



FIXER

App de búsqueda de profesionales mediante
geolocalización

Pablo Menéndez Navarro
2º DAM "A"

Tabla de contenido

1.	Introducción.....	3
a.	Contexto	3
b.	Presentación.....	3
c.	Objetivos.....	3
2.	Análisis y diseño de la aplicación	3
a.	Análisis de requisitos funcionales	3
b.	Tabla de requisitos	3
c.	Diagrama de casos de uso de la app	6
d.	Diagrama y descripción	6
	Actores.....	6
	Casos de Uso	7
	Ejemplo.....	7
e.	Diagrama de interfaces	8
	Diagramas y descripción	8
f.	Análisis de las estructuras de datos utilizadas	12
	Diagramas y descripción	12
g.	Fase de Pruebas	14
	Pruebas de validación de código	14
	Pruebas de implantación en distintos soportes	14
3.	Implementación	15
a.	Tecnologías utilizadas.....	15
4.	Puesta en Marcha. Despliegue.....	16
5.	Aplicación en el ámbito Empresarial.....	17
a.	Idea de negocio	17
b.	Planificación de necesidades de RRHH	17
c.	Planificación de necesidades de producción	18
d.	Planificación de la Prevención de Riesgos Laborales	18
e.	Previsiones de inversión y financiación	18
6.	Conclusión.....	19
a.	Valoración personal	19
b.	Posibles ampliaciones.....	19
7.	Bibliografía y webgrafía.....	19
8.	Anexos.....	20
a.	Manual de Usuario	20

b. Código completo del proyecto.....	27
--------------------------------------	----

1. Introducción

a. Contexto

El presente proyecto se basa en el desarrollo de Fixer, una aplicación móvil híbrida de búsqueda de profesionales a domicilio mediante geolocalización, respondiendo a la falta de presencia de soluciones tecnológicas en el ámbito profesional cotidiano.

Este trabajo surge como una oportunidad para aplicar los conocimientos adquiridos durante el ciclo formativo, integrando herramientas de programación en Java y SpringBoot, como TypeScript con Angular e Ionic, así como participar íntegramente en la creación de un proyecto de principio a fin monitorizado y gestionado por uno mismo.

b. Presentación

Fixer ha sido diseñada para acercar a profesionales de todo tipo al usuario, y viceversa. En este documento se detallará el proceso de análisis, diseño, implementación y despliegue, ofreciendo una visión completa del desarrollo y su aplicabilidad en un contexto real.

c. Objetivos

El objetivo general es crear una aplicación funcional en la que destaque la técnica y la organización con la que se aplican las distintas tecnologías aprendidas en clase, teniendo como referencia las siguientes premisas:

- Desarrollar una aplicación funcional que cumpla con los requisitos establecidos.
- Aplicar metodologías de diseño y desarrollo aprendidas durante el ciclo.
- Garantizar la usabilidad y escalabilidad de la solución propuesta.
- Evaluar la viabilidad de la aplicación en un entorno empresarial.
- Ganar experiencia y soltura en las tecnologías aprendidas.

2. Análisis y diseño de la aplicación

a. Análisis de requisitos funcionales

Los requisitos funcionales definen las funcionales previstas desde un comienzo que conforman la app de Fixer, los cuales, fueron elaborados mediante el análisis y el estudio tanto del entorno como los posibles usuarios, tratando de satisfacer sus necesidades.

b. Tabla de requisitos

A continuación, se muestran los requisitos planteados:

Número de requisito	RF1
Nombre de requisito	Login/Logout
Descripción del requisito	Al iniciar la app deberá aparecer una pagina de login, dando la opción de elegir si quieres una cuenta de profesional o de cliente.
Prioridad	<u>Alta</u>
Casos de uso asociados	Iniciar y cerrar sesión.

Ilustración 1. RF1 Login-Logout

Número de requisito	RF2
Nombre de requisito	Buscador de servicios con filtrado
Descripción del requisito	Una vez buscado el servicio que queremos se nos abrirá la página de los profesionales, donde podremos filtrar en base a nuestras preferencias (horario, duración, turnos, etc.)
Prioridad	<u>Media</u>
Casos de uso asociados	Buscador de servicios con filtros

Ilustración 2. RF2 Buscador de servicios

Número de requisito	RF3
Nombre de requisito	Contratación
Descripción del requisito	Cuando el cliente elige el profesional se abrirá la ventana de contratación, donde habrá que confeccionar la cita. A final de la página habrá un botón de contratar, activando así el evento, que tendrá que ser aceptado por el profesional.
Prioridad	<u>Alta</u>
Casos de uso asociados	Contratar servicio

Ilustración 3. RF3 Contratación

Número de requisito	RF4
Nombre de requisito	Gestiona
Descripción del requisito	Al finalizar el servicio, el cliente tendrá la posibilidad de valorar el servicio prestado por el profesional.
Prioridad	<u>Baja</u>
Casos de uso asociados	Puntuar al profesional

Ilustración 4. Gestión y contratación

Número de requisito	RF5
Nombre de requisito	Servicio de valoraciones
Descripción del requisito	El usuario administrador tendrá la potestad de actuar en todos los ámbitos de la app, asegurando así, el buen uso de la app (eliminar valoraciones, bloquear usuarios)
Prioridad	<u>Media</u>
Casos de uso asociados	Gestionar usuarios

Ilustración 5. Servicio de valoraciones y admin

Número de requisito	RF6
Nombre de requisito	Gestión de perfil
Descripción del requisito	Tanto los profesionales como los clientes pueden adaptar su perfil dependiendo de sus intereses (turnos, horario, localización, etc.)
Prioridad	<u>Baja</u>
Casos de uso asociados	Gestión de perfil

Ilustración 6. Gestión de perfil

Número de requisito	RF7
Nombre de requisito	Aprobación de contratación
Descripción del requisito	Cuando un cliente ofrece un trabajo, el profesional será notificado. El profesional podrá aceptar o denegar la propuesta.
Prioridad	<u>Alta</u>
Casos de uso asociados	Aceptar trabajos

Ilustración 7. Aprobación de contratación desde profesional

c. Diagrama de casos de uso de la app

El siguiente diagrama de casos de uso muestra como interactúan los actores (Cliente, Profesional, Administrador) con la app, ayudando a visualizar las funcionalidades de la app.

