

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



UNIVERSITAS
Miguel Hernández

“Prototipo de sistema de valoración de servicios
basado en contactos reales con, interfaz
Android y servicio Web”

TRABAJO FIN DE GRADO

Julio - 2023

AUTOR: Horlando Rafael Illaris Avilés

DIRECTOR/ES: Miguel Onofre Martínez Rach



RESUMEN

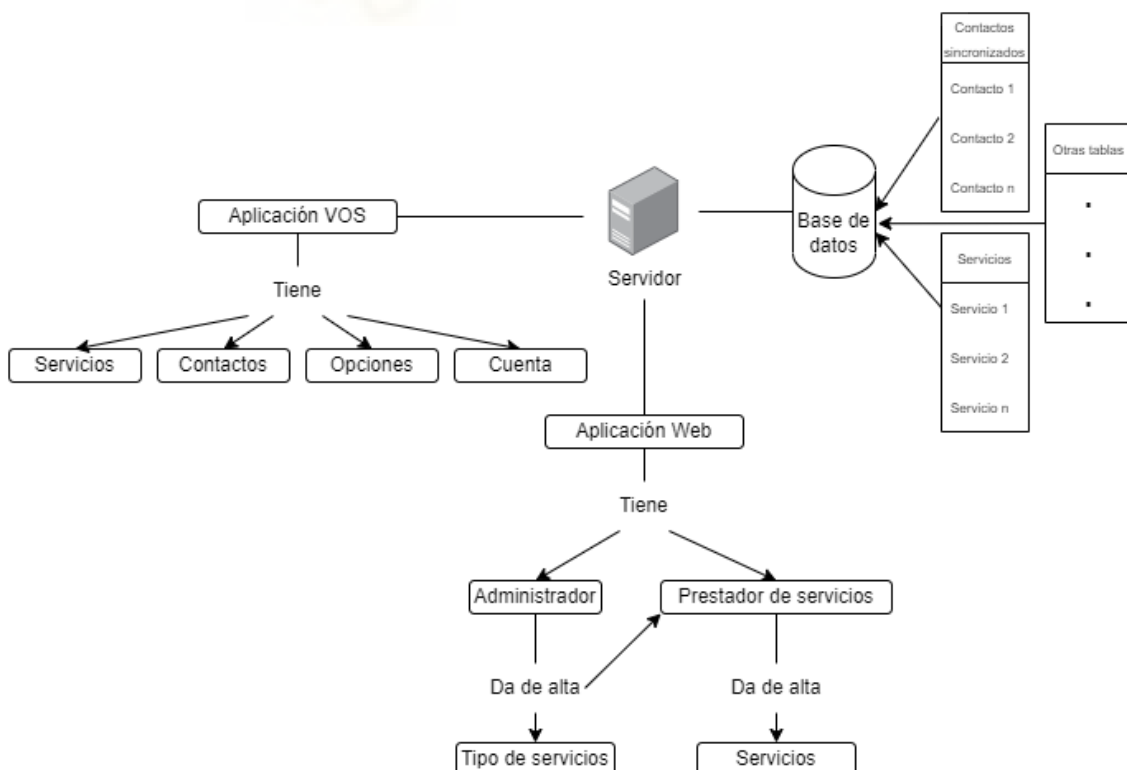
El proyecto consiste en el desarrollo de una aplicación Android para el dispositivo móvil y una aplicación Web. En el dispositivo móvil se tendrá en cuenta los contactos de la agenda, puesto que al registrarse en la aplicación, se pueda realizar un comentario y una valoración sobre un determinado servicio y poder ver los comentarios y valoraciones de nuestros contactos y de los contactos de estos previamente sincronizados.

Los servicios serán dados de alta por un prestador de servicios mediante el uso de la aplicación web. Los servicios serán de varios tipos: bar, restaurante, hotel, etc.

Tendremos un administrador que se encargará de dar de alta a los prestadores de servicios y también podrá dar de alta los tipos de servicios, todo esto mediante una aplicación web.

La aplicación Android tiene como nombre VOS, cuyo significado es, *valoración y opinión de servicios*. Lo novedoso de esta aplicación será que podremos ver las valoraciones y opiniones de nuestros contactos y de los contactos de estos en un determinado servicio. Esto nos servirá para tener una cierta confianza en el momento de elegir un servicio, gracias a dichas valoraciones y opiniones.

El siguiente esquema muestra las diferentes partes del proyecto.



Palabras clave: React Native, Laravel, Aplicación web, mysql, base de datos, servicios, locales, contactos.



ÍNDICE GENERAL

RESUMEN	3
Capítulo 1 Introducción.....	14
1.1. LAS RESEÑAS EN INTERNET	14
1.2. JUSTIFICACIÓN DEL PROYECTO	15
1.3. OBJETIVOS	16
1.3.1. Objetivos principales	16
1.3.2. Objetivos secundarios.....	17
1.3.3. Objetivos personales.....	17
Capítulo 2 Antecedentes y estado de la cuestión.....	19
2.1. APLICACIONES, PRESTADORES DE SERVICIOS Y SUS RESEÑAS....	19
2.1.1. Google maps con lugares o servicios	19
2.1.2. Four Square.....	21
2.1.3. Trip Advisor	23
2.1.4. Yelp	26
2.1.5. Resumen	28
Capítulo 3 Tecnologías.....	30
3.1. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO.....	30
3.1.1. Editor de código.....	31
3.1.1.1. Visual studio code	31
3.1.2. Administrador de código	32
3.1.2.1. Git.....	32
3.1.3. Gestión de proyecto	34
3.1.3.1. Atlassian Jira	34
3.1.4. Herramienta para diseño de base de datos.....	35
3.1.4.1. Lucidchart.....	35
3.1.5. Prototipo aplicación móvil.....	37
3.1.5.1. Figma.....	37
3.1.6. Diseño gráfico.....	39
3.1.6.1. Gimp.....	39
3.2. VALORACIÓN Y COMENTARIO.....	40
3.3. FRONTEND y BACKEND	42
3.3.1. Frontend.....	43
3.3.1.1. Html.....	43

3.3.1.2.	Bootstrap	45
3.3.1.3.	React native	49
3.3.2.	Backend	52
3.3.2.1.	Laravel.....	52
3.3.2.2.	Gestor de base de datos Mysql.....	54
3.3.2.3.	Servidor web apache	56
3.3.2.4.	Firebase	57
3.3.2.5.	Node js.....	60
Capítulo 4	Metodología, Estructura y Resultados	62
4.1.	PLANIFICACIÓN DEL PROYECTO	62
4.1.1.	Ciclo de vida.....	63
4.1.1.1.	Scrum en Jira.....	63
4.2.	DISEÑO	65
4.2.1.	Base de datos	65
4.2.1.1.	Modelo Entidad-Relación	67
4.2.1.2.	Modelo relacional.....	67
4.2.2.	Logo aplicación	68
4.2.2.1.	Diseño del logo.....	68
4.2.3.	Prototipo aplicación Android.....	70
4.2.3.1.	Pantallas del prototipo.....	70
4.3.	CAPTURA DE REQUISITOS	72
4.3.1.	Roles de usuarios	72
4.3.2.	Casos de uso	75
4.4.	IMPLEMENTACIÓN.....	90
4.4.1.	Aplicación Android - frontend	90
4.4.1.1.	Creación proyecto	90
4.4.1.2.	Screens (pantallas en React Native).....	95
4.4.1.3.	Componentes.....	98
4.4.1.4.	Librería para obtener contactos del móvil.....	99
4.4.1.5.	Contactos.....	100
4.4.1.6.	Servicios	101
4.4.2.	Aplicación móvil - backend.....	102
4.4.2.1.	Usuario	102
4.4.2.2.	Servicios	103
4.4.2.3.	Contactos.....	103

4.4.3.	Aplicación web - frontend	104
4.4.3.1.	Estructura del administrador	104
4.4.3.2.	Código IU administrador.....	105
4.4.3.3.	Código IU prestador de servicios	106
4.4.4.	Aplicación web - backend	107
4.5.	RESULTADOS.....	108
4.5.1.	Aplicación móvil	108
4.5.1.1.	Crear cuenta.....	108
4.5.1.2.	Inicio de sesión.....	109
4.5.1.3.	Menú de la aplicación	110
4.5.1.4.	Cuenta de usuario	111
4.5.1.5.	Opciones.....	112
4.5.1.6.	Contactos.....	113
4.5.1.7.	Lista de servicios.....	114
4.5.1.8.	Servicio.....	115
4.5.2.	Aplicación web	119
4.5.2.1.	Panel de prestadores de servicios - administrador	119
4.5.2.2.	Pantalla tipo de servicios - administrador	120
4.5.2.3.	Pantalla de servicios - prestador de servicios.....	121
Capítulo 5	Conclusiones y Trabajo futuro.....	123
5.1.	CONCLUSIONES	123
5.2.	TRABAJO FUTURO.....	124
Bibliografía	126

ÍNDICE DE TABLAS

Tabla 2.1. Características de aplicaciones	29
Tabla 3.1. Versiones de html	44
Tabla 4.1. Rol de usuario. Usuario app	72
Tabla 4.2. Rol de usuario. Prestador de servicios.....	73
Tabla 4.3. Rol de usuario. Administrador	74
Tabla 4.4. Caso de uso. Registro en aplicación Android.....	75
Tabla 4.5. Caso de uso. Registro en aplicación Web	76
Tabla 4.6. Caso de uso. Inicio de sesión en aplicación Android	76
Tabla 4.7. Caso de uso. Inicio de sesión en aplicación Web.....	77
Tabla 4.8. Caso de uso. Cerrar sesión en aplicación Android	77
Tabla 4.9. Caso de uso. Cerrar sesión en aplicación web.....	78
Tabla 4.10. Caso de uso. Ver lista de servicios en aplicación Android/web.....	78
Tabla 4.11. Caso de uso. Ver lista de contactos en aplicación Android.....	78
Tabla 4.12. Caso de uso. Lista prestadores de servicios en aplicación Web.....	79
Tabla 4.13. Caso de uso. Lista tipo de servicios en aplicación web.....	79
Tabla 4.14. Caso de uso. Ver un servicio en aplicación Android.....	80
Tabla 4.15. Caso de uso. Publicar reseña en aplicación Android.....	80
Tabla 4.16. Caso de uso. Editar una reseña en aplicación Android	81
Tabla 4.17. Caso de uso. Eliminar reseña en aplicación Android	81
Tabla 4.18. Caso de uso. Alta servicio en aplicación web	82
Tabla 4.19. Caso de uso. Editar servicio en aplicación web	82
Tabla 4.20. Caso de uso. Ver servicio en aplicación web	83
Tabla 4.21. Caso de uso. Baja servicio en aplicación web	83
Tabla 4.22. Caso de uso. Alta prestador de servicios en aplicación web	84
Tabla 4.23. Caso de uso. Editar prestador de servicios en aplicación web	85
Tabla 4.24. Caso de uso. Información, prestador de servicios en aplicación web	85
Tabla 4.25. Caso de uso. Baja prestador de servicios en aplicación web.....	86
Tabla 4.26. Caso de uso. Alta tipo de servicio en aplicación web	86
Tabla 4.27. Caso de uso. Editar tipo de servicio en aplicación web	87
Tabla 4.28. Caso de uso. Modificar alias y correo en aplicación Android.....	87
Tabla 4.30. Caso de uso. Cambiar color de fondo en las cards en aplicación Android .	88
Tabla 4.31. Caso de uso. Desincronizar contactos en Aplicación Android.....	88
Tabla 4.32. Caso de uso. Sincronizar contactos en aplicación Android.....	89

Tabla 4.33. Caso de uso. Eliminar cuenta en Aplicación Android..... 89



ÍNDICE DE FIGURAS

Figura 2.1. Google maps lugares o servicios.....	21
Figura 2.2. FourSquare servicios nocturnos	22
Figura 2.3. FourSquare pub	23
Figura 2.4. Trip Advisor restaurante	24
Figura 2.5. Trip Advisor comentarios	25
Figura 2.6. Yelp bicicletas.....	27
Figura 2.7. Yelp bicicletas y detalles.....	28
Figura 3.1. Visual studio code	31
Figura 3.2. Versiones en Git.....	33
Figura 3.3. Jira Scrum.....	35
Figura 3.4. Lucidchart plantillas.....	36
Figura 3.5. Figma prototipo aplicación Android	38
Figura 3.6. Lucidchart plantillas.....	39
Figura 3.7. Gimp.....	40
Figura 3.8. Estructura html.....	43
Figura 3.9. Página dinámica con html y react js.....	44
Figura 3.10. Dependencia agregada en React JS.....	46
Figura 3.11. Estructura de archivos de bootstrap en React JS.....	46
Figura 3.12. Clases de bootstrap en React JS	47
Figura 3.13. Botones de bootstrap	48
Figura 3.14. Barra de navegación de bootstrap	48
Figura 3.15. Todas las versiones de bootstrap.....	48
Figura 3.16. Esquema de React	50
Figura 3.17. Componentes en react	51
Figura 3.18. Modelo MVC	52
Figura 3.19. Laravel.....	53
Figura 3.20. Tabla imágenes con sus atributos.....	54
Figura 3.21. Tabla servicios con sus atributos.....	55
Figura 3.22. SQL Server Y Mysql.....	55
Figura 3.23. Cliente Servidor	56
Figura 3.24. Módulos de apache.....	57
Figura 3.25. OTP Firebase.....	59
Figura 3.26. Plan Spark	59

Figura 3.27. Panel usuario de autenticación	59
Figura 3.28. Node Js	61
Figura 4.1. Épicas	64
Figura 4.2. Incidencias.....	65
Figura 4.3. Epic frontend Android.....	65
Figura 4.4. Modelo E-R.....	67
Figura 4.5. Modelo relacional.....	68
Figura 4.6. Logo VOS diseño en Gimp	69
Figura 4.7. Logo VOS app.....	69
Figura 4.8. Prototipo menú app Android	70
Figura 4.9. Prototipo servicios app Android.....	71
Figura 4.10. Prototipo con un servicio y un comentario	71
Figura 4.11. Esquema usuario app.....	73
Figura 4.12. Esquema prestador de servicios	74
Figura 4.13. Esquema administrador	75
Figura 4.14. Creación proyecto en React Native	91
Figura 4.15. Ficheros del proyecto	91
Figura 4.16. Node modules.....	92
Figura 4.17. Package Json	93
Figura 4.18. Proyecto de prueba.....	93
Figura 4.19. Comando de ejecución de la aplicación.....	94
Figura 4.20. Aplicación inicial de React Native.....	94
Figura 4.21. Estructura de una aplicación en React Native.....	95
Figura 4.22. Estructura de las screens	96
Figura 4.23. Screens	97
Figura 4.24. Hook Use State en SignUpScreen.....	97
Figura 4.25. Axios en React Native.....	98
Figura 4.26. Directorio components	99
Figura 4.27. Librería Contacts.....	99
Figura 4.28. Hook useState.....	99
Figura 4.29. Permisos en Android	100
Figura 4.30. Componente flatlist	101
Figura 4.31. Hook personalizado.....	101
Figura 4.32. Llamada a la API Servicios.....	102
Figura 4.33. Rutas para React Native	103

Figura 4.34. Rutas Api de Servicios	103
Figura 4.35. Rutas de contacto	104
Figura 4.36. Rutas de contacto	105
Figura 4.37. Código de la interfaz del administrador	106
Figura 4.38. Código de la interfaz de los prestadores de servicios	106
Figura 4.39. Rutas del administrador.....	107
Figura 4.40. Rutas middleware.....	107
Figura 4.41. Ruta servicios del prestador de servicios	108
Figura 4.42. Crear cuenta en VOS app.....	109
Figura 4.43. Información password en VOS app	109
Figura 4.44. Inicio de sesión en VOS app	110
Figura 4.45. Menú en VOS app	111
Figura 4.46. Opción cuenta en VOS app	112
Figura 4.47. Opción cambiar color de las cards en VOS app.....	113
Figura 4.48. Contactos en VOS app	114
Figura 4.49. Servicios en VOS app	115
Figura 4.50. Servicio con imagen descriptiva en VOS app.....	116
Figura 4.51. Servicio con imagen descriptiva en VOS app.....	117
Figura 4.52. Comentarios en VOS app.....	118
Figura 4.53. Comentarios en VOS app.....	119
Figura 4.54. Panel del administrador.....	120
Figura 4.55. Crear usuario prestador de servicios	120
Figura 4.56. Listado del tipo de servicios.....	121
Figura 4.57. Listado de servicios.....	121
Figura 4.58. Información acerca de un servicio	122



Capítulo 1

Introducción

1.1. LAS RESEÑAS EN INTERNET

Si navegamos por internet, podemos observar que en casi todo lo relacionado con compras, lugares, contratación de servicios, ocio, etc. Se puede apreciar que casi siempre hay una parte donde podemos ver y comprobar una reseña haciendo referencia a cada uno de estos, detallando la información y características como parte de la reseña de cada usuario. Esto nos da un *feedback* importante, puesto que las personas, para poder elegir, comprar o acudir a un lugar, tendrán en cuenta estas reseñas.

Según una investigación realizada por *BrightLocal*, dice que el **85%** de los internautas leen reseñas antes de comprar y que un **65%** leen un promedio de 6 reseñas de la compra. Esto quiere decir que las reseñas son mucho más importantes de lo que parecen [1].

Hay una parte “menos buena” de las reseñas, esto es debido a que en ocasiones estas pueden ser falsas o posiblemente publicadas por error. Según un estudio realizado por la Organización de Consumidores y Usuarios (OCU), teniendo en cuenta los portales de compraventa de bienes y servicios: **Amazon**, **Tripadvisor** y **Booking**, se realizó un estudio y a partir de los resultados se obtuvo que un 8,4% de Amazon, un 6,2% de Tripadvisor y un 2,1% de Amazon de las reseñas, eran falsas [2]. Por lo tanto, podemos ver que las reseñas van a ser muy importantes para poder adquirir un tipo de producto o acudir o contratar un servicio, pero tendremos que tener en cuenta que esas reseñas no serán del todo fiables cuando al adquirir el producto o servicio pueda tener una sensación de que el producto o servicio previamente elegido, no nos satisface o no es lo que se espera a consecuencia de haber visto una reseña o reseñas que en ese momento nos habían convencido. Obviamente no podemos saber si estas reseñas son verdaderas, falsas o publicadas por error. Pero están allí y de alguna manera nos van a ayudar para poder elegir o no un determinado servicio o producto.

1.2. JUSTIFICACIÓN DEL PROYECTO

Si indagamos a través de Internet y tenemos en cuenta los servicios que nos podemos encontrar para poder contratar o acudir, estos son de varios tipos como por ejemplo, restaurante, bar, panadería, etc. Nos encontraremos con estos lugares o servicios y con sus respectivas valoraciones y comentarios, estos no necesariamente quiere decir que no sean de fiar pero no sabemos con exactitud de quienes son, normalmente son de gente “anónima” para nosotros aunque tengan algún nombre o alias.

Podemos optar por contratar algún tipo de servicio, siempre y cuando las reseñas de esos servicios nos ayuden a tomar una decisión.

Para asegurarnos o tener más confianza si queremos contratar o acudir un servicio entonces se opta por desarrollar esta aplicación *Android* que tiene como nombre **VOS** (valoración y opinión de servicios), gracias a esta aplicación podemos tener en cuenta nuestro entorno (contactos) que comentarán servicios en los que hayan acudido o contratado para posteriormente dejar una valoración y opinión. Lo interesante de esta aplicación es que nosotros podemos optar por contratar o acudir a un lugar cuando se tenga en cuenta las reseñas de nuestros contactos que necesariamente deben estar

registrados en la aplicación. Seguramente tenemos contactos de amistades con los que normalmente se mantiene una relación cercana, estos si están registrados en la aplicación podrían ver nuestra reseña y nosotros ver las suyas, de tal manera que aún más confianza nos daría elegir un servicio que ha sido valorado por uno de nuestros contactos.

En lo personal esta aplicación me ha ayudado a comprender más sobre el desarrollo de una aplicación móvil mucho más extensa en comparación con otras aplicaciones vistas en la carrera. Por otra parte, he obtenido una mejora en el conocimiento de **React Native** en el lado del cliente, **Laravel** en la parte del servidor y también en el diseño de **base de datos**.

1.3. OBJETIVOS

1.3.1. Objetivos principales

El proyecto tiene como objetivo el desarrollo de una aplicación móvil para Android teniendo pendiente el desarrollo para IOS, a consecuencia de temas económicos (no se dispone de un ordenador Apple). También tiene por objetivo satisfacer la necesidad de una persona en el momento de acudir o elegir un servicio teniendo en cuenta un cierto grado de confianza al elegirlo.

Otro de los objetivos se centraría en el que, un prestador de servicios una vez dado de alta en la aplicación pueda llegar a tener una gran reputación gracias a sus servicios prestados, este tendrá dos valoraciones, la de nuestros contactos y una global, esta última será de todos los usuarios registrados que han podido valorar su servicio.

Las funcionalidades que podemos destacar de la aplicación serían las siguientes:

- Registro con nuestra sim del móvil.
- Obtener mediante una lista, todos los servicios disponibles.
- Ver servicios en los que se muestre las valoraciones de nuestros contactos y de los contactos de estos.
- Mostrar las valoraciones mediante cards con la información (alias y número de teléfono) de cada contacto.
- Ver los contactos que tenemos en nuestro dispositivo y comprobar mediante un check los que están registrados y sincronizados en la aplicación.
- Ver información importante (valoraciones con estrellas y fotos) correspondiente a un servicio.

- Editar nuestras valoraciones de un servicio en cualquier momento.
- Cambiar el color de fondo de la card de las valoraciones de nuestros contactos, para que se distingan de la nuestra.
- Modificación de datos personales (alias y email).
- Eliminar cuenta, junto a todos sus datos asociados.

1.3.2. Objetivos secundarios

Estos objetivos buscan que el usuario se sienta conforme con el uso de la aplicación, por ello la aplicación dispone de una interfaz de usuario intuitiva y minimalista, teniendo en cuenta que solamente se trata por el momento de un prototipo. La interfaz es sencilla y su uso no tiene complicaciones, de tal manera que el usuario no tendrá problemas cuando esta aplicación se encuentre instalada en su dispositivo móvil Android.

Otro objetivo más que podemos agregar es, el número de los usuarios registrados de los contactos que tenemos agregados en nuestro dispositivo móvil puedan llegar a ser numerosos, puesto que si más usuarios de nuestros contactos se encuentran registrados y si estos dejan alguna reseña de los servicios, más información nos dan en el momento de poder elegir un servicio gracias a sus reseñas.

1.3.3. Objetivos personales

En este proyecto como objetivos, buscar obtener un conocimiento más amplio sobre el desarrollo de aplicaciones Android mediante la utilización de **React Native** como parte del *front-end*, ya que este tiene muchas ventajas que más adelante se detallaran a fondo. En el *back-end* se implementarán *Apis* desarrolladas con **Laravel** y de esta manera afianzar más conocimiento sobre este último.

Por otra parte también se busca la mejora en el conocimiento y en el diseño de la base de datos, puesto que se requiere un diseño conceptual como primera parte y posteriormente el diseño relacional, todo ello necesario y como base en el proyecto TFG.

También se busca conocer y comprender el trabajo de los servidores para poder almacenar información en una base de datos (en este caso se ha utilizado MYSQL), mediante el envío de información por parte del cliente.



Capítulo 2

Antecedentes y estado de la cuestión

2.1. APLICACIONES, PRESTADORES DE SERVICIOS Y SUS RESEÑAS

En este caso vamos a mencionar algunas aplicaciones que están relacionadas con la prestación de servicios y realizar un estudio de estas, estas aplicaciones son bastante utilizadas por la comunidad debido a su popularidad. Todas tienen algunas características en común y en lo que más nos vamos a fijar o a prestar más atención, es, en las reseñas que cada usuario realiza en un determinado servicio.

Podemos haber elegido más aplicaciones pero se ha decidido por las aplicaciones que tiene bastante popularidad. Mencionaremos algunas de sus características más importantes. Al final tendremos un comentario y valoración de estas aplicaciones mediante una tabla comparativa.

2.1.1. Google maps con lugares o servicios

Es una aplicación de Google, se encuentra disponible para Android e IOS, también hay una versión de escritorio. Esta aplicación tiene varias funcionalidades pero nos vamos a centrar en la parte de lugares o servicios y en las reseñas que son creadas por los usuarios para referenciar a cada servicio.

En esta aplicación podemos buscar restaurantes, barberías, lavaderos de coches, electricistas, etc. Donde podemos acudir para satisfacer alguna necesidad o recomendar a alguien, bien puede ser un amigo o familiar. En este caso tomamos ejemplo de un restaurante y en el podemos ver una parte con las reseñas de personas que no conocemos. Pueden ser personas reales o perfiles falsos. De alguna u otra manera estas reseñas van ser de gran importancia a la hora de tomar una decisión. Si nos fijamos en la imagen (Figura 2.1) podemos ver las estrellas (de color amarillo) y entre paréntesis el número de personas que han valorado dicho restaurante, esto es importante porque a partir de esos datos se obtiene la media de la valoración final del restaurante.

Tiene una interfaz simple y es de fácil uso. Si queremos buscar un servicio solamente debemos realizar una búsqueda en su buscador y nos aparecerá todos los lugares dependiendo de la ubicación física o teniendo en cuenta el mapa de google, en el mapa nos aparecerá un icono haciendo referencia al tipo de servicio y si lo tocamos nos abrirá una ventana con el servicio seleccionado junto a sus detalles. Para dejar una reseña necesariamente debemos estar registrados y también podemos adjuntar fotos.

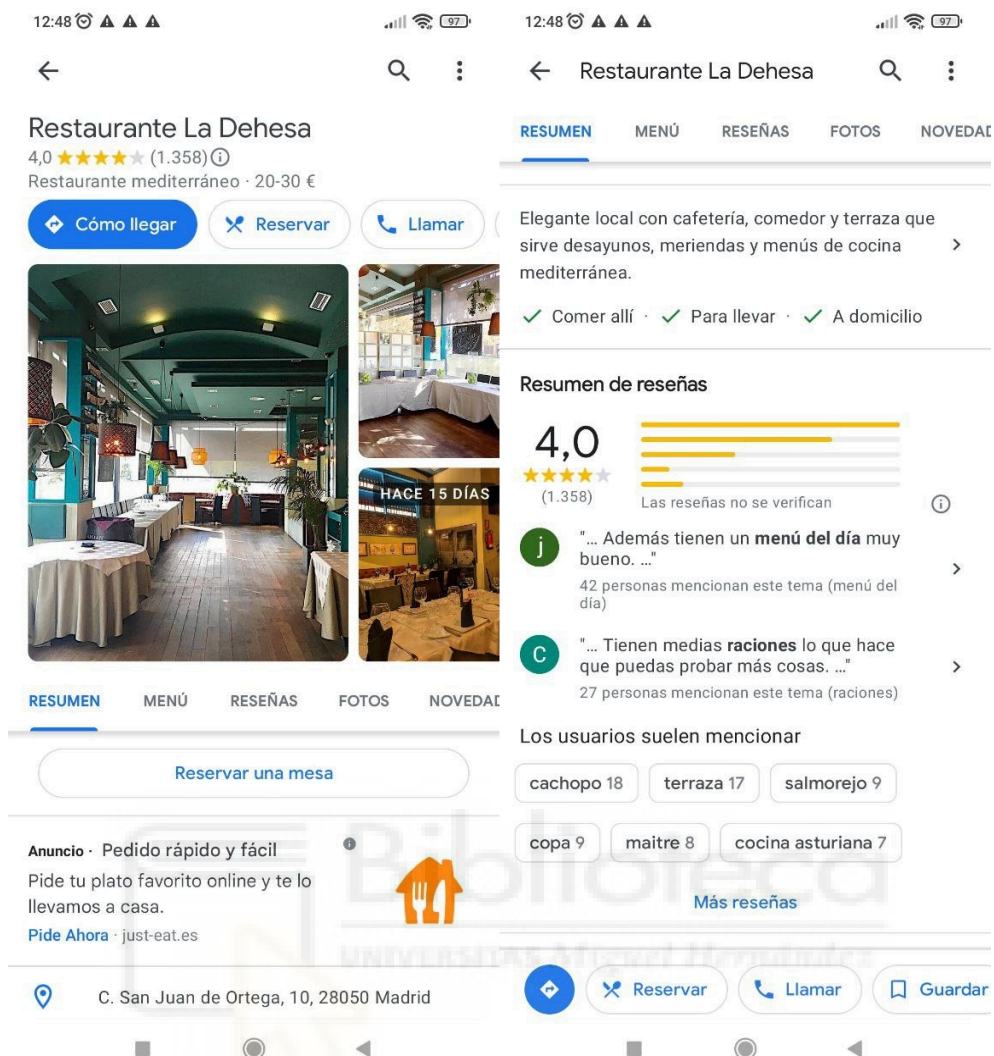


Figura 2.1. Google maps lugares o servicios

La aplicación es de mucha ayuda porque nos permite encontrar algún servicio en el que este previamente se haya dado de alta. Por lo tanto, si hay un nuevo local(servicio) este deberá darse de alta en la aplicación, caso contrario, este no aparecerá en la aplicación.

2.1.2. Four Square

Esta plataforma basada en la nube está disponible para Android, IOS y una versión de escritorio, nos permite buscar lugares y poder acudir a estos a partir de nuestra ubicación. Lo interesante de esta aplicación es que nos permite referenciar un lugar específico en función de nuestra ubicación física.

En esta aplicación también tendremos que estar registrados para poder dejar una reseña, pero se podrán ver las reseñas y lugares sin estar registrados. Lo más interesante de esta aplicación es que funciona con geolocalización y obligadamente debemos tener activada nuestra ubicación, caso contrario no podemos ver toda la funcionalidad disponible de dicha aplicación.

Si tenemos en cuenta las reseñas, en este caso también disponemos de una sección donde se puede observar las distintas reseñas de los usuarios que se encuentran registrados en la aplicación. Esta aplicación tiene una interfaz intuitiva y su uso es bastante fácil. Podemos buscar un tipo de servicio y posteriormente obtendremos los locales que mantendremos cerca de nuestra ubicación. Podemos elegir un servicio y poder ver los detalles de este junto a sus reseñas.

