



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Aplicación web para gestión  
de indicadores del portal del  
investigador UBU  
Documentación Técnica**



Presentado por Iker Gayangos Merino  
en Universidad de Burgos — 9 de julio de 2024

Tutores: Dr. César Ignacio García Osorio  
Dr. José Manuel Galán Ordax



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	6
<b>Apéndice B Especificación de Requisitos</b>	<b>11</b>
B.1. Introducción . . . . .	11
B.2. Objetivos generales . . . . .	11
B.3. Catálogo de requisitos . . . . .	12
B.4. Especificación de requisitos . . . . .	13
<b>Apéndice C Especificación de diseño</b>	<b>23</b>
C.1. Introducción . . . . .	23
C.2. Diseño de datos . . . . .	23
C.3. Diseño procedimental . . . . .	39
C.4. Diseño arquitectónico . . . . .	40
<b>Apéndice D Documentación técnica de programación</b>	<b>43</b>
D.1. Introducción . . . . .	43
D.2. Estructura de directorios . . . . .	43
D.3. Manual del programador . . . . .	46

D.4. Compilación, instalación y ejecución del proyecto . . . . .	46
D.5. Pruebas del sistema . . . . .	53
<b>Apéndice E Documentación de usuario</b>	<b>57</b>
E.1. Introducción . . . . .	57
E.2. Requisitos de usuarios . . . . .	57
E.3. Instalación . . . . .	58
E.4. Manual del usuario . . . . .	58
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>65</b>
<b>Bibliografía</b>	<b>67</b>

---

# Índice de figuras

---

B.1. Diagrama de casos de uso nivel 0. . . . .	14
B.2. Diagrama de casos de uso nivel 1. Visualización de la información. . . . .	14
B.3. Diagrama de casos de uso nivel 1. Exportación de la información. . . . .	15
B.4. Diagrama de casos de uso nivel 1. Fusión investigadores. . . . .	15
C.1. Diagrama relacional de tabla investigadores . . . . .	24
C.2. Tabla investigadores . . . . .	25
C.3. Tabla departamentos . . . . .	25
C.4. Tabla programas de doctorado . . . . .	26
C.5. Tabla áreas . . . . .	27
C.6. Tabla centros de investigación . . . . .	28
C.7. Tabla grupos de investigación . . . . .	29
C.8. Tabla tesis . . . . .	29
C.9. Tabla patentes . . . . .	30
C.10. Diagrama relacional de tabla publicaciones . . . . .	31
C.11. Tabla publicaciones . . . . .	32
C.12. Tabla citas recibidas . . . . .	33
C.13. Tabla journal impact factors . . . . .	34
C.14. Tabla journal citation indicators . . . . .	35
C.15. Tabla scopus citescore . . . . .	36
C.16. Tabla SCImago journal ranks . . . . .	37
C.17. Tabla dialnet revistas . . . . .	38
C.18. Tabla investigadores publicaciones . . . . .	39
C.19. Diagrama de secuencia . . . . .	40
C.20. Diagrama de despliegue . . . . .	41
D.1. Copiar dirección para clonar repositorio. . . . .	47
D.2. Seleccionar clonar dentro de <i>Fork</i> . . . . .	47

D.3. Clonar repositorio en <i>Fork</i> . . . . .	48
D.4. Abrir solución en Visual Studio. . . . .	49
D.5. Seleccionar solución del proyecto. . . . .	49
D.6. Abrir directorio de proyecto <i>Angular</i> . . . . .	50
D.7. Crear base de datos SQLExpress. . . . .	51
D.8. Abrir consola Visual Code. . . . .	52
D.9. Lanzar aplicación <i>.NET</i> . . . . .	52
D.10. Test SonarQube de aplicación <i>.NET</i> superados. . . . .	53
D.11. Resultados SonarQube de aplicación <i>.NET</i> . . . . .	54
D.12. Test SonarQube de aplicación <i>Angular</i> superados. . . . .	55
D.13. Resultados SonarQube de aplicación <i>Angular</i> . . . . .	55
E.1. Listado investigadores aplicación web. . . . .	59
E.2. Filtros aplicación web. . . . .	60
E.3. Columna de acciones del listado de investigadores. . . . .	60
E.4. Botón para lanzar el proceso de <i>Web Scraping</i> . . . . .	61
E.5. Captura selección de investigadores . . . . .	61
E.6. Botón para abrir ventana modal de fusión . . . . .	61
E.7. Captura ventana modal fusión investigadores . . . . .	62
E.8. Captura listado de publicaciones. . . . .	62
E.9. Botón de exportación de publicaciones. . . . .	63
E.10. Captura vista indicadores. . . . .	63

---

# Índice de tablas

---

A.1. Tareas del <i>sprint</i> 0. . . . .	2
A.2. Tareas del <i>sprint</i> 1. . . . .	2
A.3. Tareas del <i>sprint</i> 2. . . . .	3
A.4. Tareas del <i>sprint</i> 3. . . . .	3
A.5. Tareas del <i>sprint</i> 4. . . . .	4
A.6. Tareas del <i>sprint</i> 5. . . . .	5
A.7. Tareas del <i>sprint</i> 6. . . . .	6
A.8. Tareas del <i>sprint</i> final. . . . .	6
A.9. Coste personal del desarrollo del proyecto . . . . .	7
A.10. Coste del material utilizado. . . . .	8
A.11. Costes totales . . . . .	8
A.12. Licencias software. . . . .	8
B.1. CU-1 Realizar <i>Web Scraping</i> . . . . .	16
B.2. CU-2.1 Visualizar información de investigadores . . . . .	17
B.3. CU-2.2 Visualizar información de publicaciones . . . . .	18
B.4. CU-2.3 Visualizar información de indicadores . . . . .	19
B.5. CU-3.1 Exportar información de publicaciones . . . . .	20
B.6. CU-3.1 Exportar información de indicadores . . . . .	21
B.7. CU-4 Fusionar investigadores . . . . .	22
D.1. Versiones de las herramientas empleadas . . . . .	46





## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este apartado de los anexos se va a tratar la planificación temporal que ha llevado el proyecto, así como un estudio de viabilidad, tanto económica como legal.

### A.2. Planificación temporal

Para el desarrollo de este proyecto se ha decidido el uso de la metodología ágil *Scrum*. Debido al desarrollo continuo del proyecto, se han hecho las siguientes modificaciones sobre una metodología *Scrum* ideal:

- La duración de los *sprints* al comienzo será aproximadamente de dos semanas, aunque al final con el objetivo de apretar en el desarrollo, será de 1 semana.
- El equipo de desarrollo está formado por mí.
- El rol de cliente y de gestor de proyecto lo abordan mis tutores
- Entre cada *sprint* se realizará una reunión con los tutores. En donde primeramente se revisa si se ha cumplido con el *sprint* anterior, y después se definen las tareas para el próximo *sprint*.

### ***Sprint 0 - Tareas iniciales (21/03/2024 - 03/04/2024)***

El *sprint* inicial o *sprint 0* es en el que se realizó la búsqueda de información acerca de medidas de implementación y la forma en la que se iba a realizar el proyecto.

Tareas realizadas
Explorar el portal de investigación.
Búsqueda de información sobre <i>Web Scraping</i> .
Búsqueda de herramientas para la realización de <i>Web Scraping</i> .
Búsqueda de información sobre los frameworks más adecuados para la implementación.
Investigar sobre los contenedores <i>Docker</i> y su funcionamiento. Contrastar con los tutores las herramientas seleccionadas.
Investigar sobre base de datos a utilizar.
Construir la metodología de desarrollo a usar.
Explorar sobre licencias de uso de las herramientas preseleccionadas.

Tabla A.1: Tareas del *sprint 0*.

### ***Sprint 1 (03/04/2024 - 17/04/2024)***

El objetivo de este *sprint* fue la selección de *frameworks* a utilizar, la creación del repositorio remoto y la instalación de las herramientas necesarias para el desarrollo del proyecto.

Tareas realizadas
Elección de framework para <i>back-end</i> .
Elección de framework para <i>front-end</i> .
Elección de base de datos.
Instalación de Visual Studio.
Instalación de <i>.NET</i> .
Instalación de Visual Studio Code.
Instalación de <i>Angular</i> .
Creación de repositorio <i>GitHub</i> .
Subir proyecto de <i>.NET</i> a repositorio.
Subir proyecto de <i>Angular</i> a repositorio.

Tabla A.2: Tareas del *sprint 1*.

***Sprint 2 (17/04/2024 - 03/05/2024)***

Durante la realización de este *sprint* se instaló la base de datos y se realizaron pruebas sobre un despliegue futuro en un contenedor *Docker*

Tareas realizadas
Instalación SQL Management Studio.
Configuración base de datos.
Estudio de migraciones en <i>.NET</i> .
Instalación <i>Docker</i> .
Realizar una implementación de prueba en un contenedor <i>Docker</i> .

Tabla A.3: Tareas del *sprint 2*.***Sprint 3 (03/05/2024 - 17/05/2024)***

En el desarrollo de este *sprint* se comenzó la fase de *Web Scraping* en la aplicación. Lanzando este proceso en trozos de código individuales, comprobando su correcto funcionamiento.

Tareas realizadas
Instalación librería <i>Selenium</i> para la realización del <i>Web Scraping</i> .
Investigar sobre el uso más adecuado de <i>Selenium</i> .
Obtener URLs de departamentos.
Obtener un listado con todos los investigadores del portal.
Obtener la información personal de los investigadores.
Obtener departamento del investigador.
Obtener centro de investigación del investigador.
Obtener áreas del conocimiento del investigador.
Obtener programas de doctorado del investigador.
Obtener grupo de investigación del investigador.
Obtener el listado con las publicaciones del investigador.
Obtener la información general de las publicaciones.
Creación de modelos necesarios.
Realizar migración para la creación de una base de datos temporal.
Configurar el guardado de la información obtenida anteriormente en base de datos.

Tabla A.4: Tareas del *sprint 3*.

***Sprint 4 (17/05/2024 - 24/05/2024)***

En este *sprint* se generaron unos documentos con los investigadores que tenían algún error dentro del portal de investigación, para favorecer su solución interna. También se realizó la parte de *Web Scraping* de todos los indicadores disponibles en las publicaciones.

---

<b>Tareas realizadas</b>
Obtener los autores de cada publicación.
Relacionar las publicaciones de base de datos con los investigadores.
Añadir multiplicidad a los modelos de centros de investigación y departamentos.
Realizar función para obtener investigadores con múltiples centros de investigación.
Realizar función para obtener investigadores con múltiples departamentos.
Realizar función para obtener investigadores que se encuentren en el listado de un centro de investigación al que no están asociados.
Realizar función para obtener investigadores que se encuentren en el listado de un departamento al que no están asociados.
Obtener el indicador <i>Citas recibidas</i> de las publicaciones.
Obtener el indicador <i>Journal Impact Factor</i> de las publicaciones.
Obtener el indicador <i>SCImago</i> de las publicaciones.
Obtener el indicador <i>Scopus</i> de las publicaciones.
Obtener el indicador <i>Journal Citation Indicator</i> de las publicaciones.
Obtener el indicador <i>Dimensions</i> de las publicaciones.
Obtener el indicador <i>Dialnet</i> de las publicaciones.
Creación de modelos necesarios.
Actualización de estructura de base de datos mediante migración.

---

Tabla A.5: Tareas del *sprint 4*.

***Sprint 5 (24/05/2024 - 31/05/2024)***

El objetivo de este *sprint* era la obtención de las tesis y de las patentes de los investigadores, y el inicio del desarrollo de la aplicación web.

---

**Tareas realizadas**

---

Obtener el listado de las tesis del investigador.  
Obtener la información de las tesis.  
Asociar las tesis a sus investigadores.  
Obtener listado de patentes.  
Obtener información de las patentes.  
Asociar las patentes a sus creadores.  
Inicio de proyecto *Angular*.  
Instalación *Angular Material*.  
Instalación *FlexLayout*.  
Creación modulo principal.  
Configuración rutas.  
Creación de *header*.  
Creación de *footer*.

---

Tabla A.6: Tareas del *sprint* 5.

***Sprint* 6 (31/05/2024 - 14/06/2024)**

Es este *sprint* se preparó toda la funcionalidad para la conexión entre *back-end* y *front-end*, además de todos los listados, filtros, paginación y exportación a *Excel*.