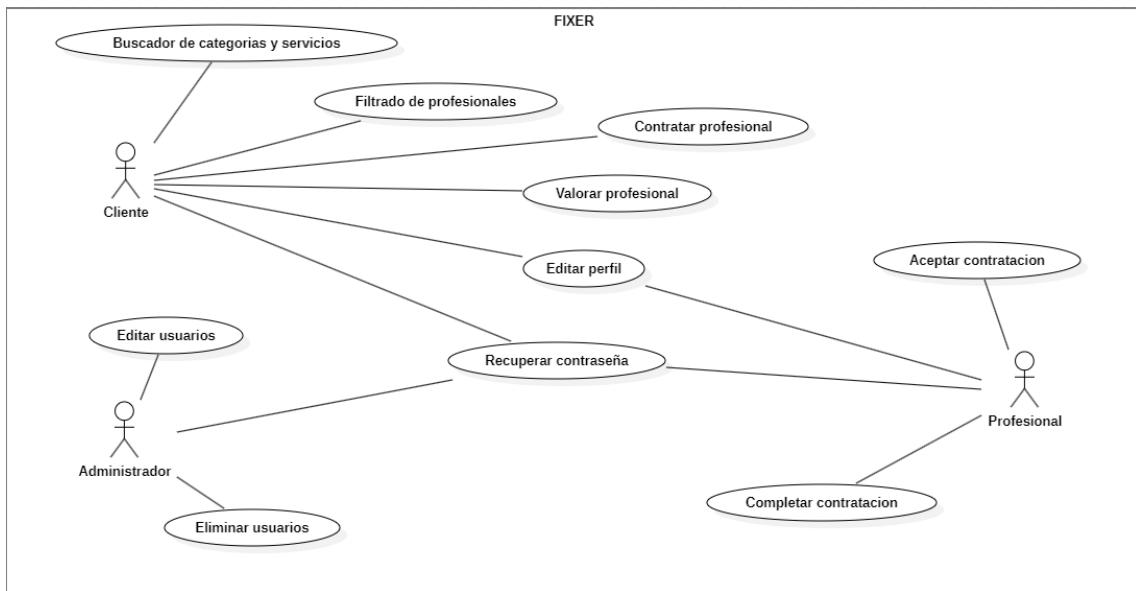


Ilustración 8. Diagrama de casos de uso

d. Diagrama y descripción

Actores

El diagrama presenta tres actores principales que interactúan con el sistema Fixer. El primero es el **Cliente**, quien tiene la capacidad de explorar y gestionar la oferta de profesionales disponibles en la plataforma, además de poder añadir como favorito a cualquier profesional.

El segundo actor es el **Profesional**, encargado de recibir y gestionar las contrataciones propuestas por los clientes, además de revisar las valoraciones recibidas.

Por último, el **Administrador** tiene un rol de supervisión y control, con la autoridad para modificar o eliminar usuarios y gestionar sus perfiles dentro de la aplicación.

Casos de Uso

1. **Buscador de categorías y servicios:** Este caso de uso permite al Cliente buscar y explorar las diferentes categorías y servicios disponibles en la plataforma, facilitando la identificación de opciones que se ajusten a sus necesidades.
2. **Añadir un profesional a favoritos:** Un cliente puede añadir a su lista de favoritos a cualquier profesional entrando en su perfil, una vez añadido se podrá visualizar en la página "Favoritos".
3. **Filtrado de profesionales:** El Cliente puede aplicar filtros para refinar la lista de profesionales según criterios específicos, como experiencia, ubicación o especialización, para encontrar el perfil más adecuado.
4. **Contratar profesional:** El Cliente tiene la posibilidad de seleccionar y contratar a un profesional de su elección, iniciando el proceso de contratación a través de la plataforma.
5. **Valorar profesional:** Tras completar un servicio, el Cliente puede evaluar la actuación del Profesional, dejando una valoración que influirá en su reputación dentro del sistema.
6. **Editar perfil:** El Profesional puede modificar su perfil para actualizar información relevante, como habilidades o disponibilidad, manteniendo su presentación lo más actualizada posible.
7. **Recuperar contraseña:** Tanto el Cliente como el Profesional pueden usar esta funcionalidad para restablecer su acceso al sistema en caso de olvidar su contraseña.
8. **Aceptar contratación:** El Profesional recibe las solicitudes de contratación y tiene la opción de aceptarlas, formalizando el acuerdo con el Cliente.
9. **Completar contratación:** Una vez aceptada, el Profesional debe completar los detalles de la contratación, como fechas o condiciones específicas, para dar inicio al servicio.
10. **Editar usuarios:** El Administrador puede modificar los datos de cualquier usuario (Cliente o Profesional) para corregir errores o actualizar información.
11. **Eliminar usuarios:** El Administrador tiene la capacidad de eliminar usuarios de la base de datos si es necesario, por ejemplo, en casos de incumplimiento de las normas de la plataforma.

Ejemplo

1. **Buscador de categorías y servicios:** Un Cliente busca "reparación de electrodomésticos" y encuentra varias categorías relacionadas, como "reparación de lavadoras" o "reparación de neveras".

2. **Añadir un profesional a favoritos:** Un cliente queda satisfecho con el servicio ofrecido y decide marcar como favorito al profesional, pudiendo acceder a él de manera más sencilla y directa.
3. **Filtrado de profesionales:** El mismo Cliente filtra los resultados por profesionales cercanos a su ciudad y con más de 3 estrellas de valoración media, reduciendo la lista a tres opciones.
4. **Contratar profesional:** El Cliente selecciona a un profesional de la lista y envía una solicitud de contratación para reparar su lavadora.
5. **Valorar profesional:** Tras finalizar el servicio, el Cliente deja una valoración de 5 estrellas al profesional por su rapidez y eficacia.
6. **Editar perfil:** El Profesional actualiza su perfil editando su nombre de usuario.
7. **Recuperar contraseña:** Un Profesional olvida su contraseña y usa la opción de recuperación, recibiendo un enlace por correo para restablecerla.
8. **Aceptar contratación:** El Profesional recibe la solicitud del Cliente y la acepta, comprometiéndose a realizar la reparación en el horario acordado.
9. **Completar contratación:** El Profesional completa los detalles de la contratación, acordando una visita a las 10:00 AM y confirmando su disponibilidad.
10. **Editar usuarios:** El Administrador edita el perfil de un Cliente para corregir su dirección tras un error reportado.
11. **Eliminar usuarios:** El Administrador elimina a un Profesional de la plataforma tras detectar múltiples quejas por incumplimiento de las normas de la aplicación.

e. Diagrama de interfaces

El diseño de la interfaz de la app prioriza una estética atractiva, moderna y minimalista, utilizando una paleta de colores suave y tipografías elegibles, tratando de reducir la fatiga visual.

Se ha decidido utilizar una disposición de las páginas en distintos tabs, buscando mejorar la experiencia del usuario.