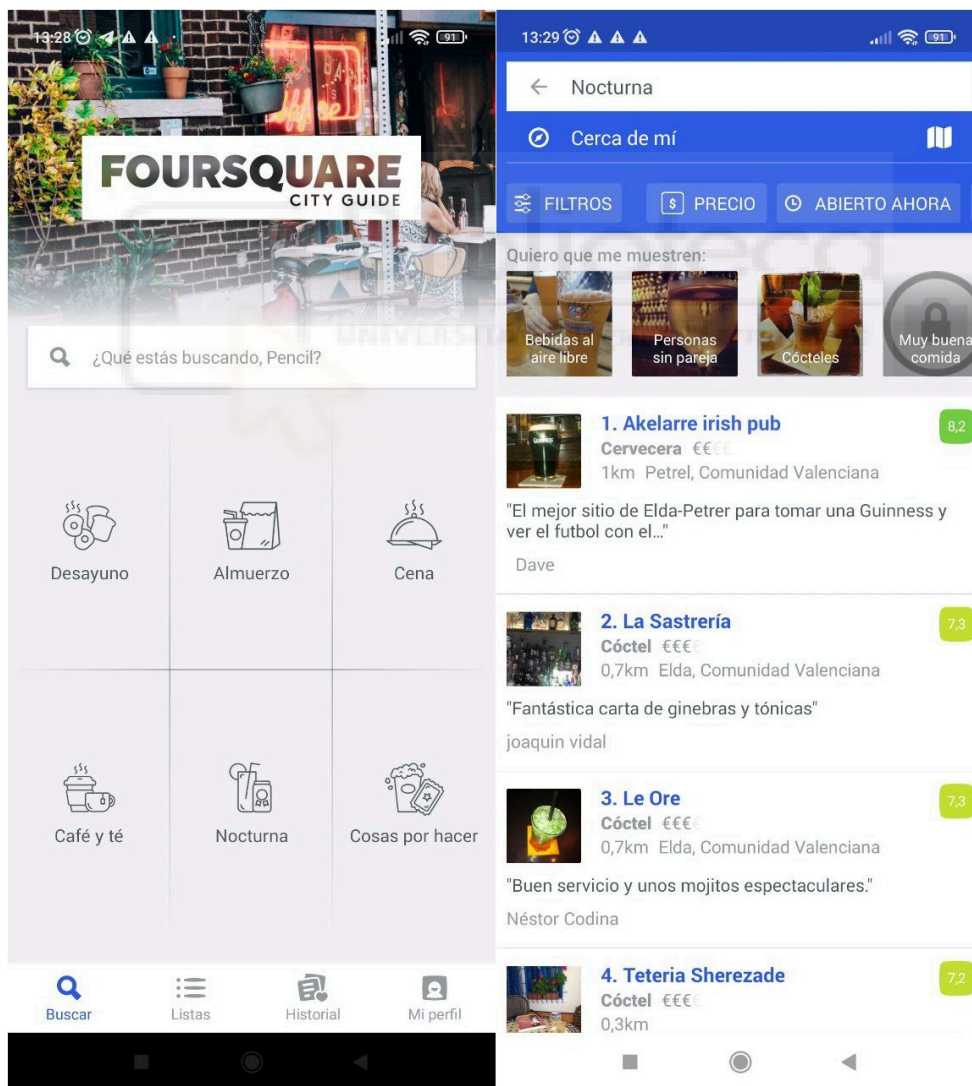


Figura 2.2. FourSquare servicios nocturnos

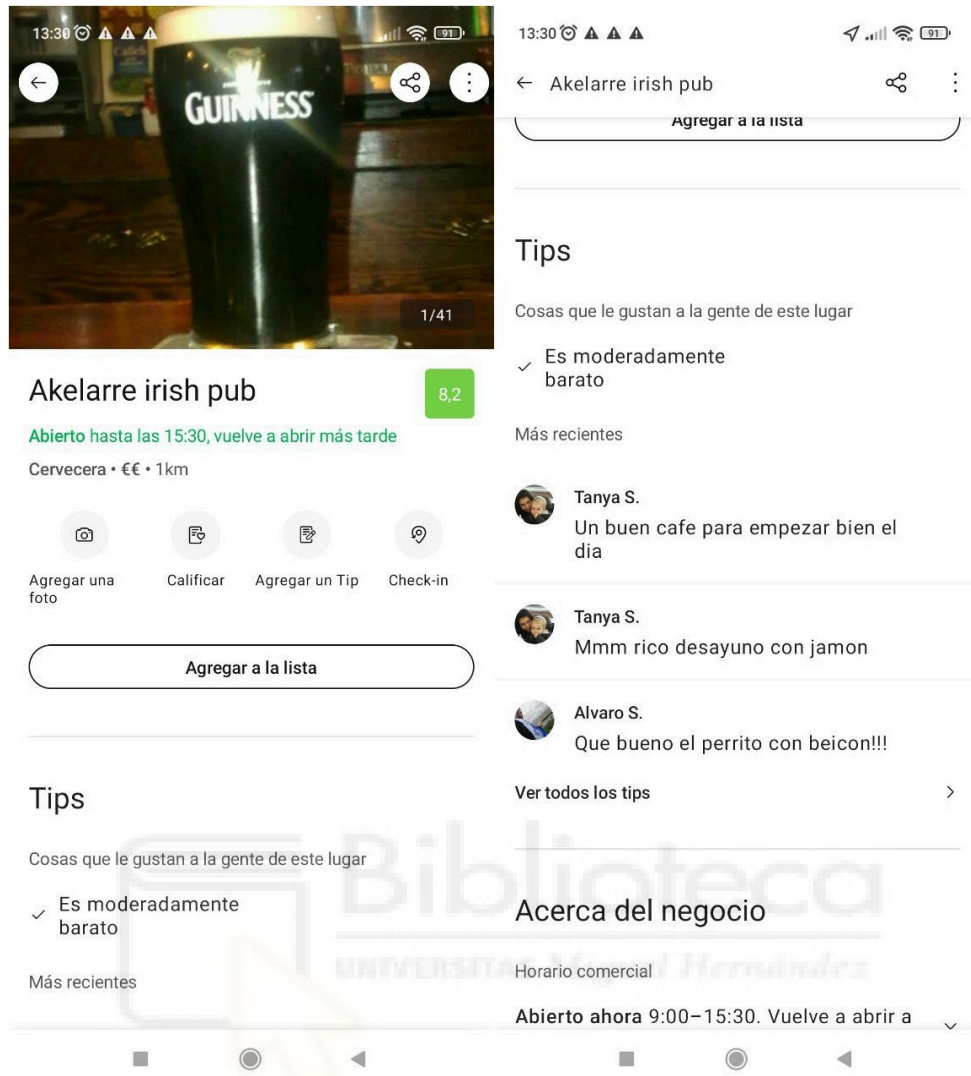


Figura 2.3. FourSquare pub

2.1.3. Trip Advisor

Esta aplicación está disponible para Android, IOS y dispone de una versión de escritorio.

Este portal permite que personas de todo el mundo puedan dejar reseñas de los distintos servicios ofrecidos a estos y de esta manera ayudar al resto de personas a poder tomar una decisión para elegir un servicio. Podemos encontrar varios servicios y estos pueden ser: hoteles, restaurantes, bares, cruceros, etc. Es un portal completo y su uso es fácil. Para poder utilizar todas las funcionalidades debemos estar registrados, aunque si queremos ver o buscar servicios no será obligatorio estar registrado.

En la aplicación disponemos de un buscador en el cual podemos escribir el servicio a buscar y obtendremos los resultados mediante una lista. En esta lista podemos ver un resumen de una parte de su información, en este caso podemos entrar en un servicio y ver sus detalles junto a sus reseñas y más. Si nos fijamos en la parte de las reseñas (Figura 2.5) estas tienen mejor información, puesto que están más detalladas.

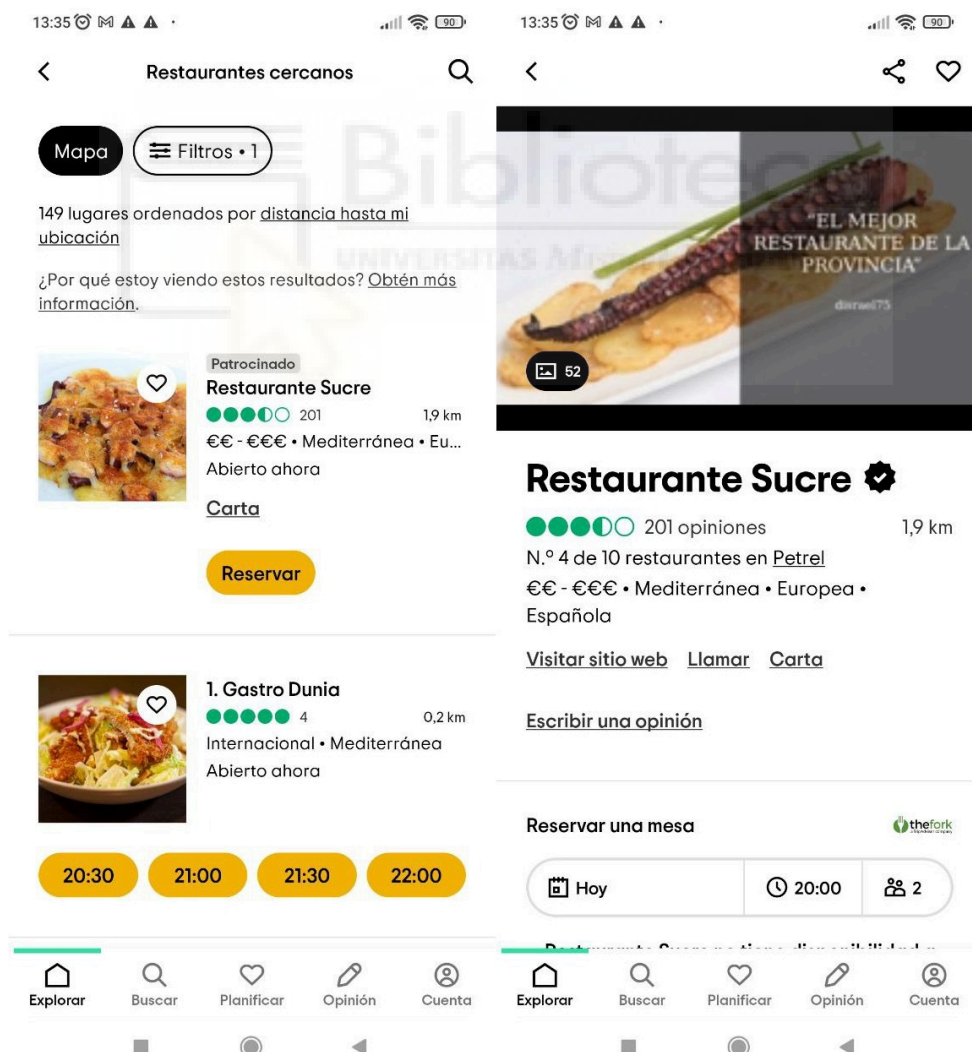


Figura 2.4. Trip Advisor restaurante

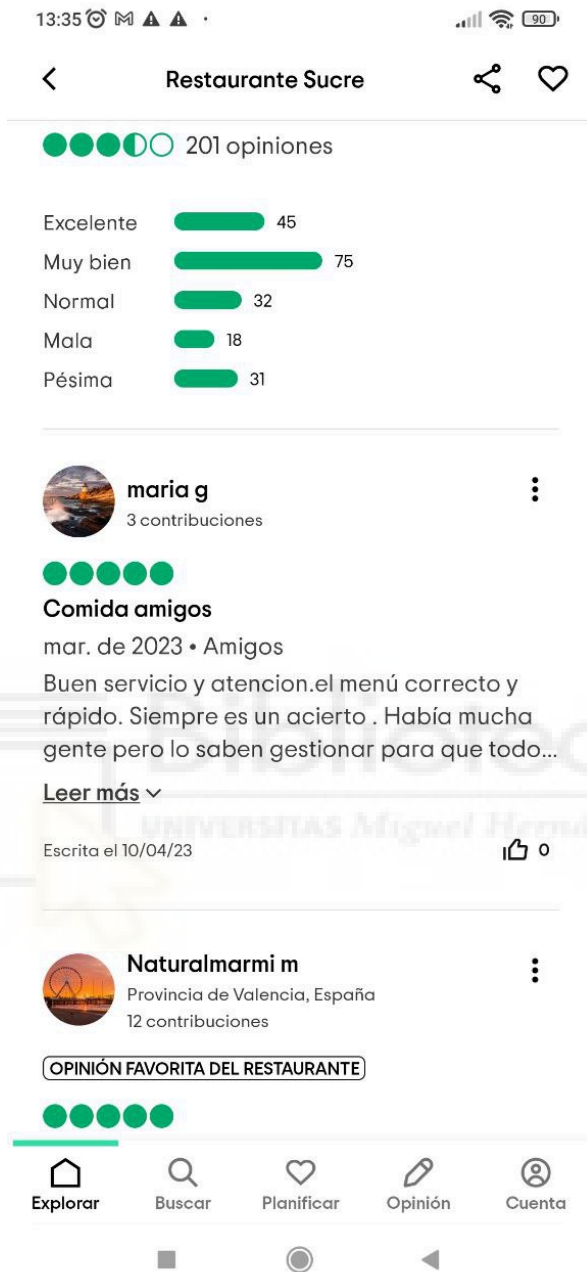


Figura 2.5. Trip Advisor comentarios

2.1.4. Yelp

Esta aplicación está disponible para Android, IOS y tiene disponible la versión de escritorio.

Yelp tiene muchas funcionalidades similares como las aplicaciones comentadas anteriormente. De esta aplicación podemos destacar que, podemos buscar servicios sin activar la ubicación pero en este caso debemos introducir un código postal para tener una referencia de los servicios que se pueden encontrar a partir de dicho código. Como podemos observar en las siguientes imágenes, esta aplicación también dispone de un buscador para poder obtener el tipo de servicio que necesitamos o también tenemos unas opciones de acceso rápido para buscar un tipo de servicio. En este caso buscamos un servicio relacionado con las bicicletas, como se puede apreciar en la *figura 2.6*, se encuentran listados los distintos servicios a partir de la búsqueda. Si seleccionamos un servicio podemos ver sus detalles.



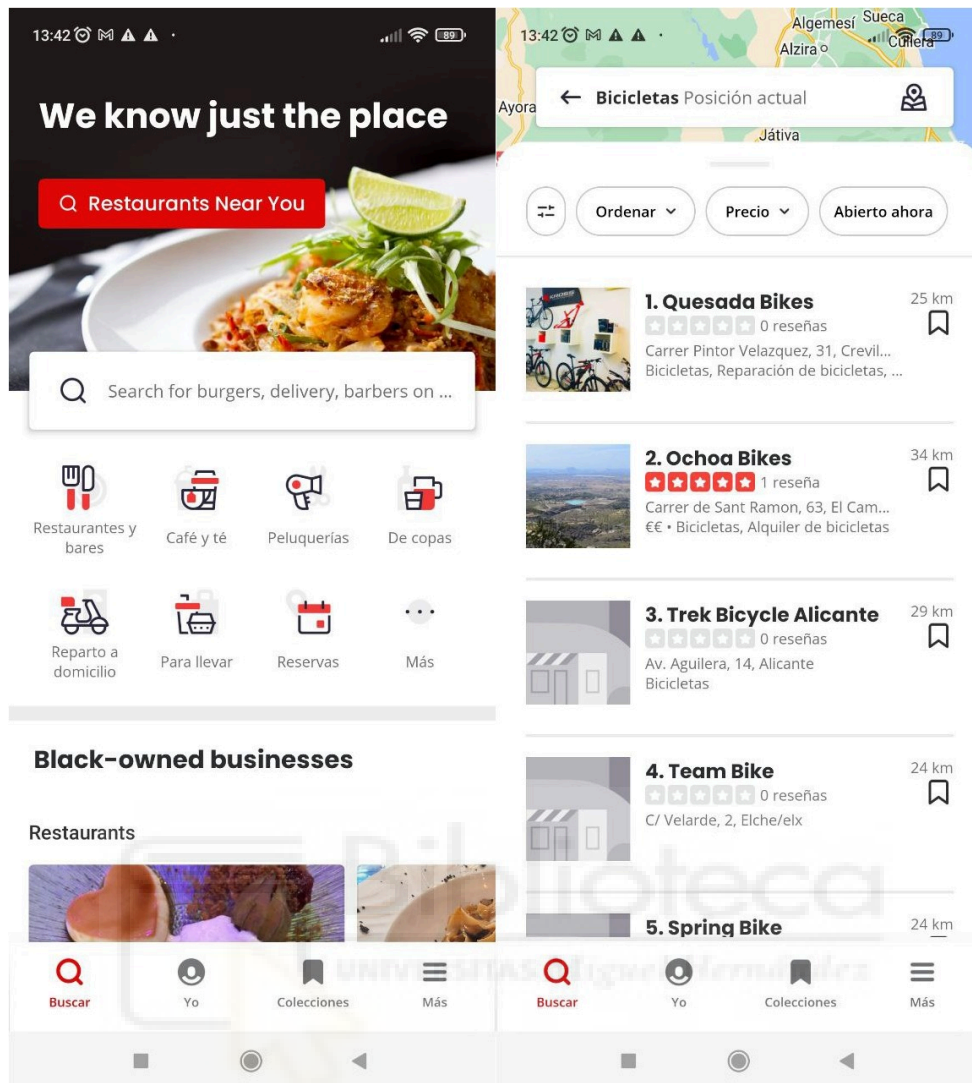


Figura 2.6. Yelp bicicletas

En la siguiente imagen vemos los detalles de un servicio, tenemos varias opciones pero nos vamos a fijar en las reseñas. En dicha reseña tenemos varias opciones que en las anteriores aplicaciones no teníamos, la opción de útil y divertida, características de esa aplicación. La aplicación tiene una interfaz intuitiva y su uso no es complicado, pero será necesario tener una cuenta para poder usar todas las funcionalidades, aunque si no tenemos ninguna cuenta también podemos ver solamente, los servicios y sus reseñas.

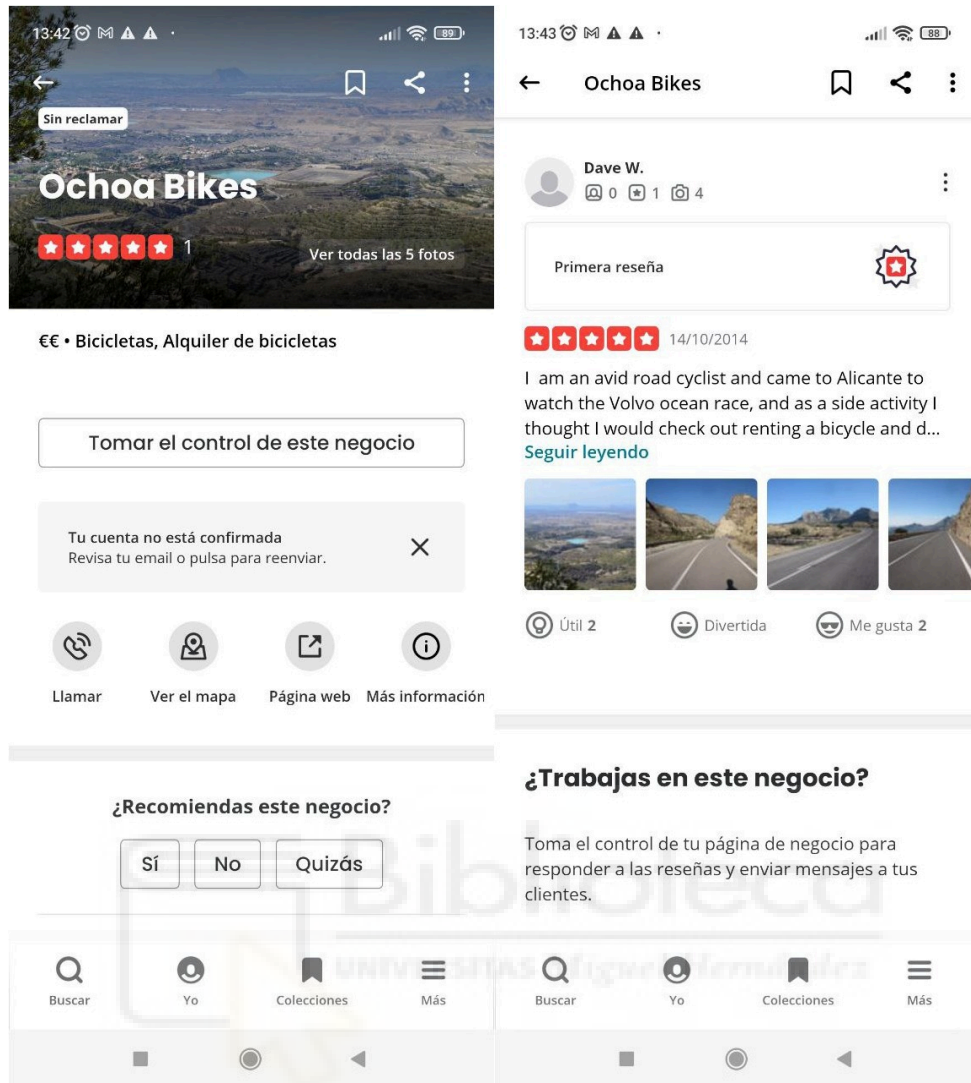


Figura 2.7. Yelp bicicletas y detalles

2.1.5. Resumen

En estas aplicaciones tendremos varias cosas en común pero la que más nos interesa son las reseñas de las personas que han podido contratar o acudir a un servicio. Con dichas reseñas tomaremos una decisión al momento de elegir un servicio de entre los varios que podemos encontrar. En las aplicaciones mencionadas anteriormente tenemos la misma pregunta, ¿son comentarios de fiar? nosotros tenemos la última decisión.

En la tabla que presentamos a continuación hacemos una pequeña comparación con las características más importantes que a mi parecer he visto en las aplicaciones, teniendo en cuenta la aplicación para Android que se está desarrollando a consecuencia del TFG.

Tabla 2.1. Características de aplicaciones

Nombre App	Google maps	FourSquare	Trip Advisor	Yelp
Plataforma	Android, IOS, y web	Android, IOS, y web	Android, IOS, y web	Android, IOS, y web
Funcionalidad mediante contactos telefónicos	No	No	No	No
Fiabilidad de los comentarios	Buena	Buena	Buena	Buena
Geolocalización obligatoria para su uso	No	Sí	No	No
Valoración en Google Play Store	3,9 de 5	4,1 de 5	4,4 de 5	3,9 de 5

La aplicación que se quiere desarrollar tiene muy en cuenta las tres filas centrales de la tabla. La integración de los contactos no necesariamente deberían tener las aplicaciones mencionadas, pero es una novedad en la aplicación Android desarrollada para el TFG. En este caso con las aplicaciones mencionadas anteriormente se han probado y en ninguna tenemos esa funcionalidad u opción (funcionalidad con los contactos del dispositivo móvil). La parte de contactos será fundamental en **VOS** porque tendremos en cuenta solamente dichas reseñas a partir de los contactos que se encuentren registrados en la aplicación.

En cuanto a fiabilidad de los comentarios, no se dispone de un mecanismo que compruebe la veracidad de un comentario dado en la aplicación **VOS**. Lo que será necesario para comprobar o verificar dicho comentario en un determinado servicio, si se desea, será contactar con el contacto que ha dejado su comentario.

La **geolocalización** en **VOS** no está implementada de momento porque se necesitaría rediseñar la base de datos ya que está diseñada para poder ser utilizada con una funcionalidad básica y no muy compleja.



Capítulo 3

Tecnologías

3.1. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO

Muchas de las herramientas que se ha utilizado en este proyecto han sido en consecuencia de las diferentes ventajas que tienen y por el conocimiento que ya se tenía sobre estas, además de que son bastante conocidas en el mercado laboral.

A continuación vamos a explicar cada una con sus respectivas características junto a sus ventajas y desventajas.

3.1.1. Editor de código

3.1.1.1. Visual studio code

Este editor de código es una de las herramientas más populares para el desarrollo de software, está desarrollado por Microsoft y está disponible para las siguientes plataformas: Windows, Linux y macOS.

Visual Studio Code, también conocido como *VS Code*, tiene muchas características que podemos destacar, nos puede ayudar a escribir código de manera rápida utilizando atajos con el teclado e instalando plugins, los cuales nos podemos descargar e instalar y sacar mucha ventaja al utilizarlos en el momento de codificar, también nos ayuda a depurar el código y a corregirlo haciendo uso del método intelli-sense.

De las principales características podemos destacar las siguientes:

- Gran variedad de temas para poder instalarlos fácilmente y ver los cambios al instante
- Soporte para múltiples lenguajes de programación (c++, c, javascript, etc.)
- Soporte multiplataforma (Windows, Linux y Mac)
- Amplia variedad de extensiones dependiendo de los lenguajes de programación utilizado para trabajar más rápido
- Se pueden abrir simultáneamente varios proyectos

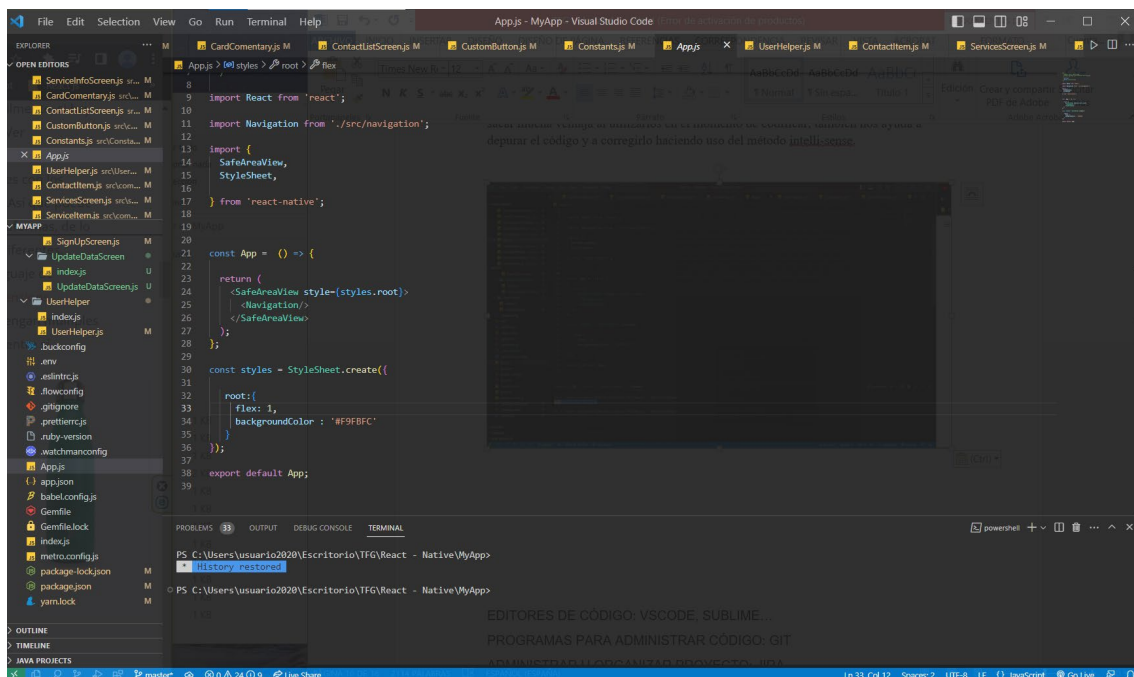


Figura 3.1. Visual studio code

Ventajas

- Gratuito y de código abierto
- Personalizable (instalando extensiones y temas)
- Compatibilidad con varios lenguajes de programación
- Comunidad amplia
- Disponible para Windows, macOS y Linux
- Auto-completa código mediante atajos de teclado o los llamados snippets
- Vinculación disponible a Git
- Gran variedad de extensiones y temas
- Terminal incorporada

Desventajas

- Lento en algunos proyectos grandes
- Interfaz de usuario algo confusa al llegar de otro IDE
- Algunos problemas de estabilidad

3.1.2. Administrador de código

3.1.2.1. Git

Git, es un gestor de código fuente y controlador de versiones de código enfocado en proyectos. Sirve de gran ayuda a los desarrolladores de programas, aplicaciones de todo tipo, etc. Git nos permite realizar varias versiones de un proyecto (por ejemplo, desarrollo de aplicación móvil) en el que podremos volver a la versión que queramos en cualquier momento, manteniendo los cambios en cada versión sin perjudicar al resto de versiones. Está disponible para las siguientes plataformas: Linux, Windows y macOS.

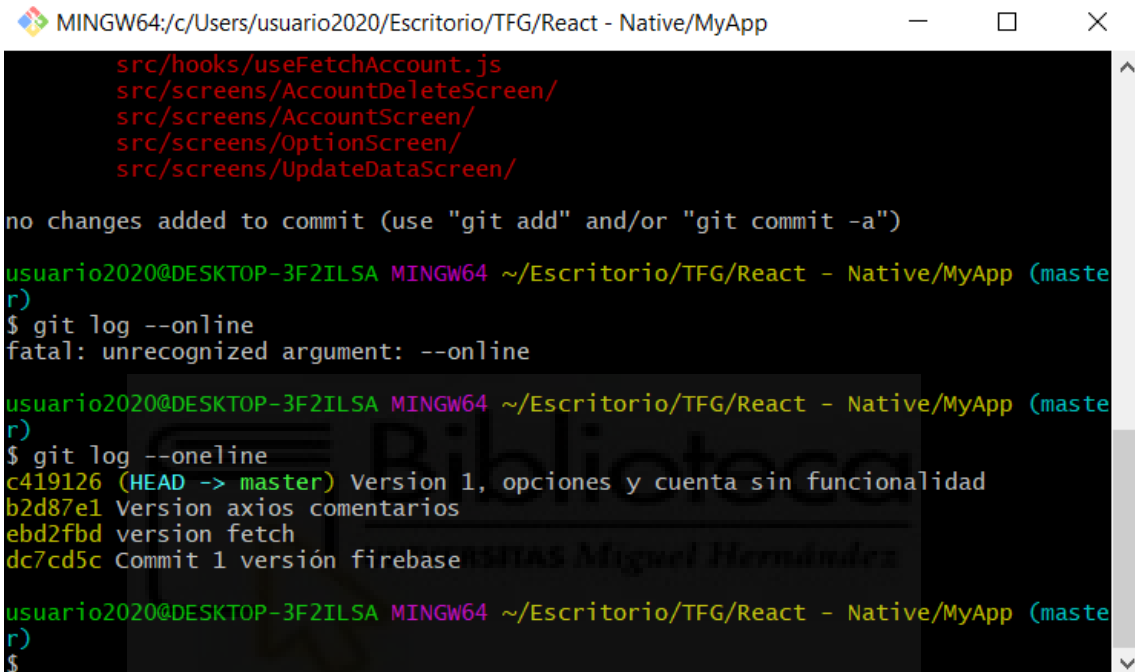
Inicialmente en el mantenimiento del núcleo de Linux, los cambios en la parte del software eran en forma de parches y archivos, esto se realizaba entre los años (1991-2002).