Tareas realizadas
Realizar cambios en <i>Web Scraping</i> debido a cambios en portal de investigación.
Añadir listado de investigadores.
Añadir filtrado por nombre o apellidos.
Añadir filtrado por investigador relacionada.
Añadir modal con listado de publicaciones del investigador.
Añadir modal con vista de indicadores de la publicación.
Habilitar exportación a <i>Excel</i> de los listados.
Añadir modal para fusión de información de investigadores.

Tabla A.7: Tareas del *sprint* 6.

***Sprint* Final (14/06/2024 - Actualmente)**

Tareas realizadas
Realizar pruebas de rendimiento.
Configurar implementación en <i>Docker</i> .
Realizar memoria y anexos.

Tabla A.8: Tareas del *sprint* final.

**A.3. Estudio de viabilidad**

En este apartado se aborda ´la viabilidad del desarrollo de este proyecto, en concreto, nos centraremos en los aspectos económicos y legales.

**Viabilidad económica**

En este apartado se describirán los costes del proyecto y se determinará si el proyecto es viable económicamente,

### Costes de personal

En la realización del proyecto se ha contado solamente con un desarrollador web. El salario medio de un programador junior sin experiencia laboral en la ciudad de Burgos, oscila sobre los 18500 € anuales. Dividiendo este sueldo anual entre 14 pagas, nos quedarían 1320 €/mes.

Constando con que la fase de análisis y desarrollo comenzó a mediados de marzo, y la aplicación se terminó a finales de junio, serían unos 3 meses y medio.

Hay que añadir a este coste la contribución a la seguridad social. Todo ello se desglosa en la siguiente tabla:

Concepto	Coste
Salario mensual (Neto)	1320 €
Contingencias comunes (23,60 %)	311,52 €
Mecanismo Equidad Intergeneracional (0,58 %)	7,65 €
Tipo general (5,5 %)	72,60 €
FOGASA (0,2 %)	2,64 €
Formación profesional (0,6 %)	7,92 €
<b>Total mes</b>	<b>1722,33 €</b>
<b>Total 3 meses y medio</b>	<b>6028,15 €</b>

Tabla A.9: Coste personal del desarrollo del proyecto

### Costes material

El hardware que se ha necesitado para el desarrollo del proyecto es un ordenador de mesa con unas especificaciones por valor de 1000€. Planteamos una amortización del terminal a lo largo de 5 años de uso, por lo que para un periodo de 3 meses y medio el coste serían 58,33€.

En cuanto al coste de software, todos los programas y herramientas utilizadas son gratuitas, por lo que el coste de esta parte es 0€.

Concepto	Coste
Ordenador	58,33 €
Software utilizado	0 €
<b>Total</b>	<b>58,33 €</b>

Tabla A.10: Coste del material utilizado.

### Costes totales

Concepto	Coste
Personal	6028,15 €
Material	58,33 €
<b>Total</b>	<b>6086,48 €</b>

Tabla A.11: Costes totales

### Viabilidad legal

Para estudiar la viabilidad legal del proyecto se presenta a continuación (ver figura A.12), una lista con los programas y librerías utilizados con sus correspondientes licencias.

Dependencia	Licencia
Angular	MIT
Angular Material	MIT
FlexLayout	MIT
.NET	MIT
EntityFramework	EULA
Selenium	Apache
Visual studio 2022	EULA
Visual studio code	EULA
SQL Server Management Studio 20	EULA
Docker	Apache

Tabla A.12: Licencias software.



Como se puede apreciar en la tabla anterior, todas las licencias de las herramientas utilizadas son de dominio publico y código abierto, por lo que su utilización es gratuita.

La licencia seleccionada para el proyecto ha sido MIT, debido a la compatibilidad con las licencias antes mencionadas en la tabla, y la permisibilidad de reutilización de software en desarrollos futuros.



## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En este apartado se van a especificar los requisitos funcionales, los requisitos no funcionales y las historias de usuario.

### B.2. Objetivos generales

Los objetivos generales que debe cumplir la aplicación son los siguientes:

- Generación de la base de datos mediante *Web Scraping*
- Visualización y filtrado de los investigadores
- Visualización de las publicaciones del investigador, junto a sus indicadores
- Exportación a *Excel*
- Fusión de información de investigadores

## B.3. Catálogo de requisitos

En esta sección del anexo se van a listar los requisitos funcionales y los no funcionales junto con una breve descripción de cada uno.

### Requisitos funcionales

- **RF-1 Sistema de obtención de información del portal de investigación:** El usuario podrá lanzar el proceso de *Web Scraping*.
- **RF-2 Visualizar información obtenida:** Los usuarios podrán visualizar la información del portal del investigador que se ha obtenido mediante *Web Scraping*.
  - **RF-2.1 Visualizar investigadores:** El usuario podrá visualizar un listado de los investigadores, con posibilidad de filtrar por unos campos concretos.
  - **RF-2.2 Visualizar publicaciones:** El usuario podrá visualizar un listado con las publicaciones del investigador.
  - **RF-2.3 Visualizar indicadores:** El usuario podrá visualizar los indicadores de las publicaciones.
- **RF-3 Exportación de indicadores:** Los usuarios podrán exportar los indicadores.
  - **RF-3.1 Exportación de indicadores de publicación:** El usuario podrá exportar los indicadores de una publicación concreta.
  - **RF-3.2 Exportación de indicadores de todas las publicaciones de un investigador:** El usuario podrá exportar en un mismo archivo, los indicadores de todas las publicaciones de un investigador.
- **RF-4 Fusión de investigadores:** Los usuarios podrán asignar toda la información de los investigadores seleccionados a un investigador elegido.
  - **RF-4.1 Selección de investigadores a fusionar:** El usuario seleccionara dentro del listado de investigadores aquellos que quiera fusionar en uno solo.

- **RF-4.2 Selección investigador principal:** El usuario elegirá dentro de los investigadores anteriores cual sera al que vaya a parar toda la información.

## Requisitos no funcionales

- **RNF-1 Usabilidad:** La aplicación tiene que ser intuitiva, visual y de fácil acceso para el usuario.
- **RNF-2 Rendimiento:** Los tiempos de ejecución del *back-end* y los tiempos de carga de la aplicación web deben ser lo más óptimos posible.
- **RNF-3 Mantenibilidad:** La aplicación tiene que tener un fácil mantenimiento y soporte.
- **RNF-4 Escalabilidad:** La aplicación debe estar abierta a poder añadir funcionalidad en *back-end* y más páginas e interfaces en la aplicación web.
- **RNF-5 Compatibilidad:** La aplicación tiene que poder ser ejecutada en los principales navegadores: *Chrome*, *Edge*...
- **RNF-6 Portabilidad:** La aplicación ha de tener una sencilla implementación en un terminal distinto.

## B.4. Especificación de requisitos

En esta sección se van a desarrollar los casos de uso relacionados con los requisitos funcionales listados anteriormente, con sus correspondientes diagramas.

### Actores

Para el uso de la aplicación, actualmente no hay distinción de roles, por lo que todos son usuarios básicos.

## Diagramas de casos de uso

### Nivel 0

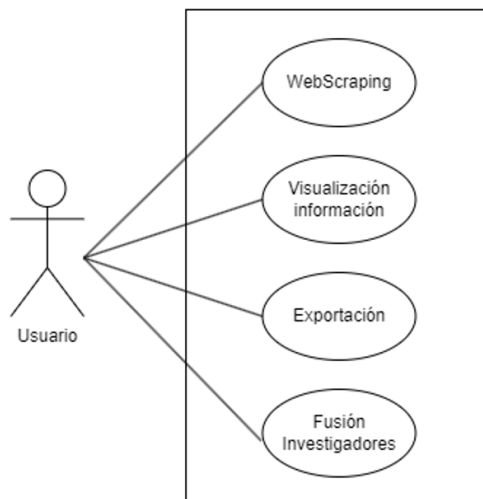


Figura B.1: Diagrama de casos de uso nivel 0.

### Nivel 1

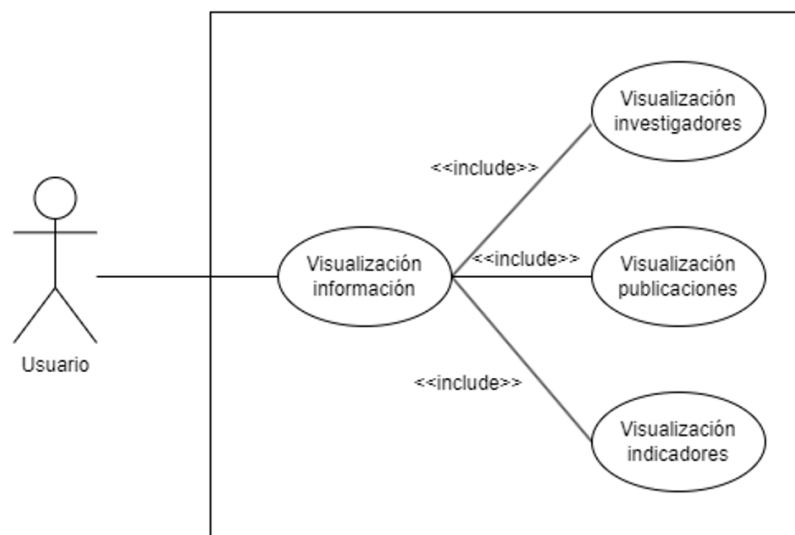


Figura B.2: Diagrama de casos de uso nivel 1. Visualización de la información.

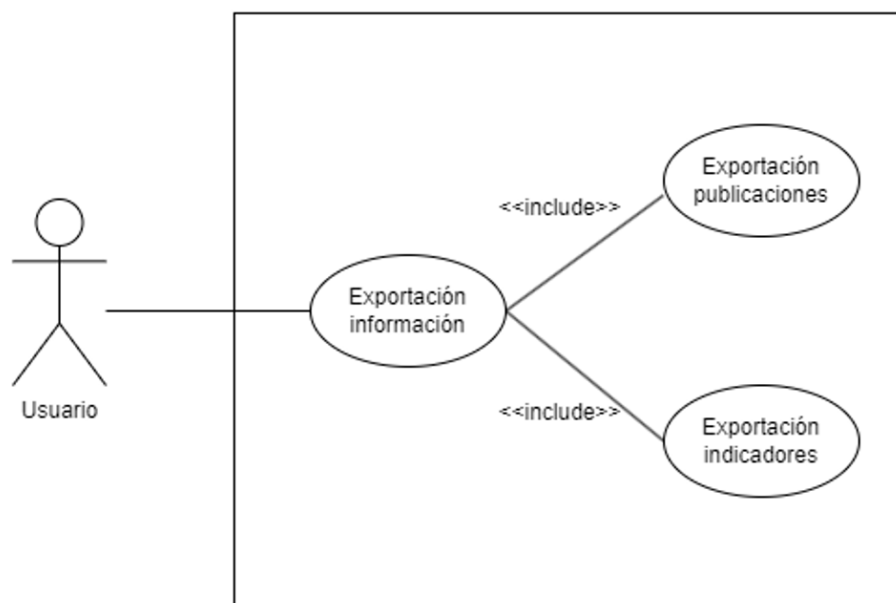


Figura B.3: Diagrama de casos de uso nivel 1. Exportación de la información.