Se han incluido elementos como barras de búsqueda, botones de filtrado, muy sencillos e intuitivos con la intención de minimizar la curva de aprendizaje tratando de llegar a la mayor cantidad de usuarios posible.

Diagramas y descripción

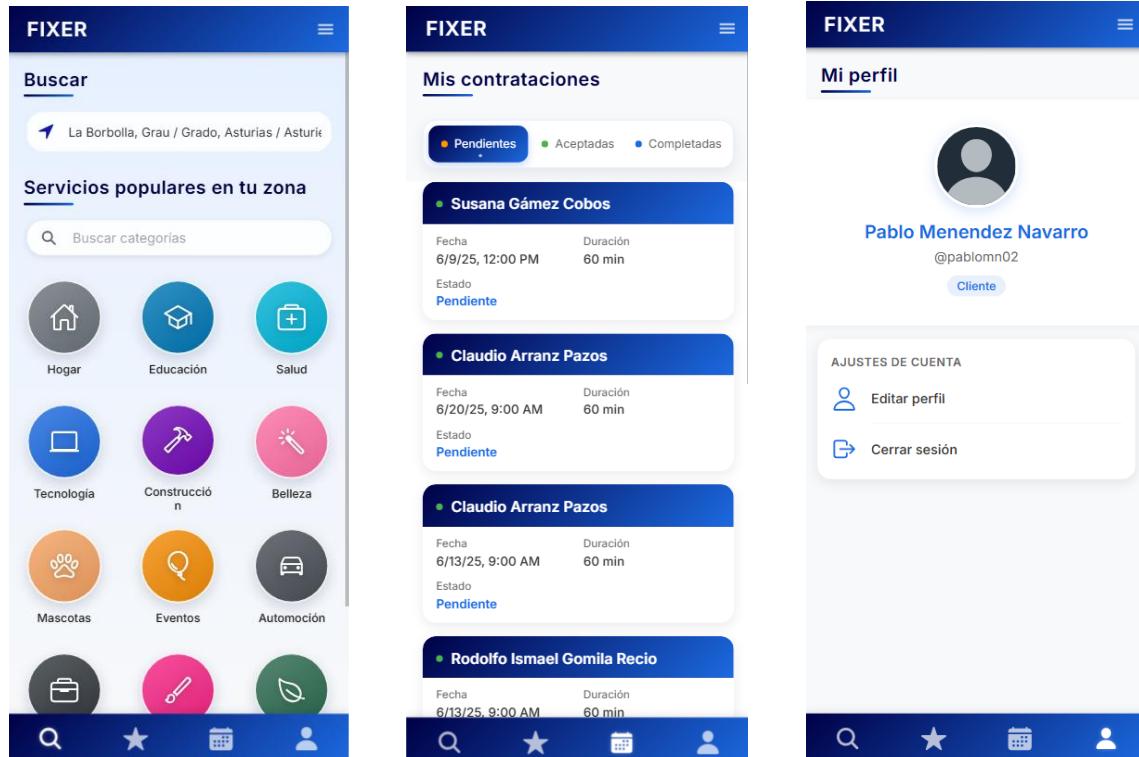
A continuación, se van a enumerar y detallar la disposición de cada una de las páginas de la app. Lo primero que hay que tener claro es que la app tiene distintas partes en función del tipo de usuario que inicie sesión en ella:

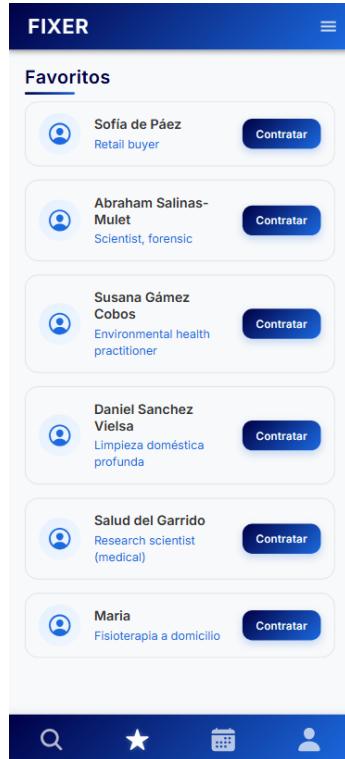
- **Inicio de sesión / Registro**

- Antes de iniciar sesión, Fixer ofrece un menú donde se puede registrar e iniciar sesión, además de una opción de recuperación de contraseña en caso de haberla olvidado.
- La página de registro está dividida en dos posibilidades; registro como cliente o como profesional.

- **Tabs de cliente:**

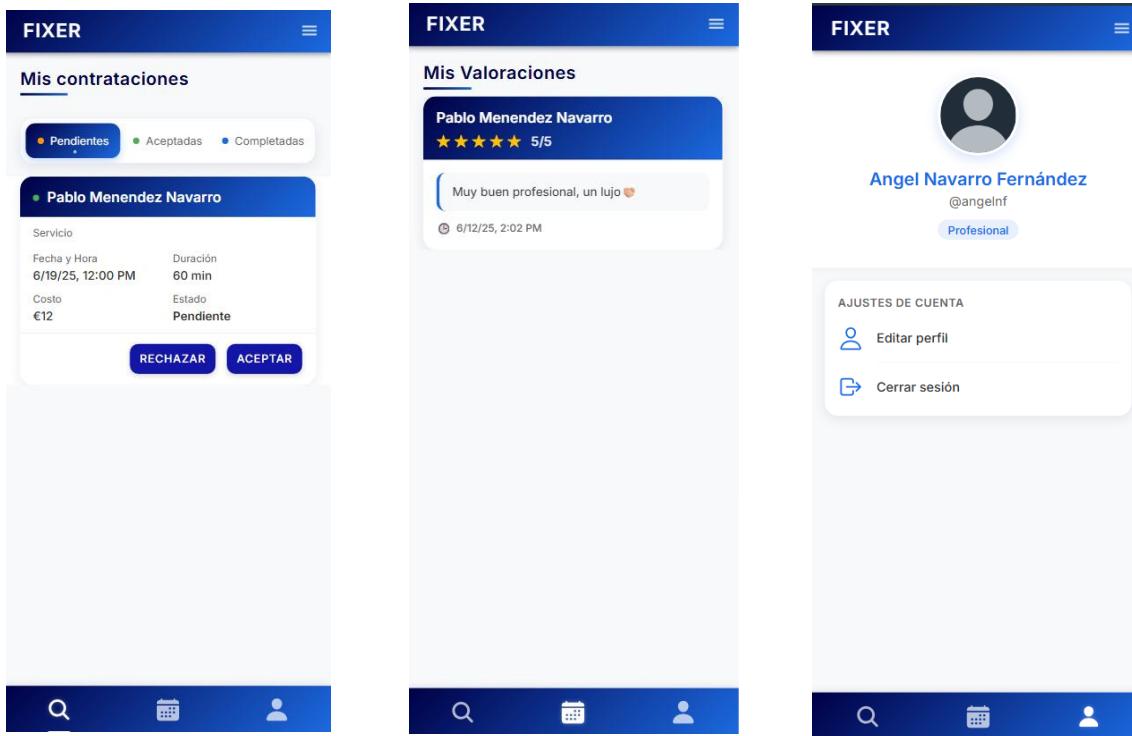
- Pantalla de inicio (también denominada *página de búsqueda*), donde el usuario visualiza la oferta de servicios que dispone la app en su zona
 - Servicios: Se accede a ella cuando se selecciona una categoría, se muestran los servicios dentro de cada categoría
 - Profesionales: una vez seleccionado un servicio, se muestran los profesionales que ofertan esa especialidad, con la posibilidad de contratarlos, añadirlos a favoritos, etc.
- Página de favoritos: Donde se muestran los profesionales favoritos del usuario desde donde también se puede acceder a la contratación después de comprobar si su ubicación está dentro del rango considerado.
- Página de contrataciones: Resumen de las contrataciones hechas desde esa cuenta, organizadas en “Pendientes”, “Aceptadas”, “Completadas”. Cuando un profesional marca una contratación como “Completada”, se activa la opción de valorar el servicio de ese profesional.
- Página de perfil: Gestión de perfil y cierre de sesión