En el 2002, Linux empieza a utilizar un sistema de control de versiones distribuido llamado BitKeeper. Este software era gratuito.

En el año 2005, surge una mala relación entre la comunidad que se dedicaba a desarrollar el núcleo de Linux y la compañía desarrolladora de BitKeeper, a consecuencia de esto la herramienta BitKeeper dejó de ser gratuita. Es por esto último que la comunidad de

desarrollo de Linux, desarrolla su propia herramienta y surge GIT, teniendo en cuenta los siguientes objetivos:

- Velocidad
- Diseño Sencillo
- Fuerte apoyo al desarrollo
- Completamente distribuido
- Capacidad para manejar grandes proyectos de manera eficiente



```
src/hooks/useFetchAccount.js
src/screens/AccountDeleteScreen/
src/screens/AccountScreen/
src/screens/OptionScreen/
src/screens/UpdateDataScreen/

no changes added to commit (use "git add" and/or "git commit -a")

usuario2020@DESKTOP-3F2ILSA MINGW64 ~/Escritorio/TFG/React - Native/MyApp (master)
$ git log --online
fatal: unrecognized argument: --online

usuario2020@DESKTOP-3F2ILSA MINGW64 ~/Escritorio/TFG/React - Native/MyApp (master)
$ git log --online
c419126 (HEAD -> master) Version 1, opciones y cuenta sin funcionalidad
b2d87e1 Version axios comentarios
ebd2fbd version fetch
dc7cd5c Commit 1 versión firebase

usuario2020@DESKTOP-3F2ILSA MINGW64 ~/Escritorio/TFG/React - Native/MyApp (master)
$
```

Figura 3.2. Versiones en Git

Git es el gestor de código más conocido y popular, es por ello que muchos programadores y empresas lo utilizan hoy en día. A continuación vamos a listar las ventajas y desventajas que posee.

Ventajas

- Sistema distribuido
- Rápido y ligero
- Creación ramas y mezcla de estas
- La integridad de la información está asegurada
- Permite flujos de trabajo muy flexibles
- Es gratuito y de código abierto

Desventajas

- Complejidad para los sistemas centralizados tradicionales
- Curva de aprendizaje alta
- Comandos y conceptos confusos

3.1.3. Gestión de proyecto

3.1.3.1. Atlassian Jira

Jira, es un software que nos permite gestionar proyectos con el fin de mantener un orden y una organización en el desarrollo de software. Este software surge a consecuencia de los problemas que podían aparecer al realizar seguimientos en el proceso de desarrollo de un determinado software. Otra característica importante de esta herramienta es que nos va a permitir realizar seguimientos de *incidencias* en el desarrollo del software.

El objetivo de Jira es, mantener la productividad de los miembros que conformen un equipo de trabajo para el desarrollo de un software y de esta manera obtener buenos resultados en dicho desarrollo.

Esta plataforma tiene una versión basada en la nube, también dispone de un instalador para obtenerlo de manera local, se puede instalar para las siguientes plataformas: Windows, macOS y Linux.

Esta herramienta dispone de tableros que tienen características especiales para realizar un seguimiento en nuestro proyecto, los tableros que dispone son, **scrum** y **kanban**. Kanban está basado en tareas visuales que gestionan los flujos de trabajo. Scrum nos permite estructurar y gestionar el trabajo en el desarrollo de software mediante, valores, principios y prácticas [9].

Nos vamos a centrar en las características del tablero de Jira Scrum porque este se ha aplicado en el desarrollo del proyecto, ya que este está orientado a proyectos.

Jira Scrum tiene unos términos o características y de los que vamos a explicar, los más importantes que son:

- **Sprint:** Es el plazo en el que un equipo trabaja para completar una parte del trabajo que se ha establecido previamente.
- **Backlog:** Es una lista de trabajo que debe estar ordenado por prioridades.
- **Historias de usuario:** Es la descripción o explicación de una función de software perteneciente al trabajo que se tiene que realizar.
- **Incidencia:** Son tareas que se deben llevar a cabo en el trabajo.
- **Epic:** Se compone de historias de usuario que pueden contener historias de usuario más pequeñas.

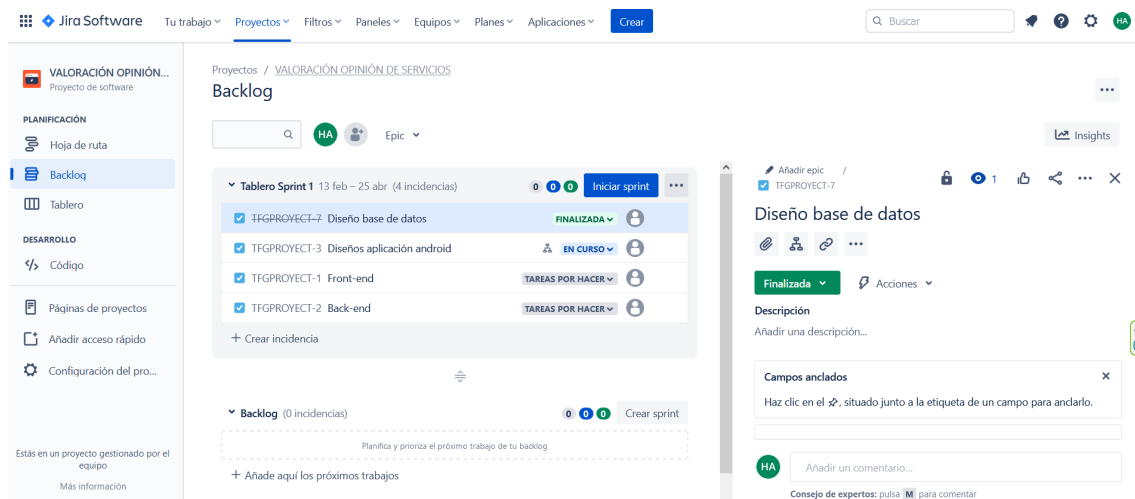


Figura 3.3. Jira Scrum

Scrum es conveniente usarla en proyectos que se requieren descomponer, estos están basados en prioridades de negocio, proyectos grandes con poca planificación, a esto podemos añadir que los proyectos se pueden descomponerse en pequeñas partes con la intención de que sean más manejables.

Ventajas

- Planificación flexible al momento de trabajar
- Elaboración de informes
- Interfaz sencilla, intuitiva y de fácil uso
- Versión gratuita de 30 días
- Recomendado para equipos de hasta 10 miembros
- Herramienta en línea y aplicación móvil disponible
- Planificación de proyectos con facilidad, con Scrum y Kanban

Desventajas

- Alto costo de la licencia
- Las distintas funcionalidades pueden resultar confusas para principiantes
- Costosa para empresas de poco presupuesto
- Al principio su configuración es complicada

3.1.4. Herramienta para diseño de base de datos

3.1.4.1. Lucidchart

Lucidchart es una aplicación de diagramas inteligente que permite diseñar diagramas de varios tipos: de clases, de componentes, de objetos, de casos de uso, etc. Nos permite colaborar con un equipo en tiempo real para la creación de diagramas de flujo, maquetas, diagramas UML entre otros, en resumen, la aplicación nos permite la creación de diagramas, visualización de datos y la colaboración entre varios miembros de un equipo.

Esta herramienta está disponible solamente como plataforma web.

Una de las características de esta herramienta es la integración que te permite realizar con otras herramientas, tenemos al menos quince plataformas en las que te permite realizar integraciones [12].

La curva de aprendizaje al principio puede ser complicada pero con el tiempo y la práctica esa dificultad desaparece porque las herramientas están bien definidas y explicadas para su uso. Desde el panel de usuario disponemos de los diagramas creados y su acceso rápido mediante un click.

Cuando estamos en un proyecto, disponemos de diagramas en la parte izquierda, esto es de gran ayuda porque los tenemos a “mano” y podemos seleccionarlos sin complicaciones. También se dispondrá de un guardado automático.

Con accesos rápido a sus funciones y su simplicidad con algunas herramientas, esta herramienta es bastante útil y “poderosa”.

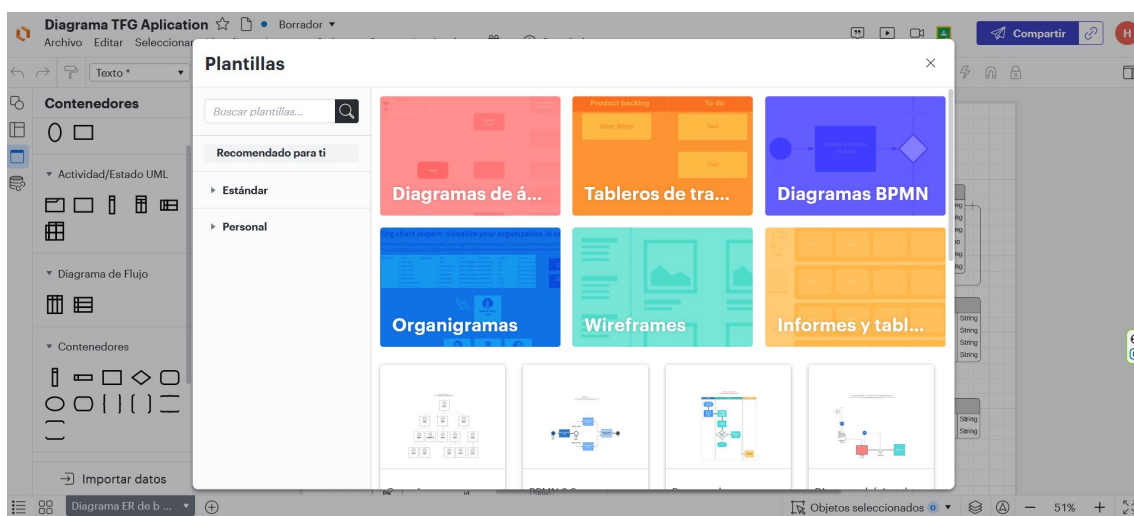


Figura 3.4. Lucidchart plantillas

Ventajas

- Fácil de usar gracias a su interfaz
- Variedad de plantillas y formas para poder trabajar con ellas
- Colaboración en tiempo real
- Integración con herramientas de Google (Drive, Docs, Slack) y Microsoft Office.
- Herramienta en línea
- Almacenamiento de proyectos en la nube

Desventajas

- Planes con precios para obtener el resto de funcionalidades
- Limitaciones en la versión gratuita
- Conexión mediante internet (herramienta basada en la nube)
- Curva de aprendizaje alta, dependiendo del tipo de herramientas a utilizar

3.1.5. Prototipo aplicación móvil

3.1.5.1. Figma

Figma es una herramienta que nos permite realizar diseños y prototipos (webs y aplicaciones), también podemos elegir plantillas para organizar ideas y organizar reuniones y tenemos la posibilidad de crear moodboards (representación visual de lo que se quiere transmitir) entre otros.

Esta herramienta está disponible en línea, como aplicación instalable de escritorio para Windows y macOS, también tenemos disponible la versión móvil de IOS y Android.

Si hacemos referencia al prototipo de páginas web y aplicaciones podemos aplicar varias funcionalidades de manera rápida y fácil, como dar colores, bordes, elegir una tipografía con sombras, el tamaño de la ventana etc. Todo esto de manera sencilla.

En los diseños web tenemos la opción de poder exportar un diseño de Figma a una web funcional con mediante el uso de una herramienta externa.

En el prototipado de las aplicaciones podemos crear maquetas de manera rápida y ver los resultados para poder tener una idea que lo que queremos para el futuro. Todo esto moviendo ventanas y agregando componentes de imagen, colores, formas etc.

En figma trabajaremos en base a un *frame* que suele ser de varios tipos: Android, IOS, Apple, Surface, web, etc. Estos frames nos permiten emular al dispositivo en el que vamos a mostrar el prototipo. En este caso si nos fijamos en la figura que se muestra a continuación, vemos la parte del menú como prototipo de la aplicación VOS en un Android.



Figura 3.5. Figma prototipo aplicación Android

Figma es bastante fácil de usar, tiene varias herramientas que están bien definidas y colocadas para que no resulte un problema en el momento de diseñar un prototipo. La imagen que vemos a continuación es parte del prototipo de la aplicación VOS y en ella podemos ver algunas herramientas en la parte superior izquierda que nos van a permitir trabajar en el diseño.

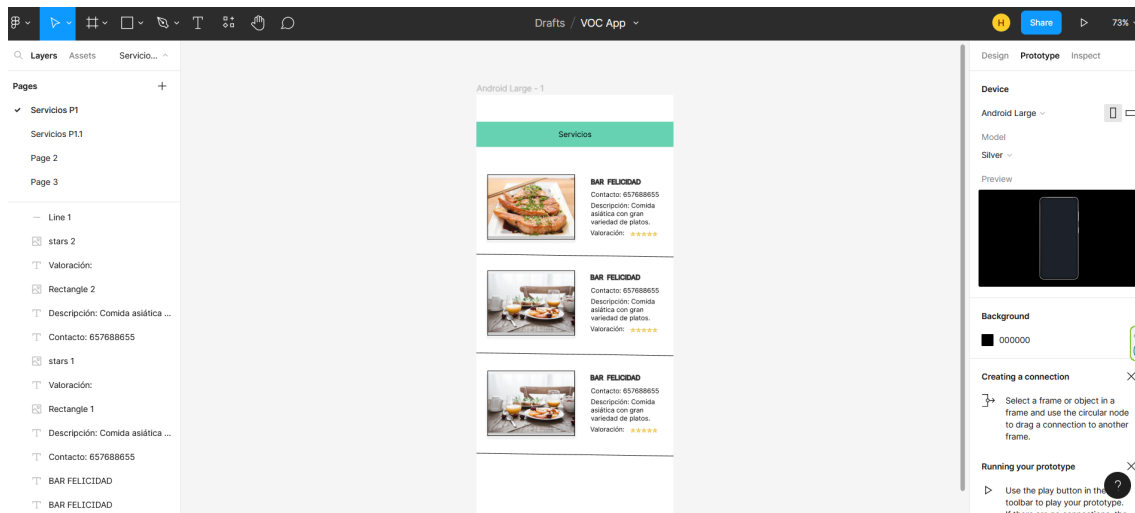


Figura 3.6. Lucidchart plantillas

Ventajas

- Gratuito
- Creación de documentos sin límite
- Fácil uso
- Trabajo en equipo
- Amplia variedad de plantillas de la comunidad
- Interacciones y animaciones
- Aplicación disponible para Windows y IOS
- Almacenamiento de diseños en la nube

Desventajas

- Software poco conocido
- Compra por parte de Adobe, dudas en el futuro acerca de su plan gratuito

3.1.6. Diseño gráfico

3.1.6.1. Gimp

Gimp es un software de código abierto que nos va a permitir el retoque, edición de imágenes y la creación de gráficos. Algunas empresas lo utilizan para diseñar logotipos o gráficos en ciertos proyectos, puesto que es un software que no tiene mucho que envidiar a otros programas que hacen uso de licencias comerciales. Gimp tiene su propio

formato de almacenamiento y soporta ficheros gráficos de varios tipos, jpg, gif, png, tiff etc.

Esta herramienta está disponible para las siguientes plataformas: Windows, macOS y Linux.

Es una herramienta que puede ser descargada y posteriormente poder instalarla en un ordenador y poder seguir una variedad de tutoriales de todo tipo que están disponibles en la página oficial de Gimp.



Figura 3.7. Gimp

Ventajas

- Edición de imágenes avanzada, gratuita
- Posibilidad de ejecución en la nube, disco externo o en local
- Compatibilidad con ficheros de Photoshop
- Enmascaramiento de imágenes (eliminar fondos de manera sencilla)
- Menú inteligente y de fácil acceso
- Interfaz intuitiva
- Es más rápido en comparación con Photoshop (al trabajar con fotos pequeñas)

Desventajas

- Menos herramientas que Photoshop
- Dificultad al crear formas atractivas
- No es buena opción al trabajar con varias capas
- Compatible solamente con imágenes RGB de 8 bits
- Extensiones externas limitadas en comparación con Photoshop

3.2. VALORACIÓN Y COMENTARIO

Las herramientas mencionadas anteriormente han sido elegidas por las ventajas que tienen y por el conocimiento de algunas, además de la facilidad de uso (en mi caso).

También tenemos que tener en cuenta que el desarrollo del proyecto es con fines educativos y no comerciales, es por ello que se han elegido estas al ser gratuitas. A continuación se pasará a justificar la elección de cada una de ellas.

Herramienta **VS Code**, se ha optado por esta herramienta porque es una herramienta con bastantes ventajas y es por ello que actualmente es una de las más populares del mundo. Su facilidad de uso, herramientas de ayuda, interfaz, etc. Se tenía conocimiento sobre esta herramienta y se ha utilizado en la mayoría del desarrollo del proyecto. Utilizando los *snippets* (inserción rápida de una parte de código) que están disponibles de varias fuentes por parte de la comunidad se ha logrado agilizar el desarrollo y de esta manera trabajar con mayor seguridad al momento de comprobar el código cuando se compilaba en el emulador.

Con VS Code se ha desarrollado la parte front-end de la aplicación Android utilizando el framework react-native y también se ha utilizado para el desarrollo de la parte back-end del proyecto en general.

Con **Git** ya se tenía conocimiento y es de gran ayuda en el momento de optar por almacenar versiones del proyecto, en mi caso fue utilizada en gran mayoría en la aplicación desarrollada con **React Native** para guardar algunas versiones que se iban desarrollando a lo largo del TFG, por mencionar “algo”, se guardó dos versiones utilizando dos librerías distintas para consumir Apis.

En la parte de **Jira** con la metodología *Scrum*, se ha optado por elegir esta herramienta por haberla visto en una asignatura de la carrera y por las ventajas que tiene en el momento el que se mantiene un desarrollo de software. En este caso se ha iniciado el proyecto mediante la creación de incidencias para mantener una organización del proyecto e ir avanzado y priorizando tareas y seguir un cierto orden. En el proyecto se ha optado por empezar con el diseño del esquema conceptual de una base de datos, posteriormente con el diseño (prototipo, diseño logo) de la aplicación Android, después pasar con la parte del frontend de la aplicación Android y finalizando con la parte backend de la misma. En la última parte se ha procedido con la parte frontend y backend de la aplicación web.

Con la herramienta **Lucidchart** se tenía ya conocimiento de esta. Es una herramienta fácil de usar, se pueden crear esquemas a partir de plantillas que tenemos disponibles de manera gratuita. Su simplicidad y claridad, hicieron que esta herramienta fuera la candidata para el diseño del esquema conceptual de la base de datos del proyecto.

La herramienta **Figma**, es una herramienta recomendada por webs dedicadas al prototipado de aplicaciones, esta es sencilla y fácil de utilizarla. Con dicha herramienta se diseñaron las ventanas del menú principal, así como la de los servicios disponibles etc.

Mediante la herramienta **Gimp** se ha diseñado el logotipo de la aplicación Android, a partir de imágenes descargadas de una plataforma gratuita de iconos e imágenes, siendo estas libres de derechos del autor.

3.3. FRONTEND y BACKEND

En esta parte pasaremos a explicar las tecnologías correspondientes en la parte de cliente (frontend) y servidor (backend), estas tecnologías son las destinadas al desarrollo de la aplicación Android y la aplicación web. En este caso se han utilizado algunas tecnologías que ya se habían visto en la carrera. Las secciones se han dividido en la parte del cliente (frontend) y en las del servidor (backend). La elección de estas tecnologías se debe a las grandes ventajas que tienen y se comentarán en su respectiva sección.

La parte **frontend** está relacionada con la aplicación Android desarrollada para este proyecto y en el que en dicha aplicación se ha utilizado **React Native** para el desarrollo de esta. También cabe mencionar que la aplicación web de escritorio está realizada con **html** y la librería **bootstrap** haciendo uso del framework **Laravel** para crear automáticamente una parte de la aplicación web. En la cual mediante comandos se crean de forma automática las vistas (views) y por otro lado para agregarle estilos hemos hecho uso de la librería bootstrap en el que previamente mediante un comando se ha instalado dicha librería.

En la parte del **backend** se ha hecho uso de algunas de las opciones que nos permite realizar **Laravel** para generar controladores, modelos, tablas, etc. Todo esto teniendo en la cuenta la parte perteneciente a React Native y la parte de la aplicación web, cada parte con sus respectivos controladores y más opciones relacionadas con el backend que pasarán a explicarse más en detalle en el siguiente capítulo (Capítulo 4 – 4.4. Implementación).

A continuación pasaremos a nombrar cada tecnología haciendo referencia al cliente (frontend) o servidor (backend), en cada una se explicará sus características más relevantes junto a sus ventajas y desventajas.

3.3.1. Frontend

3.3.1.1. Html

Es un lenguaje basado en un conjunto de etiquetas en el que un navegador se encarga de interpretarlo para que se pueda mostrar al usuario de forma entendible. Los navegadores encargados de interpretar html son algunos de los siguientes: Safari, Chrome, Firefox, Opera, etc.

Con html podemos estructurar una página web para que esta sea capaz de mostrar su contenido de forma organizada. Tenemos que tener en cuenta que html no es un lenguaje de programación, con html podremos crear webs estáticas, pero si combinamos otros lenguajes de programación podemos crear páginas web dinámicas.

```
<!DOCTYPE html>
<html>
<title>W3.CSS</title>
<meta name="viewport" content="width=device-width,
initial-scale=1">
<link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css">
<body>

<div class="w3-container w3-teal">
  <h1>Header</h1>
</div>

<div class="w3-container">
  <p>The w3-container class can be used to display
headers.</p>
</div>

</body>
</html>
```

Figura 3.8. Estructura html

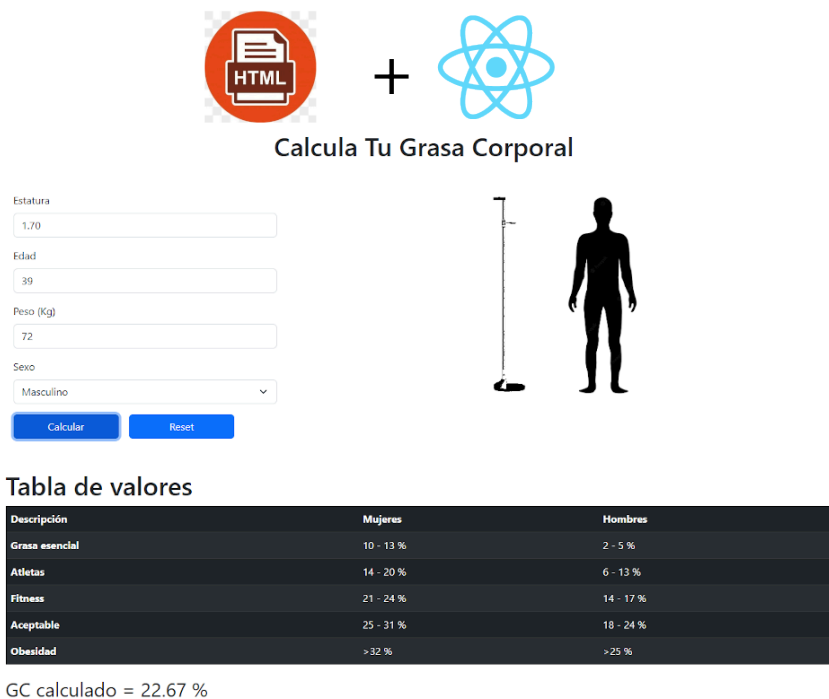


Figura 3.9. Página dinámica con html y react js

Como se ha mencionado anteriormente las etiquetas sirven para escribir código y para que este pueda ser interpretado por un navegador. Hay varios tipos de etiquetas, solamente nombraremos algunas.

- Etiquetas de título (<title>), sirve para mostrar el título de la página web.
- Etiqueta de cuerpo (<body>), nos permite indicar el desarrollo del texto o parte del cuerpo de una página web.
- Etiqueta de sección inferior conocida como footer (<footer>), esta sirve para indicar la sección final o inferior de una página web.
- Etiqueta de sección superior (<head>), indica la parte superior del texto
- Etiqueta de encabezado o título (<h1>), de esta podemos tener, h2, h3, hasta h6, dependiendo del nivel en el que nos encontremos.
- Etiqueta de párrafo (<p>), sirve para agrupar un texto.
- Etiqueta de imagen (), nos permite mostrar una imagen.

En cuanto a versiones, podemos decir que html cuenta con algunas versiones como se muestra en la siguiente tabla.

Tabla 3.1. Versiones de html

Versión	Año
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
HTML 5	2012

Html es muy utilizado para el desarrollo de páginas web y por eso sigue siendo muy utilizado por los programadores.

A continuación podemos destacar algunas de las ventajas y desventajas que posee este lenguaje.

Ventajas

- Descripción de hiper texto
- Despliegue rápido
- Compatibilidad con la mayoría de navegadores
- Permite cargar archivos pequeños

Desventajas

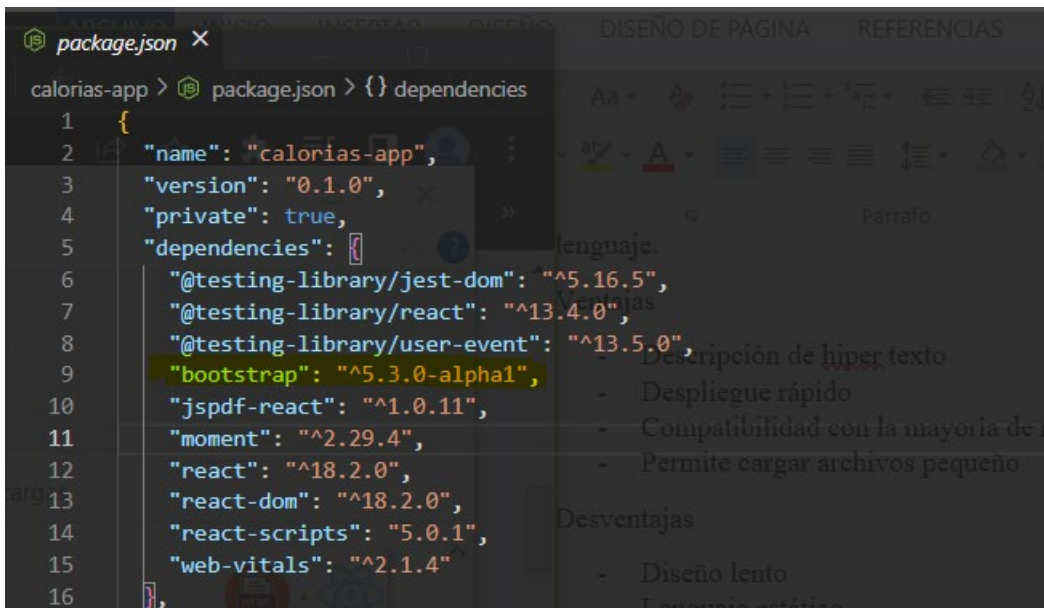
- Diseño lento
- Lenguaje estático
- Etiquetas limitadas

3.3.1.2. Bootstrap

Este *framework* css desarrollado por Twitter en 2010, es utilizado por aplicaciones frontend para dar estilos a estas aplicaciones además de usar componentes ya definidos en la web oficial que se pueden copiar y pegar en nuestro proyecto. Una característica a destacar muy importante es que, con este *framework* podemos hacer que nuestras aplicaciones sean adaptables a cualquier dispositivo puesto que usa unos estilos definidos para cumplir ese objetivo.