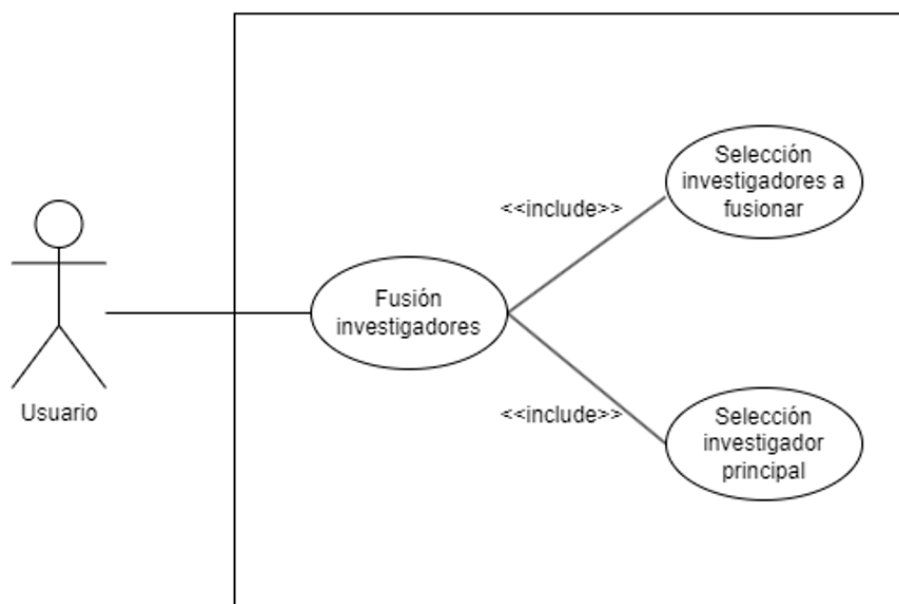


Figura B.4: Diagrama de casos de uso nivel 1. Fusión investigadores.

## Especificación de casos de uso

CU-1	Realizar <i>Web Scraping</i>
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-1
<b>Descripción</b>	El usuario lanza el proceso de <i>Web Scraping</i>
<b>Precondición</b>	No se haya lanzado un proceso de <i>Web Scraping</i> previo
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. El usuario presiona el botón para lanzar el proceso de <i>Web Scraping</i>.</li> <li>3. El usuario espera que se complete el proceso, aproximadamente unas 14 horas.</li> </ol>
<b>Excepciones</b>	El usuario cierra la aplicación web mientras se está realizando el proceso
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Realizar *Web Scraping*



CU-2.1	Visualizar información de investigadores
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-2.1
<b>Descripción</b>	El usuario accede a la información de los investigadores
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. El usuario accede a la aplicación web.</li><li>2. En la página principal aparece un listado con los investigadores. El usuario filtra por los campos que precise, y obtiene un listado paginado con la infracción de los investigadores que cumplen estos filtros.</li></ol>
<b>Excepciones</b>	La base de datos no tiene registros
<b>Importancia</b>	Alta

Tabla B.2: CU-2.1 Visualizar información de investigadores

CU-2.2	Visualizar información de publicaciones
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-2.2
<b>Descripción</b>	El usuario accede a la información de las publicaciones de los investigadores
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. En el listado de investigadores, el usuario presiona el botón asociado para ver las publicaciones del investigador que precise.</li> <li>3. Se despliega una ventana modal con el listado de publicaciones del investigador, el usuario accede a la información de las publicaciones.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. La base de datos no tiene registros.</li> <li>2. El investigador seleccionado no tiene publicaciones asociadas.</li> </ol>
<b>Importancia</b>	Alta

Tabla B.3: CU-2.2 Visualizar información de publicaciones

CU-2.3	Visualizar información de indicadores
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-2.3
<b>Descripción</b>	El usuario accede a la información de los indicadores de las publicaciones
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. En el listado de investigadores, el usuario presiona el botón asociado para ver las publicaciones del investigador que precise.</li> <li>3. Se despliega una ventana modal con el listado de publicaciones del investigador, el usuario presiona en el botón asociado para ver los indicadores de la publicación.</li> <li>4. Se despliega otra ventana modal con los indicadores de la publicación. El usuario accede a la información de los indicadores que precise.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. La base de datos no tiene registros.</li> <li>2. El investigador seleccionado no tiene publicaciones asociadas.</li> <li>3. La publicación seleccionada no tiene indicadores asociadas.</li> </ol>
<b>Importancia</b>	Alta

Tabla B.4: CU-2.3 Visualizar información de indicadores

CU-3.1	Exportar información de publicaciones
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-3.1
<b>Descripción</b>	El usuario exporta la información de las publicaciones
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. En el listado de investigadores, el usuario presiona el botón asociado para ver las publicaciones del investigador que precise.</li> <li>3. Se despliega una ventana modal con el listado de publicaciones del investigador, el usuario presiona en el botón de exportar.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. La base de datos no tiene registros.</li> <li>2. El investigador seleccionado no tiene publicaciones asociadas.</li> </ol>
<b>Importancia</b>	Media

Tabla B.5: CU-3.1 Exportar información de publicaciones

CU-3.2	Exportar información de indicadores
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-3.1
<b>Descripción</b>	El usuario exporta la información de los indicadores de una publicación
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. En el listado de investigadores, el usuario presiona el botón asociado para ver las publicaciones del investigador que precise.</li> <li>3. Se despliega una ventana modal con el listado de publicaciones del investigador, el usuario presiona en el botón asociado para ver los indicadores de la publicación.</li> <li>4. Se despliega otra ventana modal con los indicadores de la publicación. El usuario presiona en el botón de exportar.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. La base de datos no tiene registros.</li> <li>2. El investigador seleccionado no tiene publicaciones asociadas.</li> <li>3. La publicación seleccionada no tiene indicadores asociadas.</li> </ol>
<b>Importancia</b>	Media

Tabla B.6: CU-3.1 Exportar información de indicadores

CU-4	Fusionar investigadores
<b>Autor</b>	Anónimo
<b>Requisitos asociados</b>	RF-4, RF-4.1, RF-4.2
<b>Descripción</b>	El usuario asocia la información los investigadores seleccionados, a un solo investigador
<b>Precondición</b>	La base de datos esta rellena con registros obtenidos del <i>Web Scraping</i>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación web.</li> <li>2. En el listado de investigadores, el usuario selecciona los investigadores que se quieren fusionar y presiona en el botón de fusionar.</li> <li>3. Se despliega una ventana modal con el listado de investigadores seleccionados. El usuario marca el investigador al que se le asociará la información del resto de seleccionados.</li> <li>4. Una vez seleccionado un investigador, presionará el botón de guardar.</li> </ol>
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>1. La base de datos no tiene registros.</li> <li>2. La fusión se realiza sobre investigadores que no correspondían, y el proceso es irreversible.</li> </ol>
<b>Importancia</b>	Alta

Tabla B.7: CU-4 Fusionar investigadores

## *Apéndice C*

---

# **Especificación de diseño**

---

## **C.1. Introducción**

En este anexo se van a explicar los diferentes aspectos de diseño que se han llevado a cabo para la realización del proyecto. Vamos a diferenciar el diseño de datos, el diseño procedimental y el diseño arquitectónico.

## **C.2. Diseño de datos**

Dentro de este apartado se va a explicar la estructura de tablas de bases de datos que se ha empleado. Se realizó un estudio del portal de investigación sobre las tablas que iban a ser necesarias para guardar la información obtenida del *Web Scraping* y las relaciones entre estas tablas.

Podemos dividir la estructura de la base de datos en dos tablas principales: Investigadores y publicaciones.

### **Investigadores**

La tabla de investigadores se trata del eje principal en torno al que gira la aplicación.

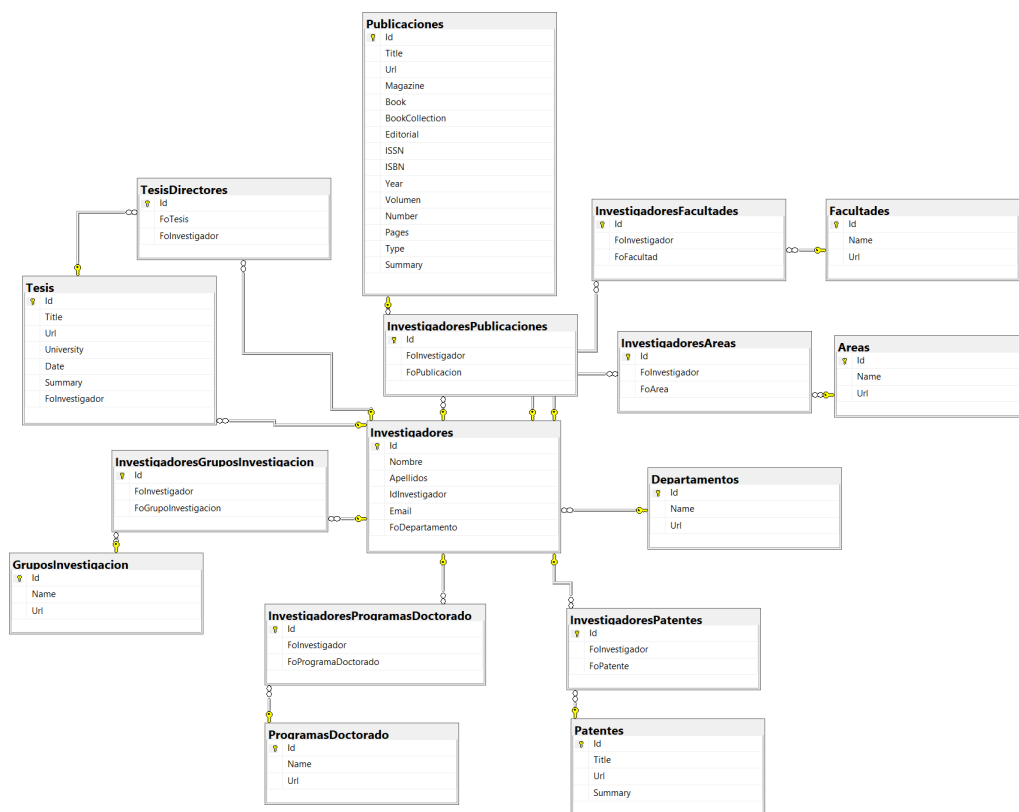


Figura C.1: Diagrama relacional de tabla investigadores

La tabla **investigadores** contiene los siguientes campos:

- **Id**: Identificador único.
- **Nombre**: Nombre del investigador.
- **Apellidos**: Apellidos del investigador.
- **IdInvestigador**: Identificador único del investigador en el portal de investigación.
- **Email**: Email del investigador.
- **FoDepartamento**: Clave foránea para relación con tabla departamentos.



Investigadores	
	Id
	Nombre
	Apellidos
	IdInvestigador
	Email
	FoDepartamento

Figura C.2: Tabla investigadores

Para profundizar en las relaciones de la tabla de investigadores, las abordaremos individualmente.

## Departamentos

La tabla **investigadores** tiene una relación 1:1 con la tabla **departamentos**, un investigador pertenece a un departamento. La tabla departamentos tiene una relación 1:N, un departamento contiene varios investigadores.

Campos de la tabla **departamentos**:

- **Id**: Identificador único.
- **Nombre**: Nombre del investigador.
- **Apellidos**: Apellidos del investigador.
- **IdInvestigador**: Identificador único del investigador en el portal de investigación.
- **Email**: Email del investigador.
- **FoDepartamento**: Clave foránea para relación con tabla **departamentos**.

Departamentos	
	Id
	Name
	Url

Figura C.3: Tabla departamentos

## Programas de doctorado

La tabla **investigadores** se relaciona con la tabla **programas de doctorado** mediante una tabla intermedia con relación N:M. Un investigador puede pertenecer a varios programas de doctorado, y un programa de doctorado contiene varios investigadores.

Campos de la tabla **programasDoctorado**:

- **Id**: Identificador único.
- **Name**: Nombre del programa de doctorado.
- **Url**: Dirección URL del programa de doctorado en el portal de investigación.

Campos de la tabla **investigadores\_programasDoctorado**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoProgramaDoctorado**: Clave foránea para relación con tabla **programas de doctorado**.

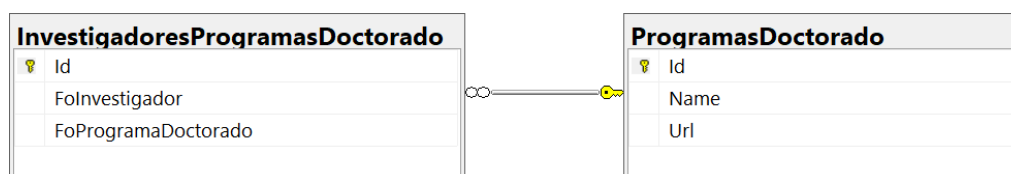


Figura C.4: Tabla programas de doctorado

## Áreas del conocimiento

La tabla **investigadores** se relaciona con la tabla **áreas** mediante una tabla intermedia con relación N:M. Un investigador puede pertenecer a varias áreas del conocimiento, y una área del conocimiento contiene varios investigadores.

