



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN INGENIERÍA DE SOFTWARE

Desarrollo de un sitio web de información
turística de Castilla y León

Autor: Pablo Maniega Alonso

Tutor: César González Ferreras



Agradecimientos

Gracias a mi familia por el cariño y paciencia a lo largo de estos años de estudios.

Gracias a mis amigos, por apoyarme en cualquier momento.

Gracias a mis profesores por la formación que me han dado. En especial a César González Ferreras, por ser el tutor de este trabajo y ayudarme en él.

Resumen

El objetivo de este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación, para la búsqueda y asesoramiento de "que hacer" en Castilla y León. Los principales puntos a desarrollar son, un mapa y tabla desde donde ver los diferentes lugares que visitar, y un sistema de favoritos y comentarios en los diferentes sitios.

La aplicación está disponible en <https://turismocyl.herokuapp.com/>

Palabras clave:

Desarrollo Web, Angular, Firebase, Bootstrap, Datos abiertos, Turismo.

Abstract

The main objective of this Degree Final Project is to develop an application, for the search and counseling of "what to do" in Castilla y León. The main functionalities to develop are, a map and a table where you can see the different places to visit, and a favorites and comments system for the different places.

The application is available at <https://turismocyl.herokuapp.com/>

Key words:

Web Development, Angular, Firebase, Bootstrap, Open data, Tourism.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIII
Lista de tablas	XV
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos y alcance	1
1.3. Entorno tecnológico	2
1.4. Que son los datos abiertos	2
2. Planificación y Presupuesto	5
2.1. Metodología del proyecto	5
2.2. Planificación inicial del proyecto	5
2.3. Costes del proyecto	6
3. Análisis	7
3.1. Actores del sistema	7

3.2. Reglas de negocio	8
3.3. Casos de Uso	9
3.3.1. Casos de Uso	9
3.3.2. Especificación de los Casos de Uso	13
3.4. Requisitos de Información	22
3.5. Requisitos No Funcionales	25
3.6. Requisitos Funcionales	26
3.7. Sobre los datos de la Junta	29
4. Diseño	33
4.1. Arquitectura lógica	33
4.2. Arquitectura física	35
4.3. Diseño Base de Datos	35
4.4. Diagramas de secuencia	40
4.5. Diagramas de interfaz de usuario	43
5. Implementación	51
5.1. Herramientas y Tecnologías	51
5.1.1. Herramientas de desarrollo	51
5.1.2. Herramientas de soporte	52
5.1.3. Tecnologías utilizadas	52
5.2. Rendimiento y precios de Firebase y API´s de Google	52
5.2.1. Firebase	53
5.2.2. API´s de Google	53
5.3. Estructura del Proyecto	55
5.4. Implementación	58
5.4.1. Proyecto Angular	58

5.4.2. Proyecto en Firebase	61
5.5. Heroku	66
6. Pruebas	67
6.1. Pruebas de Caja Negra	67
7. Conclusiones	71
8. Posibles futuras mejoras	73
A. Manual de instalación	75
Bibliografía	77

Lista de imágenes

3.1. Jerarquía de los actores	8
3.2. Diagrama Casos de Uso Usuario Anónimo	10
3.3. Diagrama Casos de Uso Usuario Registrado	11
3.4. Diagrama Casos de Uso Gerente	12
4.1. Diagrama Arquitectura Lógica	34
4.2. Diagrama Arquitectura Física	35
4.3. Diagrama de clases	36
4.4. RealTime Database Firebase	39
4.5. Authentication Firebase	40
4.6. Diagrama de secuencia CU-11	41
4.7. Diagrama de secuencia CU-05	42
4.8. Boceto de página de inicio	43
4.9. Boceto de página de mapa	44
4.10. Boceto de página de tabla	45
4.11. Boceto de página de favoritos	46
4.12. Boceto de página de subida de fichero de monumentos	47
4.13. Boceto de PopUp de inicio sesión	48
4.14. Boceto de PopUp de registrarse	49

5.1. Precio de Google Maps JavaScript API	53
5.2. Precio de Google Maps Geocoding API	54
5.3. Limite de Google Maps JavaScript API	54
5.4. Limite de Google Maps Geocoding API	54
5.5. Estructura general del proyecto	56
5.6. Estructura componente Angular ejemplo	57
5.7. Estructura carpeta shared	57
5.8. Instalación Angular	58
5.9. Crear proyecto Angular	59
5.10. Pregunta sobre routing	59
5.11. Pregunta sobre hoja de estilo	59
5.12. Lanzar aplicación Angular	60
5.13. Pregunta sobre compartir datos	60
5.14. Página principal	61
5.15. Página principal Firebase	61
5.16. Crear Aplicación Firebase nombre	62
5.17. Crear Aplicación Firebase Analytics	62
5.18. Crear Aplicación Firebase Ubicación Analytics	63
5.19. Aplicación Firebase creada	63
5.20. Authentication correo	64
5.21. RealTime Database nueva	64
5.22. Nombre aplicación web para Firebase	65
5.23. Código de Firebase a copiar	65

Lista de tablas

1.1. Objetivos	2
3.1. Actores	7
3.2. Reglas de negocio	9
3.3. Casos de Uso Usuario Anónimo	9
3.4. Casos de Uso Usuario Registrado	10
3.5. Casos de Uso Gerente	11
3.6. Especificación Caso de Uso CU-01	13
3.7. Especificación Caso de Uso CU-02	13
3.8. Especificación Caso de Uso CU-03	14
3.9. Especificación Caso de Uso CU-04	14
3.10. Especificación Caso de Uso CU-05	15
3.11. Especificación Caso de Uso CU-06	15
3.12. Especificación Caso de Uso CU-07	16
3.13. Especificación Caso de Uso CU-08	17
3.14. Especificación Caso de Uso CU-10	18
3.15. Especificación Caso de Uso CU-10	18
3.16. Especificación Caso de Uso CU-11	19
3.17. Especificación Caso de Uso CU-12	19
3.18. Especificación Caso de Uso CU-13	20

3.19. Especificación Caso de Uso CU-14	21
3.20. Especificación Caso de Uso CU-15	21
3.21. Especificación Caso de Uso CU-16	22
3.22. Definición de Usuario en DB	23
3.23. Definición de Sitio en DB	24
3.24. Definición de Comentario en DB	25
3.25. RNF Seguridad	25
3.26. RNF Usabilidad	26
3.27. RNF Robustez	26
3.28. RNF Disponibilidad	26
3.29. Esquema del 1º Conjunto de Datos	30
3.30. Esquema del 2º Conjunto de Datos	31
3.31. Esquema del 3º Conjunto de Datos	32
4.1. Tabla Usuario	37
4.2. Tabla Sitio	38
4.3. Tabla Comentario	39

Capítulo 1

Introducción

1.1. Motivación

La finalidad del presente Trabajo de Fin de Grado es, por un lado, la de, utilizando los datos abiertos relacionados con el turismo, dados por la Junta de Castilla y León, realizar una plataforma web donde los usuarios puedan buscar lugares, en Castilla y León, donde realizar actividades de ocio y, a su vez, comentar y puntuar estos lugares para que otros usuarios se hagan una idea sobre que tal está el sitio. También se permite a los usuarios guardarse los sitios que más les guste como favoritos.

Por otro lado, la de aprender más sobre el uso de las herramientas utilizadas para su desarrollo, ya que a mi parecer son de las más utilizadas a día de hoy en las empresas de desarrollo web.

1.2. Objetivos y alcance

Los principales objetivos buscados en el desarrollo de la aplicación son los siguientes:

ID	Objetivo
OB-01	Desarrollo de un sitio web de información turística de Castilla y León.
OB-02	Implementación de un mapa y una tabla, desde los cuales poder buscar y visualizar los sitios.
OB-03	Implementación de un sistema de comentarios y puntuaciones para los sitios.
OB-04	Implementación de un apartado de favoritos donde los usuarios guarden los sitios que quieran.
OB-05	Aumentar los conocimientos adquiridos en la carrera, sobre las tecnologías utilizadas en el trabajo.

Cuadro 1.1: Objetivos

1.3. Entorno tecnológico

Esta aplicación web se ha desarrollado, principalmente utilizando Angular 11, framework de javascript para el desarrollo de aplicaciones web, el cual incluye otras tecnologías como HTML5 o CSS3.

Para el diseño de la página se ha utilizado también bootstrap 4, el cual es un kit de herramientas open source para el desarrollo de páginas web responsive, el cual es utilizado en la mayoría de aplicaciones web.

Para la Base de Datos se ha utilizado la base de datos Realtime Database perteneciente a Firebase, el cual es una plataforma, perteneciente a Google, en la nube para el desarrollo de aplicaciones web y móvil. Dentro de las funcionalidades de Firebase nos encontramos Realtime Database, el cual es una base de datos NoSQL, con un formato JSON.

Para la gestión de los usuarios se ha utilizado también Firebase, en concreto su apartado de Authentication.

1.4. Que son los datos abiertos

Definición dada en la propia página de la Junta de CyL [1]:

Datos abiertos (open data en inglés) es una filosofía y práctica que persigue que determinados datos estén disponibles de forma libre a todo el mundo, sin restricciones de copyright, patentes u otros mecanismos de control. Los datos deben publicarse en bruto (sin procesar), bien estructurados y en formatos conocidos que faciliten la reutilización.

El sector público produce una gran variedad de información interesante para los ciudadanos y empresas, como por ejemplo, información social, económica, geográfica, estadística, meteorológica o turística y sobre empresas y educación. Esta información tiene unas características que la hacen particularmente atractiva para el sector de los contenidos digitales, ya que es completa, fiable y de calidad.

La apertura de datos del sector público permite que cualquier persona u organización pueda construir sobre ellos una nueva idea que resulte en nuevos datos, conocimientos, mejorar procesos, dar valor añadido a los existentes o incluso crear nuevos servicios.

Por lo tanto, tiene un considerable potencial económico y además favorece la transparencia, participación y colaboración ciudadana, necesarios para contar con un gobierno más abierto.

Capítulo 2

Planificación y Presupuesto

En este capítulo, vamos a hablar de la metodología utilizada para el desarrollo de la aplicación, así como de los costes del desarrollo de la aplicación.

2.1. Metodología del proyecto

Para el desarrollo de la aplicación, se ha optado por la metodología de "Modelo de prototipos" [30], el cual pertenece a los modelos de desarrollo evolutivo. Hemos elegido este tipo de metodología, ya que, aunque ya se habían pensado las funcionalidades básicas de la aplicación, no se habían cerrado del todo, además de que, no se tenía del todo pensado el como incluirlas en la aplicación. También, se planeaban desarrollar primero las funcionalidades, y una vez que la aplicación fuese totalmente funcional, realizar los cambios para mejorar la calidad y la estética de la aplicación. Por estos motivos, se decantó hacia el "Modelo de prototipos".

2.2. Planificación inicial del proyecto

Para la planificación inicial, no se ha seguido ningún calendario para el desarrollo. Se tenía como fecha de comienzo del desarrollo de la aplicación principios de Febrero de 2021, ya que, en esas fechas ya se habría terminado con los exámenes. Como fecha final del desarrollo finales de Mayo de 2021, para así tener tiempo para la realización de la memoria para su entrega en primera convocatoria. Habría que tener en cuenta que, durante los meses de Abril y Mayo, en su totalidad, se estarían realizando las prácticas en empresa, las cuales consumirían 7 horas diarias, de lunes a viernes, de trabajo, más el tiempo de ir y venir de la empresa, que teniendo en cuenta que no estaba ubicada en Valladolid, sería mínimo de 45 minutos ida y 45 vuelta; por lo tanto, habría que tener terminada, al menos, más de la mitad de la aplicación para finales de Marzo.

2.3. Costes del proyecto

Aunque el proyecto se ha desarrollado por una sola persona, vamos a dividir sus roles como si de dos personas se tratase.

- **Analista:** Se encarga de las tareas de análisis de requisitos, diseño de interfaces y diagramas, y la elaboración de documentos.
- **Programador:** Se encarga de la implementación de la aplicación y de las pruebas de esta.

Para los salarios de cada uno se ha escogido el salario medio en <https://es.indeed.com/>, según el cual, un analista tendría un salario medio de 14.36€/hora, y un programador de 12.35€/hora. Con estos salarios medios y las horas dedicadas a cada parte del proyecto tenemos que, por la parte del programador serían 1525,225€ ($12.35 \times 123.5 = 1525,225$) y por la parte del analista 542,09€ ($37,75 \times 14,36 = 542,09$), lo cual daría un coste total de 2067,315€. Ha este resultado habría que añadirle los costes de hardware y los costes de software, los cuales ascienden a 2176,82€, estos costes vienen dados por el ordenador, junto a la pantalla y teclado, (1000€), el dispositivo móvil (300€), el ratón (50€), y los gastos en las API´s de Google (826,82€). Todos estos costes sumados darían que, la aplicación tiene un coste total de desarrollo de 4244,135€, el cual vemos como un coste apropiado para una aplicación de esta escala.

Capítulo 3

Análisis

En esta sección del documento especificaremos los distintos tipos de actores, requisitos o reglas obtenidos del análisis para desarrollar la aplicación.

3.1. Actores del sistema

Empezamos con el análisis de los distintos tipos de actores que pueden interactuar con la aplicación.

ID	Nombre	Descripción
AC-01	Usuario anónimo	Este actor representa al usuario que accede a la aplicación sin estar registrado.
AC-02	Usuario registrado	Este actor representa al usuario ya registrado en la página y que accede como tal.
AC-03	Gerente	Este actor representa a un usuario con privilegios especiales, es decir, al encargado de subir el fichero de monumentos.

Cuadro 3.1: Actores

Su jerarquía, aunque simple, viene dada en el siguiente diagrama.

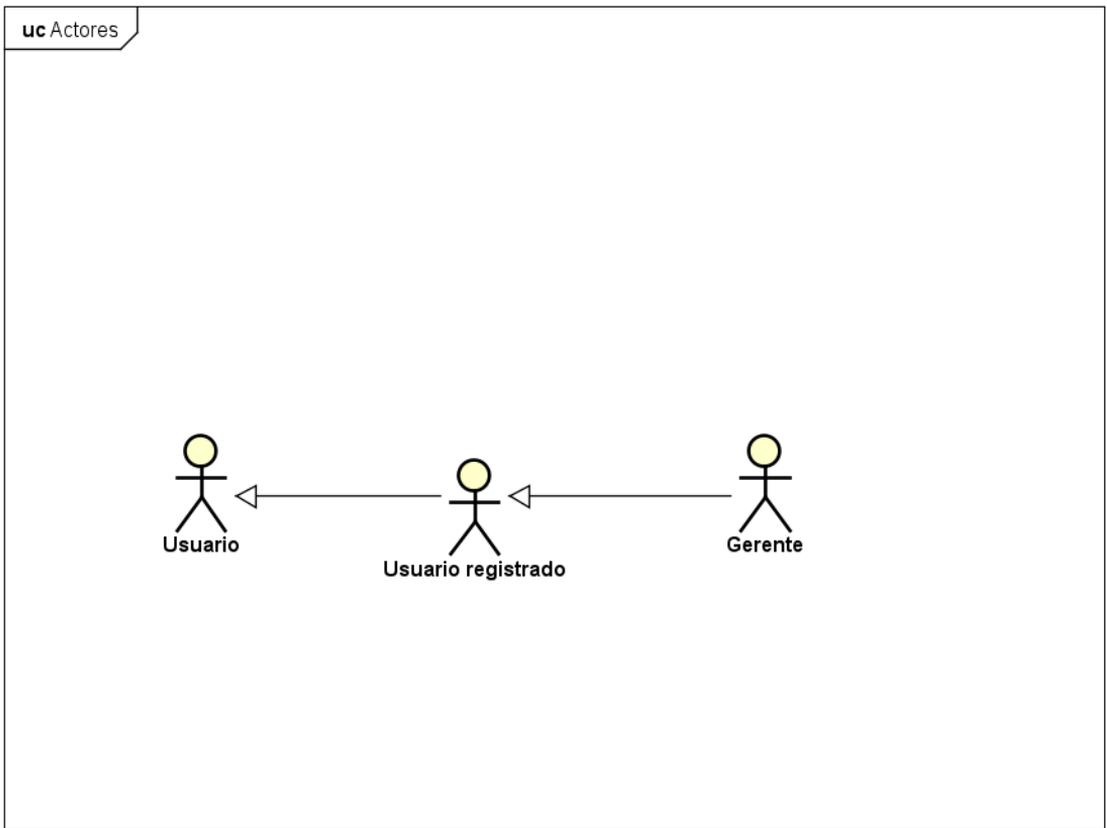


Figura 3.1: Jerarquía de los actores

3.2. Reglas de negocio

Las reglas de negocio son un conjunto de normas o restricciones que se van a aplicar al desarrollo de nuestra aplicación. Para el desarrollo de esta aplicación en concreto, tomamos en cuenta las siguientes.

ID	Descripción
RN-01	Esta aplicación está sujeta a la ley Orgánica 3/2018 de Protección de Datos Personales y garantía de los derechos digitales.
RN-02	Tanto los clientes, como el gerente, se identificarán en la aplicación de la misma manera.
RN-03	Únicamente el gerente podrá acceder al apartado para cambiar los monumentos actuales.
RN-04	El gerente podrá realizar todas las funciones del usuario registrado.

Cuadro 3.2: Reglas de negocio

3.3. Casos de Uso

En este apartado hablaremos de los casos de uso, así como de su especificación y sus diagramas. Estos estarán separados por actor, para facilitar su comprensión.