Bootstrap está compuesto por archivos CSS y JavaScript. Si queremos instalarlo en un proyecto en el que haga uso de html, podemos acudir la web oficial de bootstrap y seguir los pasos de instalación, todo esto de una manera fácil. Nos permite instalarlo en

aplicaciones que usen otras tecnologías, por ejemplo las que hagan uso de *Node Js*, *Ruby*, *Composer* y *NuGet*.



```
calorias-app > package.json > {} dependencies
1  {
2    "name": "calorias-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.16.5",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "bootstrap": "^5.3.0-alpha1",
10     "jspdf-react": "^1.0.11",
11     "moment": "^2.29.4",
12     "react": "^18.2.0",
13     "react-dom": "^18.2.0",
14     "react-scripts": "5.0.1",
15     "web-vitals": "^2.1.4"
16  },
```

Figura 3.10. Dependencia agregada en React JS

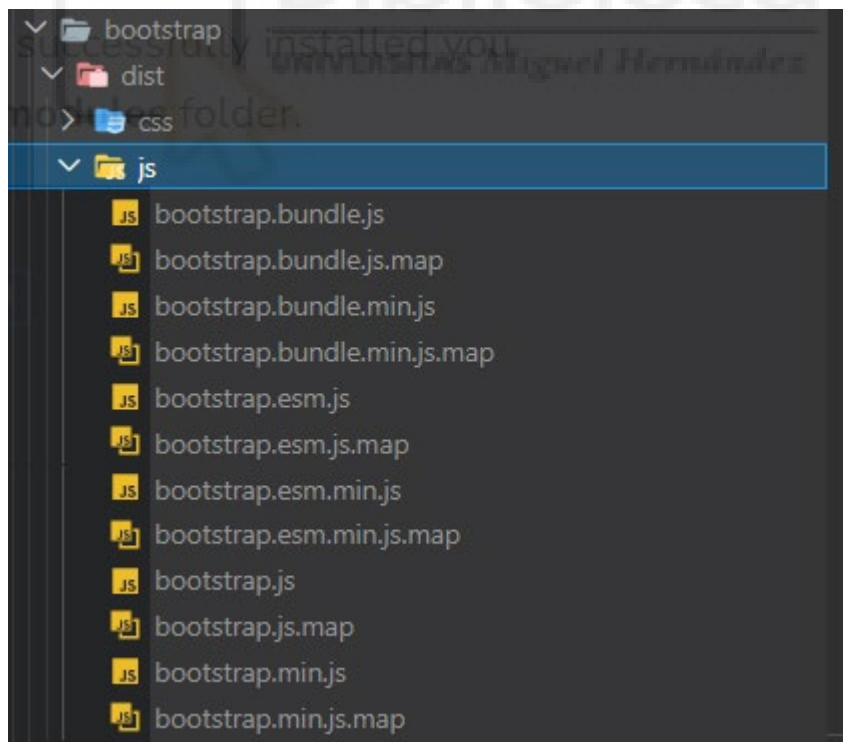


Figura 3.11. Estructura de archivos de bootstrap en React JS

```
import React from 'react'

const Formulario = ({onChange, estatura, edad, peso, sexo, onSubmit, resetValues, errorstate}) => (

  <div className='col-sm-12 col-md-4 mb-5' >
    <form onSubmit={onSubmit} >
      <div className="mb-3">
        <label htmlFor="inputestatura" className="form-label">Estatura</label>
        <input type="number"
          className="form-control input-sm"
          id="inputestatura"
          aria-describedby="estaturaHelp"
          onChange={e => onChange(e.target.value, 'estatura')}
          value={estatura}
        />
        <span className='text-danger'>{errorstate.estatura===''? null : errorstate.estatura }</span>
      </div>
      <div className="mb-3">
        <label htmlFor="inputedad" className="form-label">Edad</label>
        <input type="text"
          className="form-control"
          id="inputedad"
          aria-describedby="edadHelp"
          onChange={e => onChange(e.target.value, 'edad')}
          value={edad}
        />
        <span className='text-danger'>{errorstate.edad===''? null : errorstate.edad }</span>
      </div>
      <div className="mb-3">
        <label htmlFor="inputpeso" className="form-label">Peso (Kg)</label>
        <input type="number"
          className="form-control"
          id="inputpeso"

```

Figura 3.12. Clases de bootstrap en React JS

En la página oficial de bootstrap podemos encontrar una serie de componentes preparados para ser implementados en nuestro proyecto, teniendo en cuenta que ya tenemos instalado este *framework* en nuestro proyecto. Este caso de los distintos componentes que nos podemos encontrar tenemos, botones, cards, mensajes de alerta personalizados, barras de navegación, etc.

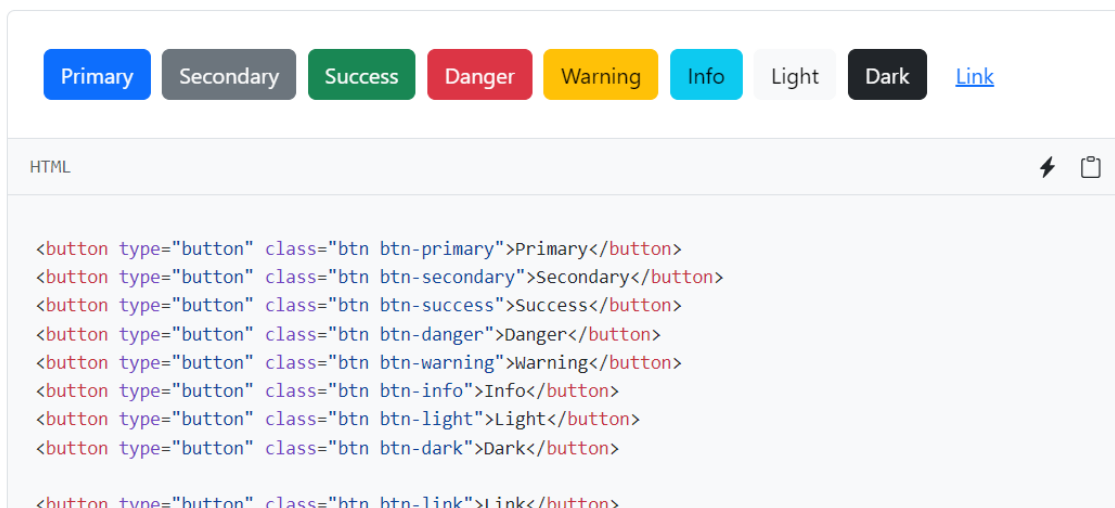


Figura 3.13. Botones de bootstrap

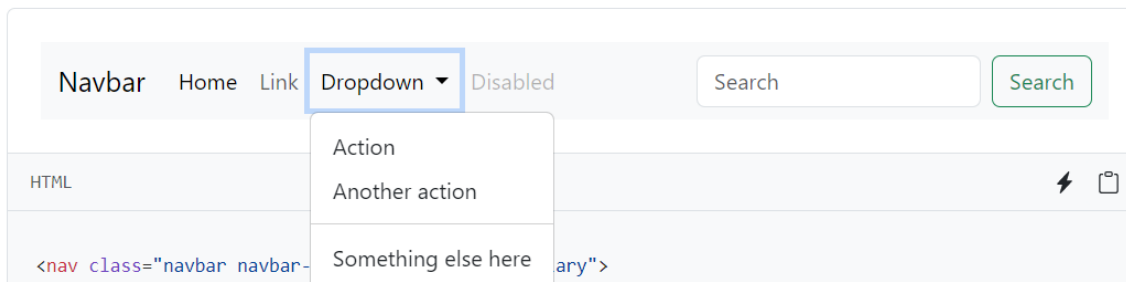


Figura 3.14. Barra de navegación de bootstrap

Sobre las versiones podemos observar que actualmente en la web oficial disponemos de la última versión, la 5.3, en la siguiente captura podemos ver las versiones de bootstrap hasta el día de hoy.



Figura 3.15. Todas las versiones de bootstrap

Finalmente podemos comentar algunas ventajas y desventajas que posee este framework.

Ventajas

- Un proyecto puede ser combinado con otros estilos css y javascript
- Incluye un sistema grid para maquetar por columnas
- Plantillas de sencilla adaptación responsive
- En la web oficial, disponemos de documentación completa, con ejemplos
- El desarrollo del diseño de interfaces es fácil

Desventajas

- Es recomendable trabajar en un proyecto desde un inicio con bootstrap, puesto que ya tenemos un proyecto y otros estilos definidos, estos podrían romperse
- No es ligero, y será necesario el uso de javascript y jquery
- Tenemos que adaptar nuestro diseño a un grid de 12 columnas
- Este framework trae por defecto anchos, altos, márgenes, etc. Muchas veces no serán los que necesitemos y deberíamos usar nuestros estilos.

3.3.1.3. React native

React native es un *framework* basado en JavaScript y ReactJS, sirve para crear aplicaciones para las plataformas IOS, Android, Web y UWP (Windows). Este *framework* fue creado por Facebook en Febrero de 2015.

Las aplicaciones realizadas con esta tecnología y las aplicaciones nativas de *Android* e *IOS*, usan mismo el paradigma fundamental de construcción de bloques de UI, esto quiere decir que, utiliza componentes visuales que permiten la interacción con el usuario.

De las características podemos mencionar algunas como las más importantes. Las aplicaciones funcionan de la misma manera que las realizadas de forma nativa, el punto a destacar es que **React Native** nos permite realizar aplicaciones que obtienen un mejor rendimiento y a su vez el desarrollo es más amigable. Otro punto a destacar es su curva de aprendizaje, pues esta es sencilla, en la web oficial disponemos de recursos y ejemplos para poder probar las aplicaciones realizadas, además disponemos de un emulador online que nos permite ejecutar las aplicaciones con su respectivo código de manera online. También cabe mencionar que este *framework* es mantenido por Facebook y la comunidad de software libre.

Para describir la ejecución de las aplicaciones en React Native podemos nombrar tres características principales que son las siguientes:

- **Código nativo:** En un proyecto podemos tener una parte con código nativo y otra parte con código de JavaScript.
- **JavaScriptCore VW:** Si estamos desarrollando una aplicación, el código que está en JavaScript, no será compilado, este será ejecutado como JavaScript y no compilado, puesto que existe una máquina virtual internamente que se encarga de ejecutar el código de JavaScript.
- **React Native Bridge:** Este se va a encargar de comunicar el código de la máquina virtual (comentada en el punto anterior) con el código nativo y las APIs nativas de la plataforma.

REACT NATIVE "Virtual DOM"

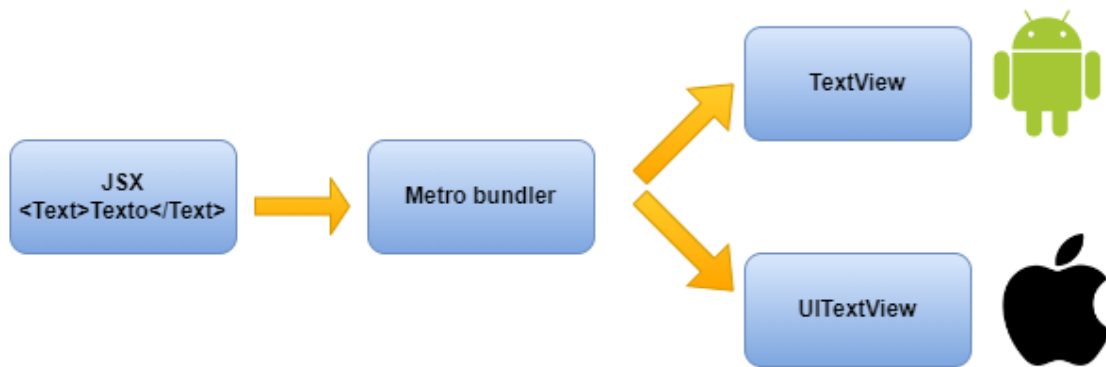


Figura 3.16. Esquema de React

Una de las capacidades que tiene al programar, es poder ver los resultados en tiempo real (recarga activa) cuando se esté ejecutando una aplicación, para aquello, debemos tener instalado un emulador Android o bien conectar nuestro dispositivo móvil, estos nos servirán de interfaz de usuario, de tal manera si realizamos alguna modificación en el código de algún proyecto, este cambio se verá reflejado en el emulador o en el dispositivo que en el que estuviésemos ejecutando la aplicación.

Otra parte que podemos destacar es que se pueden integrar funciones nativas, como por ejemplo para poder acceder al GPS o a la cámara del dispositivo.

Una de las mejores características que se tiene en el momento de trabajar con este *framework* es la reutilización del código mediante el uso de componentes, puesto nos ayudará a ahorrar tiempo en el desarrollo de una aplicación.

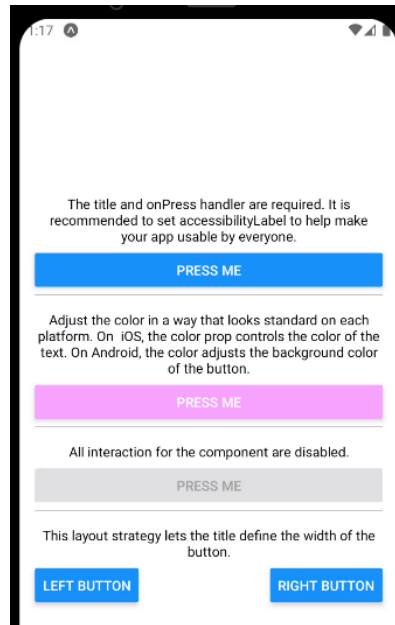


Figura 3.17. Componentes en react

Ventajas

- Se puede desarrollar aplicaciones para diferentes plataformas, utilizando el mismo código, esto hace que estas sean fáciles de mantener
- Rapidez en el desarrollo de proyectos al reutilizar componentes en otras partes del proyecto
- Utilización de JavaScript
- Código abierto
- Con la recarga activa, al realizar un cambio en el proyecto podemos ver los resultados en tiempo real
- Gran marco de actividad de los desarrolladores, esto hace que este framework sea el más popular a día de hoy
- Excelente rendimiento de las aplicaciones aprovechando el potencial de React Native
- Gran cantidad de recursos en internet
- Diseño modular

Desventajas

- Complejidad en la depuración
- Necesidad de desarrolladores nativos al implementar ciertas funciones de aplicación
- Dependencia de Facebook
- La gestión de memoria, no es del todo buena
- Un desafío al intentar buscar un muy buen rendimiento en una aplicación, al agregar funciones complejas
- Algunos componentes de terceros, son aceptables pero no lo que necesitamos en nuestra aplicación

3.3.2. Backend

3.3.2.1. Laravel

Laravel es un *framework* de PHP que utiliza el modelo MVC (modelo-visa-controlador) y nos sirve para desarrollar aplicaciones web. Este *framework* nos proporciona un entorno de trabajo que nos permite desarrollar aplicaciones en PHP. Una característica de Laravel es que podemos reutilizar código para poder desarrollar más rápidamente una aplicación y de esta manera evitar copiar el mismo código en distintas partes de la aplicación.

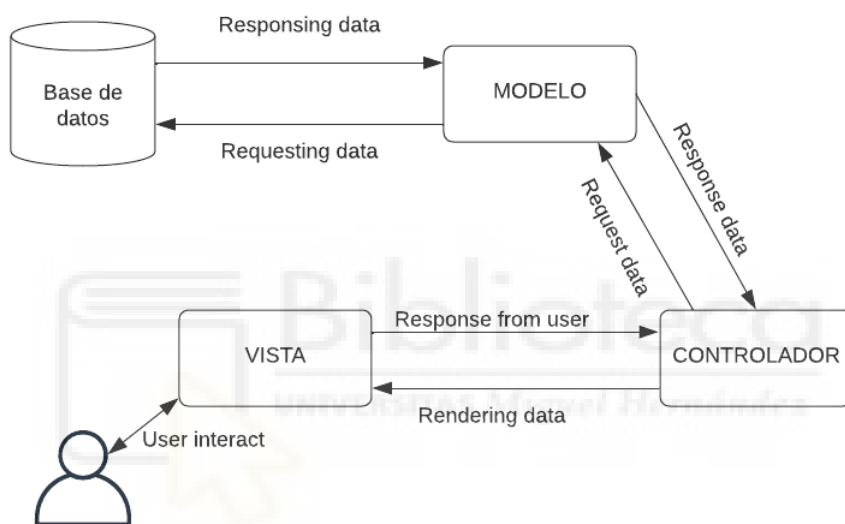


Figura 3.18. Modelo MVC

Laravel hace uso de un sistema de paquetes modulares, de esta manera nos permite desarrollar aplicaciones, que serían sencillas de actualizar y teniendo la posibilidad de agregar nuevas funcionalidades.

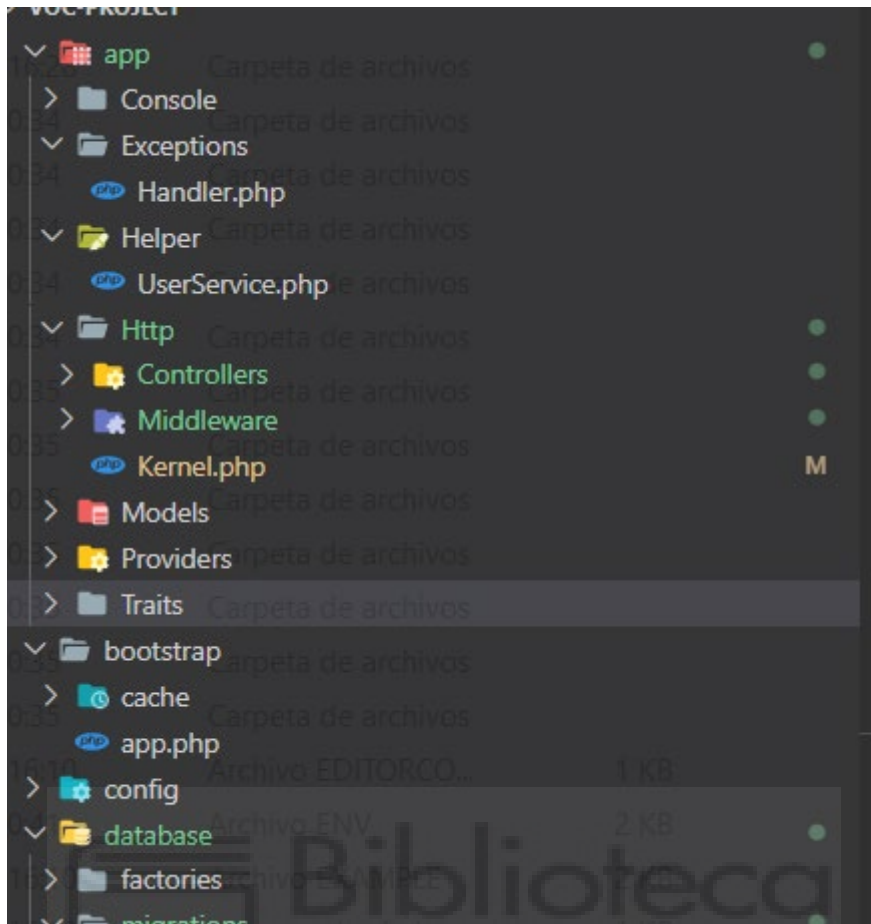


Figura 3.19. Laravel

La última versión disponible es Laravel 10, en la cual mencionamos las siguientes novedades:

- **Laravel Sanctum:** Sistema de autenticación, basada en tokens de autenticación.
- **Laravel Livewire:** Esta característica nos permite que la web no se recargue cuando se esté realizando una validación de formularios, cargar ficheros, filtración de datos, etc.
- **Laravel Process:** Gestión de procesos CLI de una manera más sencilla
- Soporte para PHP 8.2

Por último vamos a mencionar algunas de las ventajas y desventajas de este *framework*.

Ventajas

- La curva de aprendizaje es sencilla y su uso también
- Gran extensión de documentación gracias a su numerosa comunidad
- Gran cantidad de librerías y paquetes por parte de la comunidad y de las oficiales
- Seguridad alta frente a los malware o inyecciones sql

Desventajas

- Funciona con versiones superiores de php 5.4
- Las aplicaciones desarrolladas con este framework no destacan por la velocidad de navegación
- Procedencia del framework de Symfony, hereda algunas cosas que son difíciles de aprender

3.3.2.2. Gestor de base de datos Mysql

Mysql es un sistema que se encarga de la gestión de base de datos de tipo relacional, está basado en una licencia de código abierto y es propiedad de *Oracle Corporation* desde el 2010. También posee una licencia de uso comercial la cual está administrada por la misma compañía Oracle. Mysql está considerada como uno de los gestores de base de datos más populares del mundo pues tiene muchas ventajas que son muy interesantes para los desarrolladores.

Como mencionamos al principio, Mysql trabaja con bases de datos relacionales, es decir, si tenemos una base de datos, dentro de esta podemos interconectar las distintas tablas que conforman dicha base de datos para poner almacenar información u obtenerla dependiendo de las necesidades llevadas a cabo. En este caso para ser más claros las operaciones llevadas a cabo en una base de datos que se pueden realizar son las de crear, leer, actualizar y borrar, esto es el llamado CRUD cuyo significado es, create, read, update, y delete.



		id	nombre	uri	id_servicio	created_at	updated_at	deleted_at
<input type="checkbox"/>	Editar Copiar Borrar	1	bar-interior.jpg.jpg	bar-interior.jpg1676285985.jpg	1	2023-02-13 10:59:45	2023-02-13 10:59:45	NULL
<input type="checkbox"/>	Editar Copiar Borrar	2	cafe-bar.jpg.jpg	cafe-bar.jpg1676285985.jpg	1	2023-02-13 10:59:45	2023-02-13 10:59:45	NULL
<input type="checkbox"/>	Editar Copiar Borrar	3	comida.jpg.jpg	comida.jpg1676285985.jpg	1	2023-02-13 10:59:45	2023-02-13 10:59:45	NULL
<input type="checkbox"/>	Editar Copiar Borrar	4	mexican.jpg.jpg	mexican.jpg1676402393.jpg	2	2023-02-14 19:19:53	2023-02-14 19:19:53	NULL
<input type="checkbox"/>	Editar Copiar Borrar	5	tacos-mini.jpg.jpg	tacos-mini.jpg1676402393.jpg	2	2023-02-14 19:19:53	2023-02-14 19:19:53	NULL
<input type="checkbox"/>	Editar Copiar Borrar	6	tacos-pastor.jpg.jpg	tacos-pastor.jpg1676402393.jpg	2	2023-02-14 19:19:53	2023-02-14 19:19:53	NULL
<input type="checkbox"/>	Editar Copiar Borrar	7	helado-1.jpg.jpg	helado-1.jpg1677602276.jpg	3	2023-02-28 16:37:56	2023-02-28 16:37:56	NULL
<input type="checkbox"/>	Editar Copiar Borrar	8	helado-2.jpg.jpg	helado-2.jpg1677602276.jpg	3	2023-02-28 16:37:57	2023-02-28 16:37:57	NULL
<input type="checkbox"/>	Editar Copiar Borrar	9	electricity1.jpg.jpg	electricity1.jpg1678440923.jpg	4	2023-03-10 09:35:23	2023-03-10 09:35:23	NULL
<input type="checkbox"/>	Editar Copiar Borrar	10	electricity2.jpg.jpg	electricity2.jpg1679330445.jpg	4	2023-03-20 16:40:45	2023-03-20 16:40:45	NULL

Figura 3.20. Tabla imágenes con sus atributos

	id	id_tiposervicio	telefono_usuario	nombre	descripcion	valoracion_media	created_at	updated_at	deleted_at
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1		1 656656656	Bar San José	Bar de tapas.	4	2023-02-13 10:59:45	2023-02-13 10:59:45	NULL
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2		2 656656656	Mexican Food	Restaurante mexicano.	3.875	2023-02-14 19:19:53	2023-02-14 19:19:53	NULL
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3		1 656656656	Manuel	Heladería.	4	2023-02-28 16:37:56	2023-02-28 16:37:56	NULL
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	4		8 657657657	Electric	Electricista a domicilio.	5	2023-03-10 09:35:23	2023-03-10 09:38:05	NULL

Figura 3.21. Tabla servicios con sus atributos

Mysql se basa en el modelo cliente y servidor y su relación será distinguida de tal manera que un cliente haciendo uso de un sistema de registro podrá, obtener, modificar o almacenar datos.

Como Mysql es compatible con el lenguaje de consulta **sql**, podemos migrar sin problema datos desde otro motor de bases de datos mediante el uso de alguna herramienta que nos permita realizar dicha migración.

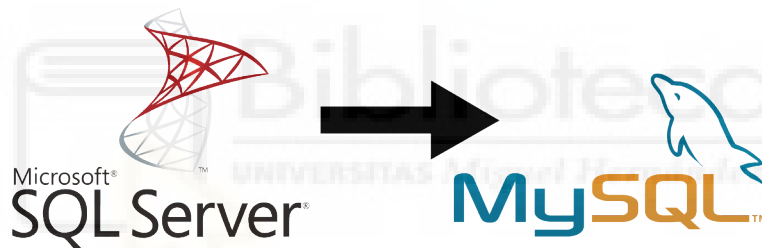


Figura 3.22. SQL Server Y Mysql

A continuación pasamos a nombrar algunas de sus ventajas y desventajas.

Ventajas

- Distribución libre y gratuita
- Open Source
- Destaca en velocidad en la realización de operaciones
- Multiplataforma
- Fácil instalación y configuración
- Posee funciones de encriptación
- El considerado uno de los gestores más rápidos del mercado

Desventajas

- La seguridad tiene unos límites dependiendo como se haya configurado

- La intuitividad no es su punto fuerte si lo comparamos con otros gestores
- No es tan eficiente cuando se trabaja con bases de datos de tamaño muy grande

3.3.2.3. Servidor web apache

El servidor web Apache es un servidor de código abierto, actualmente es uno de los que más que utilizan a nivel mundial. Fue desarrollado por Apache Software Foundation.

El nombre Apache hace referencia una tribu nativa de los Estados Unidos, los cuales destacaban por su resistencia, combate y su estrategia en la guerra. Apache mantiene una actualización constante además de su modularidad, es por ello que mantiene una popularidad entre los desarrolladores.

La función principal que realiza Apache es poder dar servicio las webs que se encuentran alojadas en dicho servidor con la intención de que la comunicación del cliente y el servidor sean de manera fluida.

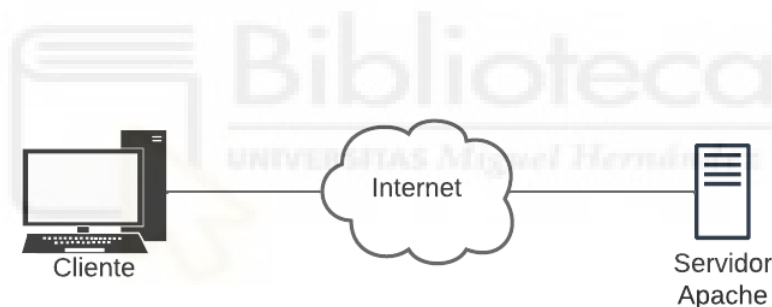


Figura 3.23. Cliente Servidor

En Apache podemos activar o desactivar algunas funcionalidades a través de sus módulos [28], pero tenemos que tener en cuenta que dependiendo del tipo de modificación podemos llegar a perjudicar la seguridad y funcionalidad de este servidor.


```

# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule auth_form_module modules/mod_auth_form.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
LoadModule authn_core_module modules/mod_authn_core.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authn_socache_module modules/mod_authn_socache.so

```

Figura 3.24. Módulos de apache

Finalmente vamos a comentar las ventajas y desventajas de Apache.

Ventajas

- Gran comunidad de desarrolladores
- Compatible con los CMS más populares del mercado, como WordPress
- Multiplataforma
- De código abierto y gratuito
- La seguridad es bastante alta

Desventajas

- Rendimiento afectado al tener páginas con un índice tráfico bastante alto, más de 10k conexiones
- Surgen brechas de seguridad al manipular módulos

3.3.2.4. Firebase

Firebase es una *plataforma* perteneciente a Google que sirve para el desarrollo de aplicaciones web y aplicaciones móviles.

Esta *plataforma* fue creada en el 2011 pero desde el 2014 pertenece a Google. Una de sus funciones a destacar es hacer que las aplicaciones sean desarrolladas de manera más rápida manteniendo la calidad de estas.

Firestore posee varias herramientas que facilitan el trabajo a los desarrolladores que trabajan en el backend. En este caso vamos a mencionar algunas herramientas y de las cuales nos centraremos en una en especial ya que para este proyecto se ha hecho uso de esta.

Las herramientas las podemos dividir en 4 grupos las cuales son: desarrollo (nos centraremos en una herramienta), crecimiento, monetización y análisis.

En este caso las herramientas de desarrollo están relacionados con:

- Autenticación de usuarios
- Base de datos
- Sistema de almacenamiento en la nube
- Reportador de fallos
- Testing de aplicaciones
- Notificaciones y mensajes a usuarios en tiempo real
- Hosting

Mediante la **autenticación de usuarios** vamos a tener en cuenta un sistema que cuenta con la generación de códigos, en este caso para ser más específicos, es el código OTP (One Time Password), lo que también se conoce como contraseña dinámica, lo que esta herramienta nos va a permitir es que mediante el uso de dicho código podemos comprobar la veracidad de un número de teléfono, puesto que llegará un sms con un código. El número de teléfono al cual se llegó el sms deberá ser válido para que de esta manera se pueda introducir el código en una aplicación que requiera dicha operación.

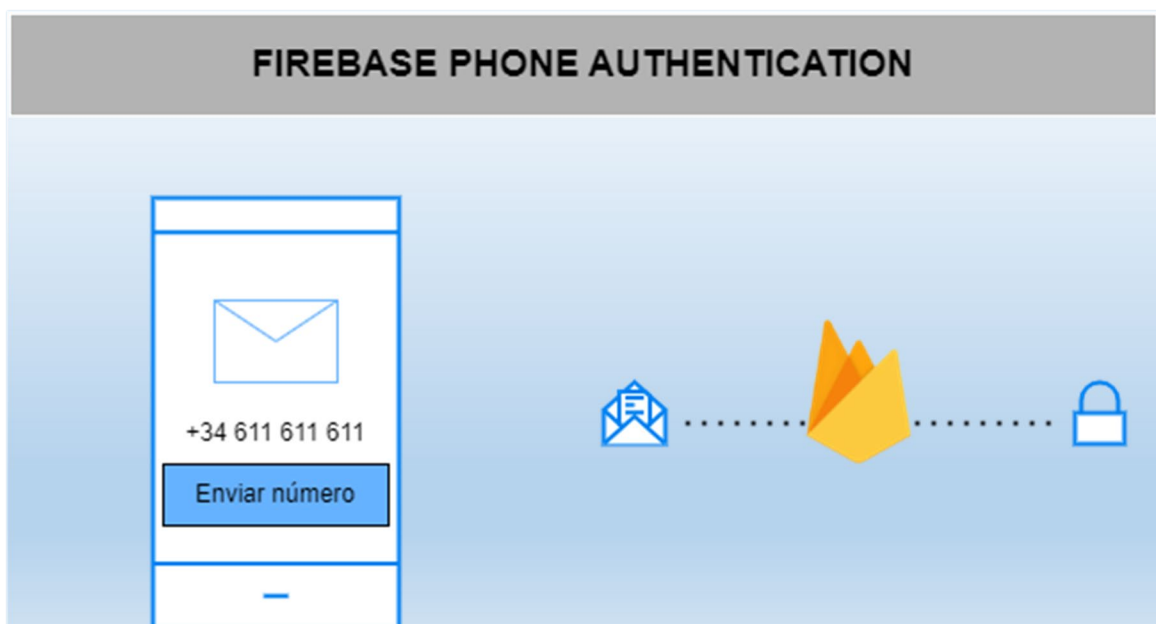


Figura 3.25. OTP Firebase

Actualmente, un usuario registrado en la plataforma con un plan gratuito tiene un límite de 50 sms por día, esto se ha limitado para la prevención de abusos en solicitud de códigos.