Campos de la tabla **areas**:

- **Id**: Identificador único.
- **Name**: Nombre del área del conocimiento.
- **Url**: Dirección URL del área del conocimiento en el portal de investigación.

Campos de la tabla **investigadores\_areas**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoArea**: Clave foránea para relación con tabla **áreas**.

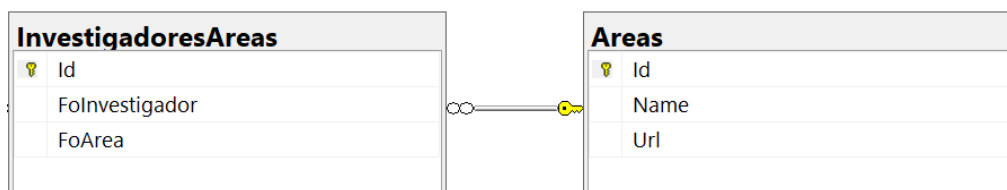


Figura C.5: Tabla áreas

## Centros de investigación

La tabla **investigadores** se relaciona con la tabla **centros de investigación** mediante una tabla intermedia con relación N:M. Un investigador puede pertenecer a varios centros de investigación, y un centro de investigación contiene varios investigadores.

Campos de la tabla **centrosInvestigacion**:

- **Id**: Identificador único.
- **Name**: Nombre del centro de investigación.
- **Url**: Dirección URL del centro de investigación en el portal de investigación.

Campos de la tabla **investigadores\_centrosInvestigacion**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoCentroInvestigacion**: Clave foránea para relación con tabla **centros de investigación**.

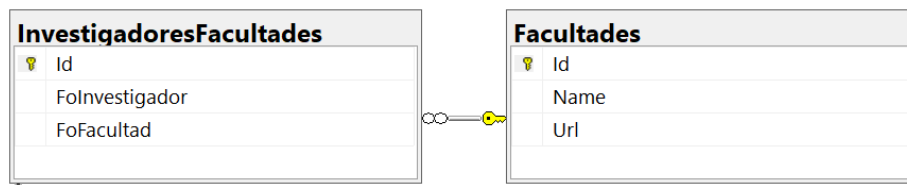


Figura C.6: Tabla centros de investigación

### Grupos de investigación

La tabla **investigadores** se relaciona con la tabla **grupos de investigación** mediante una tabla intermedia con relación N:M. Un investigador puede pertenecer a varios grupos de investigación, y un grupo de investigación contiene varios investigadores.

Campos de la tabla **gruposInvestigacion**:

- **Id**: Identificador único.
- **Name**: Nombre del grupo de investigación.
- **Url**: Dirección URL del grupo de investigación en el portal de investigación.

Campos de la tabla **investigadores\_gruposInvestigacion**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoGrupoInvestigacion**: Clave foránea para relación con tabla **grupos de investigación**.

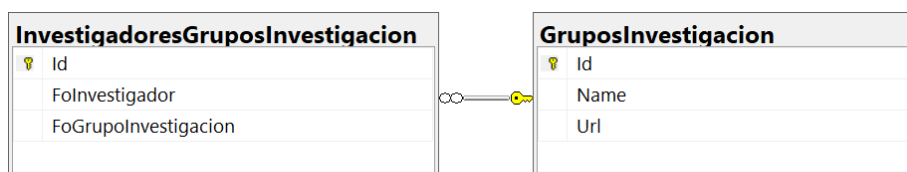


Figura C.7: Tabla grupos de investigación

## Tesis

La tabla **investigadores** tiene una relación 1:N con la tabla **tesis**, un investigador esta asociado con varias tesis (tesis realizada, tesis dirigidas y tesis que ha participado en el tribunal. a un departamento. La tabla **tesis** tiene una relación 1:1, una tesis esta realizada por un investigador.

Campos de la tabla **Tesis**:

- **Id**: Identificador único.
- **Title**: Título de la tesis.
- **Url**: Dirección URL de la tesis en el portal de investigación.
- **University**: Universidad en la que se realizó la defensa de la tesis.
- **Date**: Fecha de la defensa de la tesis.
- **Summary**: Resumen de la tesis.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.

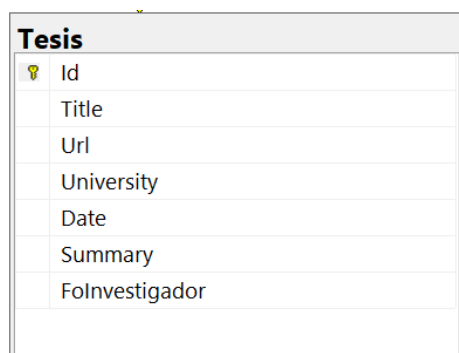


Figura C.8: Tabla tesis

## Patentes

La tabla **investigadores** se relaciona con la tabla **patentes** mediante una tabla intermedia con relación N:M. Un investigador puede haber creado varias patentes, y una patente puede ser creada por varios investigadores.

Campos de la tabla **patentes**:

- **Id**: Identificador único.
- **Title**: Título de la patente.
- **Url**: Dirección URL de la patente en el portal de investigación.
- **Summary**: Resumen de la patente.

Campos de la tabla **investigadores\_patentes**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoPatente**: Clave foránea para relación con tabla **patentes**.

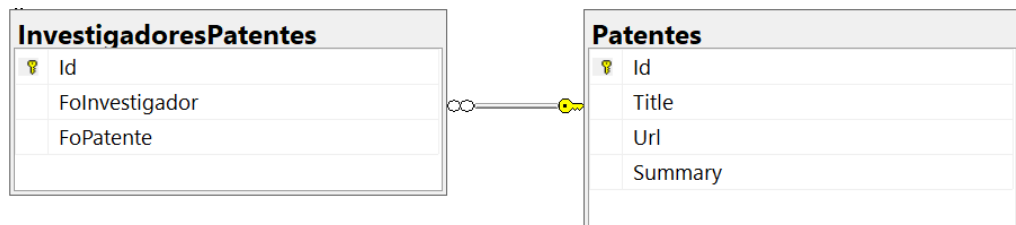


Figura C.9: Tabla patentes

## Publicaciones

La tabla de publicaciones actúa como intermediario para acceder a los indicadores requeridos.

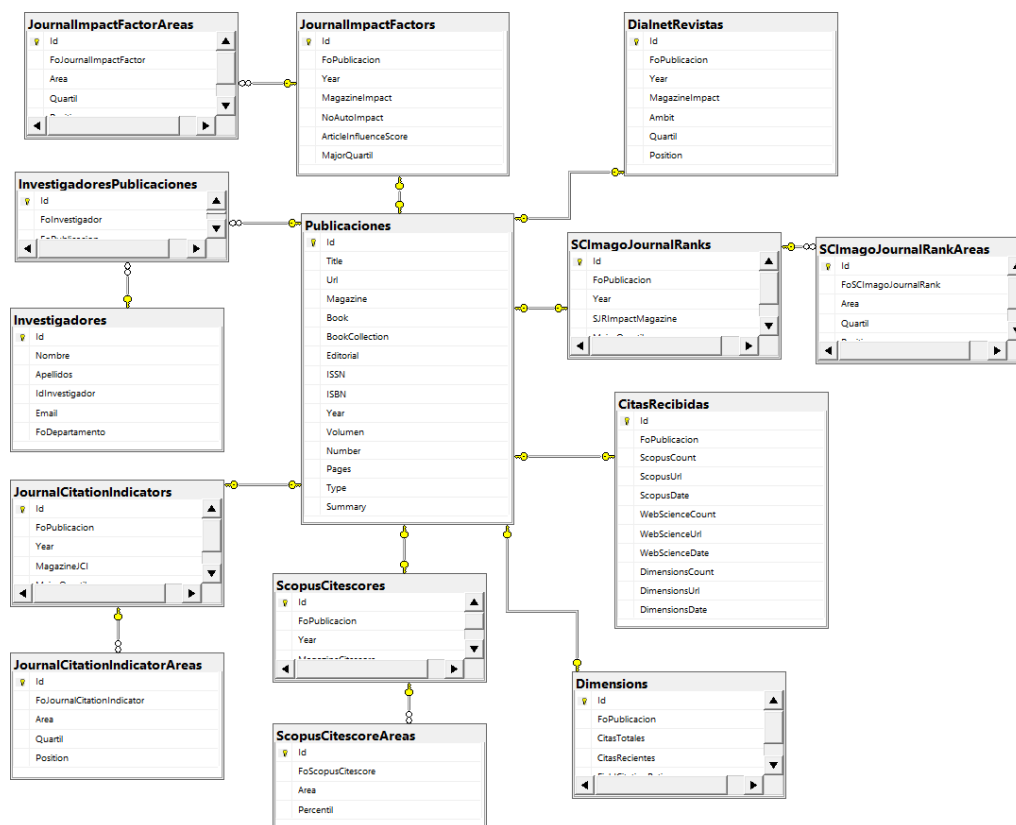


Figura C.10: Diagrama relacional de tabla publicaciones

La tabla **publicaciones** contiene los siguientes campos:

- **Id**: Identificador único.
- **Title**: Título de la publicación.
- **Url**: Dirección URL de la publicación en el portal de investigación.
- **Magazine**: Nombre de la revista en la que se ha publicado.
- **Book**: Nombre del libro en la que se ha publicado.

- **BookCollection:** Nombre de la colección de libros en los que se ha publicado.
- **Editorial:** Nombre de la editorial de la publicación.
- **ISSN:** Número ISSN.
- **ISBN:** Número ISBN.
- **Year:** Año de publicación.
- **Volumen:** Volumen de revista o libro que se encuentra la publicación.
- **Number:** Número de revista o libro donde se ha publicado.
- **Pages:** Páginas en las que se encuentra la publicación.
- **Type:** Tipo de publicación.
- **Summary:** Resumen de la publicación.

Publicaciones	
Id	
Title	
Url	
Magazine	
Book	
BookCollection	
Editorial	
ISSN	
ISBN	
Year	
Volumen	
Number	Publicaciones
Pages	
Type	
Summary	

Figura C.11: Tabla publicaciones



Para profundizar en las relaciones de la tabla de publicaciones, las abordaremos individualmente.

### Citas recibidas

La tabla **publicaciones** se relaciona con la tabla **citas recibidas** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de citas recibidas, y un registro de citas recibidas pertenece a una publicación.

Campos de la tabla **citasRecibidas**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **ScopusCount**: Número de menciones en *Scopus*.
- **ScopusUrl**: Dirección URL de la publicación en *Scopus*.
- **ScopusDate**: Fecha de última actualización de *Scopus*.
- **WebScienceCount**: Número de menciones en *Web of Science*.
- **WebScienceUrl**: Dirección URL de la publicación en *Web of Science*.
- **WebScienceDate**: Fecha de última actualización de *Web of Science*.
- **DimensionsCount**: Número de menciones en *Dimensions*.
- **DimensionsUrl**: Dirección URL de la publicación en *Dimensions*.
- **DimensionsDate**: Fecha de última actualización de *Dimensions*.


CitasRecibidas	
	Id
	FoPublicacion
	ScopusCount
	ScopusUrl
	ScopusDate
	WebScienceCount
	WebScienceUrl
	WebScienceDate
	DimensionsCount
	DimensionsUrl
	DimensionsDate

Figura C.12: Tabla citas recibidas

### Journal Impact Factors

La tabla **publicaciones** se relaciona con la tabla **journal impact factors** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de journal impact factors, y un registro de journal impact factors pertenece a una publicación. Adicionalmente, se ha creado la tabla de **journal impact factors areas** para poder guardar múltiples áreas dentro de un registro de journal impact factors.

Campos de la tabla **journalImpactFactors**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **Year**: Año del indicador.
- **Magazine impact**: Impacto en la revista.
- **NoAutoImpact**: Impacto sin autocitas.
- **ArticleInfluenceScore**: Puntuación de influencia en la revista.
- **MajorQuartil**: Cuartil mayor.

Campos de la tabla **journalImpactFactorAreas**:

- **Id**: Identificador único.
- **FoJournalImpactFactor**: Clave foránea para relación con tabla **journalImpactFactors**.
- **Area**: Nombre del área.
- **Quartil**: Cuartil en que se encuentra.
- **Position**: Posición en el área.