3.3.1. Casos de Uso

Actor AC-01: Usuario Anónimo

ID	Nombre
CU-01	Ver comentarios
CU-02	Buscar en resultados
CU-03	Ordenar resultados
CU-04	Buscar por texto
CU-05	Seleccionar por tipo
CU-06	Seleccionar por provincia
CU-07	Seleccionar por tipo y provincia
CU-08	Registrarse
CU-09	Visualizar en mapa
CU-10	Abrir ventana información

Cuadro 3.3: Casos de Uso Usuario Anónimo

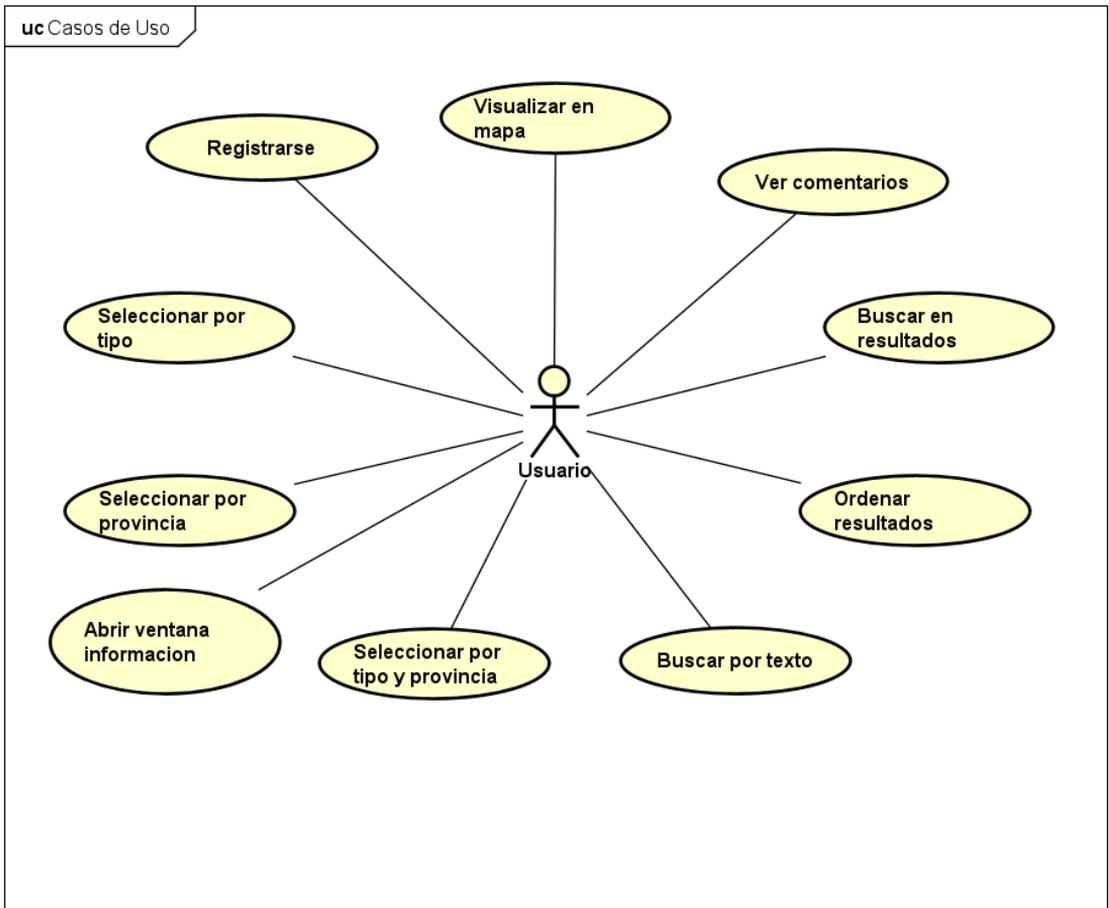


Figura 3.2: Diagrama Casos de Uso Usuario Anónimo

Actor AC-02: Usuario Registrado

ID	Nombre
CU-11	Identificarse
CU-12	Logout
CU-13	Crear comentario
CU-14	Añadir favorito
CU-15	Eliminar favorito

Cuadro 3.4: Casos de Uso Usuario Registrado

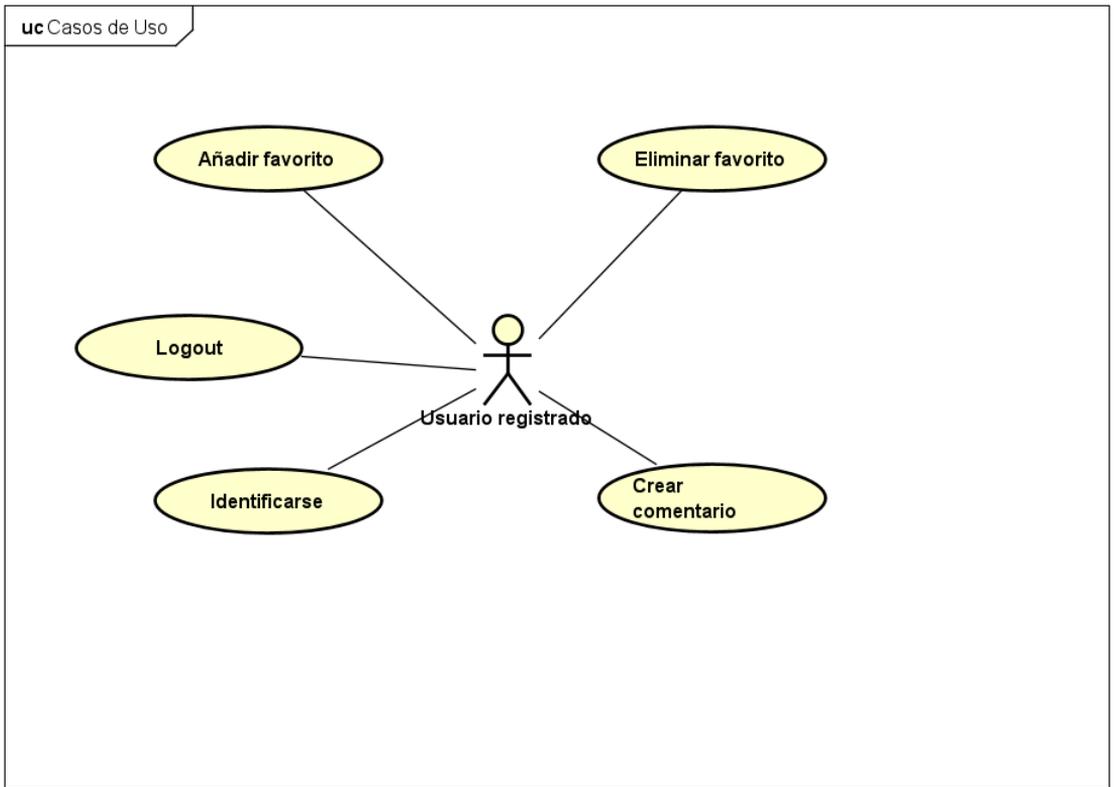


Figura 3.3: Diagrama Casos de Uso Usuario Registrado

Actor AC-03: Gerente

ID	Nombre
CU-16	Subir monumentos

Cuadro 3.5: Casos de Uso Gerente

3.3. CASOS DE USO

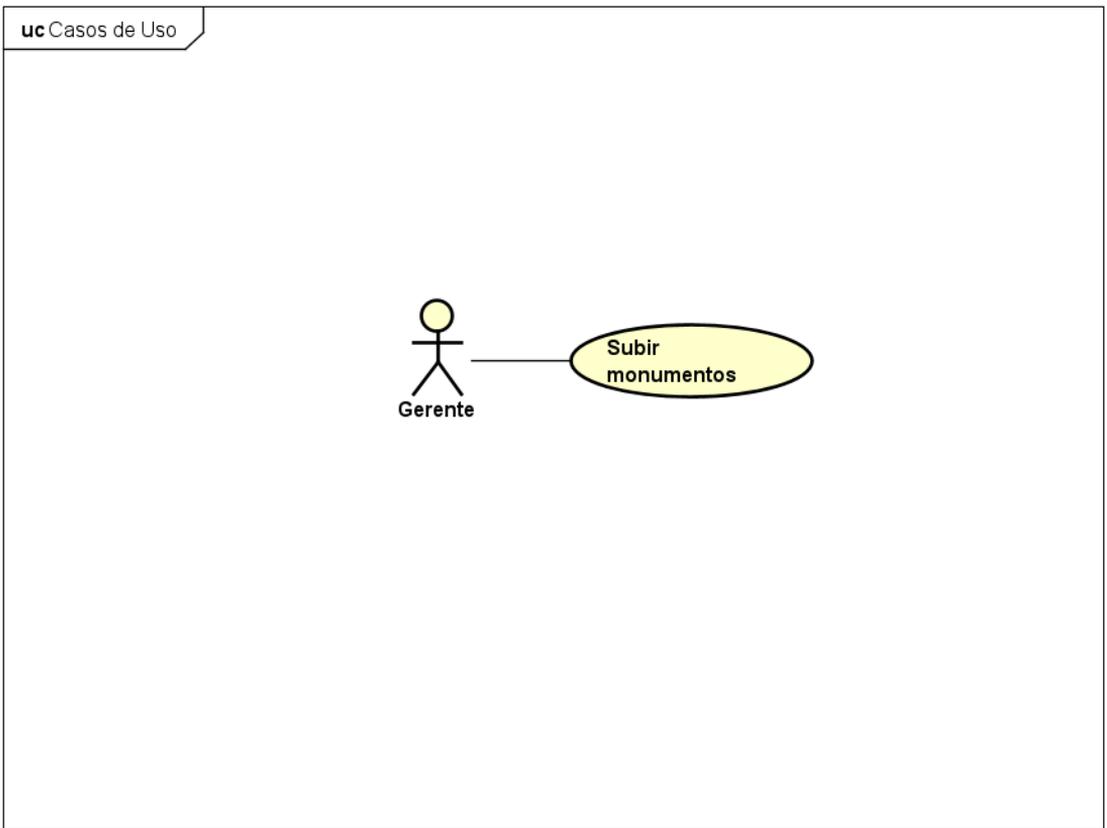


Figura 3.4: Diagrama Casos de Uso Gerente

3.3.2. Especificación de los Casos de Uso

Actor AC-01: Usuario Anónimo

CU-01	Ver comentarios
Descripción	El sistema permite al usuario visualizar los comentarios del sitio seleccionado.
Precondición	El usuario debe haber visualizado algún sitio.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo solicita ver los comentarios del sitio seleccionado. 2. El sistema muestra los comentarios del sitio pedido. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los comentarios del sitio.
Flujo alternativo	

Cuadro 3.6: Especificación Caso de Uso CU-01

CU-02	Buscar en resultados
Descripción	El sistema permite al usuario buscar en los resultados ya mostrados.
Precondición	El usuario debe haber visualizado algún sitio.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo escribe la cadena de texto por la que desea filtrar. 2. El sistema filtra los resultados por la cadena de texto y muestra los resultados correspondientes. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados filtrados.
Flujo alternativo	

Cuadro 3.7: Especificación Caso de Uso CU-02

3.3. CASOS DE USO

CU-03	Ordenar resultados
Descripción	El sistema permite al usuario ordenar por categoría los resultados ya mostrados.
Precondición	El usuario debe haber visualizado algún sitio o los comentarios de algún sitio.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo solicita al sistema ordenar los resultados por una categoría en concreto. 2. El sistema ordena los resultados por la categoría y muestra los resultados correspondientes. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados ordenados.
Flujo alternativo	

Cuadro 3.8: Especificación Caso de Uso CU-03

CU-04	Buscar por texto
Descripción	El sistema permite al usuario buscar por texto los resultados a mostrar.
Precondición	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo escribe el texto por el que desea buscar los resultados. 2. El sistema busca los resultados por el texto y muestra los resultados correspondientes. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados.
Flujo alternativo	<p>2a No existen resultados:</p> <ol style="list-style-type: none"> 1. El sistema notifica que no se han encontrado resultados. 2. Retorna al punto 1.

Cuadro 3.9: Especificación Caso de Uso CU-04

CU-05	Seleccionar por tipo
Descripción	El sistema permite al usuario seleccionar por tipo de establecimiento los resultados a mostrar.
Precondición	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo selecciona el tipo que desea visualizar. 2. El sistema busca los resultados por el tipo y muestra los resultados correspondientes. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados.
Flujo alternativo	

Cuadro 3.10: Especificación Caso de Uso CU-05

CU-06	Seleccionar por provincia
Descripción	El sistema permite al usuario seleccionar por provincia los resultados a mostrar.
Precondición	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo selecciona la provincia, cuyos sitios, desea visualizar. 2. El sistema busca los resultados por la provincia y muestra los resultados correspondientes. 3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados.
Flujo alternativo	

Cuadro 3.11: Especificación Caso de Uso CU-06

3.3. CASOS DE USO

CU-07	Seleccionar por tipo y provincia
Descripción	El sistema permite al usuario seleccionar por tipo y provincia los resultados a mostrar.
Precondición	
Flujo normal	<ol style="list-style-type: none">1. El usuario anónimo selecciona el tipo y la provincia cuyos sitios desea visualizar.2. El sistema busca los resultados por el tipo y la provincia y muestra los resultados correspondientes.3.El flujo finaliza correctamente.
Postcondición	El usuario puede visualizar los resultados.
Flujo alternativo	<p>2a Tipo seleccionado museo:</p> <ol style="list-style-type: none">1. El sistema avisa al usuario que los museos no se pueden buscar con provincias.2. Retorna al punto 1.

Cuadro 3.12: Especificación Caso de Uso CU-07

CU-08	Registrarse
Descripción	El sistema permite al usuario registrarse en la aplicación.
Precondición	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo selecciona la opción de registrarse. 2. El sistema muestra los campos que debe rellenar el usuario anónimo. 3. El usuario anónimo rellena los campos. 4. El sistema valida los campos y registra al usuario. 5.El flujo finaliza correctamente.
Postcondición	El usuario anónimo pasa a ser un usuario registrado
Flujo alternativo	<p>4a Contraseña débil:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que su contraseña es débil. 2. Retorna al punto 1. <p>4b Contraseña y repetir contraseña no igual:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que no ha escrito los dos campos de contraseña correctamente. 2. Retorna al punto 1. <p>4c Correo no valido:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que el correo introducido no es válido. 2. Retorna al punto 1. <p>4d Correo en uso:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que el correo introducido ya está en uso. 2. Retorna al punto 2.

Cuadro 3.13: Especificación Caso de Uso CU-08

3.3. CASOS DE USO

CU-09	Visualizar en mapa
Descripción	El sistema permite al usuario visualizar un sitio ya buscado en el mapa.
Precondición	Haber buscado algún sitio en la tabla o haber entrado a favoritos.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo selecciona la opción de ver en el mapa. 2. El sistema cambia al mapa y carga el sitio en él. 3.El flujo finaliza correctamente.
Postcondición	El usuario anónimo visualiza el sitio en el mapa.
Flujo alternativo	

Cuadro 3.14: Especificación Caso de Uso CU-10

CU-10	Abrir ventana informacion
Descripción	El sistema permite al usuario abrir una ventana con información en los marcadores del mapa.
Precondición	El usuario debe haber visualizado algún sitio en el mapa.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario anónimo clica un marcador en el mapa. 2. El sistema muestra una ventana con información del sitio correspondiente. 3.El flujo finaliza correctamente.
Postcondición	
Flujo alternativo	

Cuadro 3.15: Especificación Caso de Uso CU-10

Actor AC-02: Usuario Registrado

CU-11	Identificarse
Descripción	El sistema permite al usuario identificarse en la aplicación.
Precondición	Estar registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario registrado selecciona la opción de identificarse. 2. El sistema muestra los campos que debe rellenar el usuario registrado. 3. El usuario registrado rellena los campos. 4. El sistema valida los campos e identifica al usuario. 5.El flujo finaliza correctamente.
Postcondición	El usuario registrado se ha identificado en el sistema
Flujo alternativo	<p>4a Contraseña incorrecta:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que la contraseña introducida es incorrecta. 2. Retorna al punto 1. <p>4b Correo incorrecto:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que el correo introducido no es válido. 2. Retorna al punto 1.

Cuadro 3.16: Especificación Caso de Uso CU-11

CU-12	Logout
Descripción	El sistema permite al usuario salir de sesión.
Precondición	Estar identificado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario registrado selecciona la opción de logout. 2. El sistema desloguea al usuario registrado y borra la información de la sesión. 3.El flujo finaliza correctamente.
Postcondición	El usuario registrado se ha deslogueado del sistema.
Flujo alternativo	

Cuadro 3.17: Especificación Caso de Uso CU-12

3.3. CASOS DE USO

CU-13	Crear comentario
Descripción	El sistema permite al usuario crear un comentario.
Precondición	Ver comentarios (CU-01) y estar identificado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario registrado selecciona la opción de identificarse. 2. El sistema muestra los campos que debe rellenar el usuario registrado. 3. El usuario registrado rellena los campos. 4. El sistema valida los campos e identifica al usuario. 5. El flujo finaliza correctamente.
Postcondición	El usuario registrado se ha identificado en el sistema
Flujo alternativo	<p>4a Contraseña incorrecta:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que la contraseña introducida es incorrecta. 2. Retorna al punto 1. <p>4b Correo incorrecto:</p> <ol style="list-style-type: none"> 1. El sistema avisa al usuario que el correo introducido no es válido. 2. Retorna al punto 1.