- **Tabs de profesional:**

- Página de contrataciones: Muy similar al funcionamiento de la pagina de contrataciones de cliente. Ayuda a gestionar las contrataciones de una manera muy sencilla y visual.
- Página de Valoraciones: Muestra un resumen de las valoraciones que tiene un profesional, mostrando el cliente que lo valoró, el horario, el comentario y las estrellas.
- Página de Perfil: Gestión de perfil y cierre de sesión



- **Tabs de administrador:**

- Gestión de usuarios: Se muestran todos los usuarios divididos en desplegables por cada tipo de usuario. El administrador posee la capacidad de editar o eliminar cualquier usuario de la base de datos.
- Página de Perfil: Gestión de perfil y cierre de sesión

f. Análisis de las estructuras de datos utilizadas

El análisis de las estructuras de datos para la aplicación "Fixer" se enfoca en el diseño de una base de datos relacional que soporte eficientemente las interacciones entre los actores (Cliente, Profesional y Administrador), así como las funcionalidades clave como la gestión de contrataciones, valoraciones y perfiles.

Esta estructura se ha optimizado para garantizar integridad, escalabilidad y acceso rápido a la información, utilizando una arquitectura relacional que minimiza redundancias y asegura relaciones coherentes entre las entidades.

No se emplean ficheros externos, priorizando una base de datos centralizada para un manejo robusto de los datos.

Diagramas y descripción

El modelo de datos se basa en un diagrama entidad-relación que refleja las tablas principales y sus relaciones, alineado con las necesidades de la aplicación. A continuación, se describen las tablas clave y su estructura:

- Usuarios:** Contiene la información básica de todos los actores (ID_Usuario, Nombre, Apellido_Usuario, Correo, Contraseña, Tipo_Usuario [Cliente, Profesional, Administrador], ID_Usuario).
- Password_Reset_Tokens:** Gestiona los tokens para recuperación de contraseñas (Token, ID_Usuario, Expiry_Date).
- Administradores:** Almacena datos específicos de administradores (ID_Usuario, Ultimo_Acceso).

- **Clientes:** Registra preferencias de los clientes (ID_Usuario, Preferencias).
- **Profesionales:** Incluye detalles como especialidad, precio por hora, disponibilidad, experiencia, certificaciones, ubicación (latitud, longitud) y total de contrataciones (ID_Usuario, Especialidad, Precio_Hora, Disponibilidad, Experiencia, Certificaciones, Latitud, Longitud, Total_Contrataciones).
- **Categorías:** Lista las categorías de servicios (ID_Categoría, Nombre, Descripción).
- **Servicios:** Detalla los servicios asociados a categorías y profesionales (ID_Servicio, Nombre, Descripción, ID_Categoría, Servicio_ID).
- **Profesional_Servicios:** Relaciona profesionales con servicios que ofrecen (ID_Profesional_Servicio, ID_Usuario, ID_Servicio, Descripcion_Servicio).
- **Contrataciones:** Registra las contrataciones con estado, duración y costo (ID_Contratacion, ID_Profesional_Servicio, ID_Usuario, Fecha_Hora_Servicio, Estado [pendiente, aceptada, rechazada, en curso, completada], Duracion_Estimada, Costo_Total, ID_Contratacion).
- **Horas_Ocupadas:** Almacena las horas ocupadas por servicios (ID_Hora_Ocupada, ID_Profesional_Servicio, Fecha, Hora_Inicio, Hora_Fin, Estado).
- **Valoraciones:** Guarda las evaluaciones de clientes a profesionales (ID_Valoracion, ID_Usuario, ID_Usuario_Profesional, Puntuacion, Comentario, Fecha_Timestamp, ID_Valoracion).