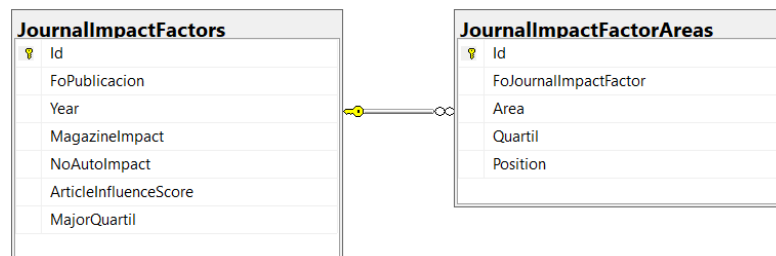


Figura C.13: Tabla journal impact factors

### Journal Citation Indicators

La tabla **publicaciones** se relaciona con la tabla **journal citation indicators** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de journal citation indicators, y un registro de journal citation indicators pertenece a una publicación. Adicionalmente, se ha creado la tabla de **journal citation indicator areas** para poder guardar múltiples áreas dentro de un registro de journal citation indicators.

Campos de la tabla **journalCitationIndicators**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **Year**: Año del indicador.
- **MagazineJCI**: Impacto en la revista.
- **MajorQuartil**: Cuartil mayor.

Campos de la tabla **journalCitationIndicatorAreas**:

- **Id**: Identificador único.
- **FoJournalCitationIndicator**: Clave foránea para relación con tabla **journalCitationIndicators**.
- **Area**: Nombre del área.
- **Quartil**: Cuartil en que se encuentra.
- **Position**: Posición en el área.

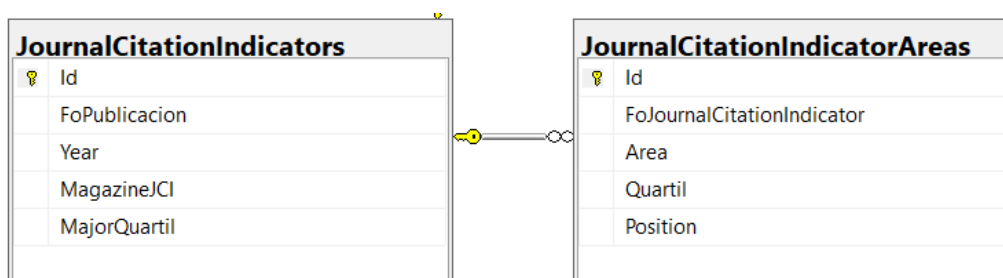


Figura C.14: Tabla journal citation indicators

### Scopus citescorres

La tabla **publicaciones** se relaciona con la tabla **scopus citescorres** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de Scopus citescorres, y un registro de Scopus citescorres pertenece a una publicación. Adicionalmente, se ha creado la tabla de **Scopus citescorres areas** para poder guardar múltiples áreas dentro de un registro de Scopus citescorres.

Campos de la tabla **scopusCitescores**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **Year**: Año del indicador.
- **MagazineCitescore**: Puntuación de *Scopus*.

Campos de la tabla **scopusCitescoreAreas**:

- **Id**: Identificador único.
- **FoScopusCitescore**: Clave foránea para relación con tabla **scopus-Citescores**.
- **Area**: Nombre del área.
- **Percentil**: Percentil en el área.

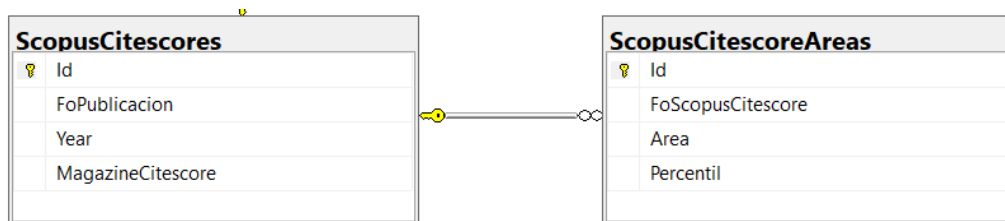


Figura C.15: Tabla scopus citescore

### SCImago journal ranks

La tabla **publicaciones** se relaciona con la tabla **SCImago journal ranks** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de SCImago journal ranks, y un registro de SCImago journal ranks pertenece a una publicación. Adicionalmente, se ha creado la tabla de **SCImago journal ranks areas** para poder guardar múltiples áreas dentro de un registro de SCImago journal ranks.

Campos de la tabla **SCImagoJournalRanks**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **Year**: Año del indicador.
- **SJRImpactMagazine**: Puntuación de impacto en la revista.
- **MajorQuartil**: Cuartil mayor.

Campos de la tabla **SCImagoJournalRanksAreas**:

- **Id**: Identificador único.
- **FoSCImagoJournalRanks**: Clave foránea para relación con tabla **SCImagoJournalRanks**.
- **Area**: Nombre del área.
- **Quartil**: Cuartil en que se encuentra.
- **Position**: Posición en el área.

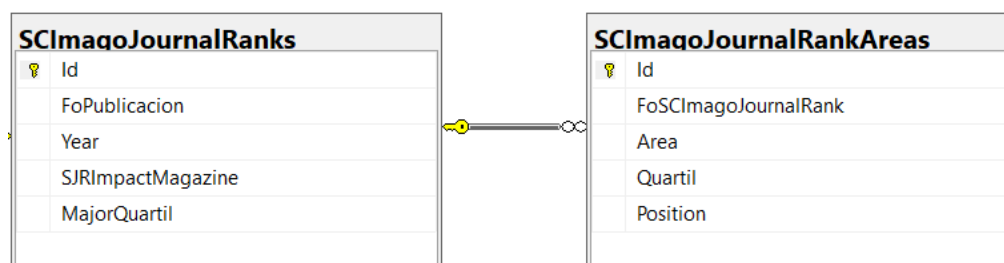


Figura C.16: Tabla SCImago journal ranks

### Dialnet revistas

La tabla **publicaciones** se relaciona con la tabla **dialnet revistas** con una relación 1:1 en ambos extremos. Una publicación tiene un registro de dialnet revistas, y un registro de dialnet revistas pertenece a una publicación.

Campos de la tabla **Dialnet revistas**:

- **Id**: Identificador único.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.
- **Year**: Año del indicador.
- **MagazineImpact**: Puntuación de impacto en la revista.
- **Ambit**: Ámbito.
- **Quartil**: Cuartil en que se encuentra.
- **Position**: Posición en el ámbito..

DialnetRevistas	
?	Id
	FoPublicacion
	Year
	MagazineImpact
	Ambit
	Quartil
	Position

Figura C.17: Tabla dialnet revistas

## Investigadores publicaciones

Para unir estas dos tablas, que con sus relaciones constituyen la estructura de la base de datos, se ha optado por incluir una tabla intermedia **investigadores publicaciones**. De esta forma, un investigador puede estar asociado a varias publicaciones y una publicación puede estar asociada a varios investigadores.

Campos de la tabla **investigadores\_publicaciones**:

- **Id**: Identificador único.
- **FoInvestigador**: Clave foránea para relación con tabla **investigadores**.
- **FoPublicacion**: Clave foránea para relación con tabla **publicaciones**.



Figura C.18: Tabla investigadores publicaciones

## C.3. Diseño procedimental

Se han tomado los casos de uso: Lanzar *Web Scraping*, visualización investigadores y fusión de investigadores. Para la explicación de su flujo se ha realizado el siguiente diagrama de secuencia.

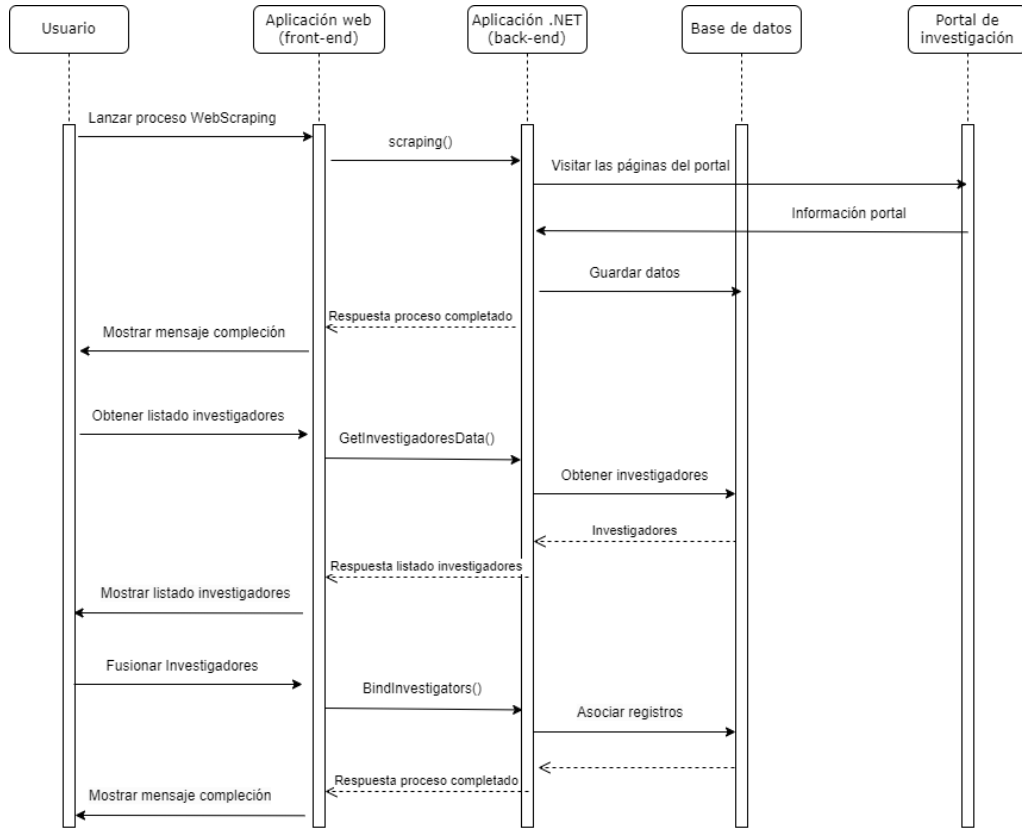


Figura C.19: Diagrama de secuencia

## C.4. Diseño arquitectónico

El despliegue del proyecto consta de un contenedor *Docker* que contiene la aplicación web, la aplicación *back-end* y la base de datos. El usuario accede al sitio web que proporciona *Angular* e interacciona con las acciones de la aplicación web. Al realizar ciertas acciones como filtrar o abrir ventanas modales con listados, la aplicación web realiza llamadas *HTTP Request* hacia la aplicación *back-end*. La aplicación *back-end* interacciona con los registros de base de datos y devuelve las llamadas *HTTP Response* hacia la aplicación web, que recibe la respuesta y completa la acción iniciada por el usuario.

También se incluye el portal de investigación al que accede la aplicación *back-end* para realizar el *Web Scraping*.



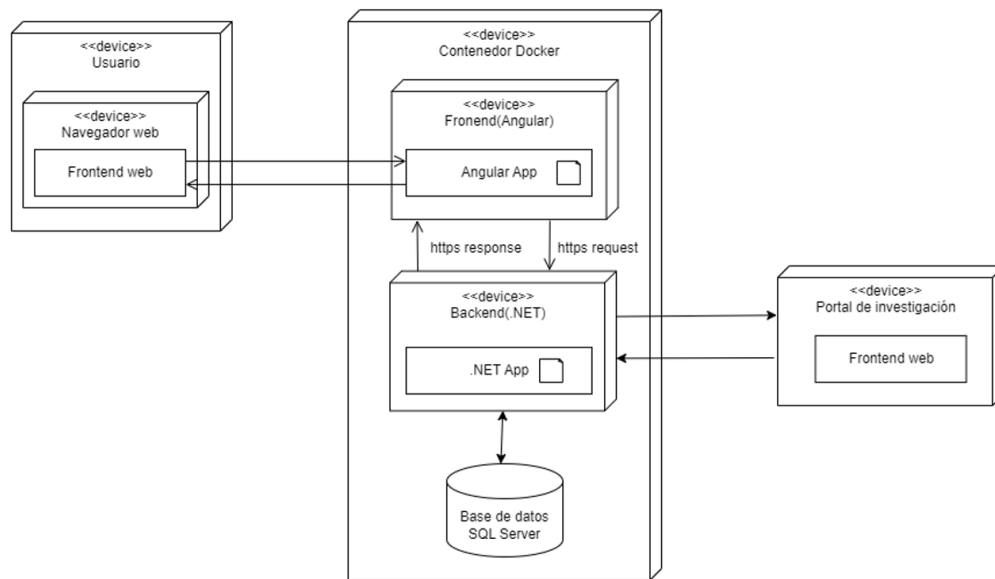


Figura C.20: Diagrama de despliegue



## Apéndice *D*

---

# Documentación técnica de programación

---

### D.1. Introducción

En este anexo se va explicar la estructura del proyecto, los requisitos necesarios para poder ejecutar la herramienta desde otro terminal y una guía de instalación para montar el entorno de desarrollo local.