Cuadro 3.18: Especificación Caso de Uso CU-13

CU-14 Añadir favorito	
Descripción	El sistema permite al usuario añadir un sitio a sus favoritos.
Precondición	Estar identificado y visualizar algún sitio.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario registrado selecciona la opción de añadir a favoritos. 2. El sistema comprueba que no exista ya ese favorito y lo añade. 3.El flujo finaliza correctamente.
Postcondición	El usuario registrado ha añadido un nuevo sitio a sus favoritos
Flujo alternativo	<p>2a Favorito existente:</p> <ol style="list-style-type: none"> 1. El sistema detecta que ya existía ese favorito, por lo tanto no lo añade de nuevo. 2. El flujo finaliza correctamente.

Cuadro 3.19: Especificación Caso de Uso CU-14

CU-15 Eliminar favorito	
Descripción	El sistema permite al usuario eliminar un sitio de sus favoritos.
Precondición	Estar identificado y tener algún favorito.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario registrado selecciona la opción de eliminar de favoritos. 2. El sistema borra ese favorito. 3.El flujo finaliza correctamente.
Postcondición	El usuario registrado ha borrado un sitio de sus favoritos
Flujo alternativo	

Cuadro 3.20: Especificación Caso de Uso CU-15

Actor AC-03: Gerente

CU-16	Subir monumentos
Descripción	El sistema permite al gerente subir nuevos monumentos.
Precondición	Estar identificado como gerente.
Flujo normal	<ol style="list-style-type: none">1. El gerente selecciona el documento a subir.2. El sistema selecciona los monumentos del documento y realiza el cambio en la base de datos.3.El flujo finaliza correctamente.
Postcondición	El gerente ha modificado los documentos.
Flujo alternativo	2a Tipo incorrecto: <ol style="list-style-type: none">1. El sistema detecta que el documento subido no es un json y avisa al gerente.2. Retorna al punto 1.

Cuadro 3.21: Especificación Caso de Uso CU-16

3.4. Requisitos de Información

En este apartado hablaremos sobre los datos utilizados en la aplicación, aunque, como se verá, en esta aplicación solo se tienen dos tipos de datos, los usuarios registrados y los sitios. Además, todos los sitios, menos los monumentos, no están guardados en la base de datos de Firebase, sino que se piden a la API de la Junta en el momento de su solicitud por parte del usuario de la aplicación. También, los sitios y favoritos se guardan de la misma forma en la DB, por lo que solo ponemos la tabla de sitio.

RI-E01 Usuario						
Descripción	Esta entidad define al usuario registrado de la aplicación.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	PK
RI-E01.1	Correo	Contiene el email del usuario.	String	Si	No	No
RI-E01.2	Contraseña	Contiene la contraseña del usuario.	String	No	No	No
RI-E01.3	UID	Contiene la clave primaria del usuario, generada por Firebase.	String	Si	No	Si

Cuadro 3.22: Definición de Usuario en DB

RI-E02 Sitio						
Descripción	Esta entidad define al sitio de la aplicación, en la DB.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	PK
RI-E02.1	email	Contiene el email del sitio.	String	Si	Si	No
RI-E02.2	accesibles_minusvalidos	Contiene si el sitio es accesible por minusválidos.	Boolean	No	Si	No
RI-E02.3	categoría	Contiene la categoría del sitio.	String	No	Si	No
RI-E02.4	cp	Contiene el código postal del sitio.	String	No	Si	No
RI-E02.5	dirección	Contiene la dirección del sitio.	String	No	Si	No
RI-E02.6	establecimiento	Contiene el tipo de establecimiento del sitio, dentro de los datos de la Junta.	String	No	Si	No

3.4. REQUISITOS DE INFORMACIÓN

RI-E02	Sitio					
RI-E02.7	localidad	Contiene la localidad del sitio.	String	No	Si	No
RI-E02.8	municipio	Contiene el municipio del sitio.	String	No	Si	No
RI-E02.9	nombre	Contiene el nombre del sitio.	String	No	No	Si
RI-E02.10	nucleo	Contiene el núcleo de población del sitio.	String	No	Si	No
RI-E02.11	num_registro	Contiene el número de registro del sitio.	String	No	No	Si
RI-E02.12	provincia	Contiene la provincia del sitio.	String	No	Si	No
RI-E02.13	telefono_1	Contiene el primer teléfono del sitio.	String	No	Si	No
RI-E02.14	telefono_2	Contiene el segundo teléfono del sitio.	String	No	Si	No
RI-E02.15	plazas	Contiene el número de plazas del sitio.	Integer	No	Si	No
RI-E02.16	latitud	Contiene la latitud del sitio.	Integer	No	Si	No
RI-E02.17	longitud	Contiene la longitud del sitio.	Integer	No	Si	No
RI-E02.18	web	Contiene la web del sitio.	String	Si	Si	No

Cuadro 3.23: Definición de Sitio en DB

RI-E03 Comentario						
Descripción	Esta entidad define a los comentarios de la aplicación.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	PK
RI-E03.1	Comment	Contiene el texto del comentario.	String	No	No	No
RI-E03.2	Score	Contiene la puntuación del comentario.	Int	No	No	No
RI-E03.3	Date	Contiene la fecha del comentario.	Date	No	No	No

Cuadro 3.24: Definición de Comentario en DB

Sobre la tabla Sitio, como pone, es la tabla correspondiente al sitio guardado en la base de datos de Firebase, no al sitio en la DB de la Junta, en esta si que tiene una clave primaria pero, esta clave primaria no nos sirve, ya que, no podemos buscar por ella en la API de datos abiertos, por lo tanto no la guardamos.

3.5. Requisitos No Funcionales

Los requisitos no funcionales definen ciertas normas, con respecto de la aplicación que deben ser cumplidas, relacionadas con la, robustez, seguridad, etc.

Seguridad	
ID	Descripción
RNF-01	Los datos ingresados por teclado serán validados correctamente por la aplicación.
RNF-02	Las contraseñas de los usuarios se almacenarán de forma segura mediante un algoritmo hash propio, definido por Firebase.
RNF-03	La contraseña de un usuario deberá tener una longitud mínima de 6 caracteres.
RNF-04	Un usuario no puede visualizar la información personal de otro usuario.
RNF-05	Los usuarios anónimos únicamente podrán acceder a las partes públicas de la aplicación.
RNF-06	Una vez el usuario cierre la sesión, sus datos de sesión deberán ser eliminados.

Cuadro 3.25: RNF Seguridad

Usabilidad	
ID	Descripción
RNF-07	La aplicación deberá estar disponible en castellano.
RNF-08	La aplicación deberá ser intuitiva y fácil de usar.
RNF-09	La aplicación deberá ser utilizable tanto en ordenador como en móvil.

Cuadro 3.26: RNF Usabilidad

Robustez	
ID	Descripción
RNF-10	El sistema estará preparado para hacer frente a situaciones anómalas.
RNF-11	El sistema mostrará mensajes de aviso si detecta fallos al validar datos introducidos por el cliente.
RNF-12	El sistema mostrará mensajes de aviso si detecta fallos en la parte de servidor.

Cuadro 3.27: RNF Robustez

Disponibilidad	
ID	Descripción
RNF-13	La aplicación deberá estar disponible 24x7.

Cuadro 3.28: RNF Disponibilidad

3.6. Requisitos Funcionales

Los requisitos funcionales definen las funciones que llevan a cabo la aplicación o sus componentes.

- CU-01: Ver comentarios
 - RF-01: El sistema habilita la opción de ver los comentarios de un sitio.
 - RF-02: El sistema muestra al usuario los comentarios.
 - RF-03: El sistema calcula la puntuación media de los comentarios.
 - RF-04: El sistema permitirá al usuario ordenar los comentarios por puntuación o fecha.
- CU-02: Buscar en resultados
 - RF-05: El sistema permitirá al usuario introducir una cadena de texto alfanumérica para buscar.

- RF-06: El sistema utilizará la cadena a partir de que tenga 2 o más caracteres de longitud.
- CU-03: Ordenar resultados
 - RF-07: El sistema permitirá al usuario ordenar, los datos resultantes, por los atributos permitidos.
- CU-04: Buscar por texto
 - RF-08: El sistema permitirá al usuario buscar resultados por una cadena de texto alfanumérica.
 - RF-09: El sistema utilizará la cadena a partir de que tenga 3 o más caracteres de longitud.
- CU-05: Seleccionar por tipo
 - RF-10: El sistema mostrará al usuario los diferentes tipos de establecimiento permitidos.
 - RF-11: El sistema permitirá al usuario seleccionar el tipo por el que desea buscar.
 - RF-12: El sistema mostrará al usuario los sitios que tengan el tipo seleccionado por el mismo.
- CU-06: Seleccionar por provincia
 - RF-13: El sistema mostrará al usuario las provincias de Castilla y León.
 - RF-14: El sistema permitirá al usuario elegir una de las provincias para buscar.
 - RF-15: El sistema mostrará al usuario los sitios pertenecientes a la provincia seleccionada.
- CU-07: Seleccionar por tipo y provincia
 - RF-16: El sistema permitirá al usuario seleccionar tanto el tipo como la provincia deseados.
 - RF-17: El sistema avisará al usuario, si el tipo seleccionado es museo, de que no se puede buscar junto a una provincia.
 - RF-18: El sistema mostrará al usuario los sitios con el tipo seleccionado y que pertenezcan a la provincia deseada.
- CU-08: Registrarse
 - RF-19: El sistema permitirá la función de registrarse.
 - RF-20: El sistema solicitará al usuario anónimo los datos pertinentes.
 - RF-21: El sistema validará los datos introducidos por el usuario.
 - RF-22: El sistema avisará al usuario si los datos introducidos no son válidos.
 - RF-23: El sistema avisará al usuario si ya existe el correo introducido.
 - RF-24: El sistema llevará a cabo el registro si no detecta ningún problema con los datos introducidos.

- CU-09: Visualizar en mapa
 - RF-25: El sistema mostrará al usuario el botón para ver el sitio en el mapa.
 - RF-26: El sistema cambiará la ventana a la de mapa.
 - RF-27: El sistema mostrará el sitio deseado en el mapa.
- CU-10: Abrir ventana información
 - RF-28: El sistema mostrará al usuario la ventana en justo encima del marcador correspondiente.
 - RF-29: El sistema mostrará al usuario, dentro de la ventana, información del sitio, siempre y cuando esta información exista en los datos de la Junta.
 - RF-30: El sistema mostrará al usuario, dentro de la ventana, tanto un botón para ver los comentarios, como otro para añadir a favoritos el sitio.
- CU-11: Identificarse
 - RF-31: El sistema permitirá la función de identificarse.
 - RF-32: El sistema solicitará los datos para la identificación.
 - RF-33: El sistema validará si los datos introducidos son correctos.
 - RF-34: El sistema avisará al usuario si la contraseña introducida no es correcta.
 - RF-35: El sistema avisará al usuario si el correo introducido es incorrecto.
 - RF-36: El sistema llevará a cabo la identificación si los datos introducidos son correctos.
- CU-12: Logout
 - RF-37: El sistema permitirá la función de desloguearse, siempre y cuando el usuario estuviese previamente identificado.
 - RF-38: El sistema borrará los datos de sesión del usuario, cuando este le de al botón.
- CU-13: Crear comentario
 - RF-39: El sistema permitirá la creación de un nuevo comentario en el sitio indicado si el usuario está identificado.
 - RF-40: El sistema permitirá introducir como comentario una cadena alfanumérica.
 - RF-41: El sistema permitirá introducir una puntuación de entre los siguientes valores: 1, 2, 3, 4 y 5.
 - RF-42: El sistema añadirá el comentario, creado por el usuario, a la DB.
- CU-14: Añadir favorito
 - RF-43: El sistema permitirá al usuario, previamente identificado, añadir un sitio como favorito.
 - RF-44: El sistema validará si el sitio no está ya como favorito.

- RF-45: El sistema añadirá el sitio como favorito, del usuario en cuestión, a la DB.
- CU-15: Eliminar favorito
 - RF-46: El sistema permitirá a un usuario, previamente identificado, eliminar uno de sus sitios favoritos.
 - RF-47: El sistema borrará el sitio de los favoritos, del usuario en cuestión, de la DB.
- CU-16: Subir monumentos
 - RF-48: El sistema permitirá al gerente subir un fichero con los nuevos monumentos.
 - RF-49: El sistema validará si el tipo de fichero es un Json.
 - RF-50: El sistema extraerá los monumentos del fichero subido.
 - RF-51: El sistema eliminará los monumentos de la DB, subidos con anterioridad.
 - RF-52: El sistema añadirá los nuevos monumentos a la DB.

3.7. Sobre los datos de la Junta

Aquí vamos a hablar sobre la estructura de los datos abiertos que utilizamos de la Junta de Castilla y León. Para esta aplicación, utilizamos 3 conjuntos de datos diferentes:

- El primero contiene la mayoría de los registros que se utilizan en la aplicación. Esta identificado en la página de los datos abiertos de la Junta por *registro-de-turismo-de-castilla-y-leon* [4].
- El segundo contiene los datos de los museos. Esta identificado en la página de los datos abiertos de la Junta por *museos* [5].
- El tercero contiene los datos de los monumentos. Este se consume de forma distinta a los otros dos, ya que, no tiene acceso mediante la API-REST, para ello esta el Gerente, el cual se descarga los datos en formato Json y los sube a la aplicación [6].

El primer tipo es el que más información contiene, ya que agrupa la gran mayoría de los tipos de establecimiento utilizados en la aplicación.

3.7. SOBRE LOS DATOS DE LA JUNTA

Nombre	Tipo	Ejemplo
establecimiento	texto	Alojam. Turismo Rural
num_registro	texto	34/000225
codigo	texto	
tipo	texto	Casa Rural de Alquiler
categoria	texto	2 ^a - 2 Estrellas
especialidades	texto	
clase	texto	
nombre	texto	LA TOBA
direccion	texto	Calle ALTA 4
cp	texto	34247
provincia	texto	Palencia
municipio	texto	Cevico Navero
localidad	texto	CEVICO NAVERO
nucleo	texto	CEVICO NAVERO
telefono_1	texto	607516354
telefono_2	texto	979797574
telefono_3	texto	
email	texto	info@latobacasarural.com
web	texto	http://www.latobacasarural.com/
q_calidad	texto	
central_reservas	texto	
posada_real	texto	
plazas	int	4
longitud	decimal	-4.1860705
latitud	decimal	41.8614695
accesibles_minusvalidos	texto	
posicion	geo_point_2d	41.8614695, -4.1860705

Cuadro 3.29: Esquema del 1º Conjunto de Datos

El segundo conjunto, el de los museos, tenía un esquema muy similar al anterior, quitando algunos atributos que nunca aparecerían o cambiándole el nombre a otros, pero a 03/06/2021 decidieron cambiar el esquema, quitando algunos datos que se utilizaban en la aplicación.

Nombre	Que es	Tipo
descripcionblob	Descripcion	texto
nombrentidad	Nombre	texto
directoriorelacionado	DirectorioRelacionado	texto
directoriorelacionado0	Latitud	decimal
column_5	Longitud	decimal
column_6	Zoom	int
directoriorelacionado1	ID DirectorioRelacionado	int
localidad	Localidad	texto
localidad0	ID Localidad	int
soloclasificar	SoloClasificar	texto
tipo_de_gestion	Tipo de Gestión	texto
horario_de_apertura	Horario de apertura	texto
requisitos_especificos_para_el_acceso	Requisitos específicos para el acceso	texto
accesibilidad	Accesibilidad	texto
servicios_disponibles	Servicios disponibles	texto
informacion_adicional	Información adicional	texto
codigo	Código	texto
directorio_superior	Directorio Superior	texto
identificador_directorio_superior	Identificador Directorio Superior	int
identificador	Identificador	int
ultimaactualizacion	ultimaActualizacion	fecha
enlace_al_contenido	Enlace al contenido	texto
posicion	Posición	geo_point_2d

Cuadro 3.30: Esquema del 2º Conjunto de Datos

3.7. SOBRE LOS DATOS DE LA JUNTA

El tercer conjunto de datos, los monumentos, es el que menos atributos y datos contiene.

Nombre	Tipo	Ejemplo
identificador	texto	23941
nombre	texto	Risco La Zorrera
tipoMonumento	texto	Yacimientos arqueológicos
identificadorBienInteresCultural	texto	696
calle	texto	Risco La Zorrera
clasificacion	texto	Yacimiento
tipoConstruccion	texto	Arte rupestre
codigoPostal	texto	5489
Descripcion	texto	Descubierto en 1986 consta de dos paneles verticales y un tercero horizontal que cubre los anteriores. Pueden distinguirse tres fases de ejecución, predominando en la primera los trazos de color rojo, en la segunda destaca una figura de forma humana, y la tercera incluye figuras animales y humanas alargadas. Se ha interpretado como la representación de un templo o altar.
email	texto	
estiloPredominante	texto	
fax	texto	
horariosYTarifas	texto	
periodoHistorico	texto	Edad del Hierro,Neolítico,Edad del Bronce
provincia	texto	Ávila
municipio	texto	Candeleda
localidad	texto	Raso (El)
telefono	texto	
latitud	texto	40.170041
longitud	texto	-5.339664
web	texto	

Cuadro 3.31: Esquema del 3º Conjunto de Datos

Capítulo 4

Diseño

4.1. Arquitectura lógica

En este apartado se describirá la arquitectura lógica de la aplicación.