De los distintos planes que tenemos disponibles, podemos elegir el plan gratuito denominado Spark, el cual tiene algunas limitaciones, una de ellas es la mencionada anteriormente. En este caso para proyectos de menor escala es una buena opción.

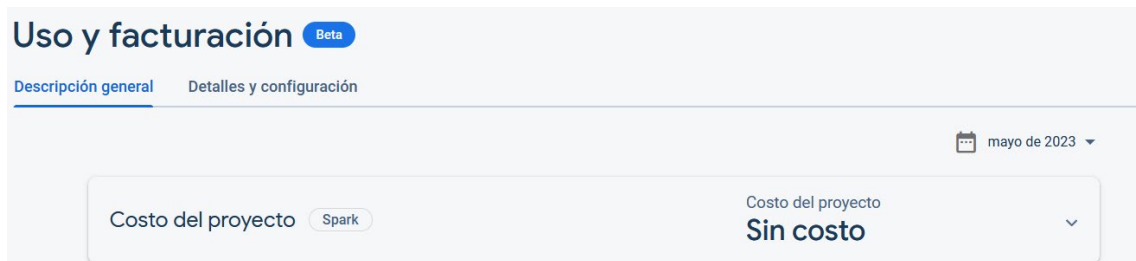


Figura 3.26. Plan Spark

Tenemos un panel donde podemos ver información sobre los usuarios que han hecho uso del código, datos sobre accesos, opciones sobre otras funcionalidades que necesariamente se deberá pasar a otro plan para poder hacer uso de estas.

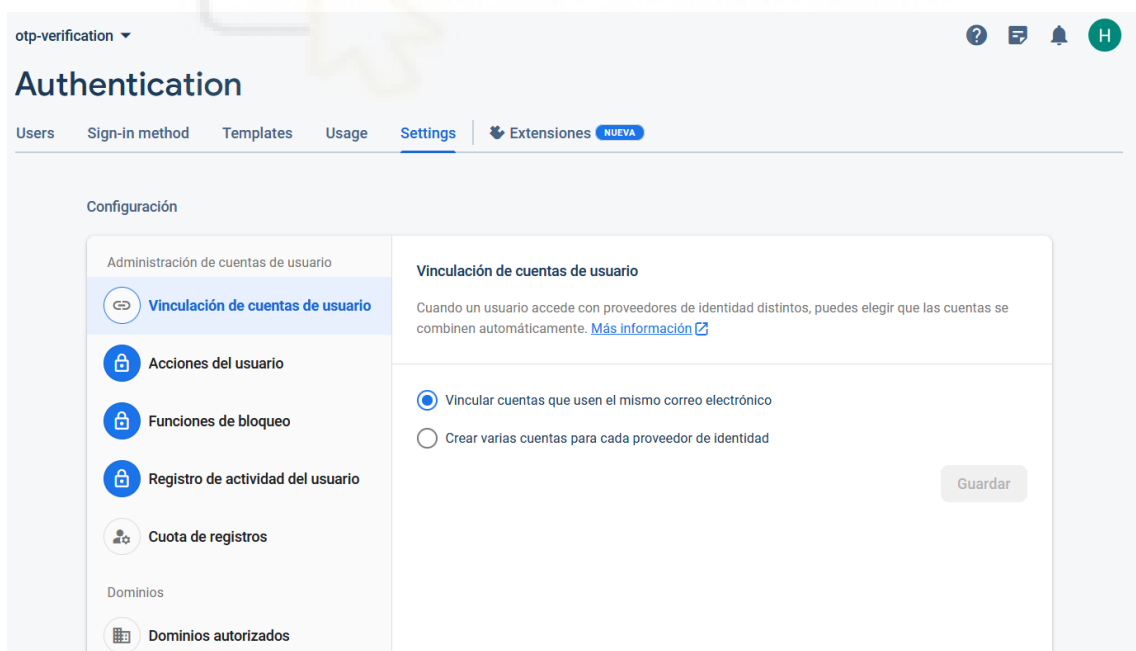


Figura 3.27. Panel usuario de autenticación

Finalmente vamos a mencionar algunas de las ventajas y desventajas que posee Firebase.

Ventajas

- Recomendable para la compartición de datos en tiempo real
- Se funcionalidades se pueden gestionar desde un único panel
- Facilidad de implementar el envío de notificaciones a usuarios
- Permite la monetización
- Soporte gratuito vía email
- Escalabilidad en los pagos, al inicio son gratuitos pero dependiendo la necesidad se pueden aplicar planes de pago si se desea
- Certificados SSL para los usuarios
- Muy buena opción para los desarrolladores backend, puesto que las infraestructuras complejas con las que cuenta firebase ya están desarrolladas

Desventajas

- No es de código abierto
- Dependencia del proveedor
- En algunos países Firebase no funciona
- Solamente están disponibles las bases de datos NoSQL
- Lentitud en las consultas
- Al comenzar, algunos servicios no son gratuitos

3.3.2.5. Node js

Node Js es un entorno en tiempo de ejecución de JavaScript que se ejecuta en el lado del servidor. Este entorno fue desarrollado por los desarrolladores de JavaScript, teniendo en cuenta que querían ir un paso más allá que JavaScript.

Inicialmente JavaScript se podía ejecutar en el lado del cliente o navegador, por lo tanto, los desarrolladores cuando querían ejecutar dicho lenguaje de programación en el servidor, hacían uso de otros lenguajes y herramientas para conseguir ese objetivo. Gracias a **Node.js** podemos ejecutar JavaScript en el lado del servidor ya que Node js está preparado para llevar a cabo ese objetivo y por lo tanto, es un gran ayuda para los desarrolladores y es por ello que es una de las herramientas que más se utilizan en el desarrollo web.



Figura 3.28. Node Js

Ventajas

- Escalable
- Rendimiento y eficacia
- Facilidad
- Open Source
- Comunidad y soporte

Desventajas

- Curva de aprendizaje
- Compatibilidad
- Módulos en desarrollo
- Documentación



Capítulo 4

Metodología, Estructura y Resultados

4.1. PLANIFICACIÓN DEL PROYECTO

En este trabajo tiene como finalidad ser parte de un proyecto académico, por lo tanto, se ha tenido en cuenta algunas etapas para que este se pueda llevar a cabo en su desarrollo. Este trabajo cuenta con una organización de desarrollo previamente analizada para posteriormente poder ser aplicada y obtener resultados aceptables.

Este proyecto consta de varias partes, las cuales se han ido desarrollando según un orden establecido. De esta forma se ha conseguido seguir unas pautas y de esta forma conseguir buenos resultados.

4.1.1. Ciclo de vida

4.1.1.1. Scrum en Jira

Como modelo de ciclo de vida se ha optado por la elección del modelo de tipo SCRUM, este modelo se adapta al proyecto que se ha desarrollado. Para tener en cuenta algunos de los motivos en el que se ha llevado a cabo la elección de este modelo. El proyecto se puede descomponer en piezas y pueden ser manejables, también se ha realizado con poca planeación previa [34].

Por otra parte cabe resaltar que el equipo de trabajo se compone solamente por el desarrollador de la aplicación. Los tiempos de desarrollo se han administrado según la dificultad de la tarea a realizar.

En los siguientes apartados se detalla con capturas algunas de las acciones llevadas a cabo para la organización y seguimiento del proyecto con el modelo SCRUM de la herramienta online JIRA.

Nuestro proyecto estaría dividido en 4 partes, pero en este caso solamente se tienen en cuenta las dos primeras partes, ya que se trata de un prototipo, los otros dos puntos se podrían aplicar si este proyecto se llevase a cabo comercialmente o desarrollarla de una manera más profunda (más extensa y con otras finalidades).

- **Épicas**
- **Incidencias**
- Versiones
- Releases

En la parte de **épicas** tenemos creadas siete (épicas): diseño de la base de datos, prototipo de la aplicación Android, diseño de logo de la aplicación, parte *frontend* de la aplicación Android y la aplicación web, finalmente la parte *backend* de la aplicación Android y de la aplicación Web.

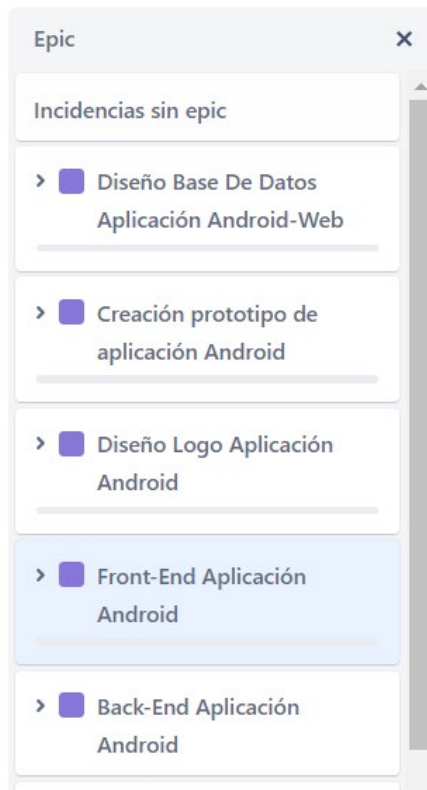


Figura 4.1. Épicas

Dentro de cada *epic* tenemos las incidencias que la componen y se ha procedido a crearlas de la siguiente forma:

- En este primer punto tenemos la *epic* que hace referencia al *diseño de la base de datos* del proyecto, este se compone de **dos incidencias** que son: modelo entidad relación y el modelo relacional.

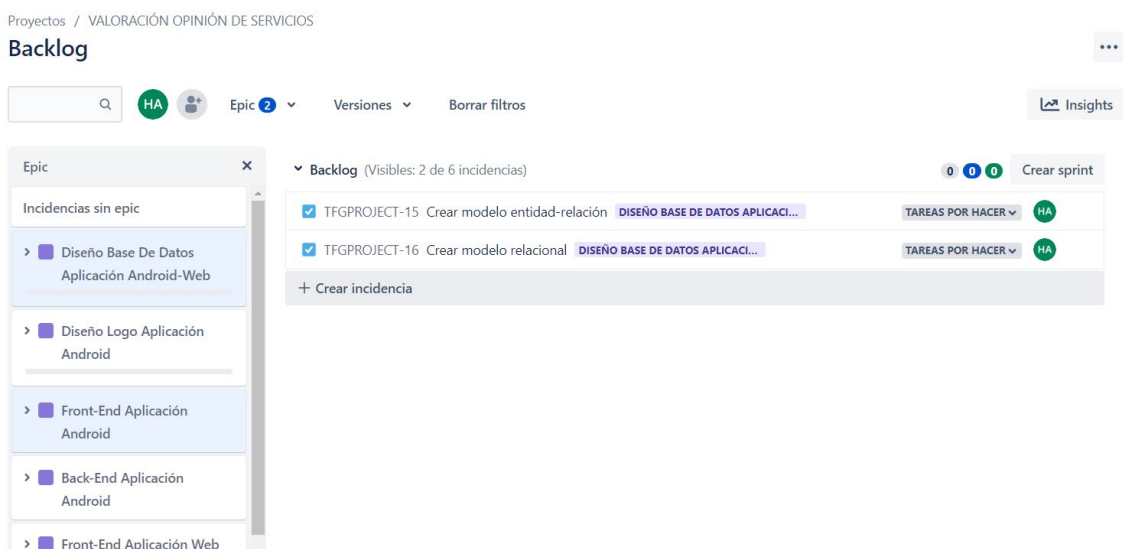


Figura 4.2. Incidencias

- En la segunda epic se elige la creación de una parte del *prototipo de la aplicación Android*, esta tiene como incidencias la creación de las pantallas: menú, servicios y la elección de un determinado servicio.
- En la tercera epic he optado por la creación del logotipo para la aplicación Android. Este cuenta con una incidencia, la cual hace referencia a la creación del logotipo mediante el uso de la herramienta de diseño **Gimp**.
- En la siguiente epic tenemos la parte del front-end de la aplicación Android y esta tiene como incidencias las pantallas de las que se compone la aplicación, el menú, los servicios, el registro de usuario, contactos, etc.

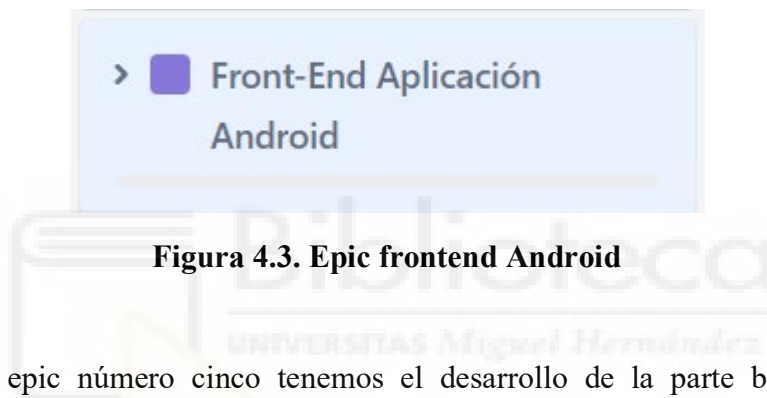


Figura 4.3. Epic frontend Android

- En la epic número cinco tenemos el desarrollo de la parte back-end de la aplicación Android, este tiene incidencias centradas en la parte del servidor haciendo uso del framework, Laravel, aquí se ha implementado la funcionalidad de, el registro de los usuarios, realización de comentarios, sincronización de contactos, etc.
- En la parte final tenemos dos epics que hacen referencia a la parte front-end y backend de la aplicación Web. En las incidencias que hacen referencia al front-end, tenemos las ventanas de login, registro, alta de servicios, etc. En las incidencias del back-end tenemos todo lo relacionado con la funcionalidad del registro de usuario, login, alta de servicios, baja de un servicio, etc.

4.2. DISEÑO

4.2.1. Base de datos

Para proceder al diseño de la base de datos de nuestro proyecto se ha tenido en cuenta las partes pertenecientes o los interesados de este, habiendo tenido un análisis no muy riguroso pero sí cuidadoso.

En primer lugar tenemos al usuario como una de las partes importantes de nuestra base de datos. Este usuario lo podemos dividir en tres tipos de usuario que son los siguientes: usuario de la aplicación Android, usuario prestador de servicios y usuario administrador. Una cosa importante que hay que tener en cuenta es que se podría crear una tabla para cada usuario con sus atributos necesarios pero en este caso como es un prototipo, se ha definido como una tabla para los tres tipos de usuarios.

A continuación pasaremos a describir las entidades participantes junto a sus atributos y las relaciones de estas con otras entidades.

- **Usuario de la aplicación:** Podrá realizar más de un comentario en un determinado servicio, almacenando la fecha y eligiendo el número de estrellas como valoración. Este también podrá obtener contactos (usuarios que deberán estar registrados en la aplicación) de su dispositivo móvil. Tiene como atributos: alias, rol, teléfono (PK), email y password (no nulo).
- **Usuario administrador:** Se encargará de dar de alta a los *usuarios prestadores de servicios* y de crear los *tipos de servicios*, de este obtendremos los siguientes datos: rol (por defecto 1), teléfono (PK), alias (no aplicable), password (no nulo) y email.
- **Usuario prestador de servicios:** Se encargará de dar de alta un servicio y este tendrá su tipo (restaurante, heladería, hotel, etc.) y a su vez tendrá la posibilidad de agregar fotos al servicio que está siendo dado de alta. Los datos que obtendremos son: rol (por defecto 0), teléfono (PK), alias (no aplicable), password (no nulo) y email.
- **Servicios:** Son los servicios que tendremos disponibles en la aplicación que serán dados de alta por un *prestador de servicios*. Cada servicio tendrá obligadamente al menos una imagen. De los servicios obtendremos los siguientes datos: id (PK), nombre, valoración media y descripción.
- **Tipo de servicio:** Son los distintos tipos (bar, restaurante, heladería, etc.) de servicios que podemos tener para un servicio. Los datos que obtendremos son: id (PK) y tipo.
- **Fotos:** Son las imágenes que pertenecerán a un determinado servicio. Los datos que obtendremos son: id (PK), nombre y uri (link de la imagen en el servidor).

4.2.1.1. Modelo Entidad-Relación

Este modelo nos va a permitir ver la información de las entidades (se encuentran dentro de rectángulos) y las relaciones (se encuentran dentro de rombos) de una manera sencilla, mediante un esquema.

Una vez tenida en cuenta la información de la descripción perteneciente a la base de datos, se ha procedido a crear el modelo E-R [35] para poder plasmar la información obtenida, en la que se puede observar en la siguiente imagen todas las relaciones junto a sus entidades y atributos.

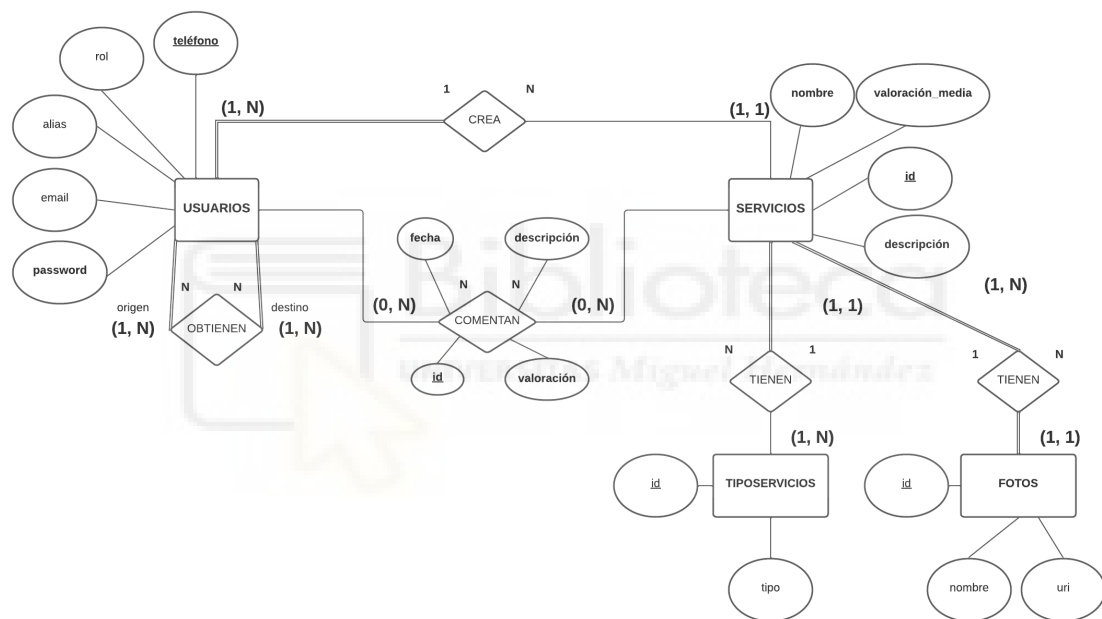


Figura 4.4. Modelo E-R

4.2.1.2. Modelo relacional

Este modelo representa la base de datos lógica con la información de las entidades y sus atributos y relaciones. Su representación se realiza mediante tablas, con sus respectivos campos y restricciones. En la siguiente imagen se puede observar dicho modelo.

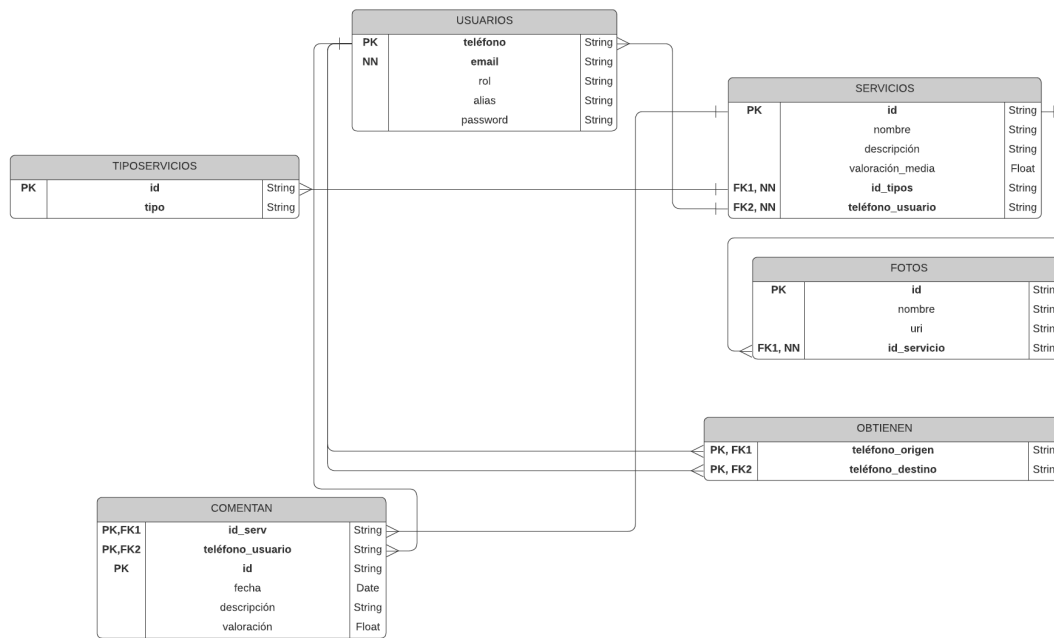


Figura 4.5. Modelo relacional

4.2.2. Logo aplicación

4.2.2.1. Diseño del logo

Para la creación del logo de la aplicación Android se ha procedido a descargarse imágenes gratis de la página *flaticon*. Como la aplicación está basada en la reseña de usuarios sobre un determinado servicio, entonces se han elegido las imágenes adecuadas para representar los servicios de entre los varios que podemos comentar.

Todo este trabajo se ha realizado con la herramienta **Gimp**, montando imágenes por capas a partir de una capa base.

La parte de logo está compuesta en la parte superior con cuatro imágenes. Un helado haciendo referencia a un servicio de tipo heladería, un plato de ensalada haciendo referencia a un restaurante, un hotel haciendo referencia a este mismo y una mesa con bebidas haciendo referencia a un bar.

En la parte media tenemos el nombre de la aplicación *VOS*, valoración y opinión de servicios, se ha aplicado unos efectos de degradado y unas sombras.

En la parte inferior tenemos una última imagen que describe a los usuarios que van a publicar una reseña en un determinado servicio.

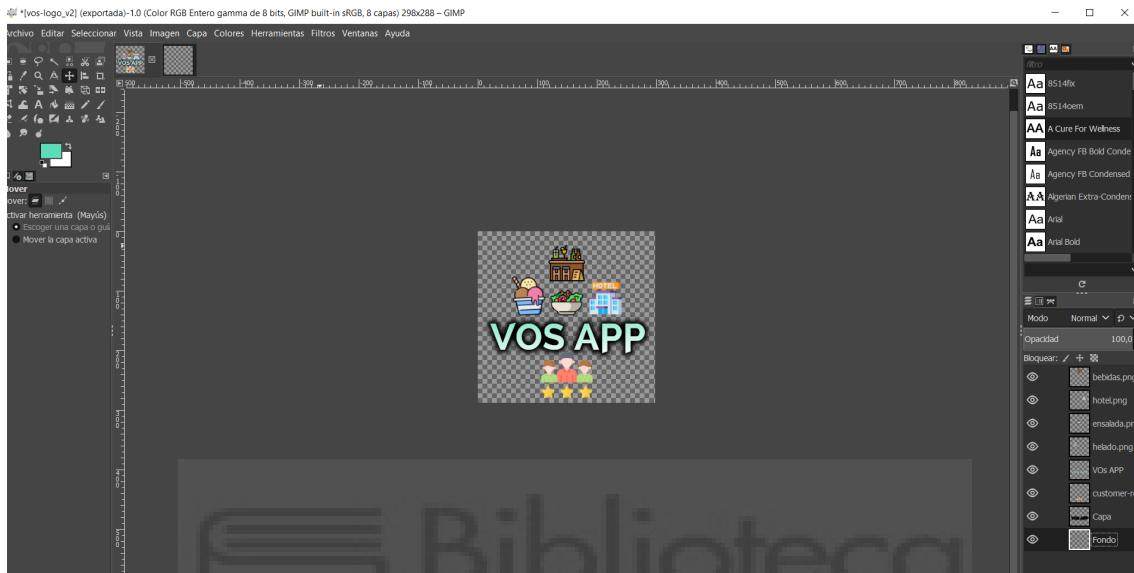


Figura 4.6. Logo VOS diseño en Gimp



Figura 4.7. Logo VOS app

4.2.3. Prototipo aplicación Android

4.2.3.1. Pantallas del prototipo

La parte de las pantallas se ha realizado mediante la herramienta online *Figma*, por la facilidad que tiene en la creación de prototipos. En la creación de las pantallas de la aplicación Android como parte del prototipo, solamente se han tenido en cuenta algunas pantallas, las más importantes como primera impresión y la idea que se quería plasmar.

En este caso como primera pantalla se ha desarrollado como prototipo, la pantalla del menú de la aplicación Android.



Figura 4.8. Prototipo menú app Android

En la siguiente imagen se puede ver un listado de servicios los cuales tienen datos que los describen para que una manera simple el usuario pueda observar los distintos servicios que tendrá disponibles.

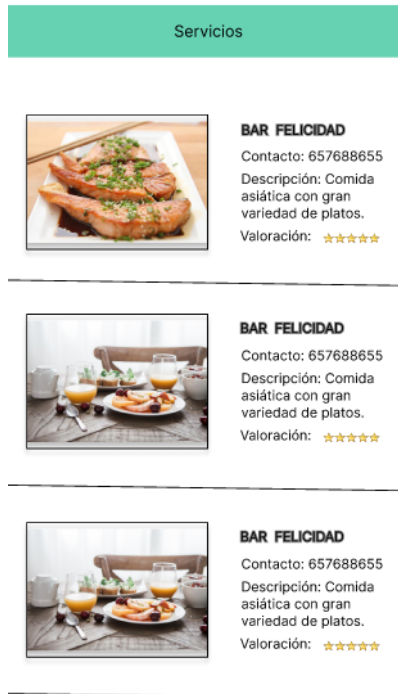


Figura 4.9. Prototipo servicios app Android

En la siguiente imagen podemos apreciar un servicio con sus datos y la realización de un comentario junto a su valoración.

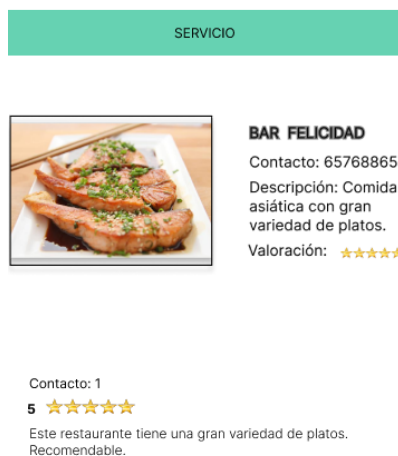


Figura 4.10. Prototipo con un servicio y un comentario

4.3. CAPTURA DE REQUISITOS

4.3.1. Roles de usuarios

El proyecto tiene tres tipos de usuarios que tienen sus roles, el primer usuario será el que está dirigido a navegar o interactuar con la aplicación Android, este se ha denominado **usuario app**, el segundo usuario será el **administrador** que se encargará de dar de alta a usuarios prestadores de servicios y a los tipos de servicios, mediante una aplicación web. Por último, tenemos al usuario **prestador de servicios** que se encargará de dar de alta a sus servicios también mediante el uso de una aplicación web.