### D.2. Estructura de directorios

La estructura de directorio raíz del repositorio consta de 5 elementos:

- `/WebScrapingAPI`: Carpeta que el proyecto de *.NET* (*back-end*).
- `/WebScrapingFront`: Carpeta que contiene el proyecto de *Angular* (*front-end*).
- `/docs`: Contiene la documentación del proyecto.
- `README`: Archivo `.txt` que contiene la información de la herramienta, así como unos vídeos demo donde se muestra el uso de la aplicación web.
- `LICENSE`: Contiene la licencia MIT.
- `docker-compose.yml`: Fichero que mediante su ejecución se construyen los contenedores *Docker*.

- `migrate.sh`: *Script* que se lanza en la ejecución de `docker-compose.yml` y realiza la migración para construir la base de datos en el contenedor.

Dentro de la carpeta que contiene el proyecto de *.NET* (`/WebScrapingAPI`) destacamos los siguientes directorios y ficheros:

- `/WebScrapingAPI/`: Destacar de esta parte del *path* que se encuentra el fichero con la solución para abrir el proyecto en *Visual Studio*.
- `/WebScrapingAPI/Dockerfile`: Fichero de configuración de la aplicación *.NET* para el despliegue a *Docker*.
- `/WebScrapingAPI/WebScrapingAPI/Program.cs`: Archivo encargado del arranque de la aplicación y de los servicios que se emplean.
- `/WebScrapingAPI/WebScrapingAPI/appsettings.json`: Archivo que alberga la configuración del proyecto.
- `/WebScrapingAPI/WebScrapingAPI/ApplicationDbContext`: Fichero usado para la configuración de los modelos y las relaciones. Fichero con mucha importancia debido a que la migración usará dicha configuración para la generación de la base de datos.
- `/WebScrapingAPI/WebScrapingAPI/Controllers`: En este directorio se encuentra la funcionalidad que se desarrolla. En este caso contamos con 2 controladores, uno se encarga de realizar el *Web Scraping* y es el encargado de enviar la información a la aplicación web.
- `/WebScrapingAPI/WebScrapingAPI/Models`: Contiene los objetos de la aplicación.

En la carpeta que contiene el proyecto de *Angular* (`/WebScrapingFront`) podemos resaltar los siguientes directorios y ficheros:

- `/WebScrapingFront/`: Destacar de esta parte del *path* los ficheros `package.json` y `package-lock.json`, que contienen la configuración de librerías utilizadas, y el fichero `angular.json` que contiene la configuración de arranque de la aplicación.
- `/WebScrapingFront/Dockerfile`: Fichero de configuración de la aplicación web para el despliegue a *Docker*.

- `/WebScrapingFront/node\_modules`: Directorio que contiene todas las librerías instaladas físicamente.
- `/WebScrapingFront/src`: Carpeta donde se encuentra el desarrollo de la aplicación.
- `/WebScrapingFront/src/assets/img`: Carpeta donde se alojan las imágenes que se utilizan.
- `/WebScrapingFront/src/styles.scss`: Fichero de los estilos generales de la aplicación.
- `/WebScrapingFront/src/app/app.module`: Fichero que contiene la importación de librerías y componentes utilizados en la aplicación, además de la configuración de rutas.
- `/WebScrapingFront/src/app/app.component`: Fichero padre de todos los módulos.
- `/WebScrapingFront/src/app/services`: Directorio con los servicios implementados. En el caso del proyecto, se encuentran el servicio de exportación a *Excel*, que permite la generación y descarga de documentos `.xlsx`, y el servicio que permite la conexión con la aplicación *back-end* mediante llamadas HTTP.
- `/WebScrapingFront/src/app/modules`: Directorio donde se encuentran los componentes de la aplicación.
- `/WebScrapingFront/src/app/modules/content-layout`: Componente padre al resto que se encarga de la enrutamiento entre componentes.
- `/WebScrapingFront/src/app/modules/material`: Componente para poder importar toda la funcionalidad de *Angular Material* a la vez.
- `/WebScrapingFront/src/app/modules/navbar`: Componente que compone el *header* de la aplicación.
- `/WebScrapingFront/src/app/modules/footer`: Componente que compone el *footer* de la aplicación.
- `/WebScrapingFront/src/app/modules/investigadores`: Componente donde se han desarrollado los listados y demás gestión y visualizado de la información referente a los investigadores.

### D.3. Manual del programador

Para conseguir compatibilidad entre todas las herramientas utilizadas en el desarrollo, es importante que se utilicen las versiones proporcionadas en la siguiente tabla:

Librería/Programa	Versión
AngularCLI	v17.3.3
Node.js	v20.12
Npm	v10.5
Angular Material	v17.3.9
FlexLayout	v15.0
.NET	v8.0
Selenium	v4.19
EntityFramework	v8.0.4

Tabla D.1: Versiones de las herramientas empleadas

### D.4. Compilación, instalación y ejecución del proyecto

Para preparar el desarrollo en entorno local, primeramente hay que instalar las herramientas necesarias.

#### Fork

*Fork* es la herramienta que utilizaremos para realizar los *commits* y llevar a cabo la gestión de las ramas.

Para su instalación, visitaremos la página oficial de **Fork**<sup>1</sup> y procederemos con su descarga.

Una vez tengamos la aplicación instalada, vamos a descargar el repositorio remoto del proyecto, que se encuentra alojado en GitHub. Enlace: <https://github.com/iker-gayangos-tfg/WebScrapingTFG>

Para obtener el repositorio accederemos al enlace del repositorio, y copiaremos la dirección para clonarlo.

---

<sup>1</sup><https://git-fork.com/>

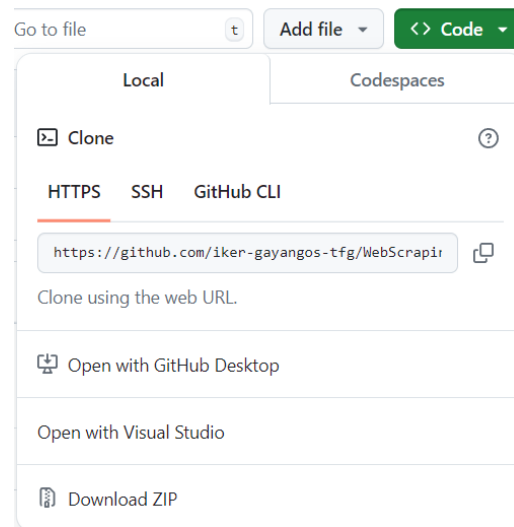
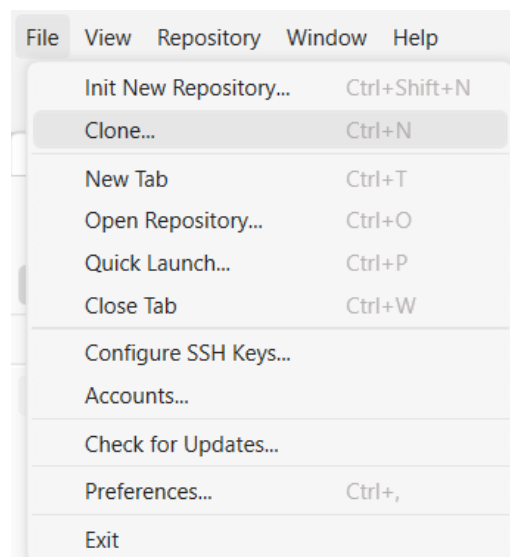
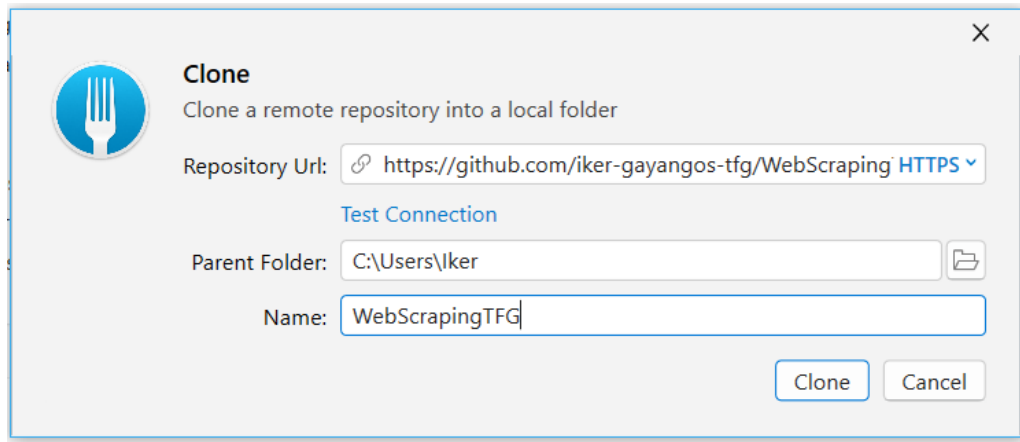


Figura D.1: Copiar dirección para clonar repositorio.

Con la dirección copiada abrimos la aplicación de *Fork*, en la parte de arriba seleccionamos *File* y después *Clone*.

Figura D.2: Seleccionar clonar dentro de *Fork*.

Finalmente, ingresamos la dirección URL copiada, el directorio donde se quiere guardar y el nombre.

Figura D.3: Clonar repositorio en *Fork*.

Siguiendo estos pasos habremos conseguido traer el directorio raíz del proyecto a una carpeta física del terminal.

## Visual Studio 2022

Es la herramienta que utilizaremos para la aplicación *.NET*. Para la instalación accedemos a la página oficial de **Visual Studio**<sup>2</sup> y la descargamos.

Tras el proceso de instalación, abriremos la aplicación y seleccionaremos la opción de abrir un proyecto o una solución.

---

<sup>2</sup><https://visualstudio.microsoft.com/es/vs/community/>



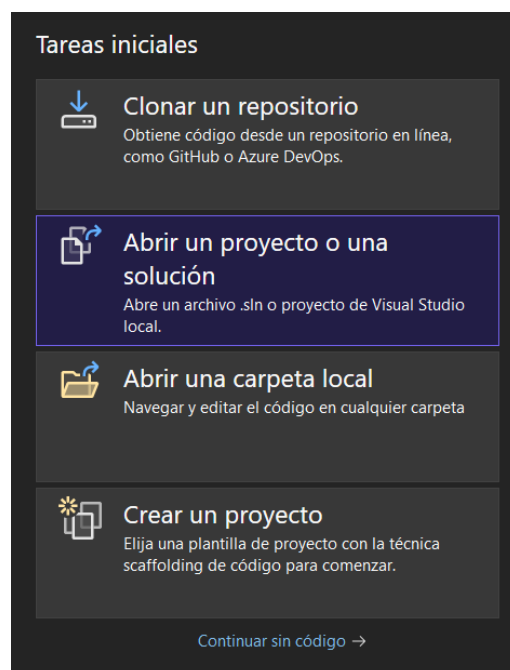


Figura D.4: Abrir solución en Visual Studio.

Se abrirá una ventana para seleccionar el archivo, nos desplazaremos hasta donde hemos clonado el proyecto previamente y en la ruta `WebScrapingTFG/WebScrapingAPI` nos encontraremos un archivo con extensión `.sln`. Lo seleccionamos y tendremos montado el proyecto de `.NET`.

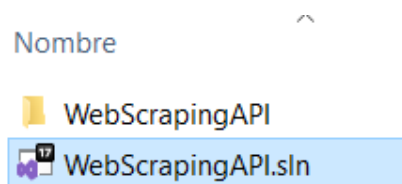


Figura D.5: Seleccionar solución del proyecto.

