



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Aplicación fullstack Mi Finquita

Fullstack application Mi Finquita

Javier Martín de León

La Laguna, 14 de junio de 2022

D. **Santiago Torres Alvarez**, con N.I.F. 43.369.478-B, profesor Contratado Doctor adscrito al Departamento de Ingeniería informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Jesús Miguel Torres Jorge**, con N.I.F. 43.826.207-Y, profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Aplicación fullstack Mi Finquita"

ha sido realizada bajo su dirección por D. **Javier Martín de León**, con N.I.F. 54.116.070-Z.

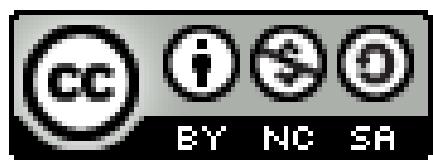
Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firman la presente en La Laguna a 14 de junio de 2022

Agradecimientos

A Santiago Torres Álvarez, Jesús Miguel Torres Jorge y José Fernando Peraza Hernández por apoyarme durante el desarrollo de mi proyecto y por proporcionarme la idea de este trabajo.

Y a mi familia y amigos, siempre mi apoyo incondicional en todo lo que hago.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El producto final realizado “Mi Finquita” surge de la realización de un proyecto fullstack destinado, por un lado, a proveer de información útil en el sector agrícola en cuanto a la monitorización y mantenimiento de explotaciones agrícolas y, por otro lado, a la visualización ágil y dinámica de información de estaciones meteorológicas, ya sean propias de la explotación, de su entorno, o de una región determinada.

El objetivo del proyecto es el de la visualización de datos tomados por estaciones meteorológicas. Además, se han incluido otras formas de visualización de datos como puede ser la obtenida a través de la utilización de satélites, Graniot o Sentinel, siendo estos importantes en cuanto a agricultura de precisión se refiere. Con todo esto se pretende establecer un entorno de prueba al cual el usuario podrá acceder y seleccionar, de las diferentes capas de información disponibles, las que quiera visualizar.

Por otro lado, cabe destacar también como finalidad el establecimiento de un rendimiento adecuado de carga de la web, así como de una estética uniforme y que garantice su usabilidad y accesibilidad adaptada.

Para este trabajo se ha utilizado el lenguaje de programación C#, para el desarrollo backend, y Angular para el desarrollo del frontend.

Palabras clave: Angular, ASP.NET Core, Aplicación, Front-End, Back-End, Graniot, Sentinel, Geomet, WMS, SIGPAC

Abstract

The final product “Mi Finquita” arises from the realization of a fullstack project that is destined to provide useful information in the agricultural sector in terms of monitoring and maintenance of agricultural operations and, on the other hand, to visualize agile and dynamic information from meteorological stations, whether they are specific to the plantation, its surroundings, or a specific region.

The objective of the project is to visualize data of the stations but there are also other ways of data visualization implemented such as the use of satellites, Graniot or Sentinel. These satellites are very important in terms of the precision agriculture. With all this, it is intended to establish a test environment in which the user can enter and select from the different layers of information the ones he wants to see.

On the other hand, it is also worth mentioning as a purpose the establishment of an adequate loading performance of the web, as well as an uniform aesthetic that guarantees its usability and adapted accessibility.

For this work, the programming language C# has been used for the backend development and the framework Angular for the development of the frontend.

Keywords: Angular, ASP.NET Core, Aplicación, Front-End, Back-End, Graniot, Sentinel, Geomet, WMS, SIGPAC

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Estado del Arte	2
1.3. Objetivos	4
1.4. Planificación del proyecto	5
1.5. Estructura del documento	6
2. Herramientas y tecnologías utilizadas	7
2.1. Software	7
3. Desarrollo Back-End	9
3.1. Servidor	9
3.2. API Mi Finquita	10
3.3. Base de datos	12
4. Desarrollo Front-End	17
4.1. Prototipo web del proyecto	17
4.2. Estructura del proyecto	19
4.3. Resultado final de la interfaz de usuario	20
5. Funcionamiento de la plataforma	23
5.1. Inicio de sesión	23
5.2. Dibujar mapa de la aplicación	24
5.3. Creación de las fincas	25
5.4. Capas de mapa	28
5.5. Uso de Graniot en la página	30
5.6. Uso de Sentinel en la página	32
5.7. Uso Geomet en la página	33
5.8. Información de las estaciones	34
6. Testing y despliegue	35
6.1. Testing Front-End	35
6.2. Testing Back-End	37
7. Presupuesto	38
8. Conclusiones y líneas futuras	40
8.1. Conclusiones y líneas futuras	40
8.2. Summary and Conclusions	41

Índice de Figuras

1.1. Aplicación de GOIA	2
1.2. Aplicación de Graniot	2
1.3. Diagrama de Gantt del proyecto	5
3.1. Estructura servidor	9
3.2. Servicios de autentificación	10
3.3. Servicios del mapa	11
3.4. Servicios perfil de usuario	11
3.5. Estructura base de datos DBeaver	12
3.6. Model E/R Geometrías	13
3.7. Model E/R Fincas	14
3.8. Model E/R Openapi	15
3.9. Model E/R Publico	16
4.1. Home del prototipo de la aplicación	17
4.2. Inicio de sesión en la demo del prototipo	18
4.3. Demo del prototipo	18
4.4. Estructura de archivos Front-End	19
4.5. Página de inicio	20
4.6. Página de inicio	21
4.7. Página de la demo	21
4.8. Página del perfil	22
5.1. Visualización de la Identity Base de datos	23
5.2. Visualización del mapa de la aplicación	24
5.3. Crear nueva finca	27
5.4. Capas de mapa	28
5.5. Capas ejemplo Topográfica y Catastro	29
5.6. Visualización Graniot NDVI	30
5.7. Visualización Graniot NDMI	30
5.8. Visualización Graniot LCI	31
5.9. Visualización Graniot OSAVI	31
5.10 Ejemplo NDVI Sentinel	32
5.11 Carga en la base de datos de los índices de Sentinel implementados	32
5.12 Ejemplo Viento	33
5.13 Ejemplo Lluvia	33
5.14 Ejemplo sensor de la estación	34

Índice de Tablas

7.1. Presupuesto recursos humanos	39
7.2. Presupuesto recursos materiales	39

Capítulo 1

Introducción

1.1. Motivación

La agricultura en nuestras islas, en cuanto a tecnología se refiere, actualmente no está al día. Hay infinidad de aplicaciones que nos ayudan a gestionar nuestras explotaciones agrícolas y, en los últimos años sobre todo, lo que más ayuda a estas plantaciones es la agricultura de precisión. La tendencia actual en el sector es avanzar hacia la Agricultura 4.0, que consiste en aplicar técnicas de digitalización, Big Data, internet de las cosas (IoT) y Robótica para conseguir producciones agrarias más eficientes y sostenibles, que incorporen gestión de la calidad y trazabilidad de los productos, y con herramientas que ayuden al agricultor en la toma de decisiones respecto a sus producciones agrícolas [1]

La agricultura de precisión no es más que una forma de gestión de la información cuyo principal objetivo es mejorar las producciones agrícolas a través de la observación, la medida y la actuación ante las diferentes fuentes de información disponibles. Requiere de un conjunto de tecnologías formado por sistemas de información geográfica, sistema global de navegación por satélite y sensores.

La utilización de la información que se obtiene mediante la agricultura de precisión ha significado una evolución en la agricultura mundial y en la actualidad se encuentran en desarrollo infinidad de aplicaciones para gestionarla.

Nos encontramos con una tecnología en continua evolución y que en la actualidad aún no se está utilizando todo lo que se debería, sobre todo en nuestras islas, lo que motiva la realización de este trabajo.

1.2. Estado del Arte

En la actualidad existe un abanico pequeño en cuanto a páginas o aplicaciones que nos permitan gestionar nuestras explotaciones agrícolas. Entre ellas resalta la aplicación de GOIA[2], la cual permite al usuario introducir datos de sus explotaciones y mantener de esta forma actualizado el cuaderno de campo.

La aplicación de GOIA[2] (Figura 1.1) me ha dado la posibilidad de tener una visión más haya de la obtención de información a partir de estaciones, incluyendo la información de diferentes satélites.

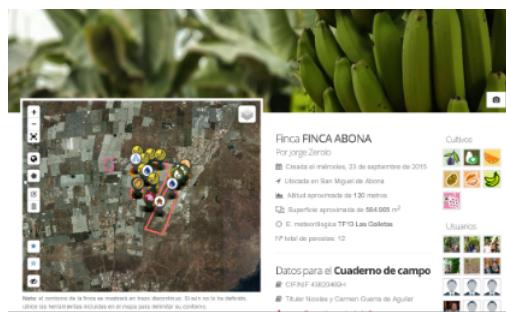


Figura 1.1: Aplicación de GOIA

Por otro lado, del resto de herramientas que existen cabe destacar GRANIOT[3] (Figura 1.2), una plataforma integradora de datos de alto valor en la producción agrícola cuya función es ayudar a la toma de decisiones basada en datos. La mencionada plataforma ha sido utilizada en el presente trabajo de fin de grado.



Figura 1.2: Aplicación de Graniot

Graniot como herramienta de gestión de explotaciones agrícolas nos da información específica de la finca que se ha configurado, lo que nos aporta a nuestro proyecto una especialización hacia el usuario sin precedentes.