Como se ha dicho anteriormente, la aplicación consiste en un Front-End desarrollado con Angular 11, y un Back-End formado, tanto por Firebase, con la base de datos y la parte de autenticación, como con los datos abiertos mantenidos por la Junta. Por ello, vamos a hablar de ambas partes.

En lo que respecta al Back-End de la aplicación, los datos abiertos de la Junta se consumen de forma fácil, ya que la propia Junta da un acceso a ellos mediante una HTTP API-REST, la cual devuelve los datos pedidos en formato Json [3].

Para la utilización de Firebase, utilizamos el paquete oficial de Google para integrar Firebase a nuestro proyecto Angular. En Firebase, como se ha dicho anteriormente, utilizamos el apartado de Realtime Database, la cual es una base de datos en tiempo real, NoSql con formato Json, y el apartado Authentication, el cual lleva a cabo la lógica con respecto a los usuarios registrados de la aplicación.

En lo que respecta al Front-End, Angular utiliza un patrón basado en componentes. Cada componente contiene 3 partes: Un archivo formato TypeScript, el cual contiene la lógica de negocio del componente, un archivo HTML, el cual tiene la parte visual del componente, y un archivo CSS, con los estilos a ser usados en el HTML. Fuera de los componentes tenemos los modelos, los cuales contienen las estructuras de los objetos a tratar, y los servicios, en los cuales está la lógica de negocio correspondiente a la comunicación con las API-REST y el servicio de Firebase.

4.1. ARQUITECTURA LÓGICA

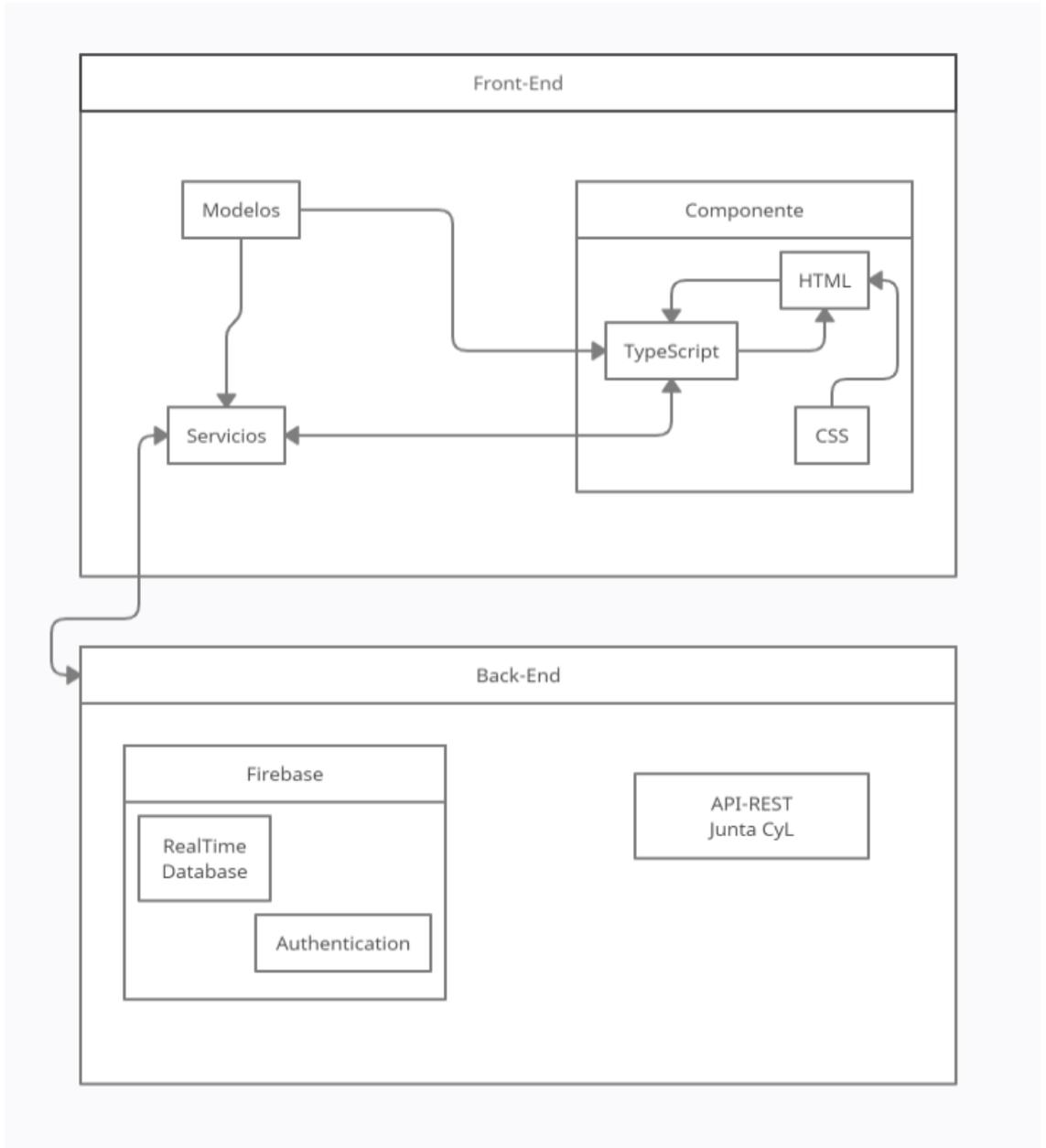


Figura 4.1: Diagrama Arquitectura Lógica

4.2. Arquitectura física

La arquitectura física representa los componentes hardware necesarios para desplegar satisfactoriamente la aplicación.

En nuestro caso tenemos una arquitectura Cliente-Servidor, en la cual el lado del cliente (Front-End) pide los datos necesarios al lado del servidor (Back-End).

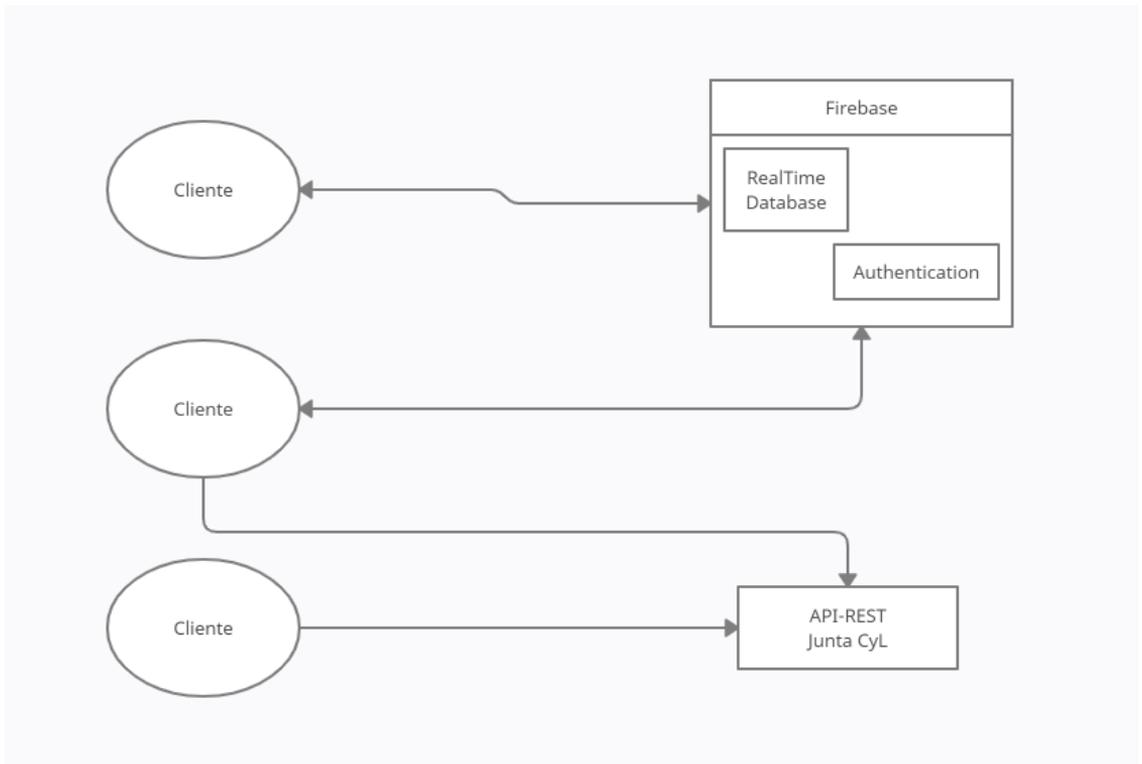


Figura 4.2: Diagrama Arquitectura Física

4.3. Diseño Base de Datos

En esta sección se mostrará el diseño de la base de datos, el cual enseñaremos a través del Diagrama de clases y, después de esto, el detalle de cada tabla.

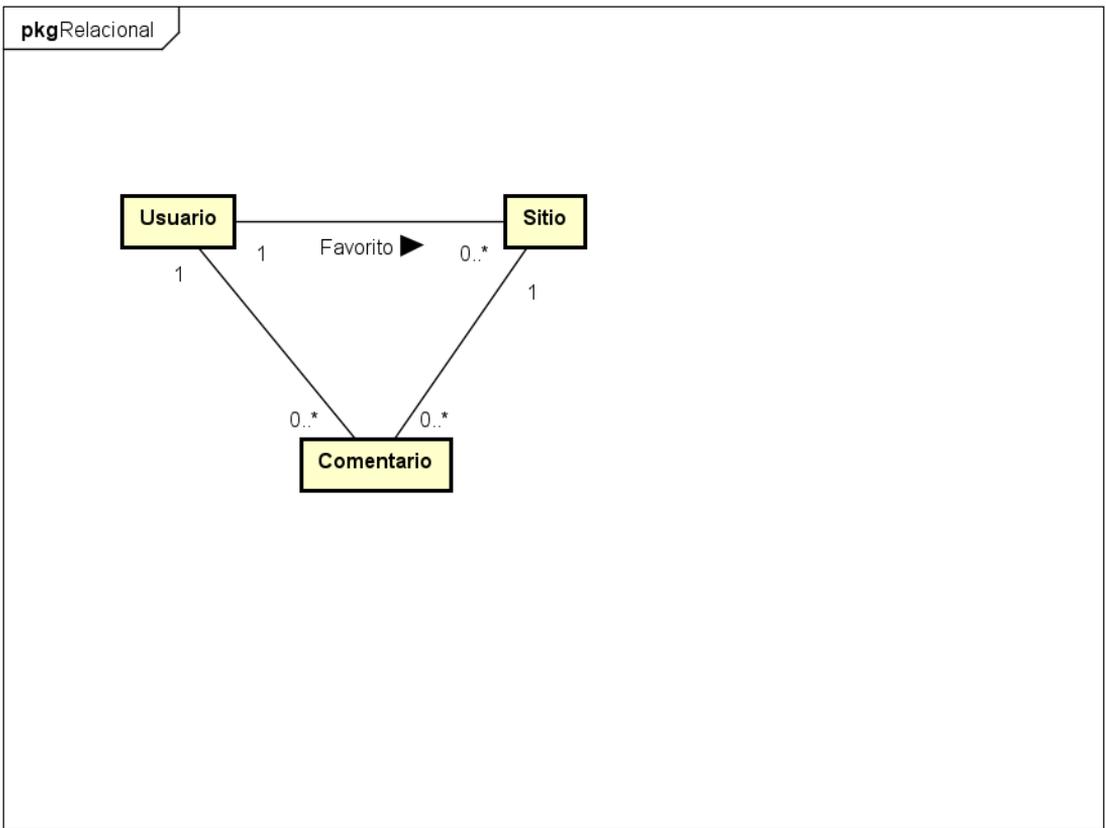


Figura 4.3: Diagrama de clases

El detalle de cada tabla viene a continuación.

Tabla Usuario					
Atributos					
Nombre	Descripción	Dominio	Unique	Null	PK
Correo	Contiene el email del usuario.	String	Si	No	No
Contraseña	Contiene la contraseña del usuario.	String	No	No	No
UID	Contiene la clave primaria del usuario, generada por Firebase.	String	Si	No	Si

Cuadro 4.1: Tabla Usuario

Tabla Sitio					
Atributos					
Nombre	Descripción	Dominio	Unique	Null	PK
email	Contiene el email del sitio.	String	Si	Si	No
accesibles_minusvalidos	Contiene si el sitio es accesible por minusválidos.	Boolean	No	Si	No
categoría	Contiene la categoría del sitio.	String	No	Si	No
cp	Contiene el código postal del sitio.	String	No	Si	No
dirección	Contiene la dirección del sitio.	String	No	Si	No
establecimiento	Contiene el tipo de establecimiento del sitio, dentro de los datos de la Junta.	String	No	Si	No
localidad	Contiene la localidad del sitio.	String	No	Si	No

4.3. DISEÑO BASE DE DATOS

Tabla Sitio					
municipio	Contiene el municipio del sitio.	String	No	Si	No
nombre	Contiene el nombre del sitio.	String	No	No	Si
nucleo	Contiene el núcleo de población del sitio.	String	No	Si	No
num_registro	Contiene el número de registro del sitio.	String	No	No	Si
provincia	Contiene la provincia del sitio.	String	No	Si	No
telefono_1	Contiene el primer teléfono del sitio.	String	No	Si	No
telefono_2	Contiene el segundo teléfono del sitio.	String	No	Si	No
plazas	Contiene el número de plazas del sitio.	Integer	No	Si	No
latitud	Contiene la latitud del sitio.	Integer	No	Si	No
longitud	Contiene la longitud del sitio.	Integer	No	Si	No
web	Contiene la web del sitio.	String	Si	Si	No

Cuadro 4.2: Tabla Sitio

Tabla Comentario					
Atributos					
Nombre	Descripción	Dominio	Unique	Null	PK
comment	Contiene el texto del comentario.	String	No	No	No
score	Contiene la puntuación del comentario.	Integer	No	No	No
date	Contiene la fecha del comentario.	String	No	No	No

Cuadro 4.3: Tabla Comentario

Al ser RealTime Database una DB NoSql, no tenemos claves primaria o foráneas. En la tabla usuario se ha marcado el uid como PK, ya que es la forma que tiene Firebase Authentication de identificar cada usuario único.

En la tabla sitio, por otro lado, se han marcado el num_registro y el nombre como únicos, ya que es lo que se usa en la aplicación para identificar un sitio.

Esto es como se ve la base de datos, RealTime Database, en Firebase.



Figura 4.4: RealTime Database Firebase

La parte de Authentication, donde están los usuarios es la siguiente.

4.4. DIAGRAMAS DE SECUENCIA



Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario ↑
a@a.com	✉	19 may. 2021	19 may. 2021	31beB1UpzIQmqHDVlQDaf4Rdlth3
pablo@mail.com	✉	18 mar. 2021	18 mar. 2021	6rOhEA2rutVlamcj2E2gVz3g4jj2
a@a.es	✉	5 abr. 2021	5 abr. 2021	BD6j20CSVxMN0xsTdzKIDCvfJUJ3
pepito@mail.com	✉	2 mar. 2021	3 jun. 2021	ls2shtZK0nZfJflwXufilCLWfgn2
prueba@mail.com	✉	3 jun. 2021	3 jun. 2021	UTfd5sS8GkXr0ytSSwBovdxMwZn2
gerentemon@mail.com	✉	28 abr. 2021	31 may. 2021	hoIDMAyWmQbpgsTZfsSF8GMx2...

Filas por página: 50 1 - 6 of 6

Figura 4.5: Authentication Firebase

4.4. Diagramas de secuencia

Dentro de los diagramas de secuencia, se van a mostrar únicamente los pertenecientes a 2 casos de uso de la aplicación, ya que, todos son muy similares entre si. Más concretamente, los diagramas pertenecientes a los CU-05 y CU-11, ya que con estos 2 se puede ver, tanto el uso de la API-REST de la Junta como de ambas partes de Firebase, RealTime Database y Authentication.

