro de los objetivos propuesto no se ha conseguido un intercambio de información en tiempo real. Se ha desarrollado un sistema de notificaciones con tecnología pull, en la cual la aplicación cliente pide al servidor de forma periódica actualizaciones. Además, se ha ofrecido la posibilidad de pedir manualmente dichas actualizaciones de manera ortogonal a las automáticas. Para solventar completamente este objetivo una posible solución sería implementar notificaciones push, esta se encuentra planteada en la sección trabajos futuros.

Con los resultados obtenidos del presente trabajo se puede concluir que la aplicación cubre los problemas y las motivaciones planteadas.

5.2. Trabajos futuros

Una vez finalizado el Trabajo Fin de Grado se consideran las siguientes ideas y funcionalidades que pueden ser implementadas en el futuro:

- Cambiar el sistema actual pull por las notificaciones push.
- Realizar la aplicación cliente en un sistema híbrido para permitir su acceso a usuarios que no sean Android.
- Realizar algún método de importación y exportación de usuarios, de forma que un mismo usuario pueda tener la aplicación con sus datos en diferentes dispositivos.
- Realizar algún sistema en el que se premie el buen uso y la colaboración en la aplicación, ya sea apareciendo en las listas de búsqueda mediante un ranking o alguna marca que los identifique como buenos usuarios.

Bibliografía

- [1] Interfaz gráfica de usuario, documentación: https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario.
- [2] Página oficial de Pencil: <https://pencil.evolus.vn/>.
- [3] Página oficial de Lucidchart: <https://www.lucidchart.com/pages/es>.
- [4] Uml, documentación: https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado.
- [5] Página oficial de Google: <https://www.google.com/intl/es/about/products/>.
- [6] Página oficial de Android Studio: <https://developer.android.com/studio/>.
- [7] Página oficial de Android: <https://www.android.com/>.
- [8] Página oficial de IntelliJ IDEA: <https://www.jetbrains.com/idea/>.
- [9] Página oficial de IBM: <https://www.ibm.com/es-es/>.
- [10] Página oficial de Eclipse: <https://www.eclipse.org/>.
- [11] James Gosling, creador de Java: https://es.wikipedia.org/wiki/James_Gosling.
- [12] Página oficial de PostgreSQL: <https://www.postgresql.org/>.
- [13] Linus Torvalds, diseñador de Git: https://es.wikipedia.org/wiki/Linus_Torvalds.

- [14] Página oficial de Git: <https://git-scm.com/>.
- [15] Página oficial de GitHub: <https://github.com/>.
- [16] Página oficial de Ruby on Rails: <https://rubyonrails.org/>.
- [17] Página oficial de Travis CI: <https://travis-ci.org/>.
- [18] Yaml, documentación: <https://es.wikipedia.org/wiki/YAML>.
- [19] Página oficial de Heroku: <https://www.heroku.com/>.
- [20] Página oficial de Indeed: <https://www.indeed.es/>.
- [21] Página oficial de LinkedIn: <https://es.linkedin.com/>.
- [22] Página oficial de InfoJobs: <https://www.infojobs.net/>.
- [23] Página oficial de CornerJob: <https://www.cornerjob.com/es/company/>.
- [24] Página oficial de Job Today: <https://jobtoday.com/es/>.
- [25] Página comparación salarial de TuSalario.es: <https://tusalario.es/salario/comparatusalario#/>.
- [26] Página comparación salarial de OCCMundial.com: <https://h3.occ.com.mx/Mercer>.
- [27] Página comparación salarial de Hays.es: <http://guiasalarial.hays.es/trabajador/calculadora>.
- [28] Página oficial de GSyC: <https://gsyc.urjc.es/>.
- [29] Página oficial de la Universidad Rey Juan Carlos: <https://www.urjc.es/>.
- [30] Repositorio de FairWage en GitHub: <https://github.com/slnruben/FairWageApp>.
- [31] Repositorio de la Memoria: <https://github.com/slnruben/MemoriaTFG-RubenSanchezRivero>.