En el futuro próximo, las plataformas que se están desarrollando agruparán todas las tecnologías aplicadas al campo, dando lugar a que en una misma aplicación podamos encontrar toda la información necesaria para gestionar nuestra explotación agrícola de la mejor forma posible, incluyendo en ellas la información de los satélites, de estaciones y de sensores.

1.3. Objetivos

El objetivo principal de este proyecto es el de dar formación y la visualización de información para la gestión agrícola. Los principales puntos a desarrollar el proyecto son los siguientes:

- Realizar una investigación en profundidad de las herramientas que pueden ser utilizadas en el proyecto.
- Prototipado de la aplicación para tener una idea base sobre la que desarrollar la experiencia de usuario.
- Configuración de la infraestructura de desarrollo del proyecto.
- Desarrollo del Back-end de la aplicación.
- Desarrollo del Front-end de la aplicación.
- Integrar estaciones meteorológicas diseñadas e implementadas en un trabajo de fin de grado de la titulación de Grado en Ingeniería Electrónica Industrial y Automática de la ULL, desarrollado en paralelo con el presente trabajo.
- Testing y despliegue de la aplicación.

Durante el desarrollo del presente trabajo, la incorporación de las estaciones propias no se pudo integrar, por lo que se modificó dicho objetivo. La no consecución en el plazo originalmente acordado de la versión final de la estación diseñada en el TFG de la otra titulación, propició que se tomara la decisión de implementar la visualización de estaciones que se encontraban ya operativas en la isla.

Por último, se han integrado otras secciones en la página en las cuales se muestra la información de los satélites en capas, mostrando los diferentes índices de interés a los que hemos podido acceder.

1.4. Planificación del proyecto

Para el desarrollo del proyecto se ha realizado una planificación de las diversas tareas a realizar. A continuación se observan dichas tareas y la franja de tiempo que se ha empleado para su realización en la figura 4.8.

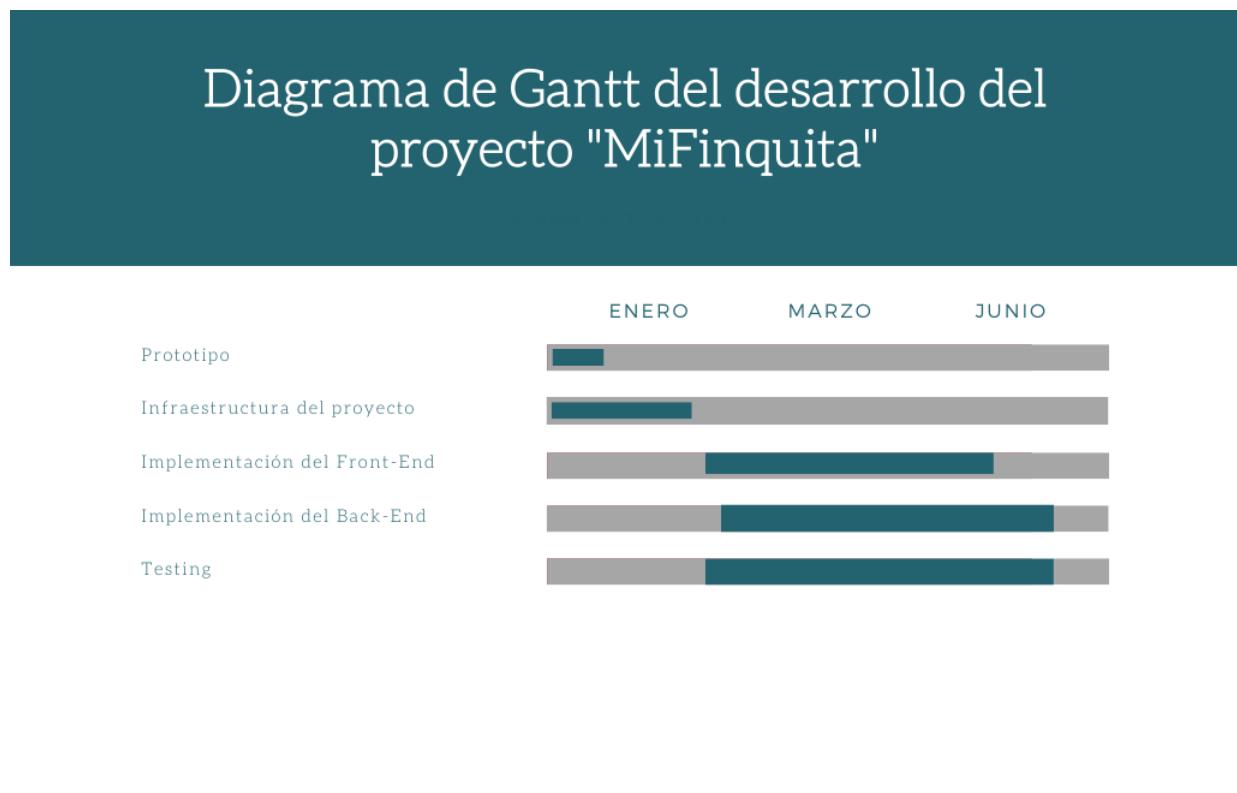


Figura 1.3: Diagrama de Gantt del proyecto

1.5. Estructura del documento

- **Capítulo 1: Introducción.**

Se indican las motivaciones y objetivos del proyecto.

- **Capítulo 2: Herramientas y tecnologías utilizadas.**

Se nombran y definen todas las herramientas y tecnologías utilizadas para el desarrollo del proyecto

- **Capítulo 3. Tecnologías utilizadas.**

Se indican las herramientas y recursos utilizados

- **Capítulo 4. Desarrollo del Front-End.**

Se explica cada uno de los pasos para el desarrollo del Front-End del proyecto.

- **Capítulo 5. Desarrollo Back-End.**

Se explica cada uno de los pasos para el desarrollo del Back-End del proyecto.

- **Capítulo 6. Funcionamiento.**

Se explica las diferentes funcionalidades del proyecto

- **Capítulo 6. Testing y Despliegue.**

Se explica las diferentes pruebas realizadas a la página y como se ha realizado su despliegue

- **Capítulo 7. Presupuesto.**

- **Capítulo 8. Conclusiones y líneas futuras.**

Capítulo 2

Herramientas y tecnologías utilizadas

2.1. Software

En cuanto a las aplicaciones software utilizadas cabe destacar las siguientes:

- El lenguaje de programación **C#[4]** para el desarrollo back-end. Se utilizó para trabajar con **ASP.NET CORE [5]** framework desarrollado por Microsoft.
- El framework **Angular[6]** para el desarrollo del front-end.
- El Sistema de Información Geográfica de Parcelas Agrícolas, **SIG-PAC [7]**, permite identificar geográficamente las parcelas declaradas por los agricultores y ganaderos, en cualquier régimen de ayudas, relacionado con la superficie cultivada o aprovechada por el ganado.
- El servidor de **Geoserver [8]** de código de abierto que permite compartir y editar datos geoespaciales.
- El **Catastro territorial[9]** es un registro administrativo dependiente del Estado en el que se describen los bienes inmuebles rústicos, urbanos y de características especiales.
- La librería de javascript **Leaflet [10]** utilizada para la creación de aplicaciones de mapas web.
- Como entorno de desarrollo integrado (IDE) se ha empleado **Visual Studio Code [11]** y **Microsoft Visual Studio [12]**.
- El servicio **Web Map (WMS)[13]** definido por el **OGC (Open Geospatial Consortium)[14]** produce mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica. Este

estándar internacional define un "mapa como una representación de la información geográfica en forma de un archivo de imagen.

- La plataforma **Graniot** [3].
- Para la base de datos **DBeaver** [15].
- Las imágenes proporcionadas por los satélites **Sentinel** [16].
- Para el almacenamiento de los ficheros y modelos:
 - **Google Drive** [17].
 - **GitHub** [18].
 - **Swagger UI**[19], conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful
- Para el Testing de la aplicación web:
 - Para las pruebas del front-end se hizo uso de **Protractor** [20].
 - Para las pruebas del back-end se hizo uso de **Postman** [21].
- Por último, para la redacción de la memoria:
 - Editor de textos en formato LaTeX **Overleaf** [22].

Capítulo 3

Desarrollo Back-End

En este apartado se explican las tecnologías utilizadas y la configuración aplicadas para el desarrollo del Back-End de nuestra aplicación web [23].

3.1. Servidor

El servidor de la aplicación se ha realizado utilizando **ASP.NET CORE**, framework modular desarrollado por windows. La utilización del framework nos da la posibilidad de tener comunicación bidireccional entre el servidor y el cliente en tiempo real. El lenguaje con el que trabaja es **C#**.

La solución creada bajo el nombre de "**WebAppMiFinquita**" está compuesto por los siguientes proyectos en la figura 3.1:

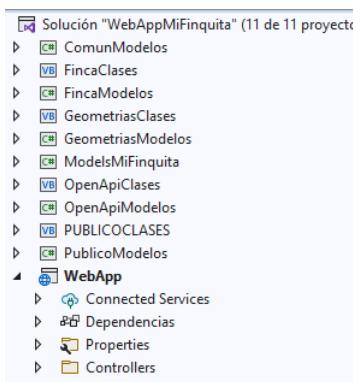


Figura 3.1: Estructura servidor

- **FincaClases/** se encuentran las clases referentes a las fincas
- **GeometriasClases/**. se encuentran las clases referentes al mapa
- **GeometriasModelos/**. se encuentran los modelos referentes al mapa
- **ModelsMiFinquita/**. se encuentran los modelos referentes a las fincas

3.2. API Mi Finquita

A continuación se señalan los diferentes servicios **ASP.NET Core Web API** para la plataforma: "**MiFinquita**".