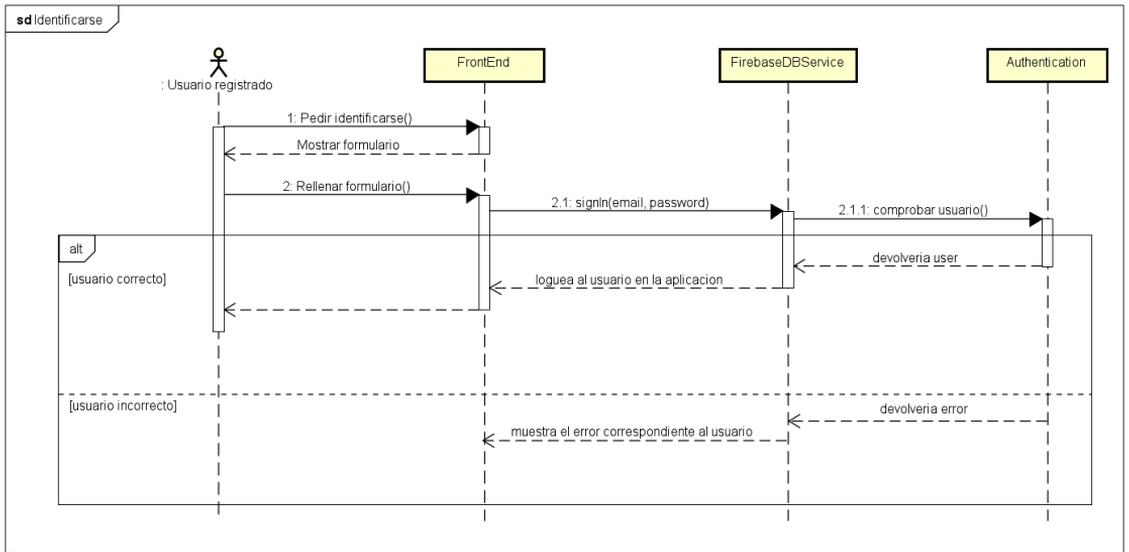


Figura 4.6: Diagrama de secuencia CU-11

4.4. DIAGRAMAS DE SECUENCIA

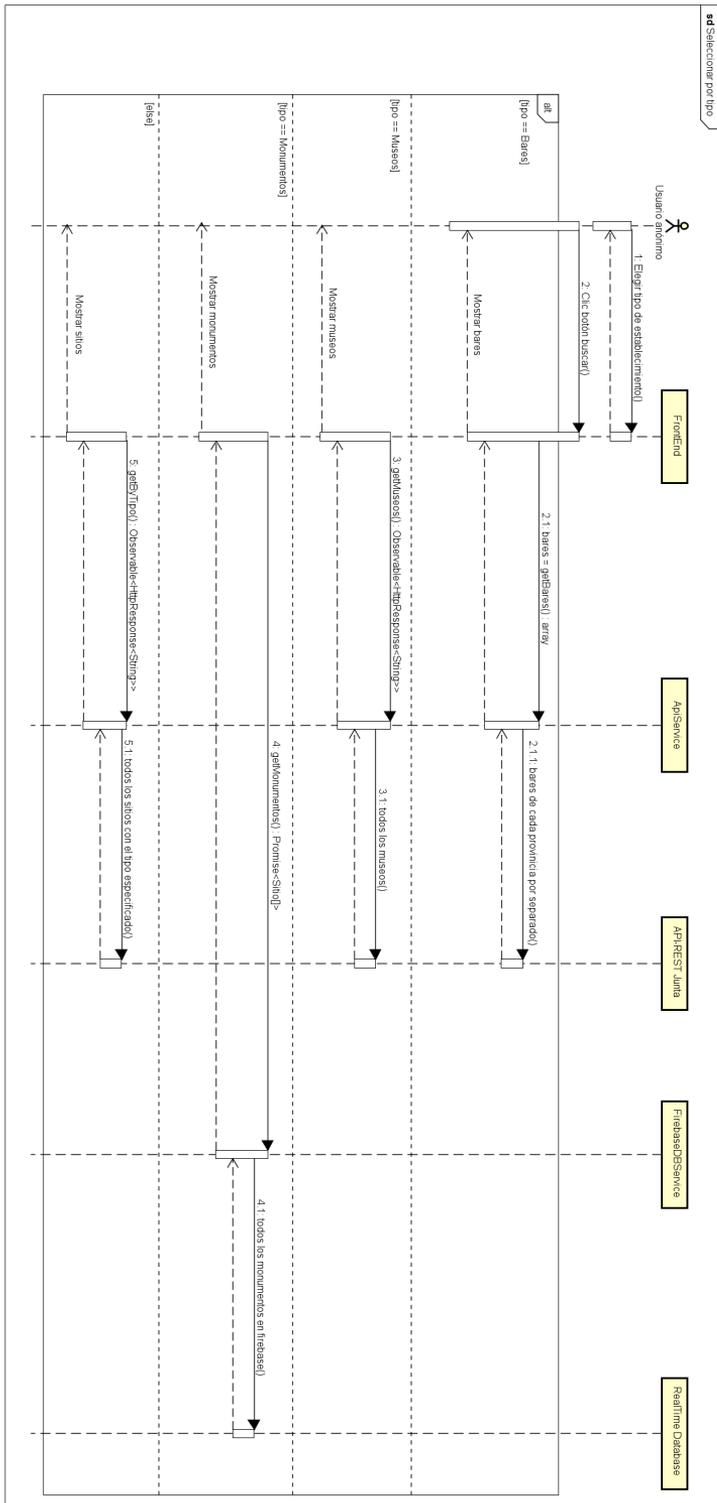


Figura 4.7: Diagrama de secuencia CU-05

4.5. Diagramas de interfaz de usuario

A continuación, los bocetos de la interfaz. Cada imagen corresponde a una pantalla de la aplicación, menos las dos últimas, ya que iniciar sesión y registrarte se hace sobre un *Modal*.

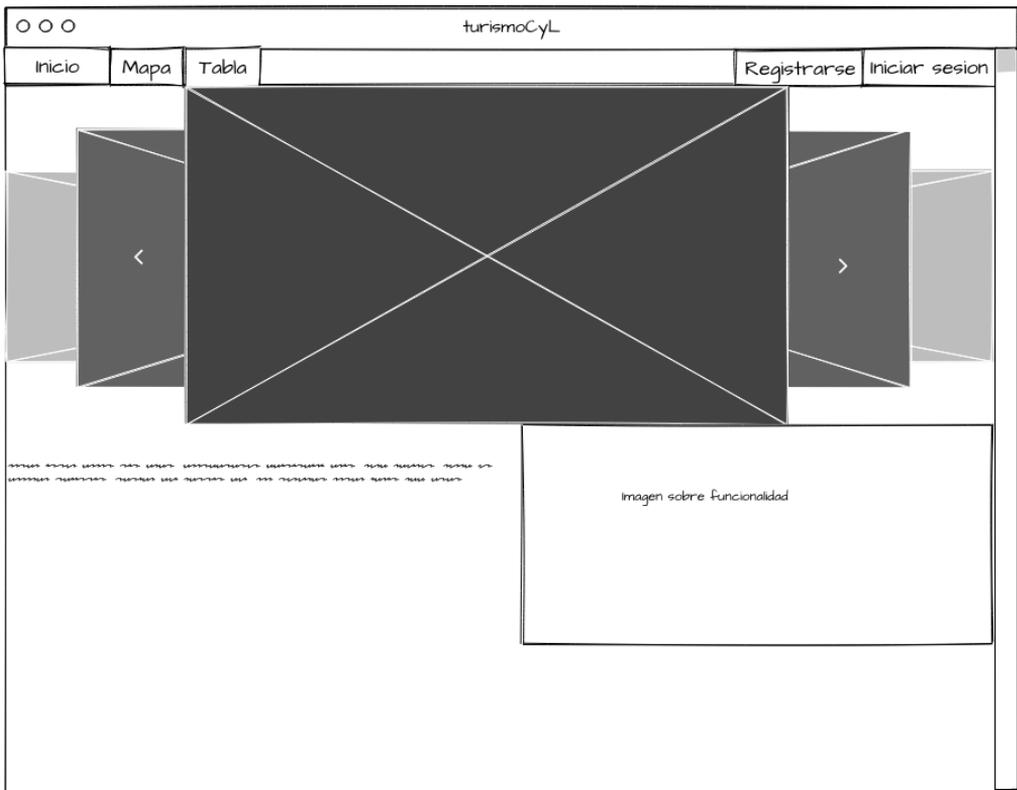


Figura 4.8: Boceto de página de inicio

4.5. DIAGRAMAS DE INTERFAZ DE USUARIO

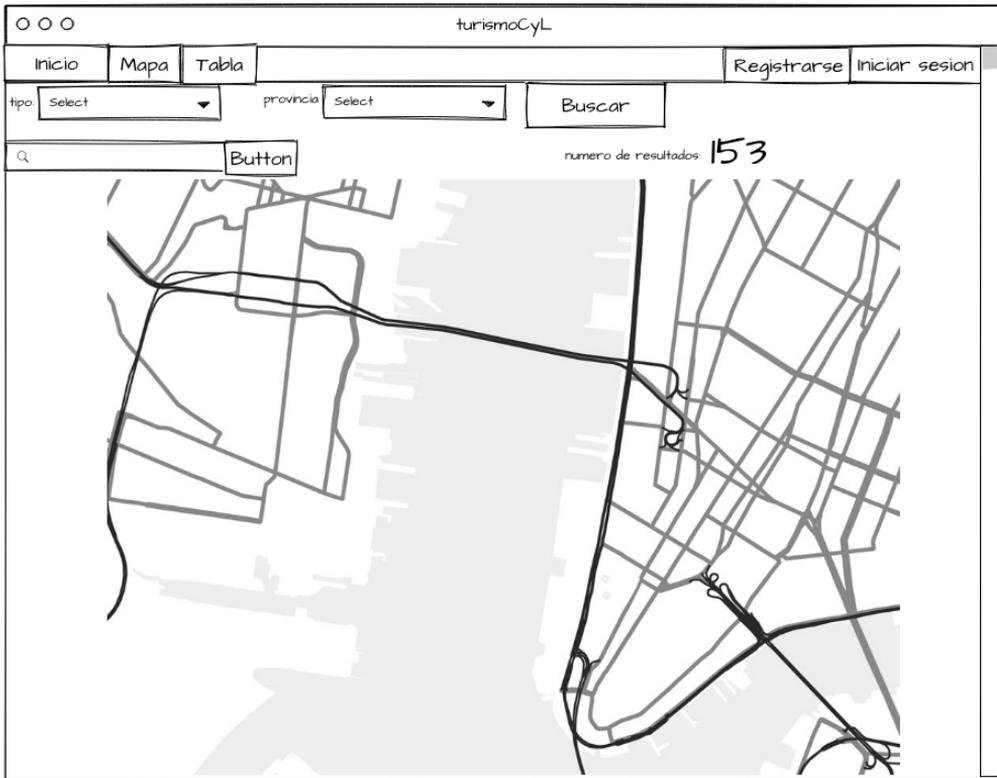


Figura 4.9: Boceto de página de mapa

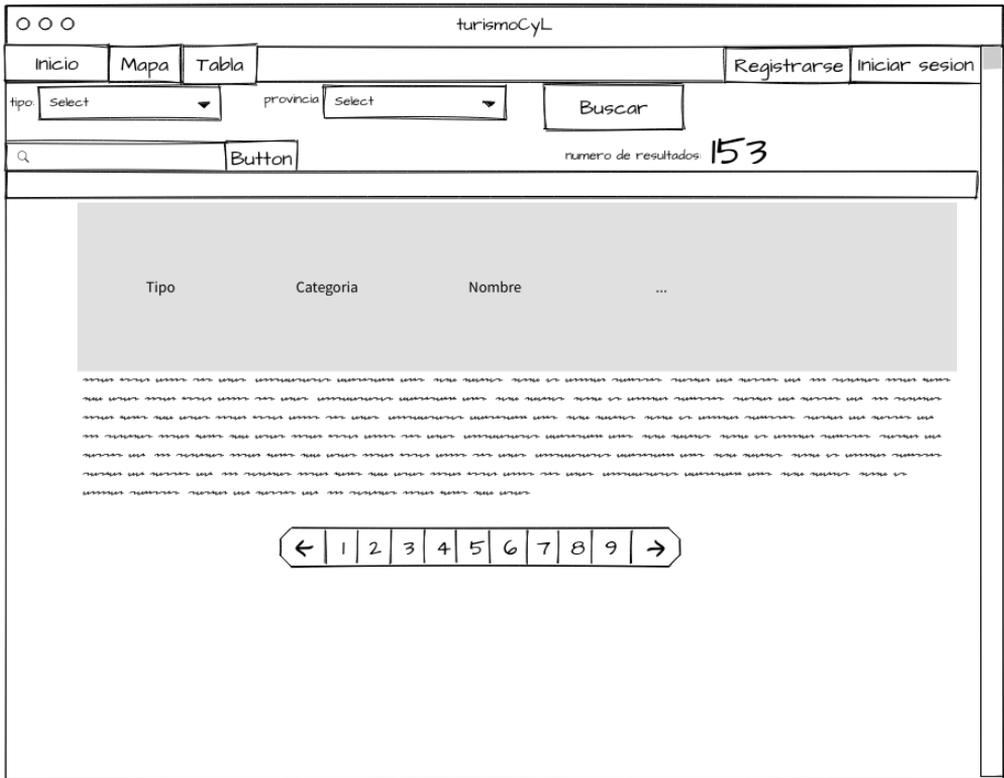


Figura 4.10: Boceto de página de tabla

4.5. DIAGRAMAS DE INTERFAZ DE USUARIO

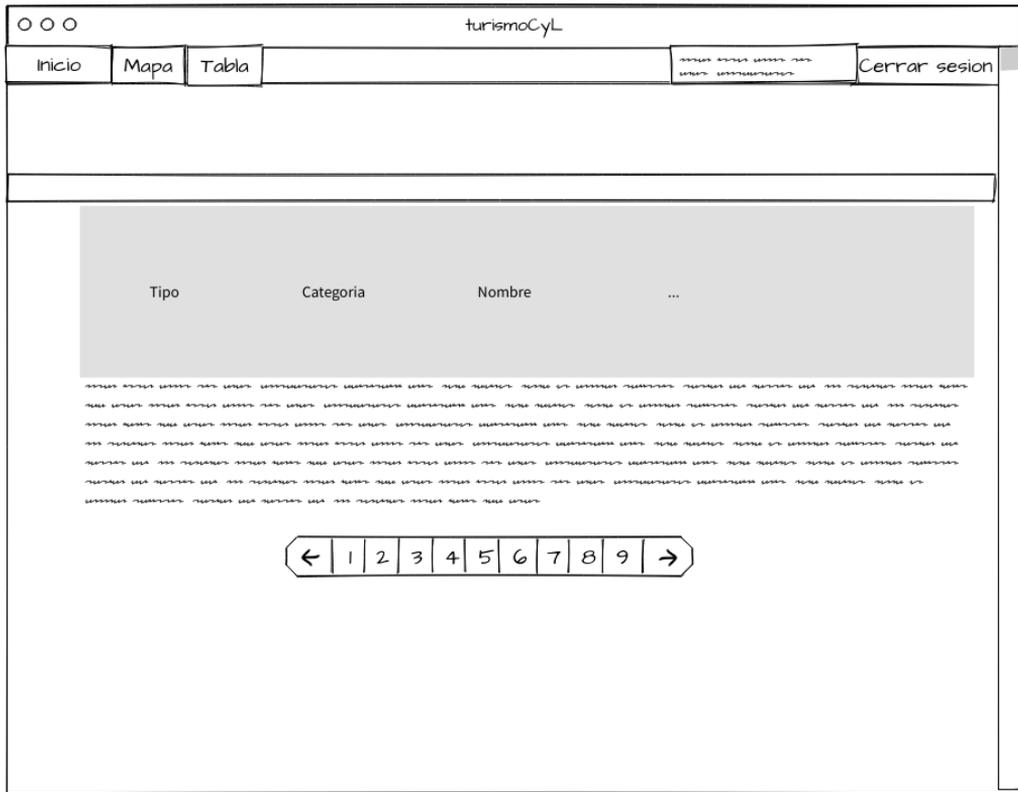


Figura 4.11: Boceto de página de favoritos

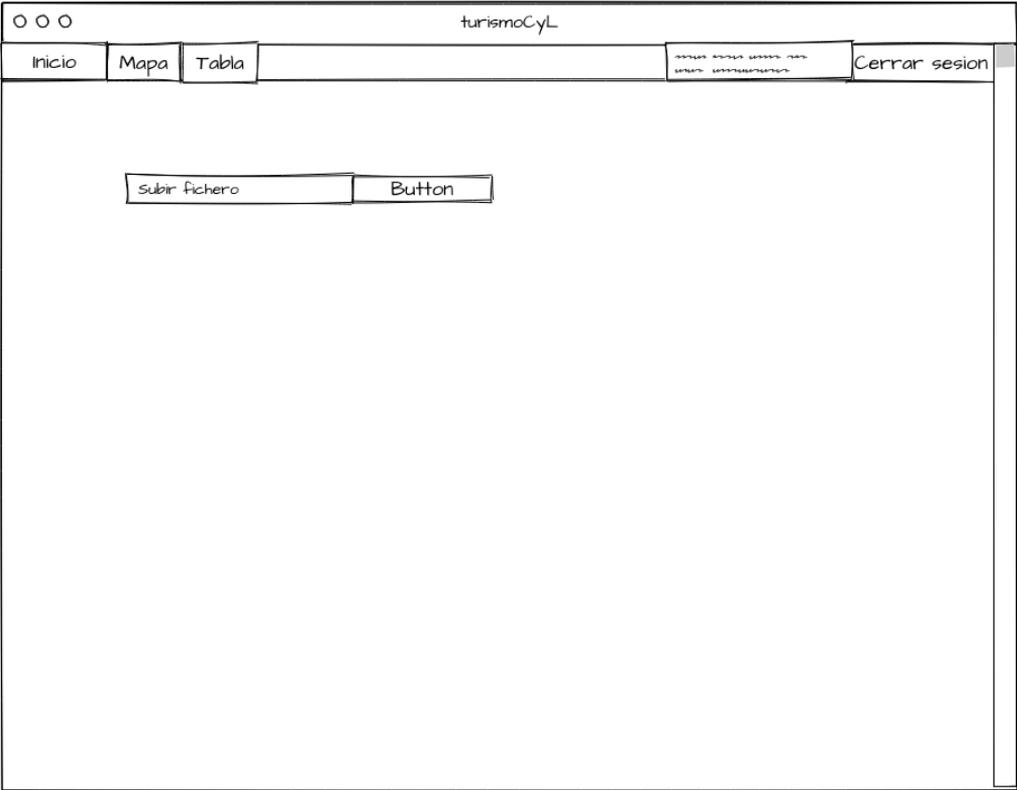


Figura 4.12: Boceto de página de subida de fichero de monumentos



Figura 4.13: Boceto de PopUp de inicio sesión

The image shows a wireframe of a registration popup form. At the top, there is a header bar with three small circles on the left and the text 'turismoCyL' on the right. Below the header, there is a navigation bar with four buttons: 'Inicio', 'Mapa', 'Tabla', and 'Registrarse', followed by 'Iniciar sesion'. The main content area is a large rectangle containing a smaller rectangle titled 'Registrarse'. Inside this rectangle, there are three input fields: 'Email', 'Password', and 'Repeat Password'. Below these fields is a dark button labeled 'Registrarse'.