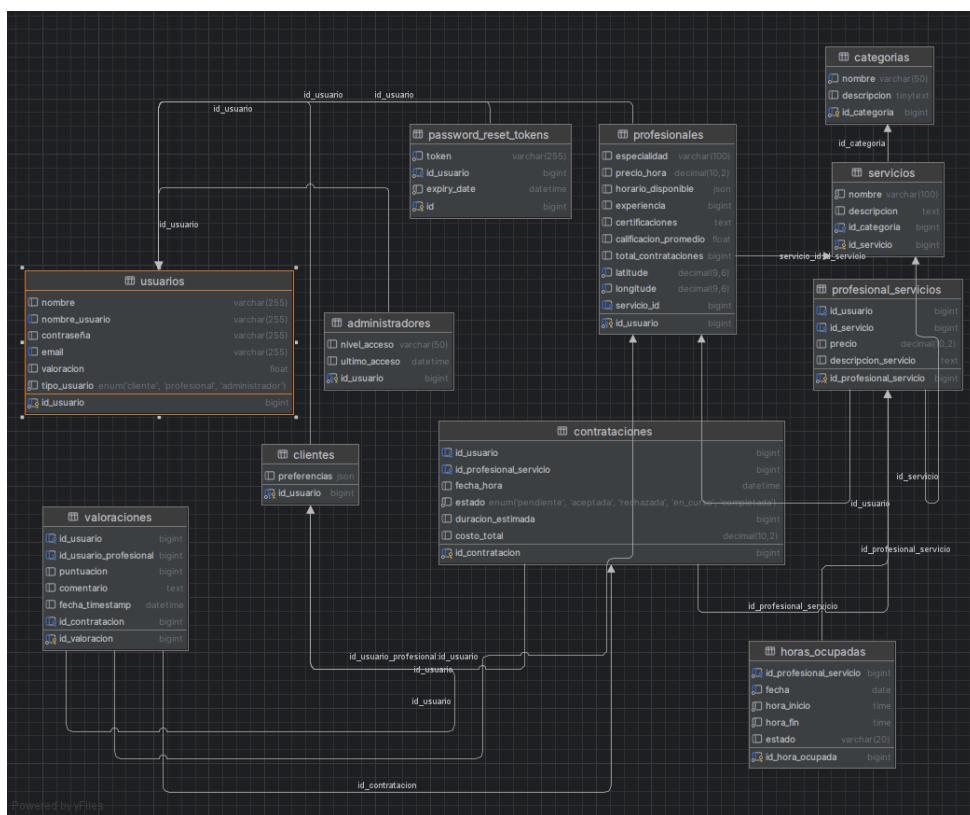


Ilustración 9. Diagrama Relacional de la BBDD

g. Fase de Pruebas

Pruebas de validación de código

La validación de código incluye un conjunto de pruebas unitarias e integración para asegurar el correcto funcionamiento de cada componente.

Se utilizaron herramientas como el entorno web y sus herramientas de desarrollador para la realización de pruebas, enfocándose en módulos clave como autenticación, gestión de contrataciones y actualización de valoraciones.

Se realizaron pruebas manuales para escenarios específicos, como la recuperación de contraseñas o la eliminación de usuarios por el Administrador. Los resultados se documentaron en un informe detallado para garantizar la calidad del código antes de la implantación.

Pruebas de implantación en distintos soportes

Las pruebas de implantación se llevaron a cabo principalmente en entornos web y móvil para evaluar la compatibilidad y rendimiento de "Fixer". En el entorno web, se probaron navegadores como Chrome, Opera y Safari, verificando la responsividad de la interfaz y el tiempo de carga de funciones como el buscador y el filtrado.



En el entorno móvil, se han utilizado dispositivos iOS y Android, tanto mediante emuladores como hardware real (Android Studio), para asegurar una experiencia fluida en diferentes resoluciones.



3. Implementación

a. Tecnologías utilizadas

- Frontend: Se empleó TypeScript junto con Ionic y Angular para desarrollar una interfaz de usuario dinámica, responsive y compatible con múltiples plataformas, facilitando una experiencia fluida tanto en entornos web como móviles.



- Backend: Se utilizó Java con SpringBoot para construir un servidor robusto y escalable, integrando JWT para implementar un sistema de autenticación seguro y eficiente.



- Base de datos: MySQL fue seleccionado como sistema de gestión de bases de datos relacionales, optimizado para almacenar y consultar eficientemente las entidades y relaciones definidas en el modelo de datos de "Fixer".



- Herramientas: Se utilizaron GitHub para el control de versiones y la colaboración en equipo, Docker para la contenedorización y despliegue consistente del entorno, Termius para la gestión remota de servidores, y OpenVPN para establecer la conexión con el servidor del instituto.



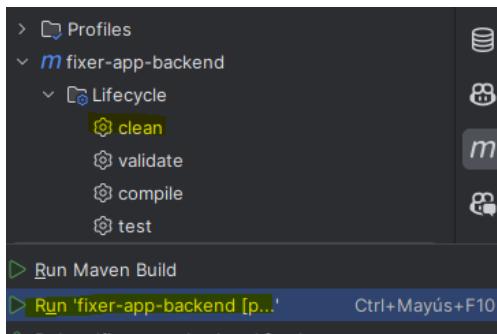
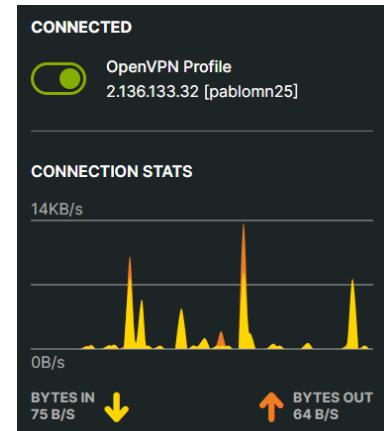
4. Puesta en Marcha. Despliegue

1. Configurar los archivos Dockerfile:

Se definieron los archivos Dockerfile para el frontend y el backend, especificando las imágenes base (Node.js para Ionic y Java para SpringBoot) y las dependencias necesarias, asegurando un despliegue consistente y reproducible en cualquier entorno compatible con Docker.

2. Configurar OpenVPN y Termius:

Se instaló y configuró OpenVPN para establecer una red privada virtual que protegiera las comunicaciones entre el servidor y los desarrolladores, mientras que Termius se utilizó para gestionar de forma remota el acceso al servidor, garantizando un control seguro y eficiente.

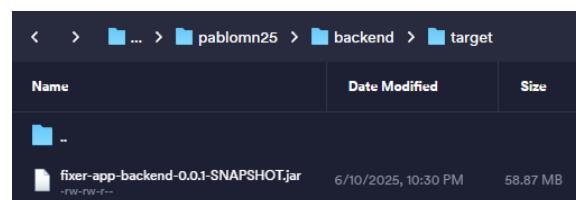


3. Generar el JAR en IntelliJ con Maven:

En IntelliJ, se ejecutaron los comandos maven clean y maven package para compilar el back, generando un archivo JAR ejecutable que contendría toda la lógica del servidor.

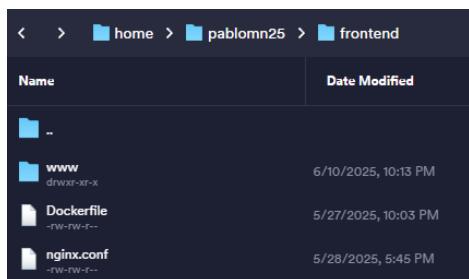
4. Copiar el JAR a la carpeta backend/target:

El archivo JAR generado se trasladó manualmente a la carpeta backend/target, preparándolo para ser integrado en el entorno contenedorizado del backend.



5. Ejecutar Ionic build --prod en Visual Studio Code:

Se lanzó el comando ionic build --prod en Visual Studio Code para construir la versión optimizada del frontend, generando la carpeta www que contiene los archivos estáticos del frontend.