- Servicio de autentificación **/api/auth** en la figura 3.2. Se encarga de la parte de autentificación de los usuarios. En nuestro proyecto trabajamos con un sistema de doble token implementado a través de la tecnología utilizada de **ASP.NET Core**

Auth		
GET	/api/auth	Test de sistema para comprobar que los servicios funcionan correctamente
GET	/api/auth/tryToken	Comprueba si el token de acceso proporcionado es válido
POST	/api/auth/login	Obtiene un token de acceso al sistema
POST	/api/auth/register	Registra un nuevo usuario en el sistema
GET	/api/auth/logout	Cierra la sesión del usuario
GET	/api/auth/refresh	Renueva un token de acceso al sistema
GET	/api/auth/reload/{id}	Renueva un login de acceso al sistema cuando se cambia de finca
POST	/api/auth/recovery	Solicita una nueva contraseña
POST	/api/auth/resetpassword	Verifica si el código de recuperación de contraseña es correcto

Figura 3.2: Servicios de autentificación

- Servicios referentes al mapa **/api/geometrias/mapa** en la figura 3.3. En él se encuentran todos los servicios referentes al mapa, como es el caso de las diferentes peticiones a los WebServices de las herramientas lo que implica la obtención de la información necesaria para montar cada capa.

Mapa	
GET	/api/geometrias/mapa/finca/{mapaid}
GET	/api/geometrias/mapa/servicio/{mapaid}/{servicioid}
GET	/api/geometrias/mapa/catastro/referencia/{referenciaid}
POST	/api/geometrias/mapa/sigpac/listarecientos
GET	/api/geometrias/mapa/sigpac/recinto/{referenciaid}
GET	/api/geometrias/mapa/finca/recintos
GET	/api/geometrias/mapa/finca/numrecintos
POST	/api/geometrias/mapa/finca/salvarrecintos
POST	/api/geometrias/mapa/catastro/posicion
POST	/api/geometrias/mapa/sigpac/posicion
POST	/api/geometrias/mapa/finca/capajson
POST	/api/geometrias/mapa/finca/capajsonservicio
POST	/api/geometrias/mapa/sigpac/listafichero
POST	/api/geometrias/mapa/sigpac/excelfichero/{tipo}

Figura 3.3: Servicios del mapa

- Servicios del perfil de usuario **/api/user/** en la figura 3.4. Se encarga de la información de dichos perfiles, tanto de actualizarlos como de cargarlos. Envía y recoge los datos que se quieran de los diferentes usuarios.

User	
GET	/api/user
PUT	/api/user
PUT	/api/user/ChangePassword
PUT	/api/user/UpdateImage

Figura 3.4: Servicios perfil de usuario

3.3. Base de datos

Para la base de datos se ha trabajado con **PostgreSQL**[24] con su extensión **PostGis**[25], que la convierte en una base de datos espacial mediante la adición de tres características: tipos de datos espaciales, índices espaciales y funciones que operan sobre ellos. Para el desarrollo de la base de datos se trabaja con la aplicación cliente de SQL **DBeaver**[15]. Esta herramienta se utiliza para la administración de bases de datos, gracias a su arquitectura de plugins nos permite modificar el comportamiento de la aplicación para proporcionar funcionalidad y características específicas de la base de datos que son independientes de la base de datos. La estructura que tenemos en dicha aplicación se muestra en la figura 4.8:

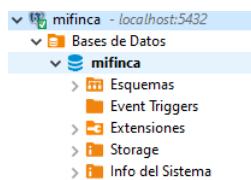


Figura 3.5: Estructura base de datos DBeaver

Por último, la aplicación mencionada en el apartado anterior **DBeaver** nos permite obtener los diagramas **E/R**[26] que componen nuestra base de datos. Entre los modelos implementados encontramos los siguientes:

- Diagrama Modelo E/R de geometrias en la figura 3.6.

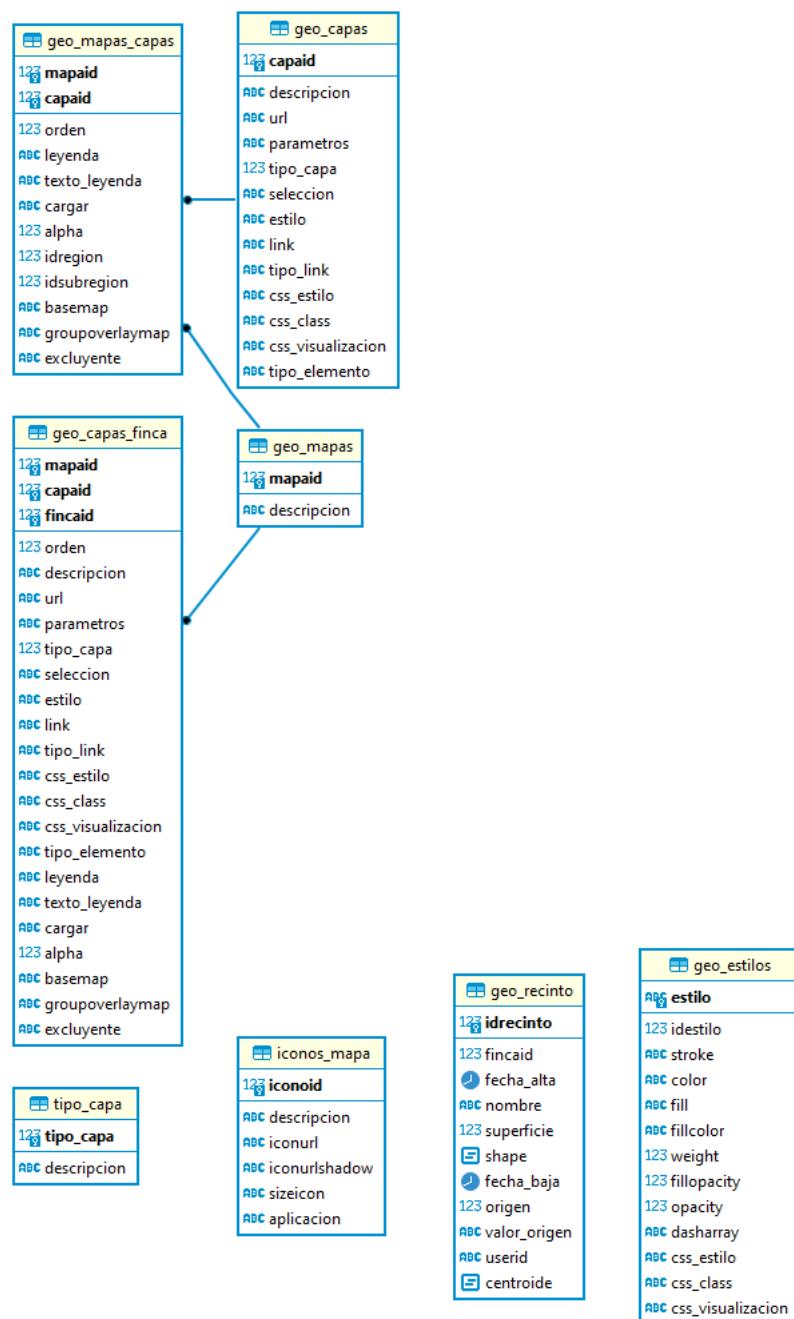


Figura 3.6: Model E/R Geometrías

- Diagrama Modelo E/R de finca en la figura 3.7.