Figura 4.14: Boceto de PopUp de registrarse

Capítulo 5

Implementación

En esta parte del documento hablaremos del desarrollo e implementación del código de la aplicación, así como de las herramientas y tecnologías utilizadas para ello.

5.1. Herramientas y Tecnologías

5.1.1. Herramientas de desarrollo

- **Visual Studio Code:** Editor de código utilizado para el desarrollo de la aplicación. Cuenta con soporte para utilizar Git.
- **Google Chrome:** Navegador utilizado para realizar las pruebas durante el desarrollo y el buen funcionamiento de la aplicación al finalizarla.
- **Git:** Sistema de control de versiones creado por Linus Torvalds. Es el sistema más utilizado en el mundo.
- **GitHub:** Portal basado en Git, el cual permite gestionar proyectos software y controlar sus versiones. Permite crear repositorios que pueden enlazarse con los repositorios Git en local. Estos repositorios pueden ser públicos o privados.
- **Heroku:** Heroku es una “Plataforma como Servicio” (PaaS: Platform as a Service), a su vez, es una de las más empleadas actualmente. Permite desplegar aplicaciones en la nube y permite una gran variedad de lenguajes de programación. También puede enlazarse con un repositorio de GitHub para utilizar la aplicación subida en este.
- **API-REST Junta CyL:** Api utilizada para sacar datos abiertos, relacionados con el turismo, de la Junta de Castilla y León. La Api en concreto es la *ODS Search V1* o lo que es lo mismo *OpenDataSoft search Api Version 1*, la cual pertenece a OpenDataSoft.
- **Google Maps JavaScript API:** Api, perteneciente a Google, utilizada para el desarrollo del apartado del mapa en la aplicación.

- **Google Maps Angular:** Módulo para Angular que, permite la utilización de la Api de Google Maps, ya que, Angular es TypeScript, lo cual da algunos problemas con la utilización directa de la Api de Google.
- **Google Maps Geocoding API:** Api, perteneciente a Google, utilizada para el desarrollo del apartado del mapa en la aplicación. En concreto para la obtención de la latitud y longitud de gran parte de los sitios de los datos de la Junta.

5.1.2. Herramientas de soporte

- **Astah UML:** Aplicación que permite el desarrollo de diagramas UML.
- **MockFlow:** Herramienta online para la realización de Wireframes. Se ha utilizado para realizar los bocetos de la interfaz.
- **Creately:** Herramienta online para la realización de diagramas y gráficos. Se ha utilizado para realizar los diagramas de arquitectura.
- **Overleaf:** Herramienta online para realizar documentos, basada en Latex.

5.1.3. Tecnologías utilizadas

- **Firebase:** Firebase es una plataforma en la nube, perteneciente a Google, que, permite el desarrollo y monitorización de aplicaciones web y móvil. Sus funcionalidades son varias y se pueden agrupar en 4 grupos, aunque, para esta aplicación hemos utilizado 2 de ellas.
 - **RealTime Database:** Base de datos en tiempo real, de tipo NoSQL con un formato parecido a Json.
 - **Authentication:** Sistema de autenticación de usuarios provisto por Firebase. Este permite registrar usuarios tanto por correo electrónico, como utilizando diferentes redes sociales, tales como Facebook o Twitter.
- **Angular 11:** Framework, perteneciente a Google, para facilitar la creación de aplicaciones web SPA (Single Page Application), el cual hace uso del lenguaje TypeScript.
- **TypeScript:** Lenguaje de programación basado en JavaScript, que hace una serie de cambios para reducir el número de errores y hacerlo más fácil de mantener.

5.2. Rendimiento y precios de Firebase y API's de Google

Para la aplicación, hay que tener en cuenta que, tanto para Firebase como para las API's de Google, se están utilizando, o el plan gratuito o el más barato, por lo que hay que tenerlo en cuenta a la hora de utilizar la aplicación, esto siendo sobretodo cierto a la hora de hacer muchas peticiones en el mapa, ya que, la API de Geocoding limita las peticiones por minuto.

5.2.1. Firebase

Para Firebase, de los dos planes disponibles, se está usando el plan Blaze, el cual es de pago pero, que incluye los servicios del plan Spark, el cual es gratuito. Esto es así ya que, al ser Firebase parte de Google y necesitar, tanto la API de mapas como la de Geocoding, de una tarjeta de crédito para su funcionamiento, el plan de Firebase se pone de forma automática en Blaze. Para la aplicación no es un problema, ya que, tanto RealTime Database como Authentication, se utilizan poco, por lo que no se tiene ningún gasto por la parte de Firebase.

En cuanto al rendimiento de Firebase.

- **Authenticate:** No hay límite de peticiones de autenticación en la aplicación, por lo tanto no hay problemas de rendimiento por esta parte.
- **RealTime Database:** El plan Blaze permite hasta 200.000 conexiones simultáneas con la base de datos y no hay un máximo de GB de almacenamiento ni de descargas mensuales; mientras que, el plan Spark permite hasta 100 conexiones simultáneas, un máximo de 10 GB de almacenamiento y un tope de 10 GB descargados por mes. Como se ha dicho antes, primero se gasta la parte gratuita del plan Spark y luego se empieza a cobrar según el plan Blaze, por lo que, para esta aplicación, por lo menos para el desarrollo de la misma y las pruebas realizadas para su defensa, no se llegarán a los topes de plan Spark.

5.2.2. API's de Google

Sobre las dos API's de Google, mapas y geolocalización, ambas cobran por su uso, pero dando 500 \$ de gasto al crear la aplicación, y 200 \$ para gastar cada mes. Durante la realización de la aplicación, y las pruebas al finalizar esta, se ha superado este gasto total de 700 \$, durante el mes de mayo, por lo que el uso de la aplicación, en su parte del mapa, ha requerido de un gasto monetario.

A continuación, los precios de cada API.

RANGO DE VOLUMEN MENSUAL (precio por CARGA DE MAPA)		
0-100,000	100,001-500,000	Más de 500,000
USD 0.007 por cada una (USD 7.00 cada 1,000)	USD 0.0056 por cada una (USD 5.60 cada 1,000)	Para obtener información sobre los precios por volumen, comunicate con el equipo de Ventas.

Figura 5.1: Precio de Google Maps JavaScript API

5.2. RENDIMIENTO Y PRECIOS DE FIREBASE Y API'S DE GOOGLE

RANGO DE VOLUMEN MENSUAL (precio por SOLICITUD)		
0-100,000	100,001-500,000	Más de 500,000
USD 0.005 por cada uno (USD 5.00 cada 1,000)	USD 0.004 por cada uno (USD 4.00 cada 1,000)	Para obtener información sobre los precios por volumen, comunicate con el equipo de Ventas.

Figura 5.2: Precio de Google Maps Geocoding API

En cuanto al rendimiento, a continuación los límites de llamadas a cada API.

Nombre de la cuota	Límite
Map loads por día	900 
Map loads por minuto	30,000 
Map loads por minuto por usuario 	300 

Figura 5.3: Limite de Google Maps JavaScript API

Nombre de la cuota	Límite
Requests por día	Ilimitado 
Requests por minuto	3,000 
Requests por minuto por usuario 	Ilimitado 

Figura 5.4: Limite de Google Maps Geocoding API

Los límites en Google Maps JavaScript API no afectan a la aplicación, para el uso dado actualmente; pero, los límites en Google Maps Geocoding API si, ya que, algunas peticiones como, todos los bares o todos los restaurantes, necesitan de mas de 3000 peticiones a la API, esto provoca un cuello de botella y hace que la página se ralentice intentando conseguir los

datos de latitud y longitud de los diferentes sitios. La forma de arreglar esto sería aumentando ese límite, pero entonces, se sobrepasaría el límite de 200 \$ muy rápido. Si la aplicación se llega a usar de forma comercial en algún momento se cambiaría ese límite pero, para el desarrollo de la misma no se ha tocado.

5.3. Estructura del Proyecto

La estructura del proyecto general es la siguiente.

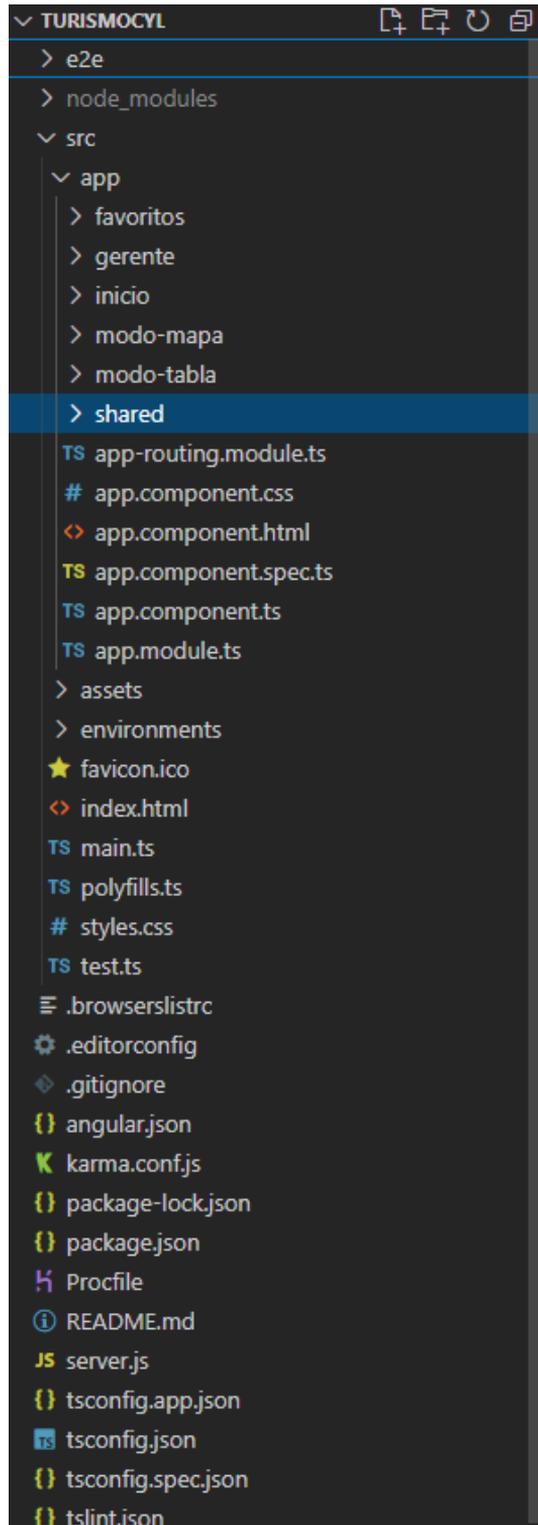


Figura 5.5: Estructura general del proyecto

Como se puede ver, la estructura es la de un proyecto Angular, donde las carpetas dentro de app, menos shared, corresponden cada una a un componente. Por ejemplo.

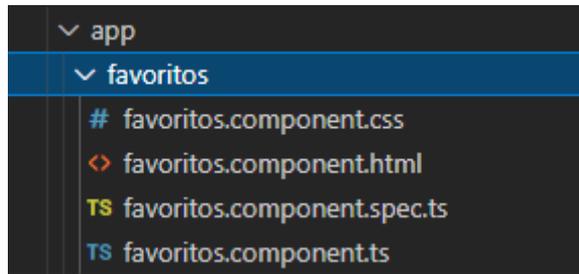


Figura 5.6: Estructura componente Angular ejemplo

El archivo `component.ts` contiene la lógica del componente, el `.html` la vista y el `.css` el estilo de la vista. El `.spec.ts` contiene los test a realizar, aunque para esta aplicación no se ha usado ninguno de estos archivos.

La carpeta `shared` contiene los servicios de la aplicación, junto a los *Modal*, los cuales están en la carpeta `dialog`.

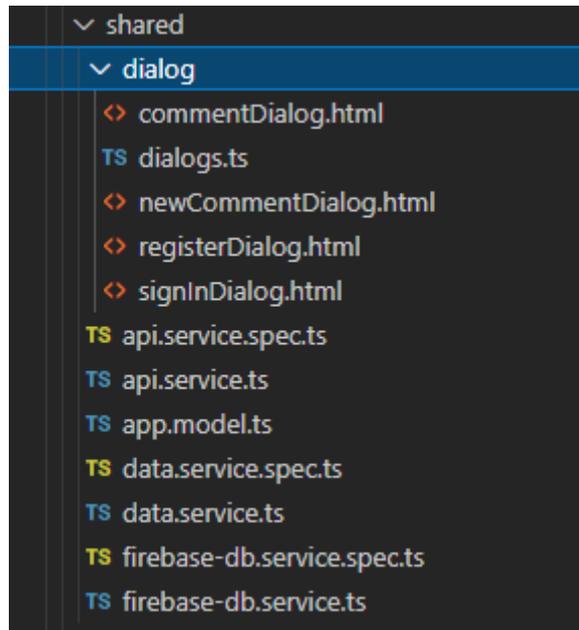


Figura 5.7: Estructura carpeta shared

Aquí también podemos encontrar el fichero `api.service.ts`, el cual contiene la lógica correspondiente a la API de datos de la Junta, `firebase-db.service.ts` tiene la lógica de Firebase,

`data.service.ts` contiene la lógica correspondiente a los datos necesarios para pasar de la tabla o favoritos al mapa, seleccionado un sitio; y, por último, `app.model.ts` contiene los modelos que se utilizan en la aplicación.

5.4. Implementación

En esta sección se describirá como crear el proyecto básico en Angular, así como la creación de un proyecto en Firebase.

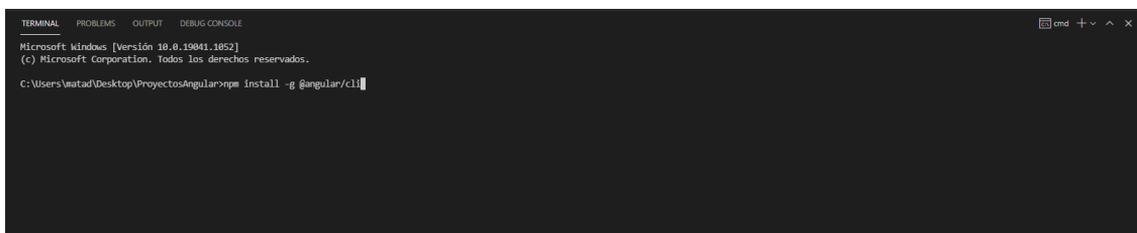
5.4.1. Proyecto Angular

Para la creación del proyecto, suponemos que tenemos instalado un IDE o un editor de texto adecuado para el desarrollo en Angular. En nuestro caso usaremos Visual Studio Code.

Para la creación básica de una aplicación Angular, primero nos instalamos Node.js, junto al gestor de paquetes NPM.

Para instalarlo, iremos a su página web (<https://nodejs.org/>) y nos instalamos la versión recomendada para el equipo. Después de descargarlo lo instalaremos.

Para instalar Angular, desde la consola de comandos ejecutaremos el siguiente comando `npm install -g @angular/cli`, lo cual instalará Angular en el equipo.

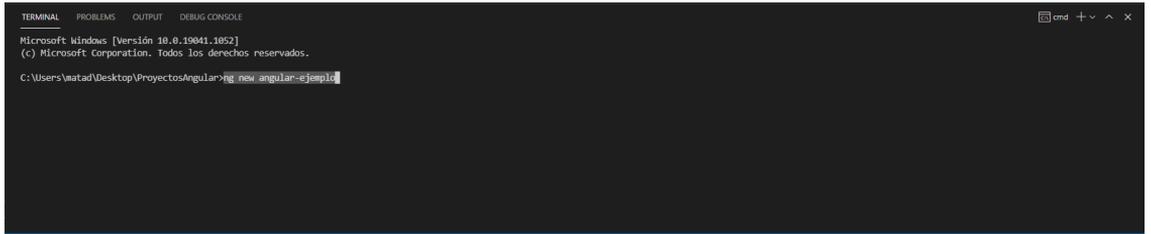


```
Microsoft Windows [Versión 10.0.19041.1052]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\matad\Desktop\ProyectosAngular>npm install -g @angular/cli
```

Figura 5.8: Instalación Angular

Una vez la instalación haya terminado, podemos crear el proyecto Angular básico con el siguiente comando `ng new angular-ejemplo`, donde `angular-ejemplo` sería el nombre del proyecto Angular a crearse.