Tabla 4.1. Rol de usuario. Usuario app

Rol de usuario	usuario app
Descripción	Podrá navegar en la aplicación VOS una vez registrado en la que podrá: <ul style="list-style-type: none">- Dejar una reseña- Ver comentarios de sus contactos y de los contactos de sus contactos de un determinado servicio- Modificar su alias y correo- Cambiar el color de fondo de las cards de las reseñas de sus contactos.- Sincronizar/desincronizar sus contactos- Otras funcionalidades
Casos de uso	C.U.3, C.U.5, C.U.7, C.U.8, C.U.11,C.U.12,C.U.13,C.U.14, C.U.23, C.U.24, C.U.25, C.U.26, C.U.27

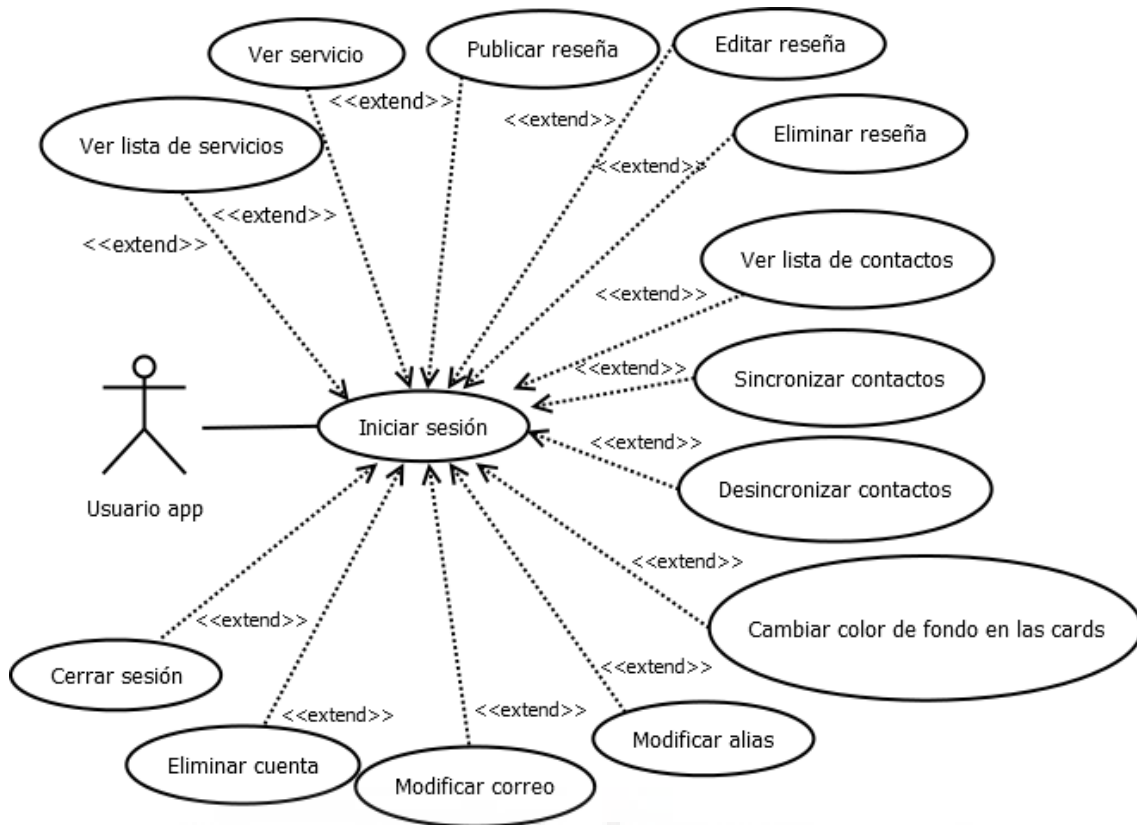


Figura 4.11. Esquema usuario app

Tabla 4.2. Rol de usuario. Prestador de servicios

Rol de usuario	Prestador de servicios
Descripción	Se encarga de dar de alta servicios y de estos podrá: <ul style="list-style-type: none"> - Dar de alta otro servicio - Editar servicio - Eliminar servicio - Ver la información de uno de sus servicios
Casos de uso	C.U.2, C.U.4, C.U.6, C.U.7, C.U.15,C.U.16,C.U.17,C.U.18

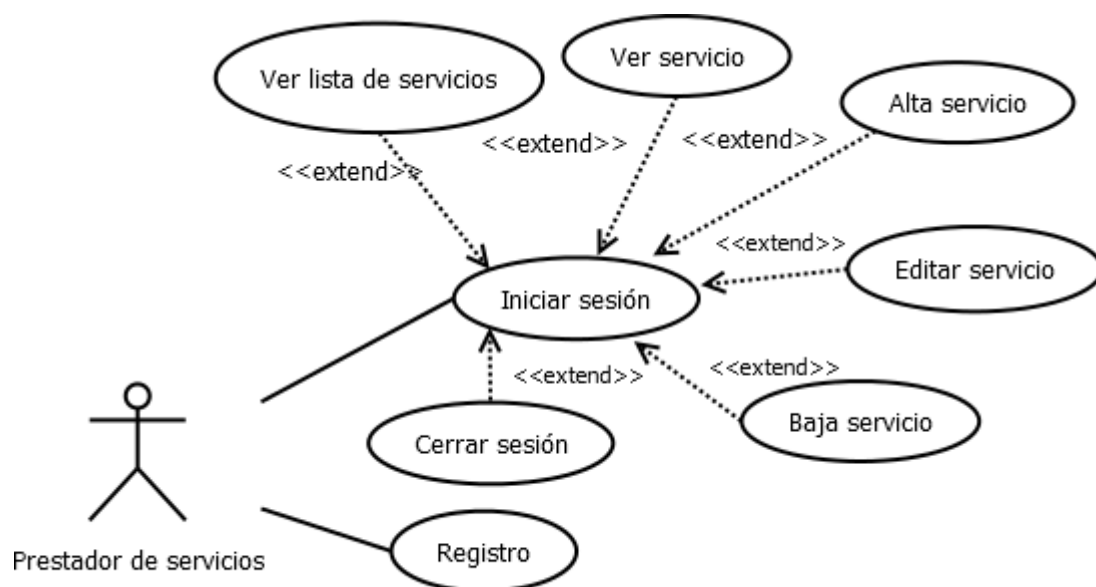


Figura 4.12. Esquema prestador de servicios

Tabla 4.3. Rol de usuario. Administrador

Rol de usuario	Administrador
Descripción	<p>Es el administrador y solamente tendrá funcionalidades sobre el prestador de servicios y se encarga de:</p> <ul style="list-style-type: none"> - Dar de alta a un prestador de servicios - Editar prestador de servicios - Eliminar prestador de servicios - Dar el alta un tipo de servicio - Editar un tipo de servicio - Otras funcionalidades
Casos de uso	C.U.4, C.U.6, C.U.9, C.U.10, C.U.19, C.U.20, C.U.21, C.U.22, C.U.23, C.U.24

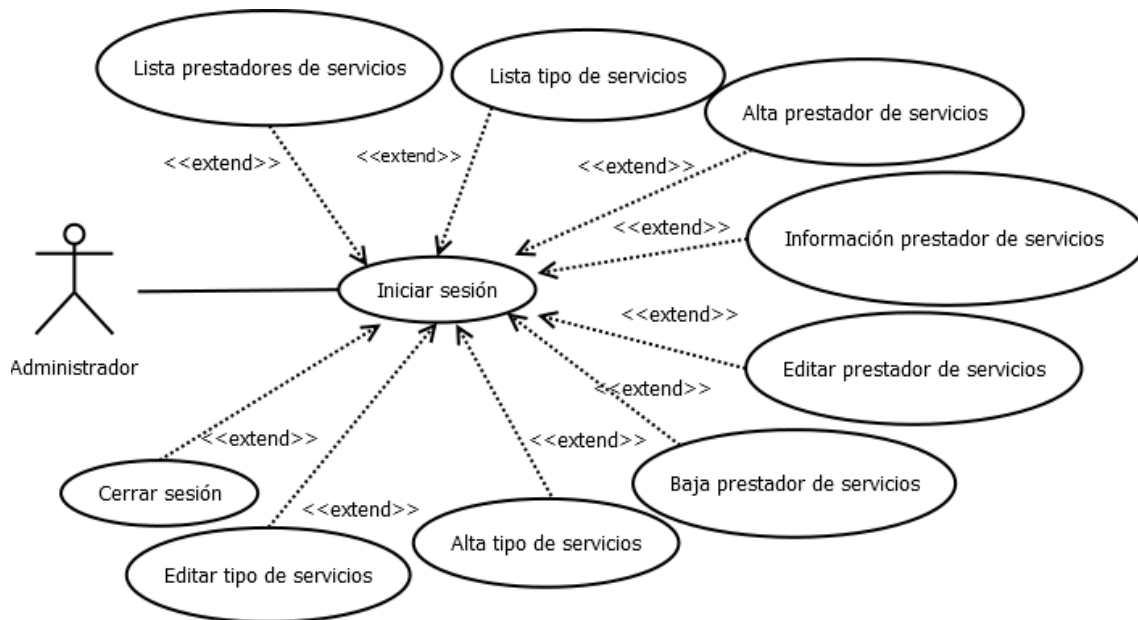


Figura 4.13. Esquema administrador

4.3.2. Casos de uso

Tabla 4.4. Caso de uso. Registro en aplicación Android

C.U.1	Registro en aplicación Android
Actores	Usuario app
Descripción	Registro en la aplicación Android mediante un formulario.
Precondición	
Secuencia normal	P1 - Introducir número móvil P2 - Introducir alias P2 - Introducir correo P3 - Introducir contraseña P4 - Introducir repetir contraseña P5 - Pulsar registrar
Poscondición	Una vez registrado, siempre que abra la aplicación pasará a la pantalla del menú principal.
Excepciones	- Si el usuario se registra con un número ya registrado, mostrará un error.

	<ul style="list-style-type: none"> - Si el correo ingresado ya se encuentra registrado, mostrará un mensaje de error. - Si la contraseña no cumple con los requisitos, mostrará un mensaje de error. - Si las dos contraseñas no coinciden al registrar, mostrará un mensaje de error.
Comentarios	Si cierra sesión, pasará a la ventana con el menú de inicio de sesión.

Tabla 4.5. Caso de uso. Registro en aplicación Web

C.U.2	Registro
Actores	Prestador de servicios
Descripción	Registro en la aplicación web mediante un formulario.
Precondición	
Secuencia normal	P1 - Introducir Nombre (no aplicable) P2 - Introducir correo P2 - Introducir número móvil P3 - Introducir contraseña P4 - Introducir repetir contraseña P5 - Pulsar registrar
Poscondición	Una vez registrado, aparecerá una opción en el menú para poder ver los servicios y agregar servicios.
Excepciones	<ul style="list-style-type: none"> - Si el usuario se registra con un número ya registrado, mostrará un error. - Si el correo ingresado ya se encuentra registrado, mostrará un mensaje de error. - Si las dos contraseñas no coinciden al registrar, mostrará un mensaje de error.
Comentarios	

Tabla 4.6. Caso de uso. Inicio de sesión en aplicación Android

C.U.3	Inicio sesión
Actores	Usuario app
Descripción	Inicio de sesión mediante un formulario.

Precondición	C.U.1
Secuencia normal	P1 - Introducir número móvil P2 - Introducir contraseña P3 - Pulsar entrar
Poscondición	Una vez identificado, será redirigido al menú principal de la aplicación.
Excepciones	<ul style="list-style-type: none"> - Si el usuario se registra con un número ya registrado, mostrará un error. - Si la contraseña no coincide con el del usuario registrado, mostrará un mensaje de error.
Comentarios	

Tabla 4.7. Caso de uso. Inicio de sesión en aplicación Web

C.U.4	Iniciar sesión
Actores	Prestador de servicios y Administrador
Descripción	Inicio de sesión mediante un formulario.
Precondición	C.U.2
Secuencia normal	P1 - Introducir correo P2 - Introducir contraseña P3 - Pulsar entrar
Poscondición	Una vez identificado, será redirigido al menú principal de la aplicación web.
Excepciones	<ul style="list-style-type: none"> - Si el usuario se identifica con un correo no registrado, mostrará un error. - Si la contraseña no coincide con el de prestador de servicios/administrador, mostrará un mensaje de error.
Comentarios	

Tabla 4.8. Caso de uso. Cerrar sesión en aplicación Android

C.U.5	Cerrar sesión
Actores	Usuario app
Descripción	Cerrar sesión mediante un botón.

Precondición	C.U.3
Secuencia normal	P1 - Cerrar sesión mediante la acción en un botón
Poscondición	Al cerrar sesión, aparecerá la ventana de iniciar sesión.
Excepciones	
Comentarios	

Tabla 4.9. Caso de uso. Cerrar sesión en aplicación web

C.U.6	Cerrar sesión
Actores	Prestador de servicios y Administrador
Descripción	Cerrar sesión en la aplicación web.
Precondición	C.U.4
Secuencia normal	P1 - Cerrar sesión mediante la acción en un botón
Poscondición	Al cerrar sesión, aparecerá la ventana de iniciar sesión.
Excepciones	
Comentarios	Se cerrará sesión automáticamente al pasar un tiempo por inactividad del usuario.

Tabla 4.10. Caso de uso. Ver lista de servicios en aplicación Android/Web

C.U.7	Ver lista de servicios
Actores	Usuario app y prestador de servicios
Descripción	Cuando se dirija a la opción servicios, estos estarán listados.
Precondición	C.U.3 y C.U.4 respectivamente
Secuencia normal	P1 - Pulsar en la opción servicios
Poscondición	
Excepciones	- Si no hay servicios disponibles, se mostrará un mensaje describiendo el motivo.
Comentarios	La lista de servicios estará disponible en el menú principal.

Tabla 4.11. Caso de uso. Ver lista de contactos en aplicación Android

C.U.8	Ver lista de contactos
--------------	------------------------

Actores	Usuario app
Descripción	Cuando se dirija a la opción contactos, estos estarán listados mediante un check los que están sincronizados, con un radio button los que no están sincronizados. Habrá otro listado con los contactos que no tienen la aplicación instalada.
Precondición	C.U.3
Secuencia normal	P1 - Pulsar en la opción contactos
Poscondición	Se verán los contactos de nuestro dispositivo móvil
Excepciones	Al entrar en la opción contactos por primera vez tendremos que: <ul style="list-style-type: none"> - Si se acepta los términos y condiciones se sincronizaran los contactos, caso contrario no podremos ver el listado de estos.
Comentarios	La lista de contactos estará disponible en el menú principal.

Tabla 4.12. Caso de uso. Lista prestadores de servicios en aplicación Web

C.U.9	Lista prestadores de servicios
Actores	Administrador
Descripción	Se podrá visualizar los prestadores de servicios dados de alta.
Precondición	C.U.4
Secuencia normal	P1 - Pulsar en la opción usuarios
Poscondición	Se verán listados los usuarios dados de alta
Excepciones	
Comentarios	La lista de usuarios estará disponible en el menú.

Tabla 4.13. Caso de uso. Lista tipo de servicios en aplicación web

C.U.10	Lista de tipo de servicios
Actores	Administrador
Descripción	Se podrá visualizar los tipos de servicios creados.
Precondición	C.U.4
Secuencia normal	P1 - Pulsar en la opción tipo de servicios

Poscondición	Se verán listados los tipos de servicios dados de alta
Excepciones	
Comentarios	La lista de tipo de servicios estará disponible en el menú.

Tabla 4.14. Caso de uso. Ver un servicio en aplicación Android

C.U.11	Ver un servicio
Actores	Usuario app
Descripción	Se podrá ver en detalle un servicio seleccionado.
Precondición	C.U.3 y C.U.7
Secuencia normal	P1 - Seleccionar un servicio de la lista de servicios
Poscondición	Se verá el servicio en detalle con sus reseñas
Excepciones	
Comentarios	Se verán las reseñas de los contactos y de los contactos de estos siempre que estén disponibles.

Tabla 4.15. Caso de uso. Publicar reseña en aplicación Android

C.U.12	Publicar reseña
Actores	Usuario app
Descripción	Se realizará un comentario y una valoración teniendo en cuenta el número de estrellas.
Precondición	C.U.3, C.U.7 y C.U.11
Secuencia normal	P1 - Pulsar sobre un icono de mensaje P2 - Se abre una ventana P3 - Introduce el comentario P4 - Elige el número de estrellas P5 - Enviar
Poscondición	Su comentario se verá publicado
Excepciones	
Comentarios	El comentario tendrá unas opciones y unos detalles.

Tabla 4.16. Caso de uso. Editar una reseña en aplicación Android

C.U.13	Editar reseña
Actores	Usuario app
Descripción	Se editará el comentario y una valoración teniendo en cuenta el número de estrellas.
Precondición	C.U.3, C.U.7, C.U.11 y C.U.12
Secuencia normal	P1 - Se elige un comentario(solamente los nuestros) P1 - Pulsar sobre el icono de editar P2 - Se abre una ventana P3 - Introduce el comentario P4 - Elije el número de estrellas P5 - Actualizar
Poscondición	El comentario se verá actualizado
Excepciones	
Comentarios	El comentario tendrá unas opciones y unos detalles.

Tabla 4.17. Caso de uso. Eliminar reseña en aplicación Android

C.U.14	Eliminar reseña
Actores	Usuario app
Descripción	Se eliminará la reseña.
Precondición	C.U.3, C.U.7, C.U.11 y C.U.12
Secuencia normal	P1 - Pulsar sobre el icono de eliminar P2 - Se abre una ventana confirmando la eliminación P3 - Pulsa sobre ok para eliminar
Poscondición	La reseña se eliminará
Excepciones	
Comentarios	

Tabla 4.18. Caso de uso. Alta servicio en aplicación web

C.U.15	Alta servicio
Actores	Prestador de servicios
Descripción	Se agregará un servicio de un tipo.
Precondición	C.U.4
Secuencia normal	<p>P1 - Pulsar sobre la opción Servicios</p> <p>P2 - Pulsar sobre agregar servicio</p> <p>P3 - Introducir nombre servicio</p> <p>P4 - Introducir tipo</p> <p>P5 - Introducir descripción</p> <p>P6 - Dirección (no aplicable)</p> <p>P7 - Ciudad (no aplicable)</p> <p>P8 - Seleccionar imágenes</p> <p>P9 - Crear</p>
Poscondición	Se creará un servicio
Excepciones	<ul style="list-style-type: none"> - Si no se introduce un nombre, mostrará un mensaje de error - Si no introduce la descripción, mostrará un mensaje de error - Si no elige al menos una imagen, mostrará un mensaje de error
Comentarios	El prestador de servicios será redirigido al menú con el resto de servicios dados de alta pertenecientes a este mismo prestador.

Tabla 4.19. Caso de uso. Editar servicio en aplicación web

C.U.16	Editar servicio
Actores	Prestador de servicios
Descripción	Se editará un servicio.
Precondición	C.U.4 y C.U.15
Secuencia normal	<p>P1 - Pulsar sobre la opción Servicios</p> <p>P2 - Pulsar sobre editar servicio</p>

	<p>P3 - Introducir nombre servicio</p> <p>P4 - Introducir tipo</p> <p>P5 - Introducir descripción</p> <p>P6 - Dirección (no aplicable)</p> <p>P7 - Ciudad (no aplicable)</p> <p>P8 - Seleccionar imágenes</p> <p>P9 - Actualizar</p>
Poscondición	Se actualizarán los datos de un servicio
Excepciones	
Comentarios	El prestador de servicios será redirigido al menú con el resto de servicios dados de alta pertenecientes a este mismo prestador.

Tabla 4.20. Caso de uso. Ver servicio en aplicación web

C.U.17	Ver servicio
Actores	Prestador de servicios
Descripción	Se mostrará los datos de un servicio.
Precondición	C.U.4 y C.U.15
Secuencia normal	<p>P1 - Pulsar sobre la opción Servicios</p> <p>P2 - Pulsar sobre un botón de color azul claro de un determinado servicio</p> <p>P3 - Se mostrará una ventana con los datos: nombre, descripción, tipo de servicio, fecha en que dio de alta y las imágenes correspondientes a dicho servicio.</p>
Poscondición	Se mostrará una ventana con los detalles de un servicio
Excepciones	
Comentarios	La opción de ver los detalles de un servicio se encuentra en el menú mediante un listado de todos los servicios.

Tabla 4.21. Caso de uso. Baja servicio en aplicación web

C.U.18	Baja servicio
---------------	---------------

Actores	Prestador de servicios
Descripción	Se eliminará un servicio.
Precondición	C.U.4 y C.U.15
Secuencia normal	P1 - Pulsar sobre la opción Servicios P2 - Pulsar sobre eliminar servicio P3 - Mensaje de confirmación, aceptar
Poscondición	Se eliminará un servicio
Excepciones	
Comentarios	El servicio será eliminado y no aparecerá junto al resto de servicios.

Tabla 4.22. Caso de uso. Alta prestador de servicios en aplicación web

C.U.19	Alta prestador de servicios
Actores	Administrador
Descripción	Se dará de alta a un usuario denominado prestador de servicios.
Precondición	C.U.4
Secuencia normal	P1 - Pulsar sobre la opción usuarios P2 - Pulsar sobre agregar usuario P3 - Introducir nombre (no aplicable) P4 - Introducir correo P5 - Introducir teléfono P6 - Introducir password P7 - Repetir password P8 - Dirección (no aplicable) P9 - Ciudad (no aplicable) P10 - Rol (prestador de servicios, por defecto)
Poscondición	Se dará de alta a un usuario y será redirigido al menú, visualizando una lista con los usuarios dados de alta
Excepciones	- Si al introducir un correo y este ya está registrado, mostrará un mensaje de error

	<ul style="list-style-type: none"> - Si al introducir los dos passwords, no coinciden, mostrará un mensaje de error - Si al introducir un número de teléfono y este ya está registrado, mostrará un mensaje de error
Comentarios	La opción usuarios se encuentra en el menú de la aplicación.

Tabla 4.23. Caso de uso. Editar prestador de servicios en aplicación web

C.U.20	Editar prestador de servicios
Actores	Administrador
Descripción	Se editará a un prestador de servicios.
Precondición	C.U.4 y C.U.19
Secuencia normal	<p>P1 - Pulsar sobre la opción usuarios</p> <p>P2 - Pulsar sobre el icono de editar un usuario</p> <p>P3 - Introducir nombre (no aplicable)</p> <p>P4 - Introducir correo</p> <p>P5 - Introducir teléfono</p> <p>P6 - Dirección (no aplicable)</p> <p>P7 - Ciudad (no aplicable)</p> <p>P8 - Rol (prestador de servicios, por defecto)</p>
Poscondición	Se editarán algunos datos de un prestador de servicios y será redirigido al menú, visualizando una lista con los prestadores de servicios dados de alta y los datos actualizado del mismo.
Excepciones	<ul style="list-style-type: none"> - Si al introducir el nombre o correo como campos vacíos, mostrará un mensaje de error
Comentarios	La opción usuarios se encuentra en el menú de la aplicación.

Tabla 4.24. Caso de uso. Información, prestador de servicios en aplicación web

C.U.21	Información del prestador de servicios
Actores	Administrador
Descripción	Se obtendrán los datos de un prestador de servicios.

Precondición	C.U.4 y C.U.19
Secuencia normal	<p>P1 - Pulsar sobre la opción usuarios</p> <p>P2 - Pulsar un botón de color azul claro</p> <p>P3 - Se mostrará una ventana con los datos del usuario: número de teléfono, rol y la fecha de alta</p> <p>P4 - Botón atrás para volver al menú</p>
Poscondición	Se mostrará una ventana con los datos del usuario
Excepciones	
Comentarios	La opción de ver los detalles de un usuario se encuentra en el menú mediante un listado de todos los prestadores de servicios dados de alta.

Tabla 4.25. Caso de uso. Baja prestador de servicios en aplicación web

C.U.22	Baja prestador de servicios
Actores	Administrador
Descripción	Se dará de baja a un prestador de servicios.
Precondición	C.U.4 y C.U.19
Secuencia normal	<p>P1 - Pulsar sobre la opción usuarios</p> <p>P2 - Pulsar un botón de rojo</p> <p>P3 - Se mostrará una ventana confirmando la baja</p> <p>P4 - Aceptar</p>
Poscondición	El prestador de servicios quedará eliminado de la lista de todos los prestadores de servicios disponibles
Excepciones	
Comentarios	La opción de eliminar un prestador de servicios se encuentra el menú, listando todos los prestadores de servicios y cada uno tendrá un botón de color rojo para poder darlo de baja.

Tabla 4.26. Caso de uso. Alta tipo de servicio en aplicación web

C.U.23	Alta tipo de servicio
Actores	Administrador

Descripción	Se dará de alta a un tipo de servicio
Precondición	C.U.4
Secuencia normal	P1 - Pulsar sobre la opción tipo servicios P2 - Pulsar sobre agregar tipo servicio P3 - Introducir nombre del nuevo tipo de servicio P4 - Crear
Poscondición	Se dará de alta un nuevo tipo de servicio
Excepciones	- Si al introducir un tipo de servicio y este ya está registrado, mostrará un mensaje de error
Comentarios	La opción de tipo de servicios se encuentra en el menú de la aplicación.

Tabla 4.27. Caso de uso. Editar tipo de servicio en aplicación web

C.U.24	Editar tipo de servicio
Actores	Administrador
Descripción	Se editará un tipo de servicio.
Precondición	C.U.4 y C.U.23
Secuencia normal	P1 - Pulsar sobre la opción tipo servicios P2 - Pulsar sobre el icono de color verde para editar P3 - Introducir nombre del tipo de servicio P4 - Actualizar
Poscondición	Se editará el nombre del tipo de servicio y se redirigirá al menú de la aplicación
Excepciones	- Si al introducir el nombre o correo como campos vacíos, mostrará un mensaje de error
Comentarios	La opción usuarios se encuentra en el menú de la aplicación.

Tabla 4.28. Caso de uso. Modificar alias y correo en aplicación Android

C.U.25	Modificar alias y correo
Actores	Usuario app

Descripción	Se modificará el alias de un usuario registrado en la aplicación.
Precondición	C.U.3
Secuencia normal	P1 - Pulsar sobre la opción cuenta P2 - Pulsar sobre actualizar datos P3 - Introducir alias P4 - Introducir correo P5 - Actualizar
Poscondición	Los datos se actualizarán, el usuario permanecerá en la misma pantalla
Excepciones	- Si el correo ya se encuentra registrado, mostrará un mensaje de error
Comentarios	Se podrá modificar uno o los dos campos a la vez, alias y correo.

Tabla 4.30. Caso de uso. Cambiar color de fondo en las cards en aplicación Android

C.U.26	Cambiar color de fondo en las cards
Actores	Usuario app
Descripción	Se modificará el color de fondo en las cards de las reseñas.
Precondición	C.U.3
Secuencia normal	P1 - Pulsar sobre la opción, opciones P2 - Elegimos el color con la paleta de colores P3 - Pulsamos cambiar color
Poscondición	Se redirigirá a la pantalla del menú
Excepciones	
Comentarios	El color de la card se podrá ver en la parte superior cuando se vaya eligiendo un color.

Tabla 4.31. Caso de uso. Desincronizar contactos en Aplicación Android

C.U.27	Desincronizar contactos
Actores	Usuario app

Descripción	Se desincronizará un contacto de la lista de contactos
Precondición	C.U.3
Secuencia normal	P1 - Pulsar sobre la opción contactos P2 - Pulsamos sobre el icono descriptivo de quitar contactos P3 - En la ventana elegimos el contacto mediante el check button P4 - Pulsamos en desincronizar
Poscondición	En la lista principal de contactos, el contacto desincronizado aparecerá con un check button
Excepciones	
Comentarios	Para volver a la lista de contactos principal, pulsamos sobre el icono descriptivo de quitar contactos.

Tabla 4.32. Caso de uso. Sincronizar contactos en aplicación Android

C.U.28	Sincronizar contactos
Actores	Usuario app
Descripción	Se sincronizará un contacto de la lista de contactos.
Precondición	C.U.3
Secuencia normal	P1 - Pulsar sobre la opción contactos P2 - Pulsamos sobre un check button de un contacto P3 - Pulsamos en sincronizar
Poscondición	En la lista principal de contactos, el contacto sincronizado aparecerá con un check
Excepciones	
Comentarios	El check hace referencia a los contactos sincronizados.

Tabla 4.33. Caso de uso. Eliminar cuenta en Aplicación Android

C.U.29	Eliminar cuenta
Actores	Usuario app
Descripción	Se dará de baja la cuenta de la aplicación Android
Precondición	C.U.3

Secuencia normal	P1 - Pulsar sobre la opción cuenta P2 - Pulsamos sobre eliminar cuenta P3 - Botón eliminar cuenta P4 - Al aceptar, se dará de baja la cuenta
Poscondición	Se redirigirá a la pantalla inicial de la aplicación
Excepciones	
Comentarios	

4.4. IMPLEMENTACIÓN

4.4.1. Aplicación Android - frontend

4.4.1.1. Creación proyecto

La creación de un proyecto en **React Native** es bastante sencillo, hay que ejecutar unos comandos para que nos permita crear un proyecto base y en el que a partir de este, poder trabajar.

Para poder crear un proyecto debemos tener instalado previamente *NodeJS* que es una herramienta imprescindible, ya que con este gestor tenemos un comando de nombre **npm** que nos va a permitir instalar paquetes. Otro aspecto a destacar es tener instalado un emulador Android o un dispositivo móvil para poder ejecutar la aplicación.

Para poder crear un proyecto con la última versión de React Native debemos ejecutar el siguiente comando en una terminal.

```
npx react-native@latest init NuevoProyecto
```

Nos aparecerá la siguiente ventana, en la que informa los pasos realizados para poder crear el proyecto o aplicación.

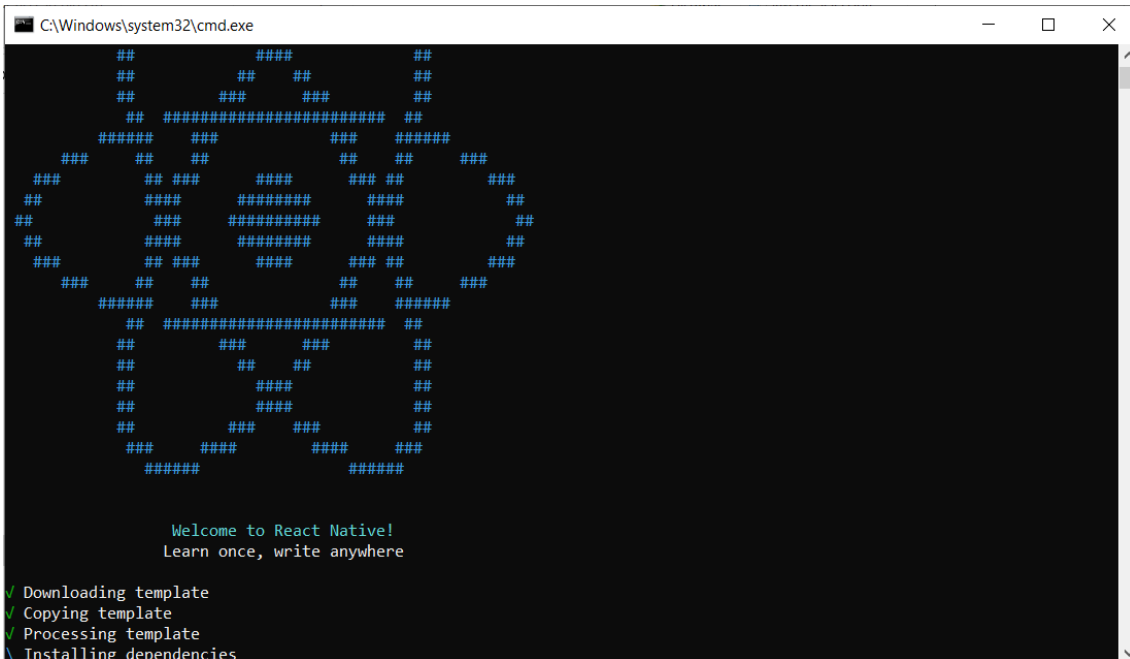


Figura 4.14. Creación proyecto en React Native

La siguiente imagen corresponde a los ficheros y carpetas creadas mediante la ejecución del comando mencionado anteriormente.