6. Mover la carpeta www al frontend:

La carpeta www generada se copió al directorio del frontend, asegurando que los archivos estáticos estén disponibles para ser servidos por el servidor web.

7. Insertar el script de la base de datos en la carpeta db:

Se colocó el script SQL de la base de datos en la carpeta db, permitiendo la inicialización o actualización de la base de datos MySQL con las tablas y datos necesarios.

8. Utilizar la dirección de la base de datos:

Se configuró la conexión a la base de datos utilizando la dirección <https://pablomn25.iesmontenaranco.com:8000>, asegurando que el backend y el frontend puedan interactuar con los datos almacenados.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
2a55b9109300	pablomn25-backend	"java -jar app.jar"	41 hours ago	Up 41 hours	0.0.0.0:8000->8080/tcp, ::::8000->8080/tcp
30fbe742ffa4	pablomn25-frontend	"/docker-entrypoint..."	41 hours ago	Up 41 hours	0.0.0.0:80->80/tcp, ::::80->80/tcp
0c010f03e14d	mysql:8	"docker-entrypoint.s..."	2 weeks ago	Up 2 weeks	0.0.0.0:3306->3306/tcp, ::::3306->3306/tcp
		fixer_app-mysql			

5. Aplicación en el ámbito Empresarial

a. Idea de negocio

La idea de negocio de "Fixer" se centra en ofrecer una plataforma digital que conecte a clientes con profesionales independientes para servicios de reparación y mantenimiento, como fontanería, electricidad o reparación de electrodomésticos...

El modelo de negocio se basa en una suscripción mensual baja para profesionales (10-15 €) y una comisión del 5-10% por cada contratación exitosa, manteniendo costos accesibles para usuarios y profesionales.

La aplicación busca captar un mercado local inicial, expandiéndose gradualmente a medida que crezca la base de usuarios, aprovechando la demanda de servicios rápidos y confiables en un entorno digital intuitivo.

b. Planificación de necesidades de RRHH

El equipo inicial consistirá en un equipo pequeño y eficiente: un desarrollador full-stack (para mantener y mejorar la plataforma), un gestor de comunidad/marketing (para promoción y atención al cliente), y un administrador de sistemas (para gestionar servidores y seguridad).

Se contratará personal a tiempo parcial o freelance para reducir costos, con un presupuesto estimado de 2,000-3,000 € mensuales.

A medida que crezca la base de usuarios, se considerará incorporar un soporte técnico adicional y un especialista en ventas, priorizando contrataciones locales para minimizar gastos de desplazamiento.

c. Planificación de necesidades de producción

La producción se limita al desarrollo y mantenimiento de la plataforma "Fixer". No se requiere fabricación física, solo la actualización continua del software y la base de datos.

El enfoque inicial será optimizar las funcionalidades existentes (buscador, contrataciones, valoraciones) con un equipo reducido, invirtiendo unas 20-30 horas semanales de trabajo.

Se utilizarán herramientas gratuitas o de código abierto (como GitHub) y servicios en la nube de bajo costo (como DigitalOcean o Heroku) para alojar la aplicación, con un presupuesto mensual de 50-100 € para infraestructura.



d. Planificación de la Prevención de Riesgos Laborales

Dado que el trabajo es mayoritariamente remoto y digital, los riesgos laborales principales son el estrés por plazos y problemas ergonómicos (como el uso prolongado de pantallas).

Se implementarán medidas como pausas regulares (cada 2 horas), provisión de sillas ergonómicas para el equipo en oficina (si aplica), y formación básica sobre gestión del tiempo.

El presupuesto para estas medidas será mínimo, alrededor de 200-300 € iniciales para equipo y talleres, revisándose anualmente para adaptarse al crecimiento del equipo.

e. Previsiones de inversión y financiación

La inversión inicial se estima en 5,000-7,000 €, cubriendo desarrollo inicial, marketing básico (redes sociales y anuncios locales), y configuración de servidores. Los

fondos se obtendrán mediante ahorros personales o un préstamo pequeño (hasta 5,000 € a 3-5 años con intereses bajos).

Los ingresos iniciales provendrán de las suscripciones y comisiones, proyectando un punto de equilibrio en 6-12 meses con 50-100 profesionales activos. Se reinvertirán las ganancias para mejorar la plataforma, evitando endeudamiento excesivo y manteniendo un enfoque conservador en el crecimiento.

6. Conclusión

a. Valoración personal

Considero que este proyecto ha sido una excelente oportunidad para poner a prueba mis habilidades y demostrar lo que realmente sé hacer, aplicando de manera práctica las tecnologías aprendidas como TypeScript, Angular, SpringBoot y MySQL.

Además, permite desarrollar un portafolio sólido que servirá como un destacado escaparate en una entrevista de trabajo, mostrando mi capacidad para diseñar e implementar una aplicación funcional desde cero. Aunque el proyecto requiera de mucho tiempo, esfuerzo y enfrentar desafíos como la gestión de tiempo, creo que compensa con creces.

b. Posibles ampliaciones

Me han surgido muchas ideas a medida que avanzaba con este proyecto, sobre todo en torno a incluir posibles nuevas funcionalidades como la integración de pagos con plataformas como Stripe, que ofrecerá transacciones seguras y eficientes.

Otra área en la que ahondar es por ejemplo la inclusión de IA (Inteligencia Artificial) de alguna manera en la app, ya sea mediante un Chat Bot adaptado para ayudar frente a posibles incidencias de los usuarios, recomendaciones que mejorarían la experiencia del usuario.

Además, también había pensado en un sistema de envío de notificaciones por correo y mediante la app, por ejemplo, cuando a un profesional le llega una petición de contratación, cuando la marca como completada, cuando un cliente realiza una valoración de su trabajo...

7. Bibliografía y webgrafía

- Documentación de Angular.
<https://angular.io/docs>.
- Documentación de TypeScript.
<https://www.typescriptlang.org/docs>.
- Documentación de SpringBoot.

<https://docs.spring.io/spring-boot/docs/current/reference/html>.