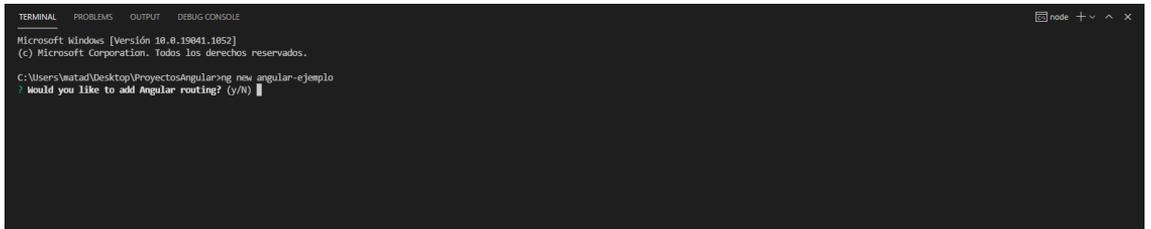


```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
Microsoft Windows [Versión 10.0.19041.1852]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\matad\Desktop\ProyectosAngular>ng new angular-ejemplo
```

Figura 5.9: Crear proyecto Angular

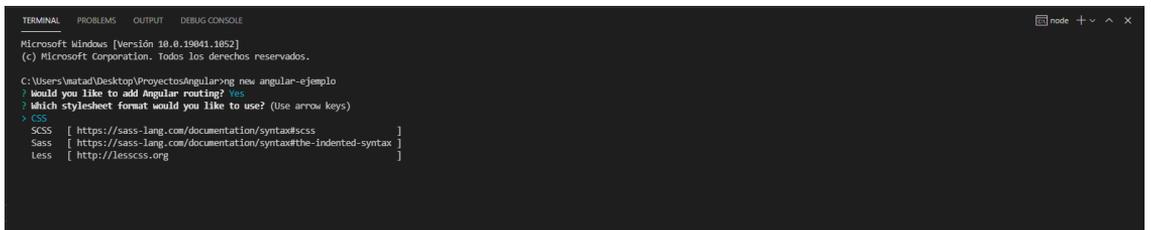
Durante la creación del proyecto, se nos harán dos preguntas; la primera es si queremos que el *routing* de la aplicación se cree de forma automática, a lo que hemos respondido que sí; la segunda es el formato que queremos para las hojas de estilo, en nuestro caso escogemos CSS.



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
Microsoft Windows [Versión 10.0.19041.1852]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\matad\Desktop\ProyectosAngular>ng new angular-ejemplo
? Would you like to add Angular routing? (y/n) █
```

Figura 5.10: Pregunta sobre routing



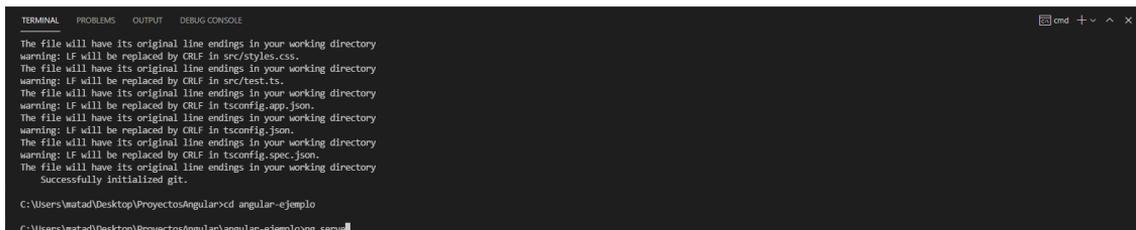
```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
Microsoft Windows [Versión 10.0.19041.1852]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\matad\Desktop\ProyectosAngular>ng new angular-ejemplo
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SASS [ https://sass-lang.com/documentation/syntax#scss ]
  Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]
```

Figura 5.11: Pregunta sobre hoja de estilo

Una vez haya terminado el proceso, tendremos la aplicación creada, con la estructura de Angular, y, para poder lanzar la aplicación, sólo tenemos que movernos al directorio raíz de la misma y ejecutar el comando *ng serve*.

5.4. IMPLEMENTACIÓN

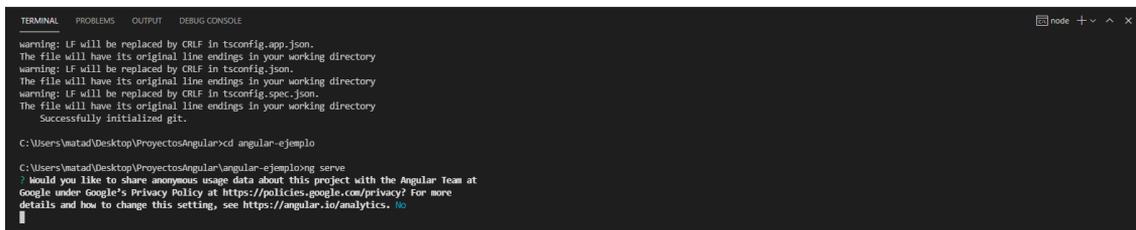


```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
Successfully initialized git.

C:\Users\matad\Desktop\ProyectosAngular>cd angular-ejemplo
C:\Users\matad\Desktop\ProyectosAngular\angular-ejemplo>ng serve
```

Figura 5.12: Lanzar aplicación Angular

La primera vez que lanzemos la aplicación, nos preguntará si queremos compartir datos anónimos de la misma con Google y el equipo de Angular, a lo cual hemos respondido que no.



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
Successfully initialized git.

C:\Users\matad\Desktop\ProyectosAngular>cd angular-ejemplo
C:\Users\matad\Desktop\ProyectosAngular\angular-ejemplo>ng serve
Would you like to share anonymous usage data about this project with the Angular Team at
Google under Google's Privacy Policy at https://policies.google.com/privacy? For more
details and how to change this setting, see https://angular.io/analytics. io
```

Figura 5.13: Pregunta sobre compartir datos

Cuando la aplicación se lance en el navegador, se habrá lanzado en <http://localhost:4200/> y veremos lo siguiente.

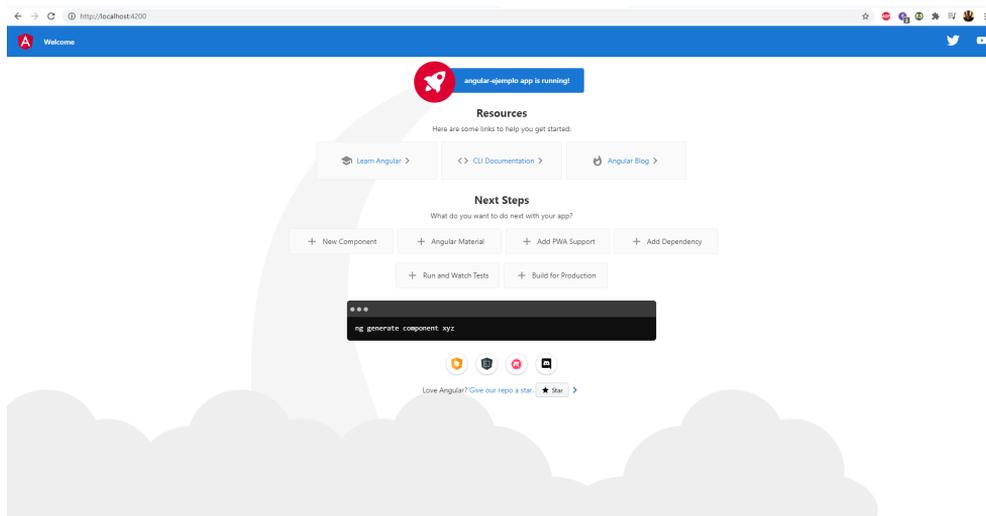


Figura 5.14: Página principal

5.4.2. Proyecto en Firebase

Para crear un proyecto en Firebase iremos a la siguiente página <https://console.firebase.google.com/> y, si es el primero que creamos, nos aparecerá lo siguiente.

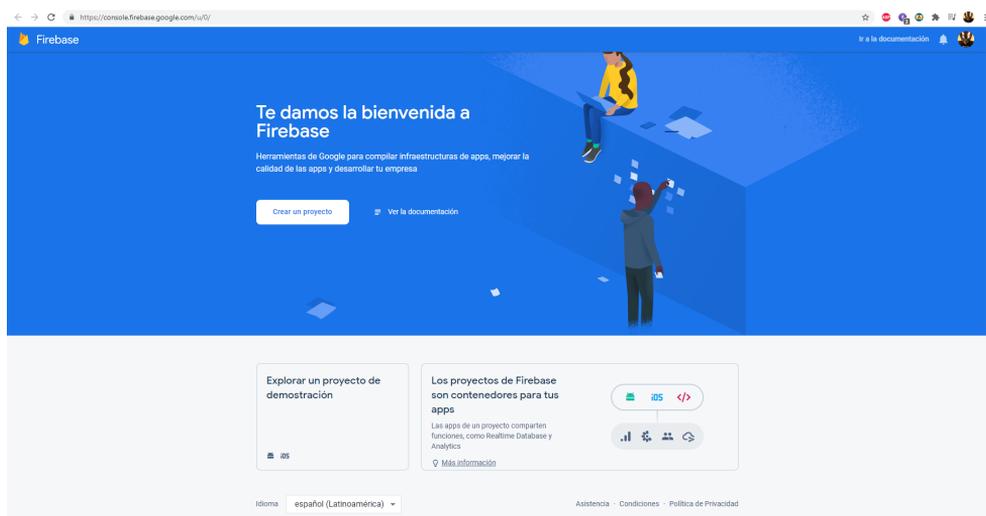


Figura 5.15: Página principal Firebase

Al seleccionar crear proyecto nos pedirá el nombre del mismo, si queremos añadir Google Analytics, y, si seleccionamos que sí, la ubicación para ello.

5.4. IMPLEMENTACIÓN

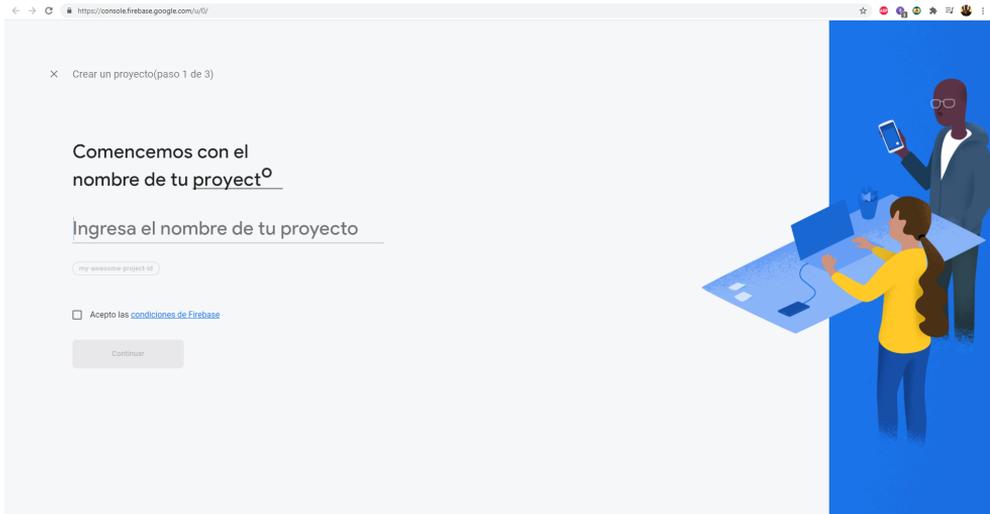


Figura 5.16: Crear Aplicación Firebase nombre

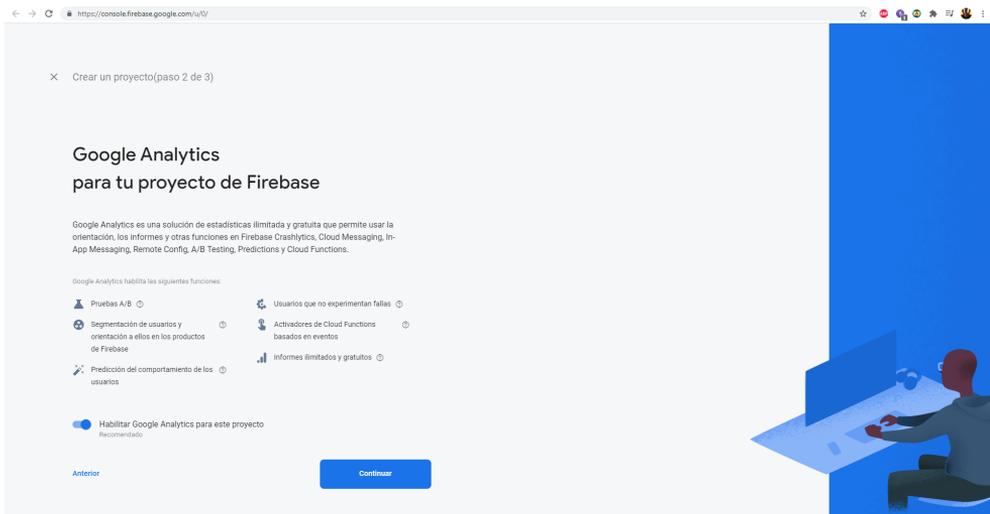


Figura 5.17: Crear Aplicación Firebase Analytics

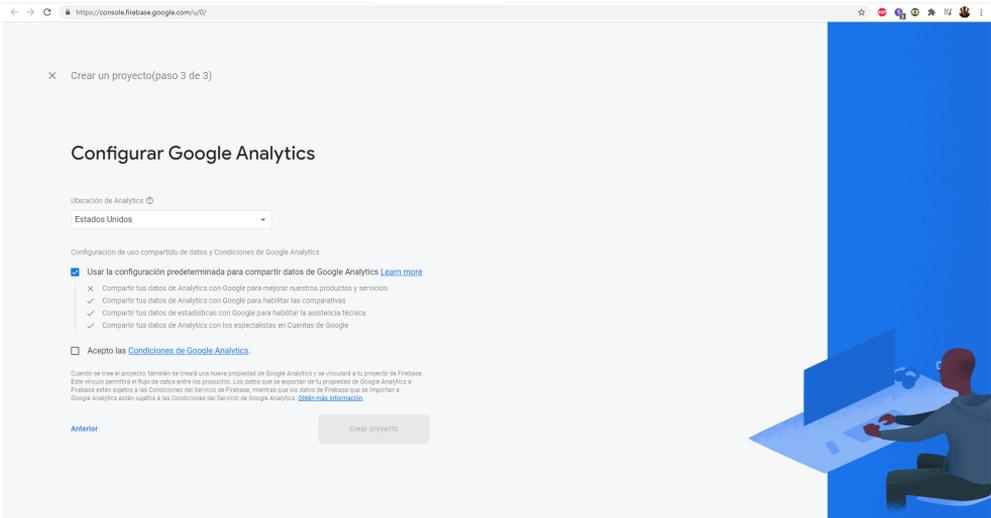


Figura 5.18: Crear Aplicación Firebase Ubicación Analytics

Una vez creada la aplicación, en el menú izquierdo seleccionamos Compilación, y dentro de ello Authentication.

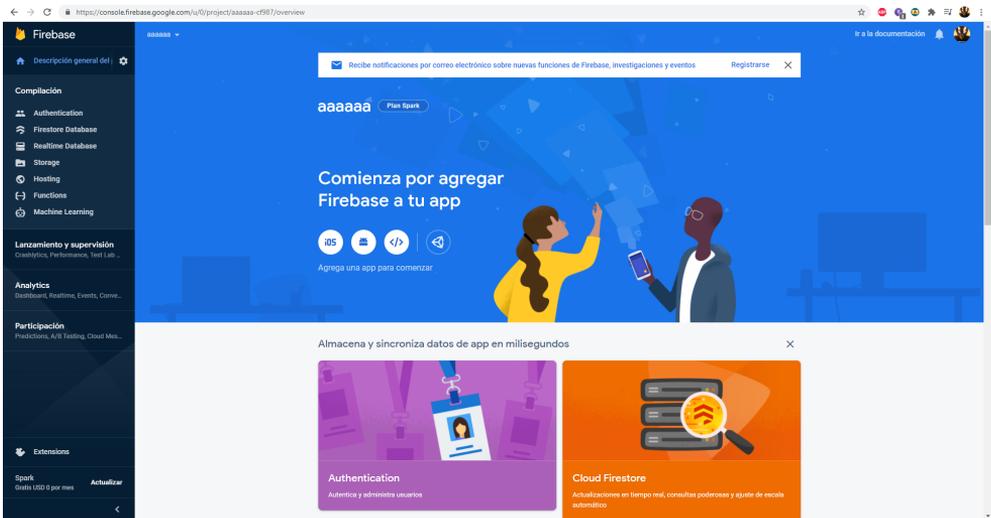


Figura 5.19: Aplicación Firebase creada

5.4. IMPLEMENTACIÓN

Dentro del apartado de Authentication, añadimos el servicio y seleccionamos, de entre las opciones posibles para la autenticación la de correo para habilitarla.

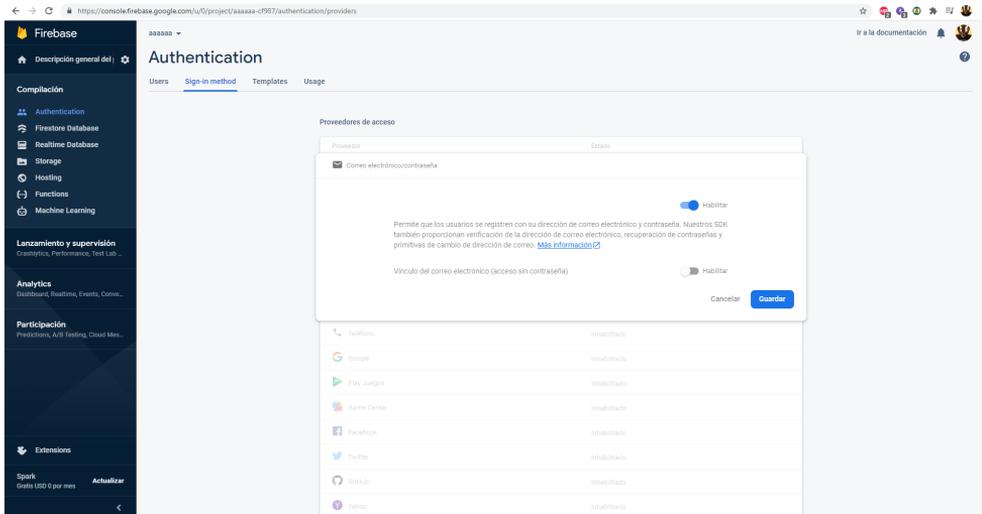


Figura 5.20: Authentication correo

Después, en el menú lateral, seleccionamos RealTime Database y añadimos el servicio con las opciones predeterminadas, con lo cual, nos queda algo como lo siguiente.