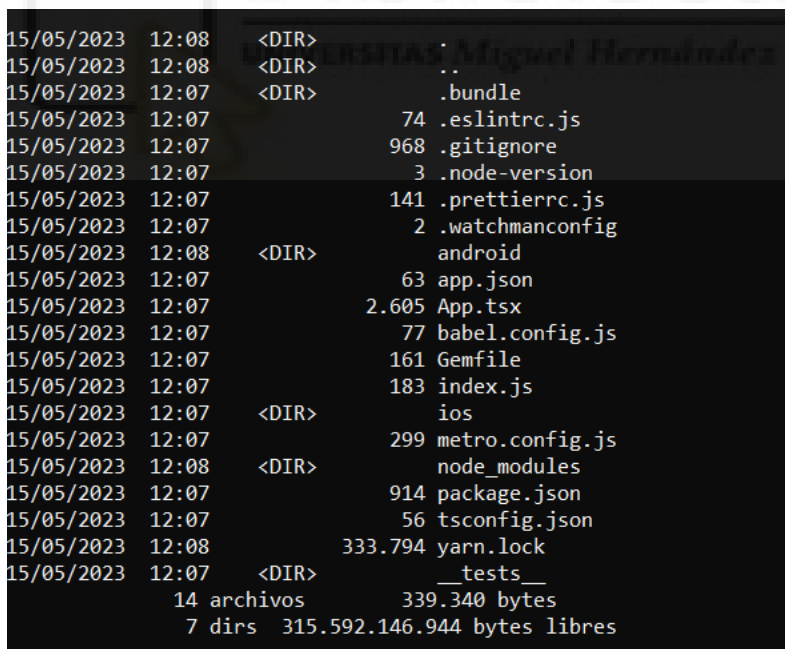


Figura 4.15. Ficheros del proyecto

De los ficheros y carpetas creados podemos mencionar dos partes importantes que forman parte del proyecto en los que tienen en cuenta a la hora de desarrollar, estos son, *node_modules* y *package.json*:

node_modules: Este es un directorio que se crea en la carpeta raíz de nuestro proyecto, en este directorio se almacenarán todos los paquetes que son instalados mediante el comando npm.

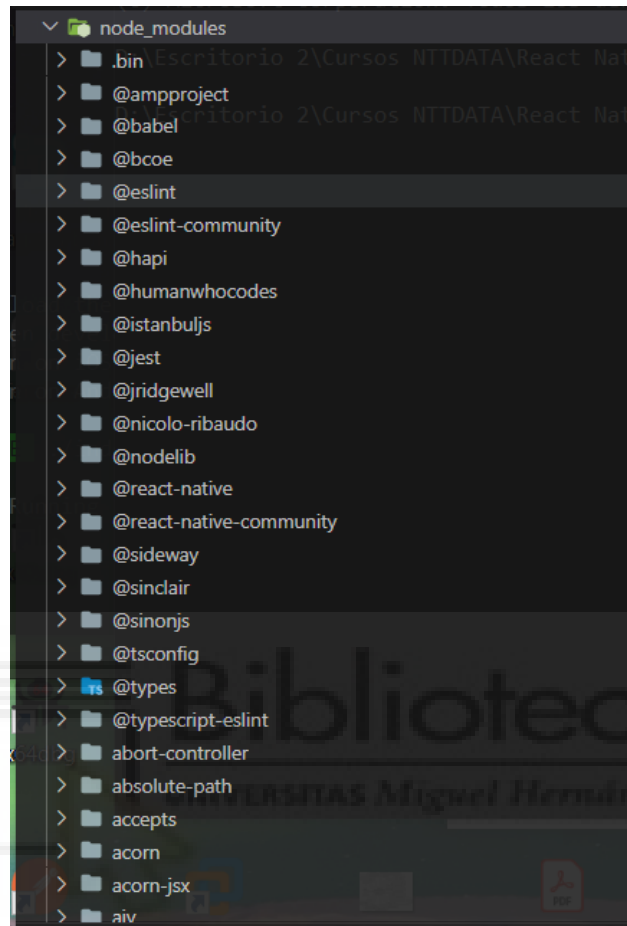


Figura 4.16. Node modules

package.json: Este es un fichero en formato json que también se crea en la raíz de la carpeta de nuestro proyecto. Este fichero contiene atributos como el nombre del proyecto, la versión, comandos para poder ejecutar la aplicación en Android o IOS, la versión de **React Native** y dispone de información de los paquetes instalados.

```

package.json 1 x
package.json > {} devDependencies
1  {
2    "name": "NuevoProyecto",
3    "version": "0.0.1",
4    "private": true,
5    "scripts": {
6      "android": "react-native run-android",
7      "ios": "react-native run-ios",
8      "lint": "eslint .",
9      "start": "react-native start",
10     "test": "jest"
11   },
12   "dependencies": {
13     "react": "18.2.0",
14     "react-native": "0.71.8"
15   },
16   "devDependencies": {
17     "@babel/core": "^7.20.0",
18     "@babel/preset-env": "^7.20.0",
19     "@babel/runtime": "^7.20.0",
20     "react-native-community/eslint-config": "^3.2.0",
21     "@tsconfig/react-native": "^2.0.2",
22     "@types/jest": "^29.2.1",
23     "@types/react": "^18.0.24",
24     "@types/react-test-renderer": "^18.0.0",
25     "babel-jest": "^29.2.1",
26     "eslint": "^8.19.0",
27     "jest": "^29.2.1",
28     "metro-react-native-babel-preset": "0.73.9",
29     "prettier": "^2.4.1",
30     "react-test-renderer": "18.2.0",
31     "typescript": "4.8.4"
32   },
33   "jest": {
34     "preset": "react-native"
35   }
36 }
37

```

Figura 4.17. Package Json

Debemos ingresar en la nueva carpeta creada y ejecutar el siguiente comando

`npx react-native run-android`

```

15/05/2023 11:24 <DIR> .
15/05/2023 11:24 <DIR> ..
25/10/2022 15:51 <DIR> ListaDeContactos
23/01/2023 14:40 <DIR> ListaDeContactosIOS
13/06/2022 17:27 <DIR> My App Images
03/02/2023 20:24 <DIR> MyApp
15/03/2023 11:10 2.115.091 MyApp Documentation.docx
15/05/2023 11:25 <DIR> NuevoProyecto
1 archivos 2.115.091 bytes
7 dirs 6.745.141.248 bytes libres
C:\Users\usuario2020\Escritorio\TFG\React - Native>

```

Figura 4.18. Proyecto de prueba

En este caso se ha procedido a abrir el proyecto con *Visual Code*, para ver la estructura del proyecto. En este caso se puede trabajar a partir de este, borrando algunos ficheros innecesarios para posteriormente crearlos desde cero y agregar lo que realmente nos interesa y lo necesario.

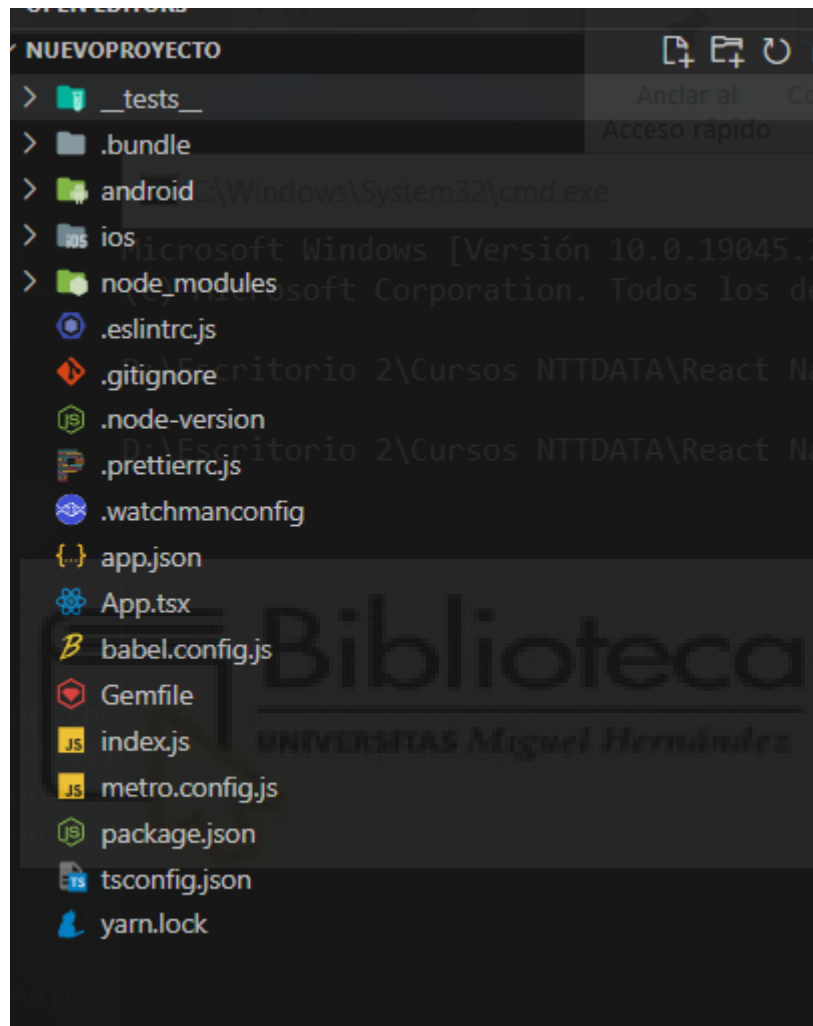


Figura 4.21. Estructura de una aplicación en React Native

4.4.1.2. Screens (pantallas en React Native)

Para comprender un poco como se ha desarrollado la parte del usuario al realizar un registro, vamos a ver como se ha estructurado la aplicación Android.

El primer lugar tenemos que hablar de las pantallas de las cuales se compone la aplicación Android.

Estás están definidas de la siguiente manera, tenemos un nombre de la carpeta *padre* con nombre **screens** y dentro de estas tenemos sus *hijas* con los nombres que describen el tipo de pantalla. Con las screens podremos navegar de una hacia otra haciendo uso de la librería *useNavigation* [37]. Los componentes se encuentran dentro de las screens, es decir, una screen se compone de varios componentes o de uno. En la siguiente imagen se puede observar una parte de las screens de la aplicación Android.

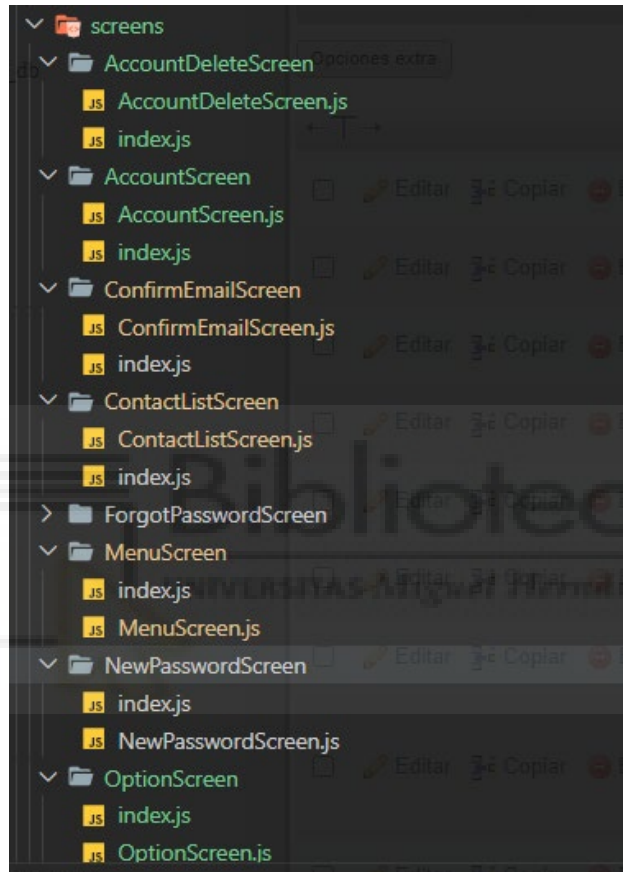


Figura 4.22. Estructura de las screens

Para hacernos una idea de la aplicación, si tenemos en cuenta la parte de registro podemos observar las siguientes imágenes, la primera tiene como screen la parte de inicio de sesión y otra de registro. En la imagen que sigue a continuación de la primera, hace referencia a una parte de la lógica de una screen en la que tenemos unas variables con una función (Hook) que permitirá modificar el valor de estas.

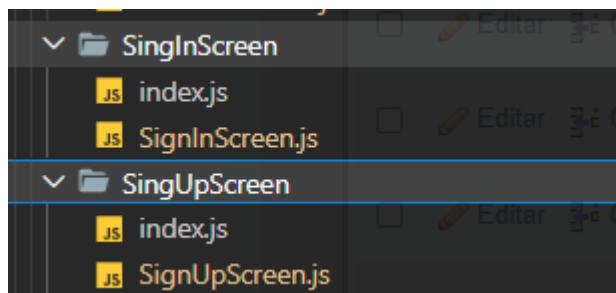


Figura 4.23. Screens

```
/** Datos del usuario para el registro */
const [state, setState] = useState({
  telefono: '',
  email: '',
  alias: '',
  password: '',
  passwordRepeat: ''
})
```

Figura 4.24. Hook Use State en SignUpScreen

Para poder solicitar un recurso a un servidor, en este caso se ha optado por utilizar la librería de *axios* [38]. En esta parte del código tenemos una petición hacia el servidor en el que se le pasa un email en la cabecera para poder comprobar el registro del usuario con un correo que aún no está registrado. En la respuesta obtenemos que, dentro de *axios* tenemos agregados unos condicionales que nos permiten verificar si un correo ya está registrado. En este caso, mientras se desarrollaba la aplicación los errores se verificaban en la terminal del ordenador que se mantiene abierta mientras la aplicación está en ejecución.

```

if (validarEmail(state.email) || state.email.length!=0){
  await axios.get(url_base+'/api/emails/'+state.email, config_json_header)
  .then(res => {
    if (res.status==200) {
      if (!res.data.status) {
        //Error, correo registrado
        setErrEmail('Email ya registrado. ');
        console.log('Email ya registrado')
        return;
      }else{
        console.log('Email válido')
        setErrEmail('');
        validEmail = 'v';
      }
    }
    setLoading(true);
    return;
  })
  .catch(err => console.log(err))
  .then(() =>{
    setLoading(true);
    return;
  })
}

```

Figura 4.25. Axios en React Native

4.4.1.3. Componentes

Los componentes son una parte muy importante del desarrollo, la reutilización de varios de estos ha permitido agilizar el desarrollo. Estos se han estructurado de la siguiente manera. A partir de una carpeta **padre** con nombre *components* se ha creado varios tipos de componentes los cuales tienen como nombre el tipo de componente.

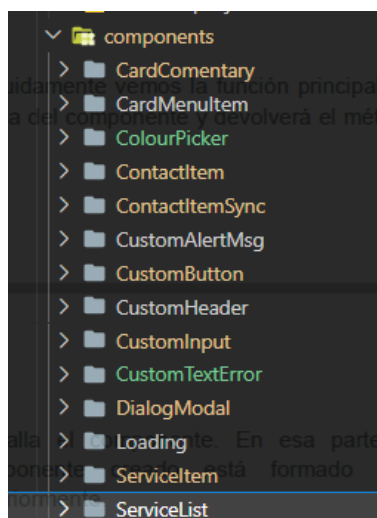


Figura 4.26. Directorio components

4.4.1.4. Librería para obtener contactos del móvil

La librería contacts, nos ha permitido obtener todos los contactos que tenemos disponibles en nuestro dispositivo. Cabe mencionar que para instalar alguna librería se requiere del siguiente comando, *npm install nombre_librería/paquete*.

```
import Contacts from 'react-native-contacts';
```

Figura 4.27. Librería Contacts

En la **figura 4.29** nos describe como obtener los contactos y almacenarlos en una variable.

- En primer lugar comprobamos que nos encontramos con un sistema operativo Android.
- Después tenemos que tener en cuenta los permisos de nuestro dispositivo, en este caso dentro del try tenemos que llamar a API (se instalada automáticamente al crear un proyecto en React Native) de nombre *PermissionsAndroid* y llamar a una función de con el nombre de **READ_CONTACTS**. En esta parte le agregamos un mensaje personalizado preguntando por los permisos.
- Cuando se acepta el mensaje que le aparecerá por pantalla al usuario, tenemos un segundo condicional que comprueba que se ha permitido el acceso a los contactos y es aquí donde se almacenan los contactos en la variable contacts mediante el *hook state* **setContacts()**.

```
const [contacts, setContacts] = useState([]);
```

Figura 4.28. Hook useState

```

if (Platform.OS === "android") {
  try {
    const granted = await PermissionsAndroid.request(
      PermissionsAndroid.PERMISSIONS.READ_CONTACTS,
      {
        title: "Vos App Permisos De Contactos",
        message:
          "Vos App necesita acceder a tus contactos " +
          "para que puedas disfrutar de la aplicación.",
        buttonNeutral: "Preguntar más tarde",
        buttonNegative: "Cancelar",
        buttonPositive: "OK"
      }
    );
    if (granted === PermissionsAndroid.RESULTS.GRANTED) {
      Contacts.getAll()
        .then((contacts) => {
          //Almacenamos los contactos de nuestro móvil en un array
          setContacts(contacts);
        });
    }
  }
}

```

Figura 4.29. Permisos en Android

4.4.1.5. Contactos

Para mostrar el listado de los contactos lo que se ha procedido es que a partir del código perteneciente a la imagen número, una vez almacenado el objeto de datos en la variable **contacts**, se pasa a implementar un componente denominado *FlatList*, este nos permite listar datos a partir de un objeto con datos. En este caso tenemos un objeto de datos almacenado en la variable **contacts**. En este primer componente de tipo lista nos va a permitir mostrar los contactos registrados, a esto hay que tener en cuenta una serie de condiciones para cumplir este objetivo.

En este mismo *FlatList* nos permite agregar otro *FlatList*, este último mediante una serie de condiciones nos mostrará solamente los contactos que **no** están registrados en la base de datos.

```
return (  
  <View style={styles.container}>  
    <SafeAreaView style={{flex:2}}>  
      <FlatList  
        data={contacts}  
        renderItem={renderItem}  
        keyExtractor={FunctionHelper.keyExtractor}  
        ListFooterComponentStyle={{marginTop:15}}  
        ListFooterComponent={  
          <FlatList  
            ListHeaderComponent={<Text style={styles.titlecontactsnosync}>Invitar a AppVol</Text>}  
            data={contacts}  
            renderItem={renderItem3}  
            keyExtractor={FunctionHelper.keyExtractor}  
          </FlatList>  
        }  
      </FlatList>  
    </SafeAreaView>  
  </View style={styles.botonsincronizar}>  
)
```

Figura 4.30. Componente flatlist

4.4.1.6. Servicios

Para poder listar los servicios que previamente se hayan dado de alta por un prestador de servicios, se ha realizado casi la misma implementación que en la de los contactos. En este caso solamente se ha utilizado un *FlatList*. Este nos ha permitido mostrar los servicios mediante un objeto de datos con todos los servicios disponibles.

Primero se realiza una petición al servidor mediante un *hook* personalizado que nos permite almacenar los datos a efecto de la petición, en la variable con nombre, *data*.

```
const {data, loading, error} = useFetchGet(`${url_base}/api/json`);
```

Figura 4.31. Hook personalizado

La siguiente imagen corresponde a una llamada de API. Este fragmento de código nos indica que en la parte superior izquierda se almacenarán los datos en las variables a partir de la información recibida por parte de la API. En el código que sigue a continuación nos permite trabajar con la información recibida para almacenarla de las variables correspondientes, en caso de error, mostrará un error en pantalla.

```
const [data, setData] = useState([]);
const [loading, setLoading] = useState(false);
const [error, setError] = useState(null);

useEffect(() => {
  const getData = async () => {
    await axios.get(url, config_json_header)
      .then(res => {
        if (res.status==200) {
          setData(res.data);
          setLoading(true);
          return;
        }
        else{
          setLoading(true);
          throw new Error("Falló el fetch de servicios");
        }
      })
      .catch(err=> {alert(err);setLoading(true)})
    getData();
  }, [url]); // Con [data] obtenemos el cambio al instante
```

Figura 4.32. Llamada a la API Servicios

4.4.2. Aplicación móvil - backend

La implementación de código haciendo uso del framework Laravel, se ha procedido de la siguiente forma.

Cada entidad o miembro participante en la aplicación desarrollada, tiene definidas sus *rut*as, *modelos* y *controladores*, estas son las que podemos destacar, porque serán necesarias para poder llevar a cabo el desarrollo y la interacción de la aplicación Android. En este caso vamos a comentar lo que se ha desarrollado o aplicado en la aplicación, teniendo en cuenta al **usuario**, **servicios** y **contactos**.

4.4.2.1. Usuario

Este es el usuario registrado que tiene definidas sus rutas (routes) que serán necesarias para que el cliente (React Native) pueda hacer peticiones a las API correspondientes. Como se puede comprobar en la imagen también disponemos del controlador correspondiente.

- **register**: mediante esta funcionalidad el usuario podrá registrarse en la aplicación.

- **login:** esta funcionalidad permite al usuario realizar un login.
- **show:** esta funcionalidad permite mostrar el alias y email del usuario, teniendo en cuenta el número de teléfono del usuario.
- **getemails:** esta funcionalidad nos permite comprobar si existe un correo, teniendo en cuenta un correo.
- **update:** esta funcionalidad permite actualizar los datos del usuario teniendo en cuenta el número móvil.

```
//Rutas para React Native(app-vol)
Route::post('register', [UserController::class, 'register']);
Route::post('login', [UserController::class, 'login']);
Route::get('show/{telefono}', [UserController::class, 'show']);
Route::get('emails/{email}', [UserController::class, 'getemails']);
Route::put('update/{telefono}', [UserController::class, 'update']);
```

Figura 4.33. Rutas para React Native

4.4.2.2. Servicios

En la parte de servicios tenemos los controladores que disponen las siguientes funcionalidades.

- **index:** esta funcionalidad nos permite obtener un objeto json con todos los servicios.
- **getAvg:** Esta funcionalidad nos permite obtener la valoración media de nuestros contactos a partir del número móvil y de un determinado servicio.

```
//Ruta Api de servicios
Route::get('/json', [ServicioDatosController::class, 'index']); // Objeto con los datos de los servicios
Route::get('/valoracion/{telefono}/{id_serv}', [ServicioDatosController::class, 'getAvg']);
```

Figura 4.34. Rutas Api de Servicios

4.4.2.3. Contactos

En la parte de contactos tenemos los controladores que disponen las siguientes funcionalidades.

- **syncAllContactsOnce:** esta funcionalidad nos permitirá sincronizar todos los contactos que estén registrados en la aplicación, cuando se le da los permisos en el dispositivo móvil.
- **syncContacts:** esta funcionalidad nos permite sincronizar un determinado contacto.

- **deleteContacts:** esta funcionalidad nos permite desincronizar los contactos o contacto.
- **showContacts:** esta funcionalidad permite mostrar los contactos sincronizados.

```
//Ruta Contacto
Route::post('contactosync/{telefono}', [ContactoObtenidoController::class, 'syncAllContactsOnce']);
Route::post('contacto', [ContactoObtenidoController::class, 'syncContacts']);
Route::post('contacto/{telefono}', [ContactoObtenidoController::class, 'deleteContacts']);
Route::get('contacto/{telefono}', [ContactoObtenidoController::class, 'showContacts']);
```

Figura 4.35. Rutas de contacto

4.4.3. Aplicación web - frontend

La interfaz de usuario se ha creado a partir del motor *Blade* de Laravel [39]. Esta característica de **Laravel** nos permite crear aplicaciones reutilizando código y agregando una serie de funcionalidades como agregar condicionales, ya que de esta manera nos permite agilizar el desarrollo. En este caso en la interfaz de administrador y del prestador de servicios se ha procedido a reutilizar el menú de navegación.

4.4.3.1. Estructura del administrador

La estructura perteneciente a la parte backend relacionada con las vistas en Laravel se ha realizado como la siguiente imagen (Figura 4.36). Tenemos un directorio **admin**, el cual tiene dentro dos carpetas con sus respectivos ficheros *blade*. Este tipo de ficheros se han creado siguiendo la lógica: create, edit, index (el fichero principal) y show.

- **create:** Esta vista te permite crear un servicio o tipo de servicio.
- **edit:** Esta vista te permite editar un servicio o tipo de servicio.
- **index:** Esta vista mostrará la página inicial con los servicios o tipos de servicios.
- **show:** Esta vista mostrará un determinado servicio o tipo de servicio.

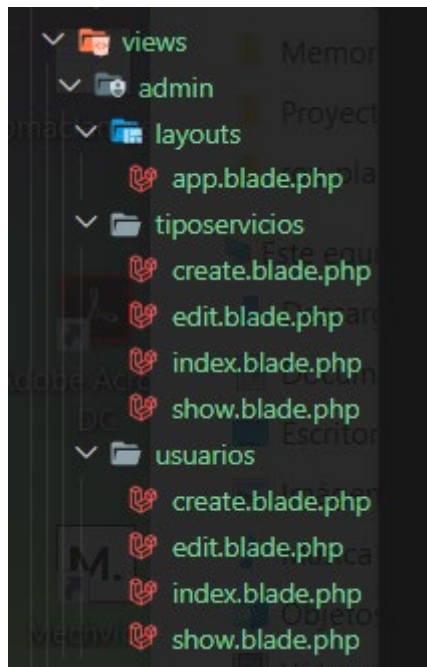


Figura 4.36. Rutas de contacto

4.4.3.2. Código IU administrador

La imagen que se muestra a continuación corresponde a una parte del código perteneciente a la parte de la interfaz de usuario (IU) en el que mediante el etiquetado de *HTML* y las clases de *Bootstrap*, se ha conseguido crear la interfaz del administrador. También se ha aplicado algunos condicionales para mostrar datos. Este código corresponde a la información en ventana de los prestadores de servicios dados de alta por el administrador y de los que se han registrado. Cabe mencionar que para la interfaz de tipo de servicios se ha procedido el desarrollo de la misma manera pero agregando sus características.

```

<div>
@if ((count($usuarios))>=1)
<div class="row pt-4">
<table class="table table-hover table-bordered">


Figura 4.37. Código de la interfaz del administrador


```

4.4.3.3. Código IU prestador de servicios

La siguiente imagen pertenece al código del prestador de servicios en el que se ha utilizado *html* y las clases de *bootstrap* para darle los estilos. También se ha utilizado las propiedades de *blade* para acceder a las variables en PHP.

```

@if ((count($servicios))>=1)
<div class="row pt-4">
<table class="table table-hover table-bordered">

```

Figura 4.38. Código de la interfaz de los prestadores de servicios

4.4.4. Aplicación web - backend

Para el administrador tenemos realizada la administración de rutas como se muestra en la siguiente imagen (Figura 4.39). En este caso como se trata del administrador se ha utilizado un middleware que permite verificar primeramente la autenticación y que es administrador. De esta forma nos aseguramos que como **administrador** pueda ingresar a las rutas de usuarios y tipos de servicios.

- La ruta **usuarios** está relacionada con el controlador **AdminController**, el cual dispone de la funcionalidad necesaria para poder realizar un CRUD [40] (create, read, update y delete) sobre los usuarios prestadores de servicios.
- La ruta **tiposervicios** está relacionada con el controlador **TipoServicioController**, el cual dispone de la funcionalidad que nos permitirá crear tipos de servicios y editarlos.

```
// Admin Routes
Route::prefix('admin')->middleware(['auth', 'isAdmin'])->group(function(){
    Route::resource('usuarios', AdminController::class);
    Route::resource('tiposervicios', TipoServicioController::class);
});
```

Figura 4.39. Rutas del administrador

Para que nuestra verificación funcione correctamente, tenemos que agregar las rutas “isAdmin” y “auth” en el fichero **kernel.php** de nuestro proyecto en Laravel, obviamente es una parte de todo el proceso, esta es una parte de las más importantes.

```
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \App\Http\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    'isAdmin' => \App\Http\Middleware\AdminMiddleware::class,
];
```

Figura 4.40. Rutas middleware

Con el prestador de servicios tenemos una ruta en el que se ha aplicado una funcionalidad para que obligadamente deba realizar una autenticación para poder dar de alta, editar o

eliminar algún servicio, es decir que debe estar “logueado” para poder acceder a la ruta servicios y poder realizar alguna operación.

```
Route::resource('servicios', ServicioController::class)->middleware('auth');
```

Figura 4.41. Ruta servicios del prestador de servicios

4.5. RESULTADOS

4.5.1. Aplicación móvil

4.5.1.1. Crear cuenta

En esta pantalla tenemos la opción de crear una cuenta, en este caso se ha procedido a utilizar un número no existente, es uno inventado. Tenemos los campos necesarios para poder crear una cuenta en la aplicación. Si se intenta registrar un correo ya registrado en la aplicación, mostrará un mensaje de error. Si el password no cumple los requisitos mínimos, aparecerá un error. En caso de que las contraseñas no coincidan, aparecerá un error.

Solo emergen... 45% 11:39

←

Crear una cuenta

677677677

Alias

Email

i

Password

Repetir Password

Registrar

Iniciar Sesión

Figura 4.42. Crear cuenta en VOS app

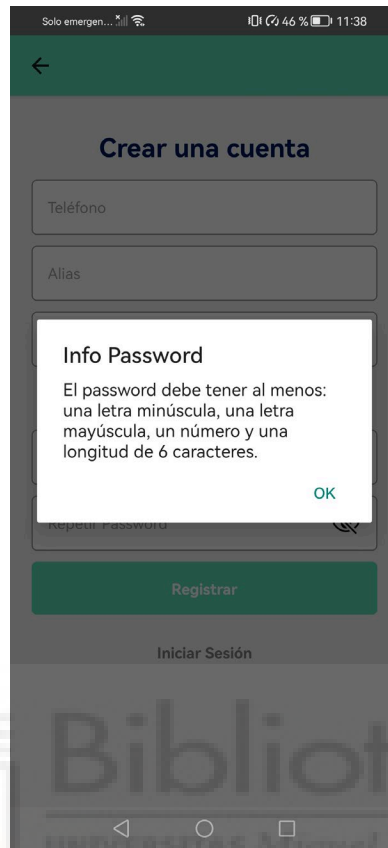


Figura 4.43. Información password en VOS app

4.5.1.2. Inicio de sesión

Esta pantalla nos permite ingresar a las opciones de la aplicación mediante una autenticación. Necesitamos el número de teléfono de la sim card, pero en este caso como es un prototipo se ha procedido a utilizar un número inventado.