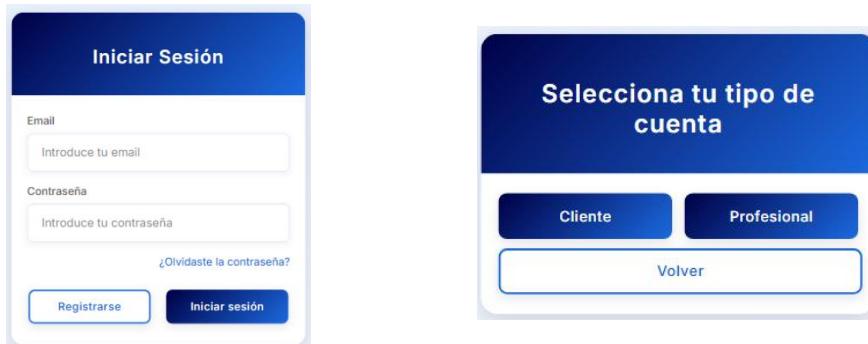
- Documentación de MySQL
<https://dev.mysql.com/doc>.
- Documentación de Ionic.
<https://ionicframework.com/docs>.
- Stack Overflow.
<https://stackoverflow.com>.
- Documentación de Docker.
<https://docs.docker.com>.
- Consultas de desarrollo a Inteligencia Artificial

8. Anexos

a. Manual de Usuario

INICIO DE SESION / REGISTRO

Esta es la página de inicio de sesión, donde introduciremos nuestro correo y contraseña. Si no tenemos una cuenta, podemos ir a la pagina de registro, donde se nos preguntará si queremos crear una cuenta de cliente o profesional.



Aquí simplemente elegimos la cuenta. Para esta prueba vamos a escoger la de profesional porque es más completa. Rellenamos los campos y pulsamos en registrar, se nos redirigirá al login.

Crear Cuenta Profesional

Nombre

Email

Nombre de usuario

Especialidad

Ej: Carpintero, Electricista, etc.

Precio por Hora (€)

Ubicación

Mapa



Coordenadas Seleccionadas

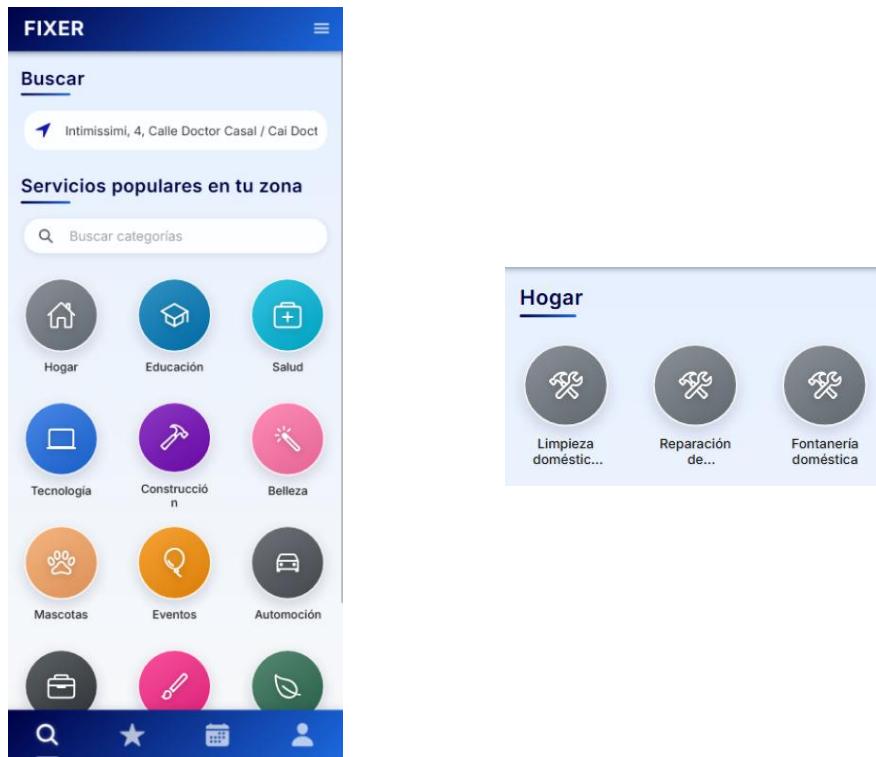
Latitud: 43.3645
Longitud: -5.8506

Seleccionar Ubicación

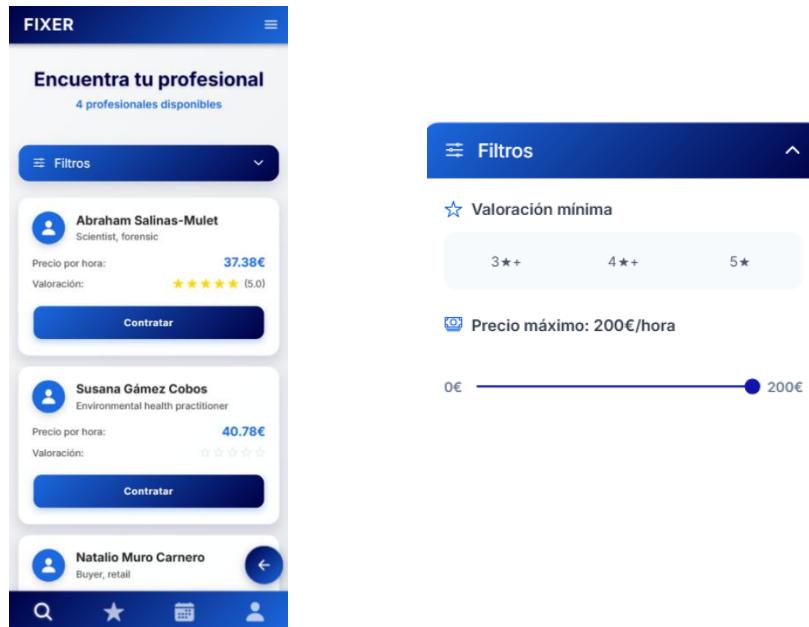
Una vez hemos introducido los datos, dependiendo del tipo de cuenta que esta iniciando sesión se nos redirigirá a unos tabs u otros, ahora hemos iniciado sesión con un cliente.

TABS DE CLIENTE

En la página de búsqueda, donde la app detecta la ubicación del usuario y muestra las categorías. Una vez seleccionamos una de las categorías, se muestran los servicios de cada categoría. Aquí vemos que la categoría seleccionada tiene 3 servicios.



Esta es la página donde se muestran los profesionales, vemos que se visualizan los profesionales con su especialidad, el precio por hora, la valoración y el botón de contratar. Podremos filtrar por estrellas de valoración o por precio por hora.