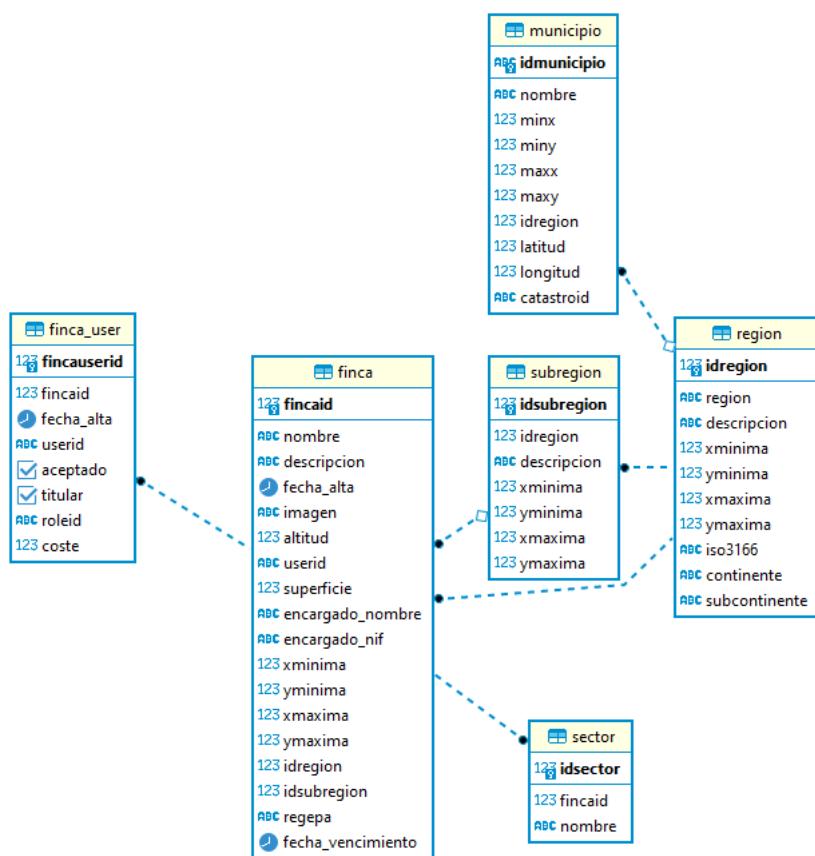


Figura 3.7: Model E/R Fincas

- Diagrama Modelo E/R de Openapi en la figura 3.8

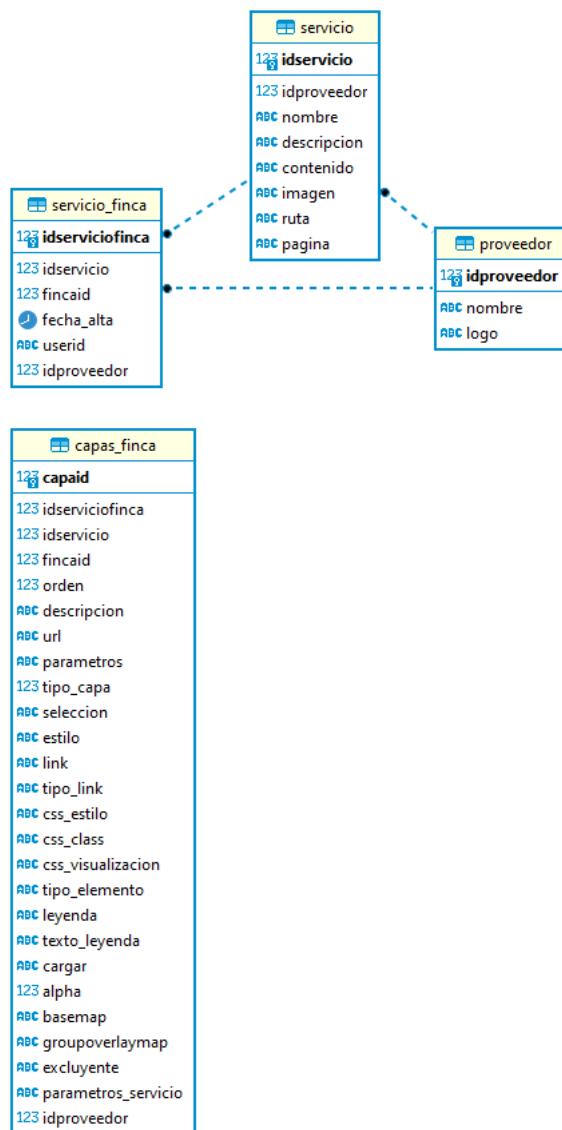


Figura 3.8: Model E/R Openapi

- Diagrama Modelo E/R de publico en la figura 3.9

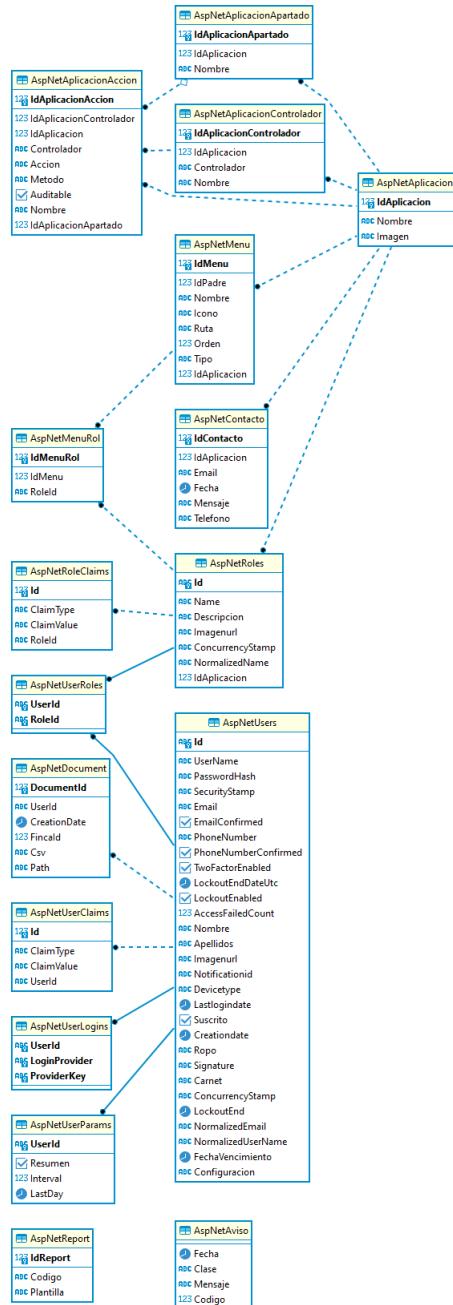


Figura 3.9: Model E/R Publico

Capítulo 4

Desarrollo Front-End

En este apartado se explican las tecnologías utilizadas y la configuración aplicadas para el desarrollo del Front-End de nuestra aplicación web [27].

4.1. Prototipo web del proyecto

La idea inicial de la página se realizó en la aplicación online de **Figma**[28], con la que obtuvimos una estructura sobre la que trabajar, en lo que a interfaz de usuario se refiere y en versión de escritorio. De dicha idea inicial destacan de la estructura una página home en la figura 4.1, una página de inicio de sesión en la figura 4.2 y la demo de la herramienta en la figura 4.3.

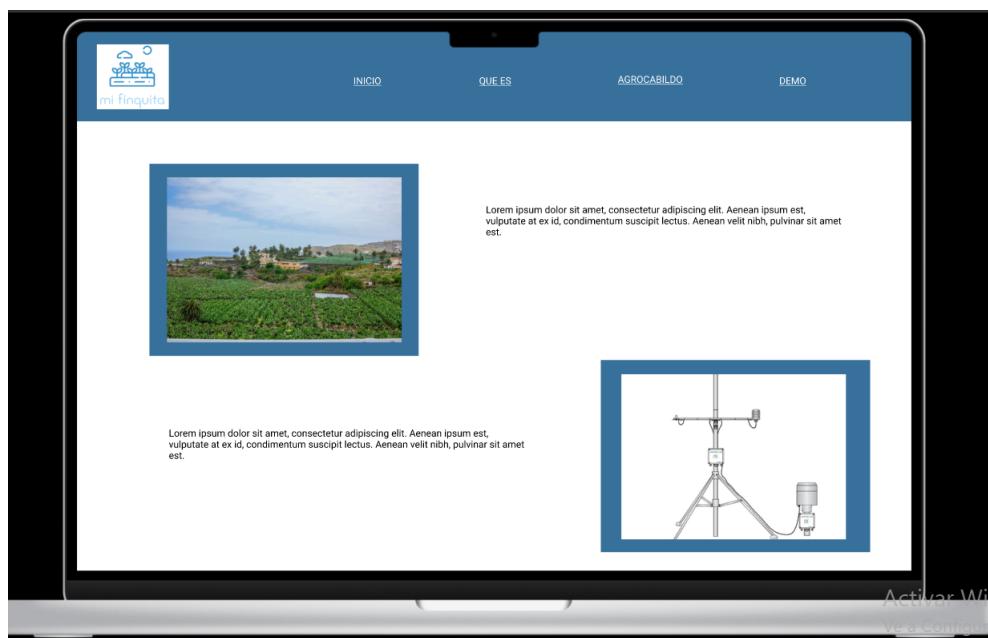


Figura 4.1: Home del prototipo de la aplicación

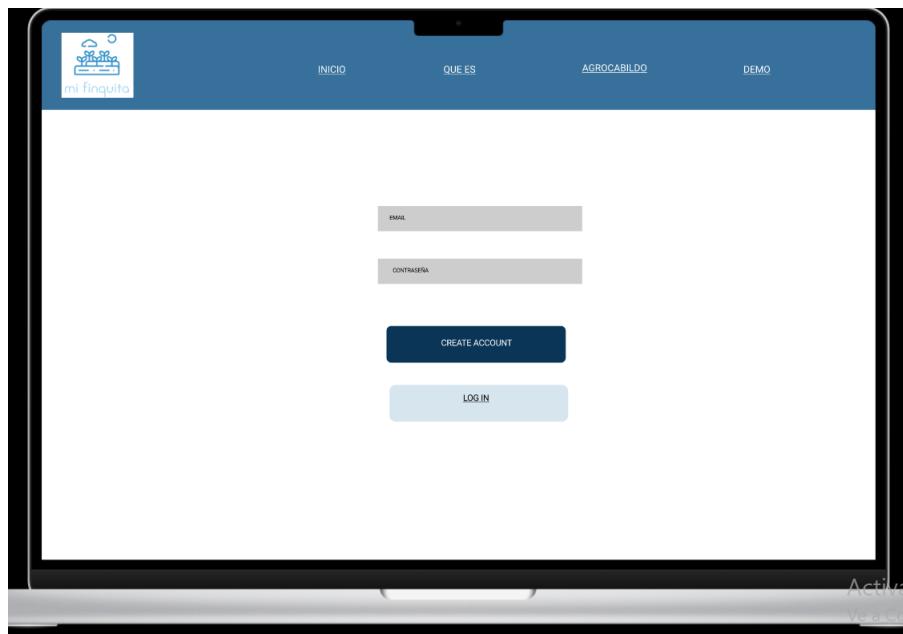


Figura 4.2: Inicio de sesión en la demo del prototipo



Figura 4.3: Demo del prototipo

4.2. Estructura del proyecto

El directorio donde se ha realizado el desarrollo del Front-End de la página se denomina src. En dicho directorio podemos encontrar la siguiente estructura en la figura 4.4:

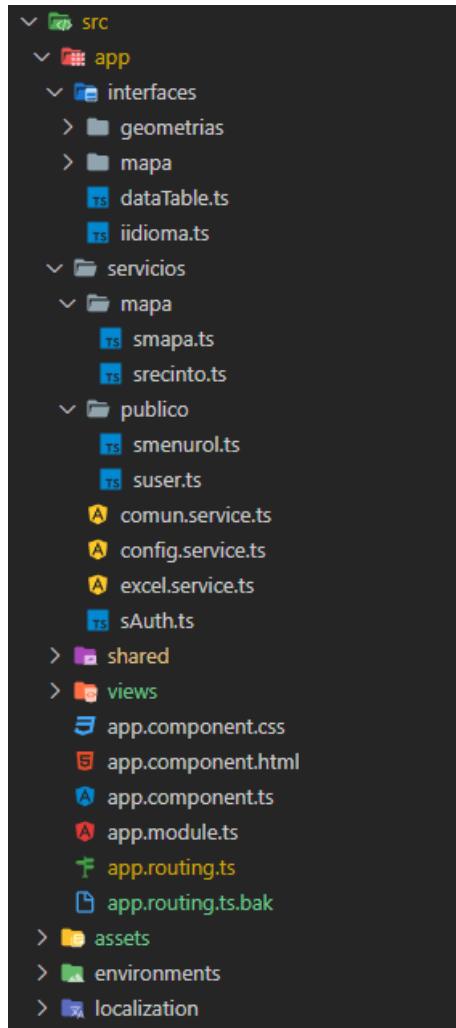


Figura 4.4: Estructura de archivos Front-End

- **Interfaces/**. Se encuentran localizados los ficheros referentes a la configuración del mapa y al recinto sobre los que trabajaremos.
- **Servicios/** Directorio donde se encuentra localizado la conexión con las apis. Exactamente con **/api/geometrías/mapa/** para el mapa , **/api/geometrías/recinto/** para el recinto y **/api/user** para los servicios del usuario y su autentificación.
- **Shared/**. Carpeta donde se encuentran localizadas las diferentes animaciones y los diferentes componentes utilizados en la página.

- **Views/**. Directorio donde se encuentran localizadas las diferentes vistas que componen la página.
- **Assets/**. Donde se encuentran las imágenes utilizadas para las distintas vistas de la aplicación.

4.3. Resultado final de la interfaz de usuario

La aplicación final que se ha desarrollado consta, al igual que el prototipo de una página de inicio en la figura 4.5, con información y acceso a la página de login de la figura 4.6, así como de una página para la demo en la figura 4.7. Por otro lado cabe destacar la existencia de una página donde se encuentra el perfil del usuario en la figura 4.8, a la vez que existen las diferentes opciones que tenemos a la hora de elegir nuestra fuente de información en el cuadro de la aplicación.



Figura 4.5: Página de inicio

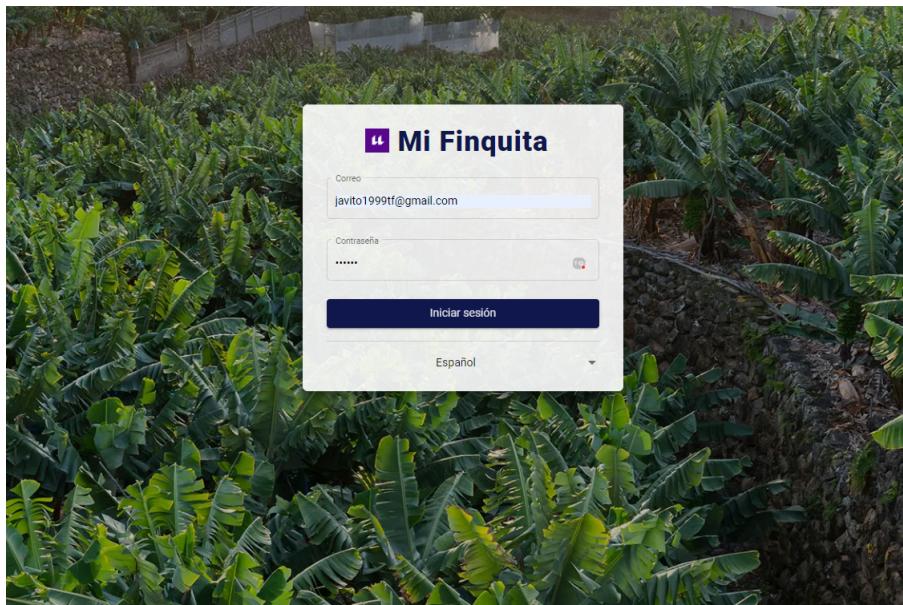


Figura 4.6: Página de inicio

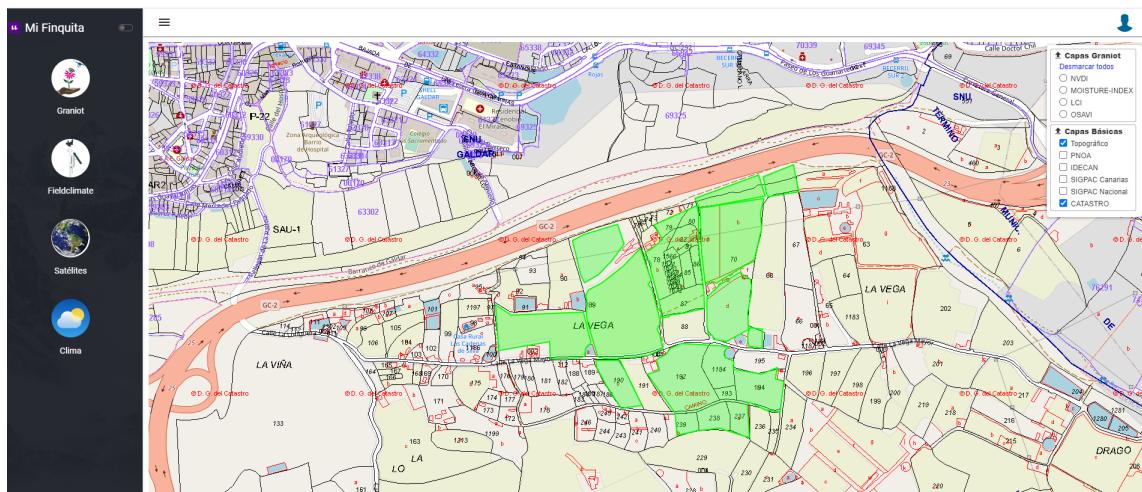


Figura 4.7: Página de la demo



Figura 4.8: Página del perfil

Capítulo 5

Funcionamiento de la plataforma

En este capítulo se expone el funcionamiento específico del proyecto desarrollado, desde el punto de vista del Front-End y del Back-End.

5.1. Inicio de sesión

En primer lugar se procedió a realizar los componentes visuales del inicio de sesión de la demo de la aplicación. En todo momento, el diseño a realizar trata de ser responsivo. Todo este proceso corresponde con el Front-End.

Desde el punto de vista del Back-End, donde se desarrollan las diferentes peticiones, como es el caso del inicio de sesión **/api/auth/login**, se ha implementado un sistema de autenticación dado por la librería utilizada de **ASP.NET Core** denominada **Identity**[29] cuya estructura se observa en la figura 5.1. Dicha librería nos permite trabajar con un sistema de autenticación de doble token, lo que da lugar a que nuestro inicio de sesión sea más seguro.

El sistema utilizado de dos token, proporciona un token al iniciar sesión al usuario que dura 5 minutos y se le concede un segundo token de refresco al mismo tiempo el cual dura alrededor de 24 horas. Evitamos de esta forma que si un tercero coge el primer token, no pueda acceder más allá del tiempo que este se encuentre operativo.

- >  [AspNetUserClaims](#)
- >  [AspNetUserLogins](#)
- >  [AspNetUserParams](#)
- >  [AspNetUserRoles](#)
- >  [AspNetUsers](#)

Figura 5.1: Visualización de la Identity Base de datos

5.2. Dibujar mapa de la aplicación

En lo que respecta a la creación del mapa utilizado en la aplicación, se ha utilizado la librería de javascript **leaflet**[10].

Leaflet es compatible con las capas del Servicio de mapas web (WMS), las capas geojson y las capas vectoriales. En nuestro proyecto se han implementado capas de cada uno de los tipos mencionados.

El modelo de visualización básico implementado por Leaflet es un mapa base como se observa en la figura 5.2, más cero o más superposiciones translúcidas, con cero o más objetos vectoriales que se muestran en la parte superior.



Figura 5.2: Visualización del mapa de la aplicación

5.3. Creación de las fincas

En cuanto a la creación de los recintos, a través del punto que nos indica el usuario, se pueden dar dos casos: en caso de que haya un recinto lo borramos y en caso de que no hay hacemos uso del **WebService** de **SIGPAC**. SIGPAC contiene la información geográfica de las parcelas agrícolas.