También será necesario el *password* para poder acceder y si el *password* no es correcto, se mostrará un error. Tenemos un icono con un ojo para mostrar u ocultar el *password* introducido. La funcionalidad de *olvidó su contraseña* no está implementada, es meramente informativa.

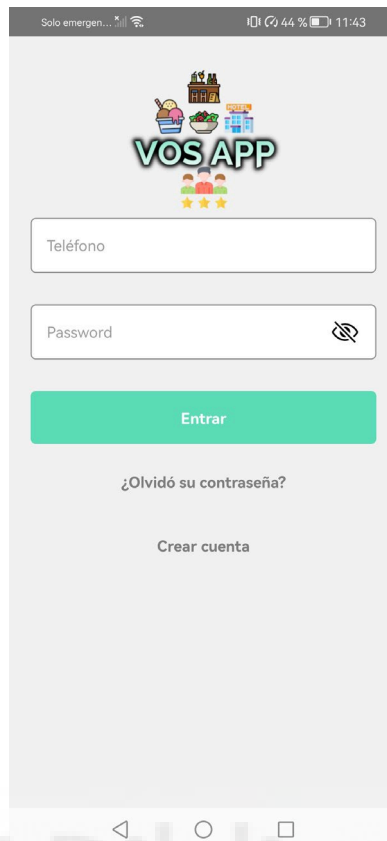


Figura 4.44. Inicio de sesión en VOS app

4.5.1.3. Menú de la aplicación

En esta pantalla tenemos el menú de la aplicación y como se puede ver en la imagen este menú dispondrá de cuatro iconos con sus respectivos nombres y de un botón para cerrar sesión. En la aplicación disponemos de las opciones que nos permitirán ingresar al tocar el icono correspondiente. Este menú de ha dividido de la siguiente manera:

- **Servicios:** Todo lo relacionado con los servicios. Cuando se elige un servicio podemos ver toda la información acerca de este además de poder dejar una opinión y valoración. También se podrá ver los comentarios de nuestros contactos sincronizados y de los contactos de estos.
- **Contactos:** Esta parte nos muestra todos los contactos que tenemos disponibles en nuestro dispositivo móvil junto a las características de cada contacto.
- **Opciones:** Esta opción nos permite ingresar a una funcionalidad que nos permite cambiar el color de las cards de los contactos con sus comentarios realizados en un determinado servicio.
- **Cuenta:** Esta opción tiene dos opciones cuando ingresas, la primera te permite actualizar datos y la otra eliminar cuenta.

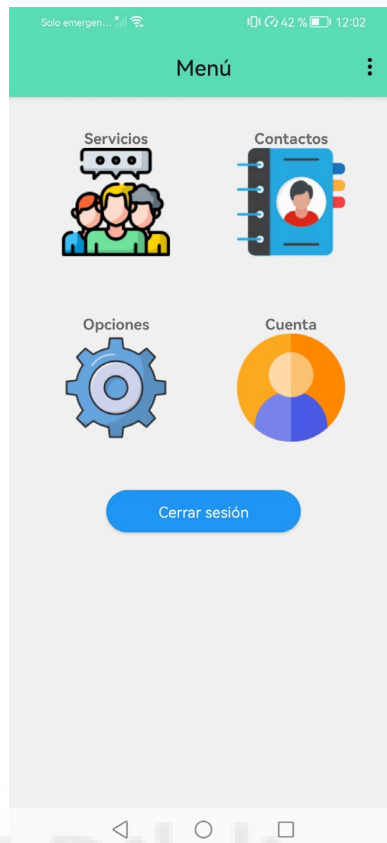


Figura 4.45. Menú en VOS app

4.5.1.4. Cuenta de usuario

Esta pantalla tiene dos opciones, la primera hace referencia a una parte de los datos de un usuario en el que podremos actualizar/modificar el correo y alias.

La segunda opción hace referencia a la eliminación de la cuenta del usuario.

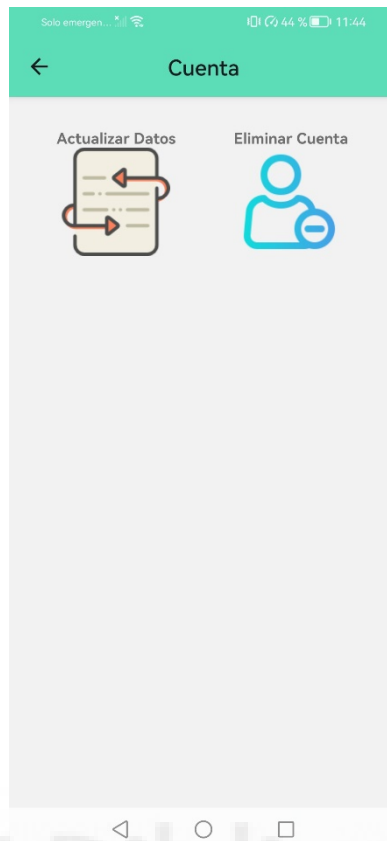


Figura 4.46. Opción cuenta en VOS app

4.5.1.5. Opciones

Esta pantalla nos va a permitir cambiar de color las cards de los comentarios de nuestros contactos y de la red de contactos de estos.

Tenemos una paleta de colores para elegir el que más nos guste y cuando lo tengamos elegido debemos pulsar en el botón *Cambiar Color*.

Al ir seleccionando los colores se verá reflejado en la card de prueba de la parte superior, para comprobar y ver el color que más nos guste.

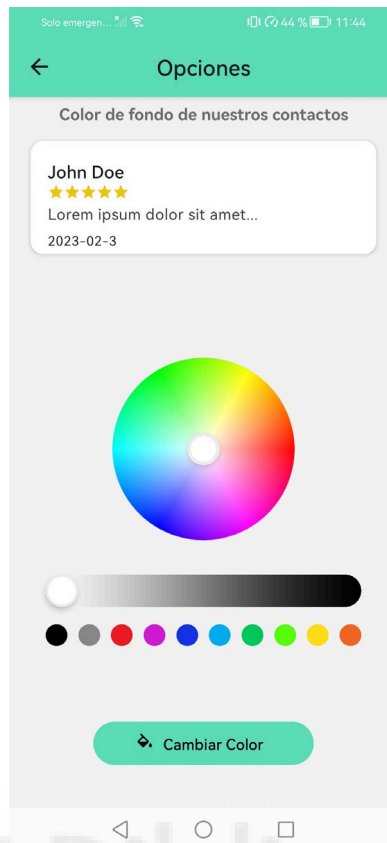


Figura 4.47. Opción cambiar color de las cards en VOS app

4.5.1.6. Contactos

En esta pantalla nos va a permitir mostrar los contactos que tenemos disponibles en nuestro dispositivo móvil. La lista la podemos dividir en dos, en la primera lista se encuentra en las primeras posiciones corresponden a los contactos que se encuentran sincronizados y se pueden comprobar mediante un check. La segunda hace referencia a los contactos que no se encuentran registrados en la aplicación y por lo tanto, no pueden ser sincronizados, estos tienen un botón de invitación que no tiene ninguna funcionalidad.

Los números que se encuentran en pantalla son inventados, la coincidencia en la vida real es puramente casual.

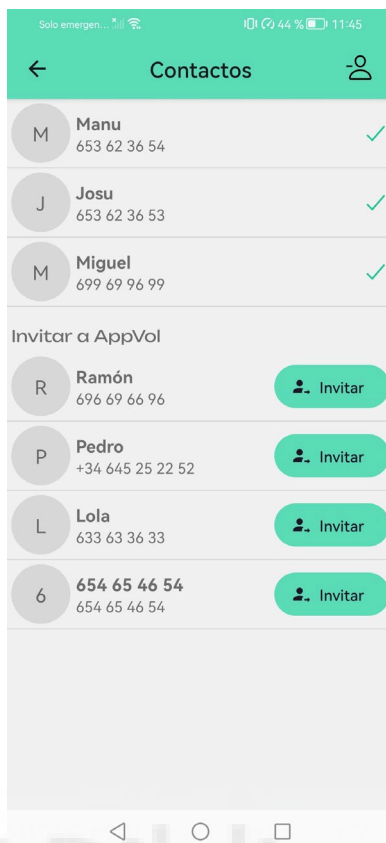


Figura 4.48. Contactos en VOS app

4.5.1.7. Lista de servicios

En la pantalla de servicios tenemos un listado con todos los servicios que se encuentran dados de alta por un prestador de servicios. En este caso tenemos varios servicios de varios tipos, restaurante, suministro eléctrico, heladería y panadería. En esta pantalla se podría haber agregado algún tipo de filtro, por ejemplo, ordenarlo por orden alfabético, como es un prototipo se ha optado por hacerlo de manera simple.

Cuando se pulse en un servicio se abrirá otra pantalla y mostrará los detalles de dicho servicio además de las reseñas de nuestros contactos y de los contactos de estos.

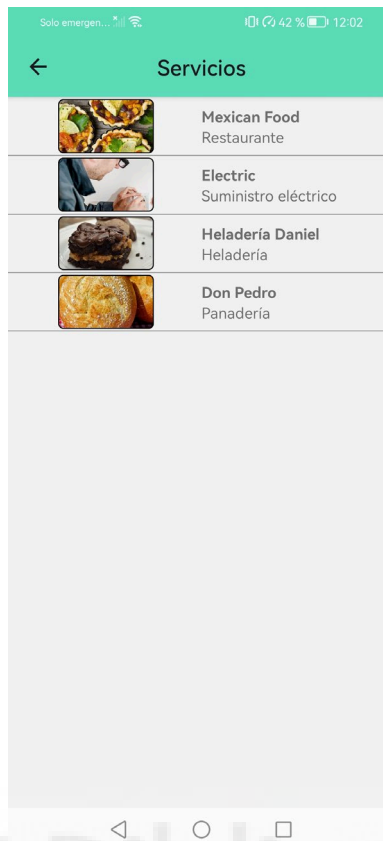


Figura 4.49. Servicios en VOS app

4.5.1.8. Servicio

Esta pantalla tiene los detalles de un determinado servicio, este servicio tiene la siguiente información:

- **Nombre del servicio:** Será el nombre del servicio dado de alta.
- **Imágenes:** Son las imágenes descriptivas del servicio, se pueden ver deslizando con el dedo hacia la derecha o izquierda de la pantalla.
- **Descripción:** La descripción de servicio, en la primera línea de la información a continuación de las imágenes.
- **Valoración global:** Es la valoración que se calculará a partir de todos los usuarios registrados en la aplicación y que hayan dejado una valoración.
- **Valoración contactos:** Lo mismo que lo comentado en el punto anterior pero en este caso se tiene en cuenta las valoraciones de nuestros contactos sincronizados.
- **Horario:** Este horario es informativo.
- **Valorar servicio:** Tenemos la opción de abrir una ventana para poder valorar el servicio.

- **Comentarios:** La parte de los comentarios en los que se puede ver reflejado nuestro comentario y los comentarios de los contactos sincronizados.

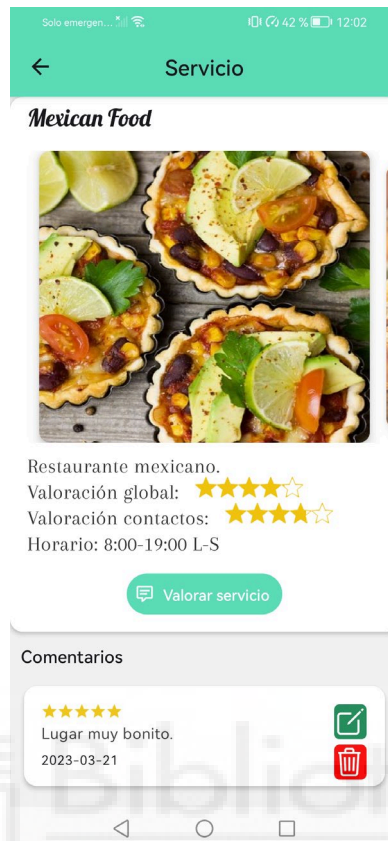


Figura 4.50. Servicio con imagen descriptiva en VOS app

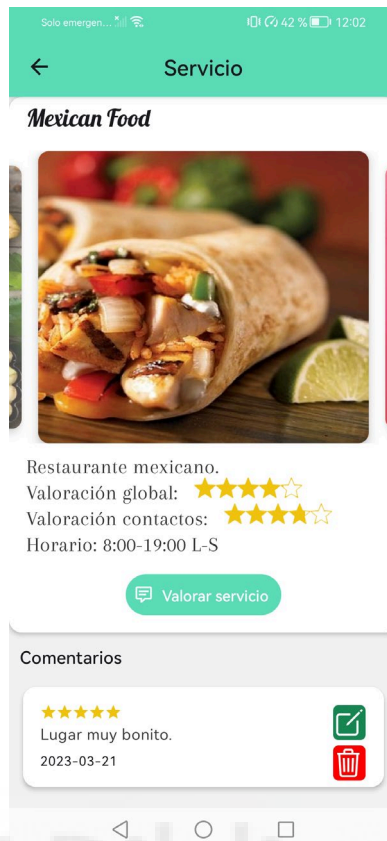


Figura 4.51. Servicio con imagen descriptiva en VOS app

En la siguiente imagen podemos ver la información acerca de los comentarios y valoraciones realizadas de los contactos. Si nos fijamos en la card de un contacto podemos tocar el icono con forma de **ojo** para ver información acerca de un contacto, dicha información corresponde con el *nombre del contacto* y su *número telefónico*.

Cada card a excepción de la nuestra, mostrará únicamente el alias y la información correspondiente. Como usuario registrado podemos ver las opciones de editar (icono de color verde) y eliminar (icono de color rojo) para poder realizar alguna acción correspondiente a uno de nuestros comentarios.



Figura 4.52. Comentarios en VOS app

En esta última parte nos vamos a fijar en los comentarios y valoraciones de los contactos de nuestros contactos. En este caso la card nos muestra únicamente el alias de los usuarios, su valoración y comentario. Por lo tanto, estos usuarios pertenecerán a un contacto que lo tengamos agregado en nuestros contactos y a su vez que se haya registrado en la aplicación.



Figura 4.53. Comentarios en VOS app

4.5.2. Aplicación web

4.5.2.1. Panel de prestadores de servicios - administrador

La siguiente imagen (Figura 4.54) tiene como objetivo mostrar la información acerca de los usuarios prestadores de servicios dados de alta por parte del administrador. El panel muestra un listado con todos los usuarios dados de alta y en ese listado cada usuario dispone de una opción en la parte derecha con unos botones de color azul claro, verde y rojo. Esos botones se aplican para mostrar (botón azul claro) un usuario con sus datos, editarlo (botón verde) y eliminarlo (botón rojo).

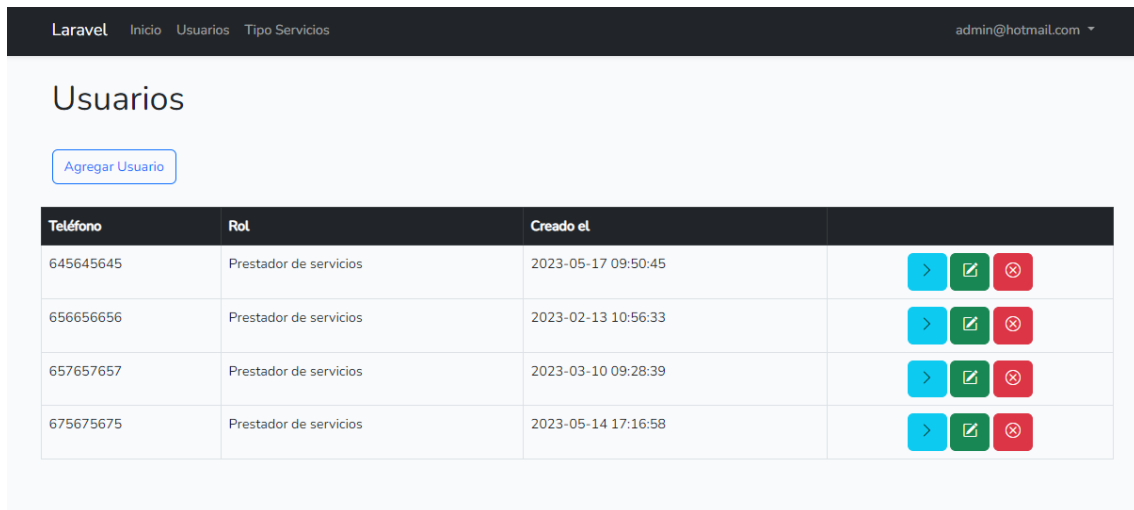


Figura 4.54. Panel del administrador

Si queremos agregar un nuevo usuario prestador de servicios debemos pulsar en el botón *Agregar usuario* que se encuentra en la parte superior izquierda del listado de los usuarios prestadores de servicios.

Una vez ingresamos en la ventana como se ve en la siguiente imagen, introducimos los datos y le damos a crear, para dar de alta un nuevo prestador de servicios.

Figura 4.55. Crear usuario prestador de servicios

4.5.2.2. Pantalla tipo de servicios - administrador

La siguiente imagen (Figura 4.56), muestra los tipos de servicios dados de alta y en el que solamente disponemos de una opción, esta nos sirve para editar. En este caso se ha obviado por agregar los botones de eliminar o mostrar un tipo de servicio. La opción

mostrar un servicio no está porque en este caso tenemos pocos datos para mostrar y la opción de eliminar no está porque obligadamente para que un usuario prestador de servicios pueda dar de alta un servicio, debe tener una variedad de tipos de servicios para poder seleccionarlos.

id	Tipo	Creado el	
2	Restaurante	2023-02-13 11:58:38	
3	Heladería	2023-03-09 19:30:47	
8	Energía	2023-03-10 09:37:31	
9	Suministro eléctrico	2023-05-17 09:47:45	
10	Panadería	2023-05-17 09:54:24	

Figura 4.56. Listado del tipo de servicios

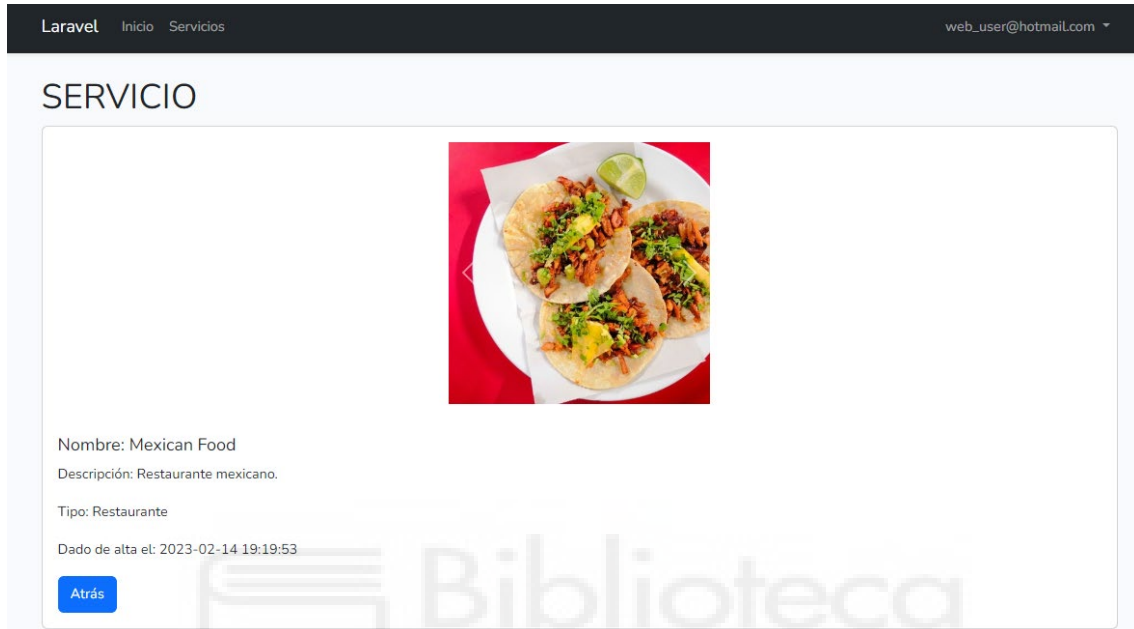
4.5.2.3. Pantalla de servicios - prestador de servicios

En el panel del prestador de servicios tenemos un listado de todos los servicios agregados, este cuenta también con tres botones de color azul claro (mostrar servicio), verde (editar servicio) y rojo (eliminar servicio). El prestador de servicios está implementado para agregar más de un servicio.

#	Nombre Local	tipo	Descripción	Creado el	
2	Mexican Food	Restaurante	Restaurante mexicano.	2023-02-14 19:19:53	

Figura 4.57. Listado de servicios

La siguiente imagen corresponde a los datos de un servicio previamente dado de alta por un prestador de servicios. Este servicio muestra unas imágenes las cuales se pueden deslizar a la derecha o izquierda mediante los botones. En la información muestra el nombre del servicio, la descripción, el tipo de servicio y la fecha en el que fue dado de alta.



The screenshot shows a web application interface. At the top, there is a dark navigation bar with the text 'Laravel Inicio Servicios' on the left and 'web_user@hotmail.com' on the right. Below the navigation bar, the main content area has a light blue background with the heading 'SERVICIO'. A large white-bordered box contains a central image of three tacos on a white plate with a red background. Below the image, the following text is displayed: 'Nombre: Mexican Food', 'Descripción: Restaurante mexicano.', 'Tipo: Restaurante', and 'Dado de alta el: 2023-02-14 19:19:53'. At the bottom left of this box is a blue button labeled 'Atrás'. A large, semi-transparent watermark 'Biblioteca' is overlaid on the right side of the page.

Figura 4.58. Información acerca de un servicio

Capítulo 5

Conclusiones y Trabajo futuro

5.1. CONCLUSIONES

En este proyecto se ha logrado completar con éxito los distintos puntos que se había planteado al inicio de este, realizando los puntos de forma ordenada para poder obtener los resultados esperados. Empezando por el diseño de base de datos, prototipo y pasando por el desarrollo en la parte del frontend y backend, en la aplicación Android y la aplicación web. Se ha logrado afianzar mucho más los conocimientos que ya previamente se tenía sobre las tecnologías mencionadas.

Una especial mención a **React Native**, porque esta tecnología ha sido nueva en el aprendizaje personal y ha servido para el desarrollo de la aplicación Android denominada VOS app. Ha sido una nueva manera de realizar el desarrollo de una aplicación para un dispositivo móvil, teniendo en cuenta todas las ventajas que dicha tecnología y la manera en el que se ha llevado a cabo el desarrollo. La parte en la que se desarrolla la aplicación y ver los resultados en tiempo real, es una de las grandes ventajas que tiene **React Native**, permitiendo corregir algún tipo de error con el editor de código de una manera más rápida.

La parte del proyecto, aplicación Android y aplicación web, ha sido desarrollada por una sola persona en la cual se ha tenido en cuenta cosas muy básicas para el desarrollo y en no complicar demasiado su desarrollo. La funcionalidad y características de VOS son una parte de lo que podría llegar a ser la aplicación si se trabajase en un equipo de desarrollo.

La parte que más trabajo ha llevado es el desarrollo de VOS app, puesto que esta dispone de muchas más funcionalidades y características que la aplicación web. En VOS y en la aplicación web se ha tenido en cuenta el desarrollo de una interfaz sencilla para que el usuario no tenga problemas en la utilización de ambas.

Con una aplicación Android y una aplicación web siendo ambas dependientes se ha conseguido un proyecto bastante novedoso, sobre todo en la parte de la aplicación Android.

5.2. TRABAJO FUTURO

Centrándonos en la aplicación VOS, se ha pensado en que puede agregar o disponer de muchas más funcionalidades, para que de alguna forma la aplicación sea mucha más completa. De esta forma puede ser mucho más beneficiosa para el usuario y sería más cómoda su interacción.

De las nuevas características o funcionalidades que se pueden agregar se ha tenido en cuenta las siguientes:

- Agregar la opción de geolocalización en la búsqueda de servicios para tener disponibles los servicios más cercanos.
- Tener una opción de llamada directa al seleccionar una opción en un comentario de un contacto y poder hablar con el mediante un sistema de llamadas integrado a la aplicación.

- En la parte de servicios, listar estos mediante diferentes filtros, como el de ciudad, posición (activar ubicación), el tipo de servicio, etc.
- En las cards, haciendo referencia a nuestros comentarios poder agregar una edición personalizada para darle un mejor aspecto y que cada card se distinga del resto.
- Agregar fotos a los comentarios en un determinado servicio.
- Colocar un botón de likes en la card de comentarios.



Bibliografía

- [1] Reseñas en internet
<https://pzt.es/conoce-la-importancia-de-las-resenas/>
20/04/2023
- [2] Reseñas falsas en internet
<https://www.newtral.es/resenas-falsas-online/20221026/>
20/04/2023
- [3] Google maps
<https://pzt.es/conoce-la-importancia-de-las-resenas/>
20/04/2023
- [4] FourSquare
<https://aulacm.com/foursquare-para-empresas/#:~:text=FourSquare%20es%20una%20plataforma%20que,acuerdo%20a%20tu%20ubicaci%C3%B3n%20f%C3%ADsica.>
20/04/2023
- [5] TripAdvisor
<https://www.cityexpress.com/blog/importancia-tripadvisor-para-hoteles-y-viajeros#:~:text=Este%20portal%20concentra%20las%20opiniones,atracciones%20que%20van%20a%20visitar.>
20/04/2023
- [6] GIT Ventajas y desventajas
<https://www.campusmvp.es/recursos/post/que-es-git-ventajas-e-inconvenientes-y-por-que-deberias-aprenderlo-bien.aspx>
21/04/2023
- [7] GIT Historia
<https://uniwebsidad.com/libros/pro-git/capitulo-1/una-breve-historia-de-git>
21/04/2023
- [8] Jira
<https://openwebinars.net/blog/que-es-jira/>
21/04/2023
- [9] Jira: Kanban vs Scrum

- <https://www.atlassian.com/es/agile/kanban/kanban-vs-scrum>
21/04/2023
- [10] Jira
<https://withpact.com/ventajas-y-desventajas-de-jira/>
21/04/2023
- [11] Lucidchart
<https://technologyadvice.com/products/lucidchart-reviews/>
24/04/2023
- [12] Lucidchart: Integraciones
https://lucid.app/documents#/integrations-marketplace?folder_id=home
24/04/2023
- [13] Lucidchart: Ventajas y desventajas
<https://ventajas.org/lucidchart/>
24/04/2023
- [14] Figma ventajas y desventajas
<https://recursosparaemprendedores.com/que-es-figma-ventajas-y-desventajas/>
24/04/2023
- [15] Gimp ventajas y desventajas
<https://suclic.com/gimp-caracteristicas-ventajas-y-desventajas/>
24/04/2023
- [16] Html que es
<https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>
24/04/2023
- [17] Html ventajas y desventajas
<https://economipedia.com/definiciones/lenguaje-html.html#:~:text=Ventajas%3A%20Permite%20describir%20hipertexto%2C%20tiene,y%20las%20etiquetas%20son%20limitadas.>
24/04/2023
- [18] React Native que es
<https://openwebinars.net/blog/react-native-que-es-para-que-sirve/>
01/05/2023
- [19] React Native que es
<https://codigofacilito.com/articulos/que-es-react-native>
01/05/2023
- [20] React Native ventajas y desventajas

- <https://blog.back4app.com/es/react-native-ventajas-y-desventajas-reveladas/>
01/05/2023
- [21] Laravel que es
<https://axarnet.es/blog/que-es-laravel>
01/05/2023
- [22] Bootstrap que es
<https://rockcontent.com/es/blog/bootstrap/>
01/05/2023
- [23] Bootstrap versiones
<https://getbootstrap.com/docs/versions/>
01/05/2023
- [24] Bootstrap ventajas y desventajas
<https://soyhorizonte.com/blog/ventajas-y-desventajas-de-usar-bootstrap/>
01/05/2023
- [25] Mysql que es
<https://openwebinars.net/blog/que-es-mysql/>
01/05/2023
- [26] Mysql ventajas y desventajas
<https://codigosql.top/ventajas-y-desventajas-de-mysql/>
01/05/2023
- [27] Apache que es, ventajas y desventajas
<https://www.webempresa.com/hosting/que-es-servidor-apache.html>
01/05/2023
- [28] Apache módulos
<https://httpd.apache.org/docs/2.4/es/mod/>
01/05/2023
- [29] Firebase que es
<https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>
01/05/2023
- [30] Firebase autenticación
<https://droidmentor.com/phone-number-firebase-auth/>
01/05/2023
- [31] Firebase desventajas
https://blog.back4app.com/es/que-es-firebase/#Limitaciones_de_Firebase
01/05/2023

- [32] Node JS
<https://www.lucushost.com/blog/que-es-node-js/>
01/05/2023
- [33] Ciclos de vida de un software
<https://evotic.es/software-a-medida/ciclo-de-vida-del-software/>
01/05/2023
- [34] Cuando usar Jira
<https://www.proyectum.com/sistema/blog/cuando-usar-scrum-en-tus-proyectos/>
01/05/2023
- [35] Modelo Entidad Relación
<https://keepcoding.io/blog/modelo-entidad-relacion/>
01/05/2023
- [36] Node modules
https://lenguajejs.com/npm/administracion/carpeta-node_modules/
01/05/2023
- [37] Use Navigation
<https://reactnavigation.org/>
01/05/2023
- [38] Axios
<https://blog.logrocket.com/using-axios-react-native-manage-api-requests/>
01/05/2023
- [39] Blade Laravel
<https://styde.net/laravel-6-doc-plantillas-blade/>
01/05/2023
- [40] Crud Laravel
<https://blog.hubspot.es/website/que-es-crud>
01/05/2023