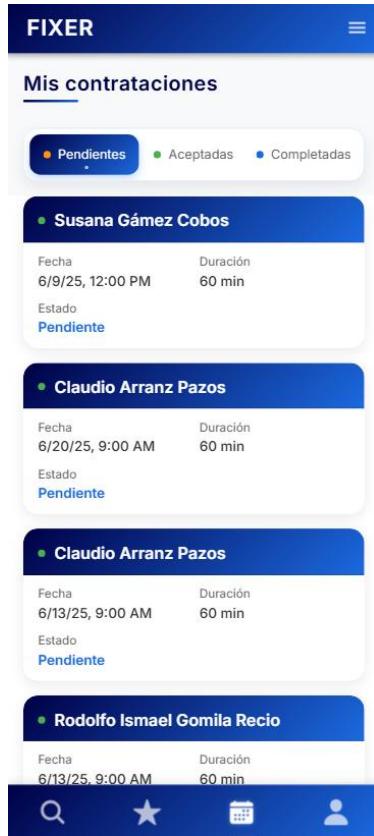


Si pulsamos en contratar, se nos abrirá el modal de la contratación, donde podremos ver los horarios disponibles del profesional, vemos que, para el día seleccionado, tiene ocupadas las 9.00 y las 12.00. Para contratar al profesional, pulsamos en la hora que queramos y confirmamos la contratación.

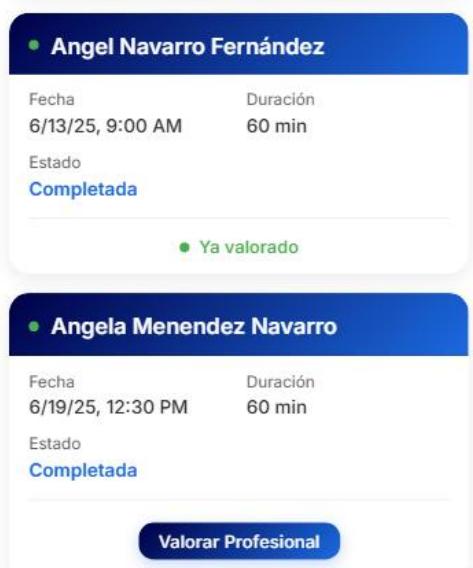


Vamos ya con la pantalla de favoritos, donde se almacenarán los profesionales que hemos marcado como favoritos, y desde donde también podremos contratarlos, después de que la app confirme si estamos en la misma zona que el profesional. Vemos también el modal de visualización del perfil de profesional, donde vemos algunas de las características del profesional y donde también podemos marcarlo o desmarcarlo como favorito.

Ahora vamos con la pagina de contrataciones. Vemos que están divididas en Pendientes, Aceptadas y Completadas. Pudiendo tener un mejor control de todas las contrataciones que tenemos abiertas.

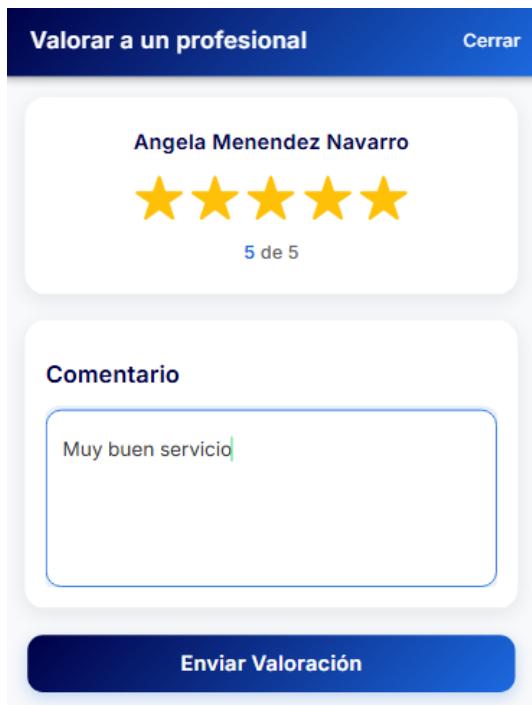


Una vez el profesional actualiza el estado de una contratación a Completada, se habilita la función de valorar el servicio:



Las contrataciones que ya han sido valoradas están marcadas como “Ya valorado”, mientras que las que no hemos valorado nos da la opción de hacerlo.

Se abrirá el modal de valoración, podemos elegir las estrellas y poner un comentario.



La pagina de Perfil, tiene un resumen de algunos de los datos del usuario (Nombre, nombre de usuario, tipo de usuario, ...). Además tiene la opción de poder cambiar o editar algunos de estos campos:

Mi perfil

Pablo Menendez Navarro
@pablomn02
Cliente

AJUSTES DE CUENTA

- Editar perfil
- Cerrar sesión

Editar Perfil

Nombre	Pablo Menendez Navarro
Correo	pablomnavarro9@gmail.com
Usuario	pablomn02
Contraseña	[Redacted]

GUARDAR

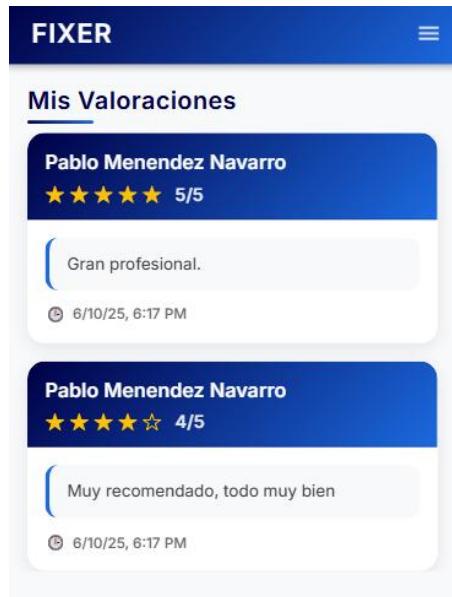
Esta página es igual para todos los tipos de cuentas

TABS DE PROFESIONAL

Una vez iniciada la sesión como profesional, se nos redirige a la página de contrataciones, donde el usuario podra visualizar las contrataciones activas



La pagina de valoraciones:



TABS DE ADMINISTRADOR

La parte de los tabs de administrador se basan en la gestion de usuarios. Esta organizada en tres desplegables divididos entre Clientes, Profesionales y Todos los usuarios.

Gestión de Usuarios

Todos los Usuarios (60)

Nombre	Usuario	Papel
Abraham Salinas-Mulet	abraham_salinas23	Profesional
Susana Gámez Cobos	emmamartinez1	Profesional
Genoveva Ripoll Agullo	abascalmiguela2	Profesional
Rodolfo Ismael Gomila Recio	ncastrillo3	Profesional
Sofía de Páez	amaya824	Profesional
Salud del Garrido	rosendo125	Profesional

Gestión de Usuarios

Todos los Usuarios (60)

Clientes (3)

Profesionales (55)

b. Código completo del proyecto

Aquí tenemos el código completo en GitHub

<https://github.com/DAM2A-PDAM-24-25/FixerApp-PabloMenendez-FrontEnd>

<https://github.com/DAM2A-PDAM-24-25/FixerApp-PabloMenendez-BackEnd>