Cuando el usuario quiere pinchar un nuevo recinto hacemos uso de los **WebService** de **SIGPAC** para encontrar el recinto seleccionado. En primera instancia al seleccionar el usuario el recinto que quiere mostrar se realiza un **GetFeatureInfo** para obtener la referencia del posible recinto en el WebService de SIGPAC. A continuación, con dicha referencia en otro WebService nos va a devolver un **.json**[30] con la información que necesitamos. En nuestro proyecto a partir de dicho json generamos el **geojson** [31], que es un formato diseñado para representar elementos geográficos sencillos junto con sus atributos no espaciales.

El geojson que generamos es necesario para que **leaflet**[10], una librería de **javascript**[32] utilizada para aplicaciones de mapa web, pueda pintar el recinto requerido por el usuario a través de la información que le hemos pasado.

El geojson obtenido se guarda en la base de datos de **PostgreSQL** en un campo de **PostGis**.

```
#Referencia SIGPAC

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "provincia": "35 - LAS PALMAS",
        "municipio": "9 - GÁLDAR",
        "agregado": 0,
        "zona": 0,
        "poligono": 3,
        "parcela": 9000,
        "recinto": 1620,
        "superficie_ha": 0.7519,
        "pendiente": 8.8,
        "coef_regadio": 100
      },
      "admisibilidad": null
    }
  ]
}
```

```

        "incidencias": [
      "112 - Plataneras"
    ],
    "uso_sigpac": "FY - FRUTALES",
    "region": ""
  }
}
],
}

# Cogemos referencia sigpac provincia,municipio,agrado,zona,polygono,parcela,recinto
# https://sigpac.mapa.gob.es/fega/ServiciosVisorSigpac/query/recinfo/35/9/0/0/3/9000/1620.json

[{"provincia":35,"municipio":9,"agregado":0,"zona":0,"poligono":3,
"parcela":9000,"recinto":1620,"superficie":1861.7956768409167,
"pendiente_media":39,"coef_regadio":100,"admisibilidad":null,
"incidencias": "112,190", "uso_sigpac": "FY", "region":null,
"srid":4258,"wkt": "POLYGON((-15.649468639275053 28.142347534010412, -15.649487423959371
28.142370380565247, -15.649531520316392
28.142362427815044, -15.649566233132257
28.142476003447978, -15.649558186504732
28.142496106605748, -15.649535387726747
28.142511479855266, -15.64943614598728
28.142533948386248, -15.649369090757911
28.142551687008986, -15.64933422203864
28.14256114682547, -15.649299353319368
28.142596624070947, -15.649142444082644
28.142640378446302, -15.648649599881445
28.14278055069778, -15.648635393392915
28.142764613346134, -15.648607784240411
28.142732365648143, -15.648596953982679
28.142721203467023, -15.64866711889831
28.142653657058098, -15.648727791308033
28.142599299574602, -15.648742880411023
28.142587134079616, -15.648760150485346
28.142581530777008, -15.648920787154632
28.142498279195173, -15.649071290102379
28.14244944622439, -15.649347460417298
28.142377103690187, -15.649468639275053
28.142347534010412)})]

```

#Construimos GeoJson

```
{"type": "FeatureCollection", "features": [{"type": "Feature", "geometry": {
  "type": "Polygon", "coordinates": [[[[-15.649468639275053, 28.142347534010412]
, [-15.649487423959371, 28.142370380565247]
, [-15.649531520316392, 28.142362427815044],
[-15.649566233132257, 28.142476003447978],
[-15.649558186504732, 28.142496106605748],
[-15.649535387726747, 28.142511479855266],
[-15.649436145987281, 28.142533948386248],
[-15.649369090757911, 28.142551687008986],
[-15.64933422203864, 28.142561146825471],
[-15.649299353319368, 28.142596624070947],
```

```

[-15.649142444082644, 28.142640378446302],
[-15.648649599881445, 28.142780550697779],
[-15.648635393392915, 28.142764613346134],
[-15.648607784240411, 28.142732365648143],
[-15.648596953982679, 28.142721203467023],
[-15.64866711889831, 28.142653657058098],
[-15.648727791308033, 28.142599299574602],
[-15.648742880411023, 28.142587134079616],
[-15.648760150485346, 28.142581530777008],
[-15.648920787154632, 28.142498279195173],
[-15.649071290102379, 28.14244944622439],
[-15.649347460417298, 28.142377103690187],
[-15.649468639275053, 28.142347534010412]]], "properties": {"origen": 2
, "texto": "Recinto Sigpac", "valor_origen": "35,9,0,0,3,9000,1620"
, "nombre": "35,9,0,0,3,9000,1620",
"href": "https://sigpac.mapa.gob.es/fega/ServiciosVisorSigpac/LayerInfo?
layer=recinto&id=35,9,0,0,3,9000,1620"}]}

```

Como se observa, en el .json que nos devuelve el WebService de SIG-PAC la parte correspondiente a la información geográfica viene dada en formato **WKT**[33], formato de codificación específicamente diseñado para la caracterización y almacenamiento de objetos geométricos espaciales en formato vectorial. Por otro lado, destacar que encontramos al final el geojson que se ha generado expuesto en los ficheros mostrados.

En la figura 5.3 se puede observar el recinto seleccionado en amarillo que se quiere añadir y en la esquina superior derecha el botón de guardado.



Figura 5.3: Crear nueva finca

5.4. Capas de mapa

En cuanto a las opciones que podemos encontrar sobre la visualización del mapa de la aplicación, se han implementado bajo el nombre de capas básicas como se observa en la figura 5.4.

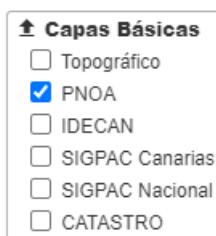


Figura 5.4: Capas de mapa

- El **Topográfico**[34] representa el relieve de la superficie terrestre.
- El Plan Nacional de Ortofotografía Aérea **PNOA**[35], con el cual obtenemos ortofotografías aéreas digitales de todo el territorio español.
- Sistema de Información Geográfica de Parcelas Agrícolas **SIGPAC CANARIAS**[7], nos permite inspeccionar geográficamente cualquier zona dedicada a la agricultura y/o a la ganadería de Canarias.
- Sistema de Información Geográfica de Parcelas Agrícolas **SIGPAC NACIONAL**[36], nos permite inspeccionar geográficamente cualquier zona dedicada a la agricultura y/o a la ganadería del territorio español.
- **Catastro**[9], permite el acceso a toda la información catastral y a la edición del parcelario.
- La Infraestructura de datos espaciales de Canarias **IDECAN**[37] utilizamos el WMS de las ortofotografías.

Todas las capas mencionadas se pueden fusionar con otra que queramos al mismo tiempo como puede ser el caso de la capa topográfica junto con la de catastro. Dicho ejemplo se puede observar en la figura 5.5.

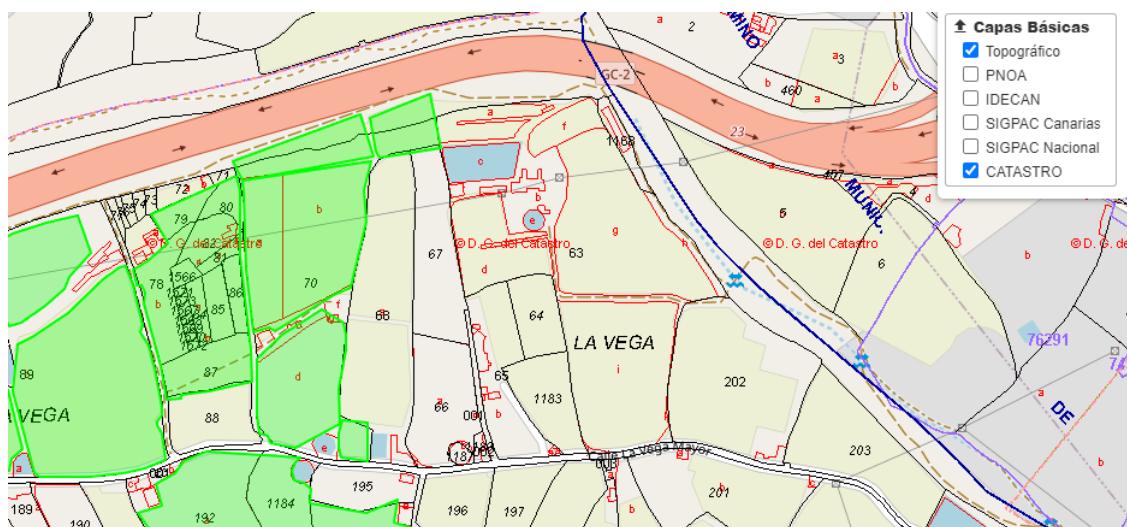


Figura 5.5: Capas ejemplo Topográfica y Catastro

5.5. Uso de Graniot en la página

Para el uso de Graniot en nuestro proyecto se ha conseguido acceso a una plantación localizada en Las Palmas. Esta herramienta es de pago pero, gracias a algunas gestiones con la empresa, se ha obtenido una cuenta de prueba.