## Visual Studio Code

Es la herramienta que usaremos para la aplicación *front-end* (*Angular*). Para la instalación accedemos a la página oficial de **Visual Studio Code**<sup>3</sup> y la descargamos.

---

<sup>3</sup><https://code.visualstudio.com/download>

Con la aplicación instalada, vamos a preparar el proyecto de *Angular*. Para ello abrimos la aplicación de Visual Studio Code, y en la parte de arriba seleccionamos *File* y *Open Folder*.

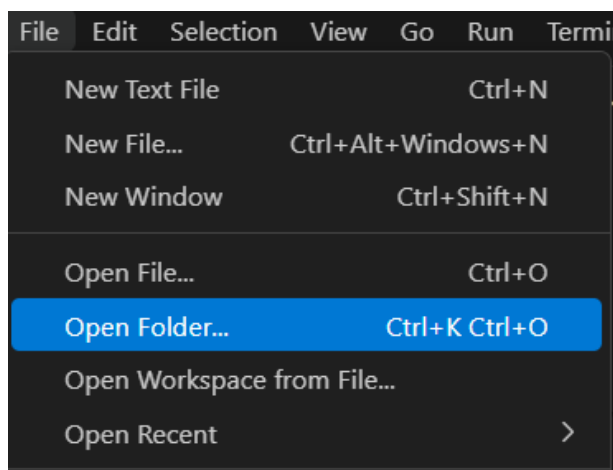


Figura D.6: Abrir directorio de proyecto *Angular*.

Se abrirá una ventana para seleccionar el directorio, nos desplazaremos hasta donde hemos clonado el proyecto previamente y en seleccionaremos el directorio *WebScrapingFront*.

Una vez tengamos el proyecto abierto, vamos a instalar las librerías necesarias. Para ello pulsamos CTRL+J, para abrir un terminal y ejecutaremos el comando: `npm install`. Con ello tendremos el proyecto de *Angular* preparado.

## SQL Server Management Studio 20

Es la herramienta que nos permite acceder a la base datos. Para su instalación accedemos a la página oficial de **SQL Server**<sup>4</sup> y descargamos la aplicación.

Vamos a crear una base de datos para uso local. Para ello arrancamos la aplicación y creamos una base de datos SQLExpress.

---

<sup>4</sup><https://learn.microsoft.com/es-es/sql/ssms>

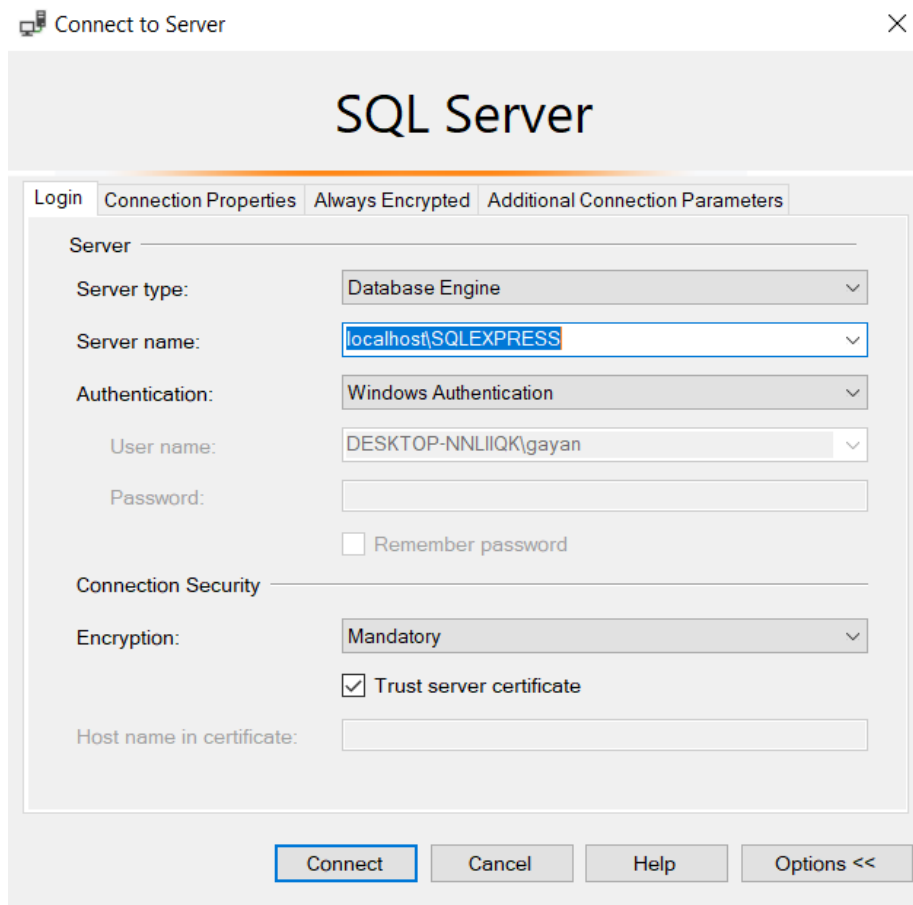


Figura D.7: Crear base de datos SQLEXPRESS.

Una vez en la ventana que se aprecia en la imagen, presionamos sobre conectar y estaría configurada.

Para conectar esta base de datos que hemos creado para el entorno de desarrollo local debemos abrir el proyecto de *.NET* en Visual Studio y acceder al fichero *appsettings.json*. Debemos sustituir el código de la línea 10, en el apartado de *DevelopConnection* por:

```
"DevelopConnection": "Server=(local)\\sqlexpress;  
Database=WebScrapingDev; Trusted_Connection=True;  
MultipleActiveResultSets=True;"
```

Con este cambio conseguimos apuntar la aplicación *.NET* hacia la base de datos que hemos creado.

Para construir la estructura de tablas en esta base de datos lo haremos con una migración. Para ello, con el proyecto *.NET* abierto en Visual Studio, en la parte de arriba seleccionamos Herramientas, Administrador de paquetes NuGet y abrimos la consola.

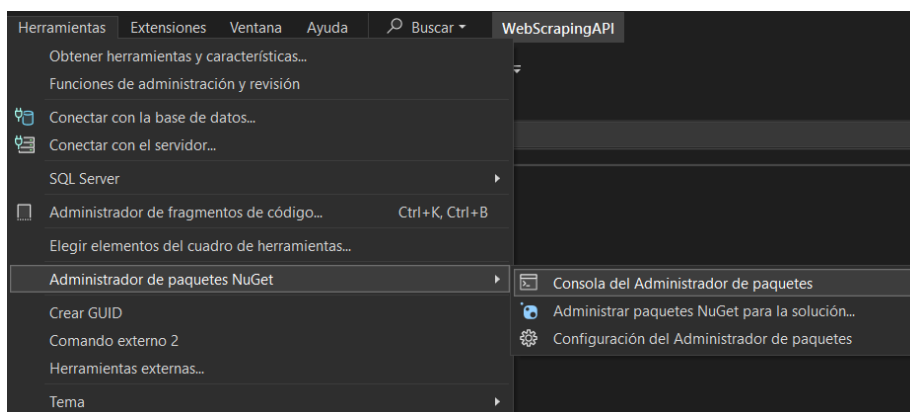


Figura D.8: Abrir consola Visual Code.

En la consola escribiremos:

**add-migration v1.0** Para generar la migración

**update-database** Para aplicar la migración en la base de datos de desarrollo.

## Lanzamiento de la aplicación

Una vez desarrollados los pasos anteriores tendremos preparado el entorno para lanzar la aplicación. Para ello, hay que ejecutar tanto la aplicación *back-end*, como la aplicación *front-end*.

Para ejecutar la aplicación *back-end*, tenemos que abrir la solución en Visual Studio del proyecto de *.NET*. En la parte superior seleccionaremos el módulo *IIS Express* y presionamos en el triángulo verde.

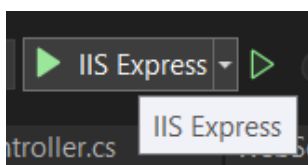


Figura D.9: Lanzar aplicación *.NET*.

Con la aplicación *.NET* lanzada, pasamos a ejecutar el proyecto de *Angular*. Abrimos el proyecto en Visual Studio Code y abrimos el terminal con CTRL+J. Escribimos en el terminal:

```
ng serve -o
```

Tras la ejecución del comando, esperamos a que compile y se abrirá una nueva ventana del navegador predeterminado, en la que estará la aplicación lanzada.

## D.5. Pruebas del sistema

En este apartado se van a explicar las pruebas que se han realizado sobre la aplicación.

### SonarQube

SonarQube es una herramienta de código abierto que permite encontrar problemas de calidad y seguridad en el código [1].

Para la instalación de la herramienta se ha utilizado una imagen *Docker*, con la que se ha creado un contenedor y se ha ejecutado. Con ello, se abre en el puerto 9000 del *localhost*, una interfaz realizar el análisis y posteriormente los resultados.

### Aplicación *back-end*

Primero se ha sometido al análisis al proyecto de *.NET*. Como resultado se ha concluido que esta aplicación ha pasado los test.

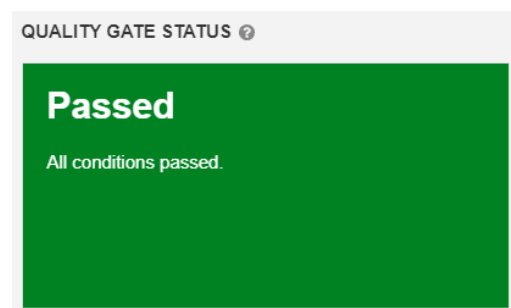


Figura D.10: Test SonarQube de aplicación *.NET* superados.

Incluyo también la tabla con los resultados de SonarQube:

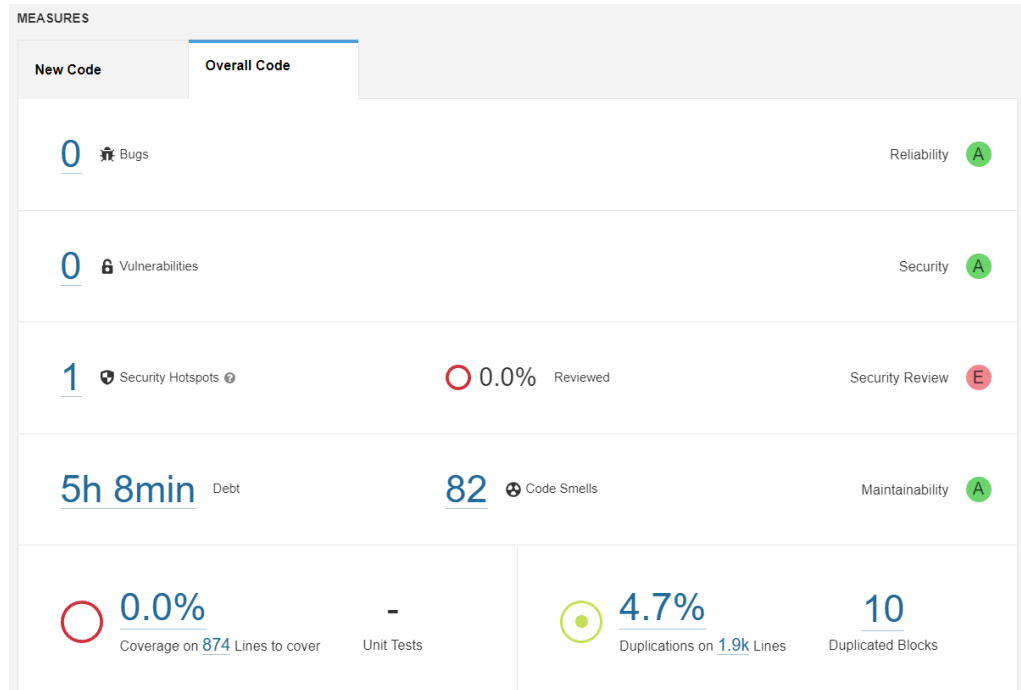


Figura D.11: Resultados SonarQube de aplicación .NET.