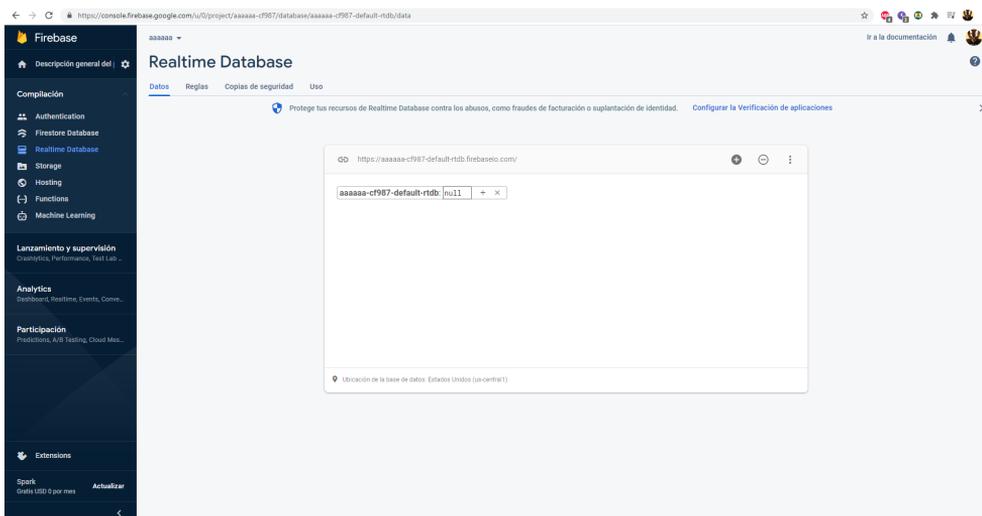


Figura 5.21: RealTime Database nueva

Una vez creada la base de datos, volvemos a la página de inicio de la aplicación y seleccionamos añadir aplicación web. Primero nos pedirá el nombre de la aplicación y después, nos dará un código que hay que copiar en la aplicación para poder conectarla con Firebase.

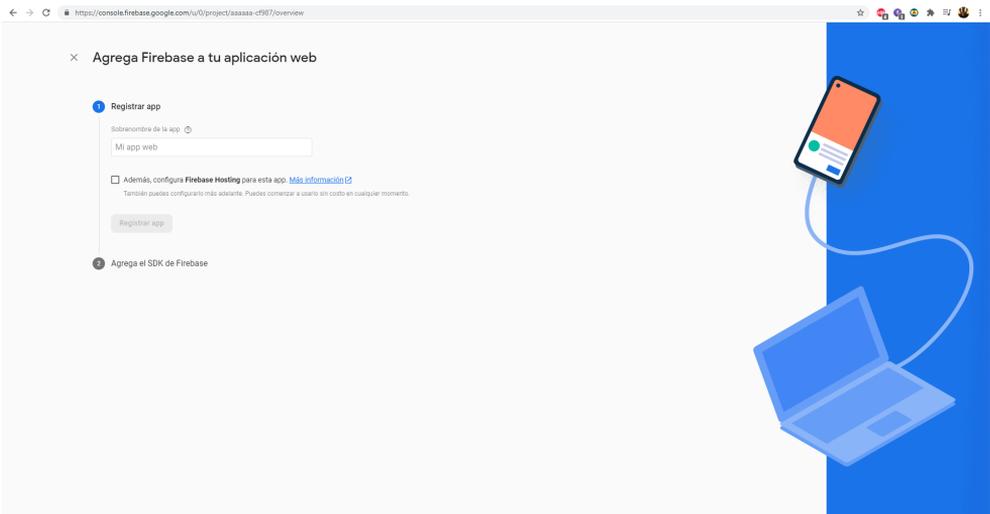


Figura 5.22: Nombre aplicación web para Firebase

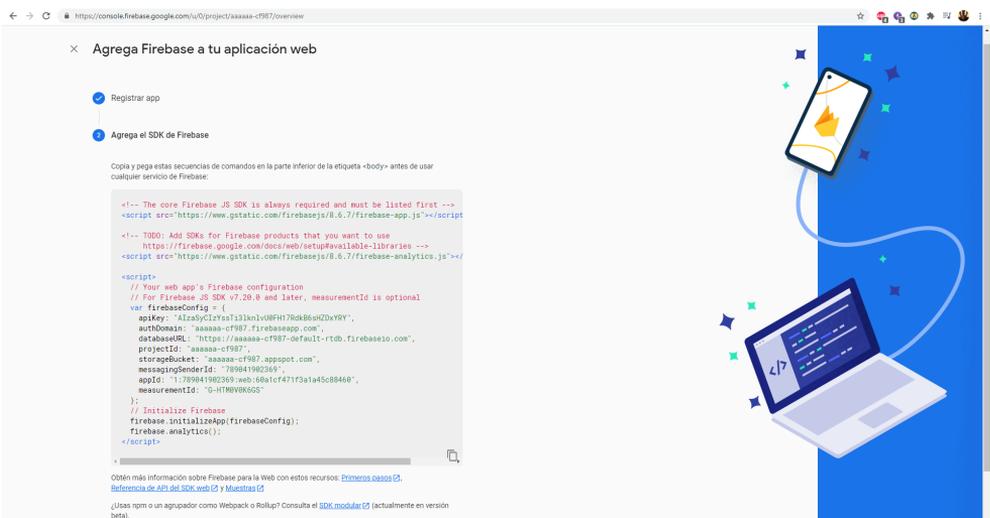


Figura 5.23: Código de Firebase a copiar

5.5. Heroku

Para el despliegue del Front-End de la aplicación hemos escogido Heroku para realizar el despliegue. Nos hemos decantado por esta herramienta ya que, como hemos explicado en el apartado de Herramientas de Desarrollo, Heroku permite conectar la aplicación en Heroku con un repositorio de GitHub, con lo que, todo cambio realizado en el repositorio se va a ver aplicado en Heroku. Además, al buscar posibles herramientas donde desplegar el Front-End, vimos que, desplegar una aplicación Angular en Heroku es muy sencillo.

Como ya se ha dicho en el resumen y el abstract, la aplicación está desplegada en <https://turismocyl.herokuapp.com/>.

Capítulo 6

Pruebas

En este capítulo se encuentran las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación. Para ello se han llevado a cabo pruebas de caja negra.

6.1. Pruebas de Caja Negra

Las pruebas de caja negra se llevan a cabo de tal forma que, no es necesario saber como funciona la aplicación o tener el código durante el desarrollo de las pruebas.

A continuación se encuentran las pruebas de caja negra realizadas, todas ellas con resultado satisfactorio.

- PCN-01: Registrar nuevo usuario.
- PCN-02: Buscar sitios, de primer conjunto de datos, por tipo en el mapa.
- PCN-03: Buscar sitios, de segundo conjunto de datos, por tipo en el mapa.
- PCN-04: Buscar sitios, de tercer conjunto de datos, por tipo en el mapa.
- PCN-05: Buscar sitios por provincia en el mapa.
- PCN-06: Buscar sitios, de primer conjunto de datos, por tipo y provincia en el mapa.
- PCN-07: Buscar sitios, de segundo conjunto de datos, por tipo y provincia en el mapa.
- PCN-08: Buscar sitios, de tercer conjunto de datos, por tipo y provincia en el mapa.
- PCN-09: Buscar sitios por texto en el mapa.

- PCN-10: Ver ventana con información al clicar un marcador.
- PCN-11: Abrir ver comentarios, desde una ventana con información abierta.
- PCN-12: Error al intentar añadir a favoritos, desde una ventana con información abierta, si no has iniciado sesión previamente.
- PCN-13: Buscar sitios, de primer conjunto de datos, por tipo en la tabla.
- PCN-14: Buscar sitios, de segundo conjunto de datos, por tipo en la tabla.
- PCN-15: Buscar sitios, de tercer conjunto de datos, por tipo en la tabla.
- PCN-16: Buscar sitios por provincia en la tabla.
- PCN-17: Buscar sitios, de primer conjunto de datos, por tipo y provincia en la tabla.
- PCN-18: Buscar sitios, de segundo conjunto de datos, por tipo y provincia en la tabla.
- PCN-19: Buscar sitios, de tercer conjunto de datos, por tipo y provincia en la tabla.
- PCN-20: Buscar sitios por texto en la tabla.
- PCN-21: Ordenar por atributo en una tabla con sitios previamente buscados.
- PCN-22: Pasar de grupo de 5 sitios, en una tabla con sitios previamente buscados.
- PCN-23: Buscar, por texto, entre los sitios previamente buscados.
- PCN-24: Ir al mapa, desde la tabla, con un sitio seleccionado.
- PCN-25: Abrir ver comentarios, desde una tabla.
- PCN-26: Error al intentar añadir a favoritos, desde una tabla, si no has iniciado sesión previamente.
- PCN-27: Error al intentar añadir un comentario si no has iniciado sesión previamente.
- PCN-28: Ordenar los comentarios por fecha o puntuación.
- PCN-29: Iniciar sesión.
- PCN-30: Añadir un sitio a favoritos tanto, desde el mapa o desde la tabla.
- PCN-31: Añadir un nuevo comentario.
- PCN-32: Ver tus sitios favoritos.
- PCN-33: Borrar un sitio de favoritos.
- PCN-34: Ir al mapa, desde favoritos, con un sitio seleccionado.
- PCN-35: Abrir ver comentarios, desde favoritos.
- PCN-36: Ordenar por atributo en favoritos.
- PCN-37: Buscar, por texto, entre los sitios favoritos.

- PCN-38: Salir de sesión.
- PCN-39: Iniciar sesión como gerente.
- PCN-40: Actualizar los monumentos.

Capítulo 7

Conclusiones

Este proyecto tenía cinco objetivos principales, el primero hace referencia al propósito de la aplicación, los 3 siguientes hacían referencia al desarrollo de la aplicación en si, mientras que el último referenciaba la mejora de los conocimientos adquiridos en la carrera. Después de haber finalizado el trabajo, concluyo que se han cumplido los 5 objetivos planeados.

Sobre el primero concluyo que, se ha cumplido el objetivo de desarrollar una aplicación web con el propósito de dar información turística sobre diferentes sitios para visitar en Castilla y León.

Sobre los 3 siguientes, se han desarrollado todas las funcionalidades de la aplicación, completando así una aplicación web sobre sitios que visitar en Castilla y León utilizando los datos abiertos dados por la Junta,

En relación al último, si que he desarrollado más mis conocimientos en las tecnologías utilizadas, las cuales ya había usado en la carrera.

Sin embargo, considero que la aplicación tiene margen de mejora futuro en ciertos aspectos, los cuales trato en el siguiente capítulo.

Capítulo 8

Posibles futuras mejoras

Aunque se han cumplido todos los objetivos planteados al comienzo del desarrollo de la aplicación, considero que se podrían hacer ciertas mejoras, a la misma, en un futuro. Estas mejoras podrían ser:

- Si a los datos abiertos de la Junta, se añaden nuevos conjuntos que tengan que ver con el turismo, añadir estos conjuntos de datos a la aplicación.
- Cambiar los iconos de los marcadores en el mapa a un icono distinto por cada tipo de establecimiento, lo cual no se ha podido hacer por restricciones a la hora de cambiar dichos iconos.
- No limitar el uso de la API de Geocoding, para así mejorar el rendimiento de la aplicación en el apartado del mapa.

Apéndice A

Manual de instalación

Para la instalación y ejecución de la aplicación, basta con tener instalados Angular y Node.js, como viene al inicio de la sección **5.4.1. Proyecto Angular**, descargar y descomprimir la aplicación del repositorio <https://github.com/mastema98/turismoCyL>, ejecutar en la consola de comandos `npm install` para descargar los módulos necesarios para la aplicación, y por último, dentro del directorio raíz de lo que se ha descomprimido, ejecutar en la consola de comandos `ng serve`. De esta forma se tendría el Front-End de la aplicación siendo ejecutado en local.

Bibliografía

- [1] Definición de Datos abiertos <https://datosabiertos.jcyl.es/web/es/iniciativa-datos-abiertos/datos-abiertos.html> Fecha de último acceso: 23/05/2021
- [2] Sobre el Hash para la contraseña del usuario en Firebase <https://firebaseopensource.com/projects/firebase/scrypt/> Fecha de último acceso: 31/05/2021
- [3] Documentación sobre la API-REST que utiliza la Junta para los datos abiertos. <https://help.opendatasoft.com/apis/ods-search-v1/#search-api-v1> Fecha de último acceso: 10/05/2021
- [4] Primer conjunto de datos abiertos de la Junta. <https://analisis.datosabiertos.jcyl.es/explore/dataset/registro-de-turismo-de-castilla-y-leon/information/> Fecha de último acceso: 10/05/2021
- [5] Segundo conjunto de datos abiertos de la Junta, los museos. <https://analisis.datosabiertos.jcyl.es/explore/dataset/museos/information/> Fecha de último acceso: 03/06/2021
- [6] Tercer conjunto de datos abiertos de la Junta, los monumentos. <https://datosabiertos.jcyl.es/web/jcyl/set/es/cultura-ocio/monumentos/1284325843131> Fecha de último acceso: 25/05/2021
- [7] Página utilizada para realizar los bocetos de la interfaz. <https://mockflow.com/> Fecha de último acceso: 08/06/2021
- [8] Página utilizada para el diseño de la barra superior. <https://miventanafavorita.blogspot.com/2016/10/poner-barra-superior-menu-blog.html> Fecha de último acceso: 12/03/2021
- [9] Página sobre HTML. <https://tutorialehtml.com/es/> Fecha de último acceso: 15/03/2021
- [10] Página sobre HTML, CSS y Angular. <https://www.w3schools.com/> Fecha de último acceso: 20/05/2021
- [11] Página sobre el mapa en Angular. <https://timdeschryver.dev/blog/google-maps-as-an-angular-component> Fecha de último acceso: 18/05/2021

- [12] GitHub de Angular Maps. <https://github.com/robisim74/angular-maps> Fecha de último acceso: 18/05/2021
- [13] Documentación oficial de las API's de Google. <https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/reference> Fecha de último acceso: 28/04/2021
- [14] Cookies en Angular. <https://itnext.io/angular-8-how-to-use-cookies-14ab3f2e93fc> Fecha de último acceso: 25/04/2021
- [15] Documentación oficial de Firebase. <https://firebase.google.com/docs/web/setup> Fecha de último acceso: 13/04/2021
- [16] Documentación oficial de Angular. <https://angular.io/> Fecha de último acceso: 13/05/2021
- [17] Documentación oficial de Bootstrap 4. <https://getbootstrap.com/docs/4.0/getting-started/introduction/> Fecha de último acceso: 13/05/2021
- [18] Página sobre JSON en Node. <https://attacomisian.com/blog/nodejs-read-write-json-files> Fecha de último acceso: 15/05/2021
- [19] Documentación oficial de Material Design para Bootstrap 4 y Angular. <https://mdbootstrap.com/docs/angular/getting-started/installation/> Fecha de último acceso: 28/05/2021
- [20] GitHub del componente Angular para las estrellas en la puntuación. <https://github.com/BioPhoton/angular-star-rating> Fecha de último acceso: 28/05/2021
- [21] Como desplegar una aplicación Angular en Heroku. https://medium.com/@roliver_javier/como-desplegar-tu-aplicacion-de-angular-en-heroku-7b9941b6d39 Fecha de último acceso: 31/05/2021
- [22] Como desplegar una aplicación Angular en Heroku 2. <https://betterprogramming.pub/how-to-deploy-your-angular-9-app-to-heroku-in-minutes-51d171c2f0d> Fecha de último acceso: 31/05/2021
- [23] Generador de tablas para Latex, utilizado para la realización de la memoria. <https://www.tablesgenerator.com/> Fecha de último acceso: 16/06/2021
- [24] Información sobre el Hash de Firebase para Authenticate. <https://firebaseopensource.com/projects/firebase/scrypt/> Fecha de último acceso: 24/04/2021
- [25] Presentación sobre datos abiertos, realizada en la Universidad. <https://www.inf.uva.es/2021/05/26/se-celebro-en-el-salon-de-actos-del-edificio-tit-una-sesion-informativa> Fecha de último acceso: 19/06/2021
- [26] Página utilizada para la creación de diagramas. <https://creately.com/es/lp/software-de-diagrama-de-flujo/> Fecha de último acceso: 09/06/2021

- [27] Información sobre la facturación y precios de las API's de Google. <https://developers.google.com/maps/billing/gmp-billing?hl=es-419> Fecha de último acceso: 12/06/2021
- [28] Información sobre la facturación y precios de las API's de Google 2. <https://cloud.google.com/maps-platform/pricing> Fecha de último acceso: 12/06/2021
- [29] Página utilizada para la búsqueda de precios medios. <https://es.indeed.com/> Fecha de último acceso: 19/06/2021
- [30] Página de Wikipedia sobre el modelo de prototipos. https://es.wikipedia.org/wiki/Modelo_de_prototipos Fecha de último acceso: 19/06/2021