Graniot nos devuelve un geojson con diferentes cuadrados que contienen el valor de cada una de las medidas que puede hacer sobre la finca solicitada y para representarla hacemos uso de **leaflet** directamente. Entre dichas medidas encontramos las siguientes:

- El Índice de Vegetación de la Diferencia Normalizada (**NDVI**) en la figura 5.6. Es un indicador numérico que utiliza las bandas espectrales roja y cercana al infrarrojo. Se asocia con el contenido de vegetación.

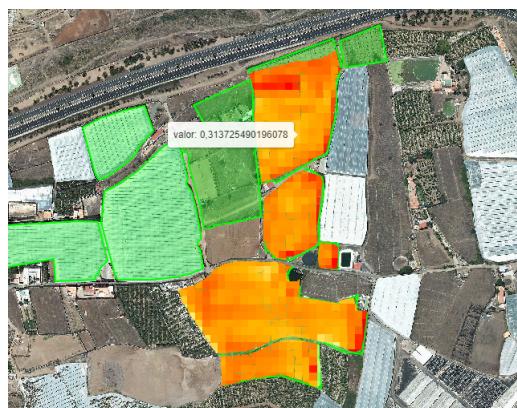


Figura 5.6: Visualización Graniot NDVI

- El Índice de Diferencia Normalizada de Humedad (**NDMI**) en la figura 5.7. Moisture Index, se utiliza para determinar el contenido de agua de la vegetación.



Figura 5.7: Visualización Graniot NDMI

- El índice de clorofila (**LCI**), en la figura 5.8. Se usa para calcular la cantidad de clorofila en las plantas.



Figura 5.8: Visualización Graniot LCI

- El índice de vegetación ajustado al suelo (**OSAVI**), en la figura 5.9. Muestra una ligera variante respecto a la fórmula tradicional del NDVI 5.6 para evitar distorsiones en los valores de análisis cuando la vegetación se encuentra sobre suelos expuestos.



Figura 5.9: Visualización Graniot OSAVI

5.6. Uso de Sentinel en la página

Para el uso de las imágenes por satélite de **SENTINEL**, se aprecia un ejemplo de uso en la figura 5.10, nos da la posibilidad de trabajar con conexión directa con el estándar de WMS definido por OGC.

El sistema que se ha implementado en el proyecto nos permite añadir cualquier capa como se puede observar en la figura 5.11, ya que las capas se han montado para que sean dinámicas y no estáticas.



Figura 5.10: Ejemplo NDVI Sentinel

102	NDVI	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "NDVI", "format": "image/jpg"}
101	Color real	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "TRUE-COLOR-S2L2A", "format": "image/jpg"}
109	Falso Color	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "FALSE-COLOR", "format": "image/jpg"}
104	Agricultura	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "AGRICULTURE", "format": "image/jpg"}
105	MOISTURE-INDEX	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "MOISTURE-INDEX", "format": "image/jpg"}
103	Falso Color Urbano	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "FALSE-COLOR-URBAN", "format": "image/jpg"}
106	SWIR	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "SWIR", "format": "image/jpg"}
107	Color Natural	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "NATURAL-COLOR", "format": "image/jpg"}
108	Geología	<input type="checkbox"/> https://services.sentinel-hub.com	{"layers": "GEOLOGY", "format": "image/jpg"}

Figura 5.11: Carga en la base de datos de los índices de Sentinel implementados

5.7. Uso Geomet en la página

El uso de la herramienta geomet en el proyecto se ha conseguido de la misma forma que con Sentinel.

En la proyecto se han utilizado dos parámetros meteorológicos, los que se consideran más importantes en el campo que nos encontramos. Dichos parámetros son los siguientes:

- El primer parámetro implementado es el del viento (ver figura 5.12), el cual nos permite visualizar los datos obtenidos de Geomet, en concreto la dirección del viento con respecto a la ubicación física de la finca del usuario.

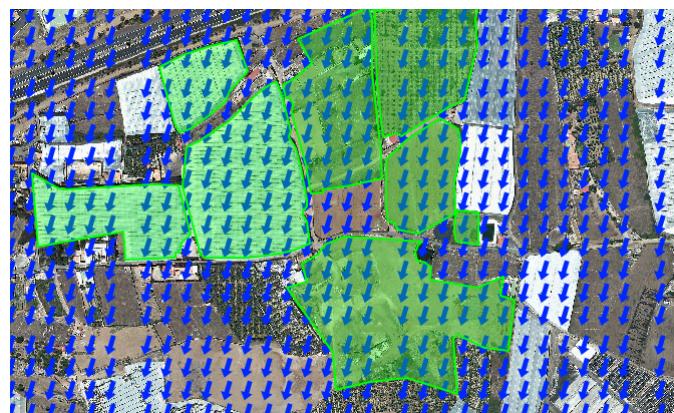


Figura 5.12: Ejemplo Viento

- El segundo y último parámetro implementado es el de la lluvia (ver figura 5.13), con el cual podemos observar si existe lluvia en el entorno de nuestro recinto.

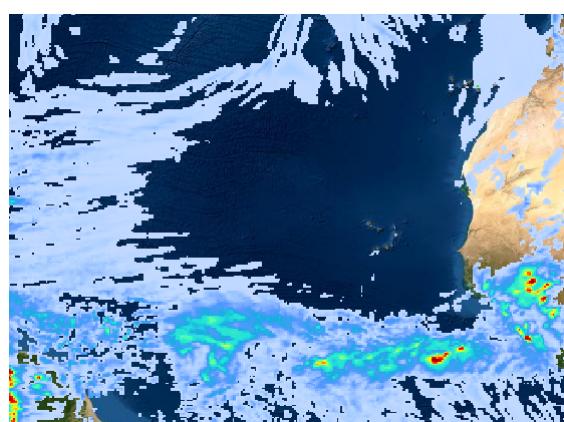


Figura 5.13: Ejemplo Lluvia

5.8. Información de las estaciones

Se ha realizado un dashboard en el que encontramos una serie de gráficas, en las que se representa la información de las estaciones.

Los datos de dichas gráficas se introducen de forma manual debido a que no se han podido conectar las estaciones al proyecto creado. Dentro de las medidas que se han establecido en los sensores de la estación utilizada tenemos la temperatura, como se observa en la figura 5.14, la humedad del suelo, la presión atmosférica, la radiación, el volumen de precipitaciones, velocidad del viento y calidad del aire.

Las medidas elegidas en las estaciones complementan la información obtenida mediante satélites, dando lugar a que se tenga datos más precisos sobre la plantación del usuario.

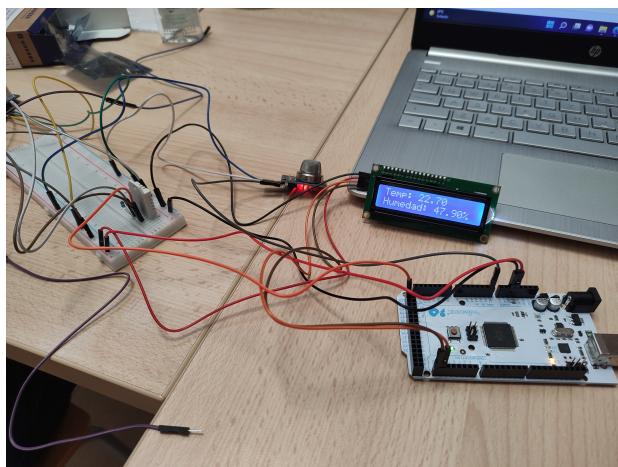


Figura 5.14: Ejemplo sensor de la estación

Capítulo 6

Testing y despliegue

6.1. Testing Front-End

Dentro del apartado de testing de la página, se han llevado a cabo pruebas de usuario a la interfaz que se ha implementado para el proyecto. Para las pruebas se ha utilizado el framework de **Protractor**[20].

Las pruebas que nos permite realizar esta herramienta son pruebas E2E. Se trata de una metodología que permite verificar que el flujo de nuestra aplicación funciona tal y como está diseñado. El objetivo que hemos logrado a través de estas pruebas, es el de identificar las dependencias del sistema y asegurar la integridad de los datos entre los componentes que lo forman.

Se han integrado pruebas para comprobar el correcto funcionamiento de funcionalidades tales como el login de la demo, selección de las diferentes capas de información, editar información del perfil y logout de la página.

Por último cabe destacar que se ha hecho uso de la herramienta de **Protractor**[20] junto con **Selenium WebDriver**[38]. Mediante la configuración que se ha utilizado, haciendo uso de **Jasmine**[39] completamos los aspectos necesarios para realizar un correcto desarrollo guiado de pruebas **TDD**[40] en nuestro Front-End.

```

#Javier Martin de Leon
#alu0101133355@ull.edu.es
#Trabajo de fin de grado
#Fichero de configuración Protractor

var SpecReporter = require('jasmine-spec-reporter').SpecReporter;

exports.config = {

  framework: 'jasmine',

  onPrepare: function() {
    jasmine.getEnv().addReporter(new SpecReporter({
      displayFailuresSummary: true,
      displayFailedSpec: true,
      displaySuiteNumber: true,
      displaySpecDuration: true
    }));
  },

  onComplete: function() {

  },

  seleniumAddress: 'http://localhost:4200/',

  specs: ['tests/funciones-spec.js'],
  jasmineNodeOpts: {

    defaultTimeoutInterval: 30000
  }

};

#Fin fichero de configuración

```

6.2. Testing Back-End

En cuanto a las pruebas realizadas en el Back-End de la aplicación, se hizo uso de la herramienta **Postman**[21]. Se trata de una plataforma API la cual permite a los desarrolladores diseñar, construir, probar e iterar sus APIS.

En nuestro proyecto se probaron las API creadas y sus peticiones.

Capítulo 7

Presupuesto

El salario promedio de un Ingeniero Informático en España es de unos **26.500€**[41] al año. En total cobra alrededor de **13,59€** hora

La herramienta Graniot utilizada en nuestra aplicación es a su vez de pago, nos cobra en relación a las hectáreas por año que queremos tener gestionadas. A su vez, cabe mencionar el coste de un despliegue de la aplicación y la herramienta que también hemos utilizado de Sentinel.

Todo esto, da lugar a que el presupuesto del proyecto se haya dividido en dos tipos de recursos. Por un lado recursos humanos 7.1 necesarios para el desarrollo del proyecto, en el cual se calcula según las horas empleadas el coste total. Y por el otro tipo de recurso que compone nuestro presupuesto es el material 7.2.

El calculo del presupuesto de recursos materiales se ha calculado para una finca de aproximadamente 5 hectáreas y con un contrato de un año. Con estos datos tenemos un presupuesto total de aproximadamente **4151€**.

Descripción	Horas	Precio €	Total €
Planificación del proyecto	27	13,59	366,93
Investigación y documentación de herramientas	30	13,59	407,70
Desarrollo Mockup	10	13,59	135,90
Desarrollo Back-End de la aplicación	130	13,59	1766,70
Desarrollo Front-End de la aplicación	103	13,59	1399,77
		TOTAL:	4077,00

Tabla 7.1: Presupuesto recursos humanos

Descripción	Coste anual	Total €
Graniot precio de hectáreas/año	7,5	50
Sentinel	24	24
	TOTAL:	74

Tabla 7.2: Presupuesto recursos materiales

Capítulo 8

Conclusiones y líneas futuras

8.1. Conclusiones y líneas futuras

La agricultura de precisión es, sin duda alguna, el futuro en cuanto a gestión de explotaciones agrícolas se refiere. El gran abanico de información que nos brinda su utilización da lugar a una mejora notable en nuestras plantaciones, a la vez que nos ayuda a tener un control total sobre las mismas.

En este proyecto se han estudiado diversas posibilidades de información que se pueden obtener a través de satélites o estaciones meteorológicas. La información obtenida mediante estos medios se ha georeferenciado y mostrado gráficamente en el caso de las estaciones.

Todo la información que hemos obtenido ayuda al agricultor, ya sea de grandes o pequeñas explotaciones, a que tenga información real y en todo momento de la situación en la que se encuentra la misma.

Otro aspecto a destacar es la gran cantidad de recursos y posibilidades con las que se puede trabajar a la hora de realizar una aplicación de este tipo. En la actualidad no existen muchas herramientas, pero todas ellas tienen un gran éxito y abarcan aspectos totalmente distintos.

La aplicación que se ha implementado tiene una enorme cantidad de posibilidades de mejora, pues se pueden incluir más métodos de información e incluso dar información según los datos obtenidos, que ayuden al agricultor directamente con consejos o advertencias. Con más tiempo y recursos se podría mejorar sobre todo las funcionalidades y realizar su adaptación a una aplicación móvil, así como tener integradas nuestras propias estaciones meteorológicas.

8.2. Summary and Conclusions

Precision farming is without a doubt the future of farm management. The great amount of information that its use gives us leads to a notable improvement in our plantations, while it also helps us to have a total control over them.

In this project various possibilities of information, that can be obtained through satellites or meteorological stations, have been studied. The information obtained through those tools have been georeferenced and also, in the case of the stations, shown by graphics.

All the obtained information helps the farmer, even in case of big or a small plantation, to get real information in order to follow at anytime its state.

Another aspect to highlight is the amount of resources and possibilities that you can work with when you are making an application of this type. At the present there are not many applications but all of them are highly successful and take into account different aspects.

The application that has been implemented has a huge number of improvement possibilities, for example, more information methods can be included and even we can also give information according to the data obtained, which help the farmer directly with advices or warnings about his plantations. With more time and resources the functionalities could be improved, the tool could be also adapted to a mobile application and we could integrate our own stations.

Bibliografía

- [1] Motivacion. <https://www.mccormick.it/es/agricultura-4-0-que-es-y-cuales-son-sus-> ONLINE Accessed: 2022-06-7.
- [2] Goia. <http://www.goia.es/>. ONLINE Accessed: 2022-06-7.
- [3] Graniot. <https://graniot.com/>. ONLINE Accessed: 2022-06-7.
- [4] (c#). https://es.wikipedia.org/wiki/C_Sharp. ONLINE Accessed: 2022-06-7.
- [5] Asp.net core. <https://docs.microsoft.com/es-es/aspnet/core/?view=aspnetcore-6.0>. ONLINE Accessed: 2022-06-7.
- [6] Angular. <https://angular.io/>.
- [7] Sigpac. <https://code.visualstudio.com/>. ONLINE Accessed: 2022-06-7.
- [8] Geoserver. <https://geoserver.org/>. ONLINE Accessed: 2022-06-7.
- [9] Catastro. sedecatastro.gob.es. ONLINE Accessed: 2022-06-7.
- [10] Leaflet. <http://leaflet-extras.github.io/leaflet-providers/preview/>. ONLINE Accessed: 2022-06-7.
- [11] Visual studio code. <https://code.visualstudio.com/>. ONLINE Accessed: 2022-06-7.
- [12] Microsoft studio. <https://visualstudio.microsoft.com/es/>. ONLINE Accessed: 2022-06-7.
- [13] Wms. https://es.wikipedia.org/wiki/Web_Map_Service. ONLINE Accessed: 2022-06-7.
- [14] Ogc. https://es.wikipedia.org/wiki/Open_Geospatial_Consortium. ONLINE Accessed: 2022-06-7.
- [15] Dbeaver. <https://dbeaver.io/>. ONLINE Accessed: 2022-06-7.
- [16] Sentinel. https://www.esa.int/Space_in_Member_States/Spain/SENTINEL_2. ONLINE Accessed: 2022-06-7.
- [17] Google drive. https://www.google.com/intl/es_es/drive/. ONLINE Accessed: 2022-06-7.
- [18] Github. <https://github.com/>. ONLINE Accessed: 2022-06-7.

- [19] Swagger ui. <https://swagger.io/tools/swagger-ui/>. ONLINE Accessed: 2022-06-7.
- [20] Protractor. https://www.esa.int/Space_in_Member_States/Spain/SENTINEL_2. ONLINE Accessed: 2022-06-7.
- [21] Postman. <https://www.postman.com/>. ONLINE Accessed: 2022-06-7.
- [22] Overleaf. <https://es.overleaf.com>. ONLINE Accessed: 2022-06-7.
- [23] Back-end. <https://github.com/alu0101133355/TFG-MI-FINQUITA>. ONLINE Accessed: 2022-06-7.
- [24] Postgre. <https://es.wikipedia.org/wiki/PostgreSQL>. ONLINE Accessed: 2022-06-7.
- [25] Postgis. <https://es.wikipedia.org/wiki/PostGIS>. ONLINE Accessed: 2022-06-7.
- [26] Modelo e/r. https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n. ONLINE Accessed: 2022-06-7.
- [27] Front-end. <https://github.com/alu0101133355/TFG-MI-FINQUITA>. ONLINE Accessed: 2022-06-7.
- [28] Figma. <https://www.figma.com/>. ONLINE Accessed: 2022-06-7.
- [29] Identity. <https://es.wikipedia.org/wiki/PostGIS>. ONLINE Accessed: 2022-06-7.
- [30] json. <https://es.wikipedia.org/wiki/JSON>. ONLINE Accessed: 2022-06-7.
- [31] geojson. <https://es.wikipedia.org/wiki/GeoJSON>. ONLINE Accessed: 2022-06-7.
- [32] javascript. <https://es.wikipedia.org/wiki/JavaScript>. ONLINE Accessed: 2022-06-7.
- [33] Wkt. https://es.wikipedia.org/wiki/Well_Known_Text. ONLINE Accessed: 2022-06-7.
- [34] topografico. https://es.wikipedia.org/wiki/Mapa_topogr%C3%A1fico. ONLINE Accessed: 2022-06-7.
- [35] Pnoa. <https://pnoa.ign.es/>. ONLINE Accessed: 2022-06-7.
- [36] Sigpac nacional. <https://sigpac.mapa.gob.es/fega/visor/>. ONLINE Accessed: 2022-06-7.
- [37] Idecan. <https://visor.grafcan.es/visorweb/default.php?svc=svcVialOrto&srid=EPSG:32628&lat=3148116.9474788574&lng=375861.7833793055&zoom=20&lang=es>. ONLINE Accessed: 2022-06-7.
- [38] Selenium. <https://es.wikipedia.org/wiki/Selenium>. ONLINE Accessed: 2022-06-7.

- [39] Jasmine. <https://www.adictosaltrabajo.com/2012/12/10/jasmine-hello-world/>. ONLINE Accessed: 2022-06-7.
- [40] Tdd. https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas. ONLINE Accessed: 2022-06-7.
- [41] Salario ingeniero informático. <https://es.talent.com/salary?job=ingeniero+inform>. ONLINE Accessed: 2022-06-7.