Como se puede apreciar en la imagen, el código no presenta *bugs* ni vulnerabilidades. El *security hotspot* que presentan las métricas, es debido a que permito el envío de peticiones a la aplicación desde cualquier origen, lo cual no es una practica recomendable pero en nuestras condiciones era los más apropiado para evitar errores de Cors. El código repetido aparece principalmente en los modelos, pero como distintos modelos tienen campos con el mismo nombre, no se ha optado por rediseñar.

### Aplicación *front-end*

Para continuar pasaremos la aplicación de *Angular* por SonarQube. Como resultado se puede ver que ha pasado los test.

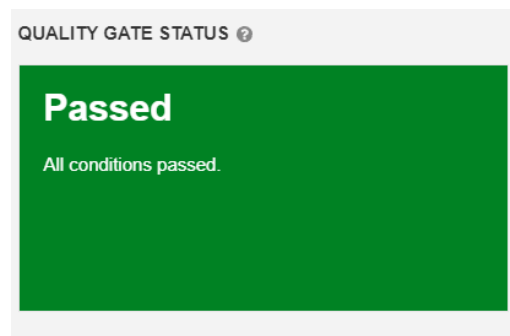


Figura D.12: Test SonarQube de aplicación *Angular* superados.

Resultados del análisis de SonarQube sobre la aplicación *Angular*:

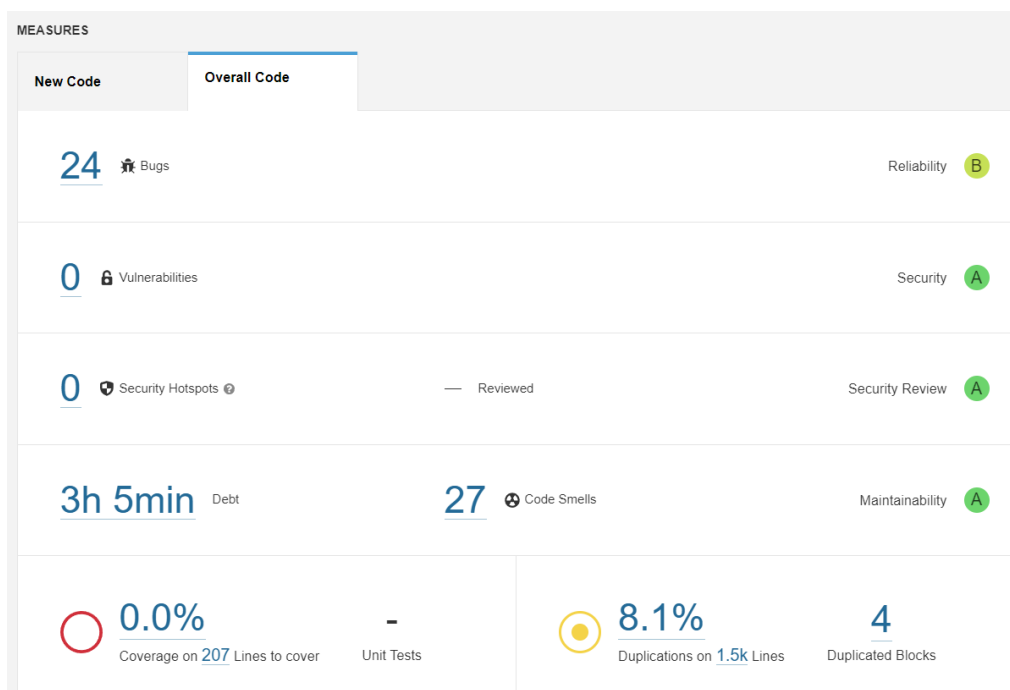


Figura D.13: Resultados SonarQube de aplicación *Angular*.

Como se puede apreciar en la imagen, el código no presenta vulnerabilidades ni *security hotspots*. Los *bugs* que aparecen en el informe son debidos a un error en la estructura de contenedores de un listado. Para finalizar comentar que el código duplicado pertenece en su mayoría a ficheros HTML que contienen las vistas, y que al mostrar información similar, la herramienta lo detecta como código duplicado.



## Apéndice *E*

---

# Documentación de usuario

---

### E.1. Introducción

En este anexo se van explicar los requisitos de los usuarios para poder montar desplegar la aplicación, así como una guía donde se explicarán todas las funcionalidades.

### E.2. Requisitos de usuarios

Los requisitos de los usuarios para poder acceder a la aplicación son los siguientes:

- Tener clonado el **repositorio del proyecto**<sup>1</sup> de GitHub en el entorno local.
- Tener instalada la aplicación *Docker Desktop*
- Tener en ejecutados los contenedores *Docker* mientras se utiliza la aplicación.

---

<sup>1</sup><https://github.com/iker-gayangos-tfg/WebScrapingTFG>

## E.3. Instalación

Para la fase de instalación, primeramente necesitamos tener descargado el repositorio del proyecto que se alberga en *GitHub*. En el apartado de manual de programador, se explica un método para obtener los directorios remotos en el terminal local.

Partiendo que ya tenemos el repositorio descargado, a continuación tenemos que instalar la aplicación *Docker Desktop*. Para ello visitamos la página oficial de **Docker Desktop**<sup>2</sup> y procedemos con su descarga. Una vez instalada la aplicación, la ejecutaremos.

En el directorio raíz del proyecto descargado, encontramos un fichero llamado `docker-compose.yml`. Este fichero se encarga de crear los contenedores *Docker* necesarios para la ejecución de la herramienta desarrollada. Para ejecutar este archivo, debemos abrir un terminal y situarnos en la carpeta raíz del proyecto descargado. Una vez nos encontremos en el directorio del proyecto, escribimos este comando:

```
docker-compose up --build
```

Tras esperar a que se creen y ejecuten los contenedores *Docker*, Encontraremos nuestra aplicación web desplegada en la dirección <http://localhost:4200/>.

## E.4. Manual del usuario

En esta sección se van a explicar las funcionalidades que contiene la web.

### Visualización investigadores

Al entrar al sitio web, nos encontramos con una página en la que se muestra un listado con los investigadores que se encuentran en base de datos.

---

<sup>2</sup><https://www.docker.com/products/docker-desktop/>

WebScrapingTFG							
<div> <div>Investigadores</div> <div>Investigador</div> <div>Investigadores relacionados</div> <div>Lanzar Web Scraping</div> </div>							
	Id	Nombre	Apellidos	IdInvestigador	Email	Departamento	Acciones
<input type="checkbox"/>	1	VICTORIA	ABAD SAN MARTIN	34937	viaspol@ubu.es	INGENIERIA ELECTROMECANICA	>
<input type="checkbox"/>	2	LANDER	ABASOLO PEREZ	131521	labasolo@ubu.es	INGENIERIA INFORMATICA	>
<input type="checkbox"/>	3	CESAR	ACERO MARQUINA	69401	cacero@ubu.es	INGENIERIA ELECTROMECANICA	>
<input type="checkbox"/>	4	FERNANDO	AQUILAR ROMERO	34946	fagular@ubu.es	INGENIERIA ELECTROMECANICA	>
<input type="checkbox"/>	5	VIRGINIA	AHEDO GARCIA	34947	vahedo@ubu.es	INGENIERIA DE ORGANIZACIÓN	>
<input type="checkbox"/>	6	LOURDES	ALAMEDA CUENCA ROMERO	34949	lalameda@ubu.es	CONST. ARQ. E ING. DE LA CONST. Y EL TERR.	>
<input type="checkbox"/>	7	ROBERTO	ALCALDE DELGADO	34952	radelgado@ubu.es	ECONOMIA Y ADMON. DE EMPRESAS	>
<input type="checkbox"/>	8	ROBERTO	ALDAY SERNA	36771	ralday@ubu.es	DIGITALIZACION	>
<input type="checkbox"/>	9	JESUS MANUEL	ALEGRE CALDERON	34954	jalegre@ubu.es	INGENIERIA CIVIL	>
<input type="checkbox"/>	10	ALVARO	ALISTE RABANO	1096525	aalistee@ubu.es	INGENIERIA INFORMATICA	>
<input type="checkbox"/>	11	ROQUE	ALONSO ALONSO	872419	roqueaa@ubu.es	INGENIERIA INFORMATICA	>
<input type="checkbox"/>	12	CARLOS	ALONSO DE ARMIÑO PEREZ	34961	caap@ubu.es	DIGITALIZACION	>
<input type="checkbox"/>	13	JESUS ANGEL	ALONSO GARCIA	36775	jalonso@ubu.es	INGENIERIA CIVIL	>
<input type="checkbox"/>	14	BELEN	ALONSO NUÑEZ	34965	balonso@ubu.es	QUIMICA	>
<input type="checkbox"/>	15	CRISTINA	ALONSO TRISTAN	34968	catristan@ubu.es	INGENIERIA ELECTROMECANICA	>

Figura E.1: Listado investigadores aplicación web.

En el listado se encuentra mostrada la información más referente a los investigadores. En la parte superior podemos apreciar 2 campos de filtrado. El primer campo *input* filtra, sin tener en cuenta los acentos o diéresis, buscando las coincidencias con el nombre y apellidos de los investigadores.

El segundo filtro, se trata de una selector de investigadores. Permite selección múltiple, y su función, es obtener el listado los investigadores con los que el/los seleccionados hayan realizado alguna publicación en conjunto.

El coordinador de doctorado puede obtener un documento *Excel* de la aplicación SIGMA, que muestra un listado de los doctorandos con sus directores de tesis. Con ayuda de estos filtros podemos encontrar el doctorando deseado, seleccionando sus directores de tesis, o si es necesario acotar más la búsqueda, escribiendo el apellido. Con ello se llega al registro del investigador y se puede acceder a su información recopilada.

Destacar en esta parte, la importancia de que el filtro de nombre y apellidos, no tenga en cuenta los acentos ni la diéresis, permitiendo buscar los investigadores escritos de distintas maneras, para facilitar su fusión, solucionando el problema explicado en en punto 5 de la memoria.



Figura E.2: Filtros aplicación web.

Para cada registro de investigador, en caso de que esté dado de alta en el portal, encontramos en la columna de acciones, un botón que permite abrir una pestaña nueva hacia la URL del investigador en el portal. En esta columna también aparece el botón para abrir las publicaciones del investigador.

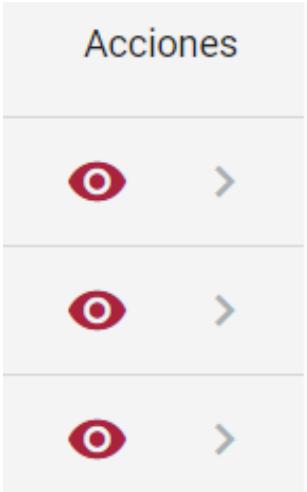


Figura E.3: Columna de acciones del listado de investigadores.

## Lanzar *Web Scraping*

En la parte superior de la página de investigadores, se encuentra un botón para lanzar el proceso de *Web Scraping*. Al presionar este botón se abre una pestaña que muestra la navegación por las páginas.

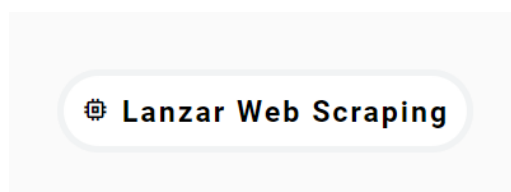


Figura E.4: Botón para lanzar el proceso de *Web Scraping*

## Fusionar investigadores

En el listado de investigadores, en la primera columna hay una casilla que permite la selección de los investigadores requeridos.

<input type="checkbox"/>	28905	Josu Arriolabengoa, Miren
<input checked="" type="checkbox"/>	1406	Josu Barrenetxea Garcia
<input checked="" type="checkbox"/>	1404	Josu Barrenetxea García
<input checked="" type="checkbox"/>	1405	Josu Barrenetxea-García
<input type="checkbox"/>	28523	Josu Narbarte

Figura E.5: Captura selección de investigadores

Al seleccionar más de un investigador, se habilita en la parte superior un botón para abrir una ventana modal,

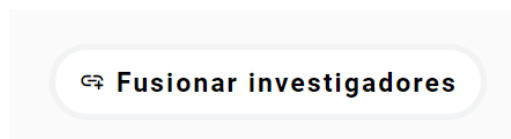


Figura E.6: Botón para abrir ventana modal de fusión

Dentro de la ventana modal, seleccionaremos el investigador al que se va a volcar la información del resto de seleccionados y guardamos.

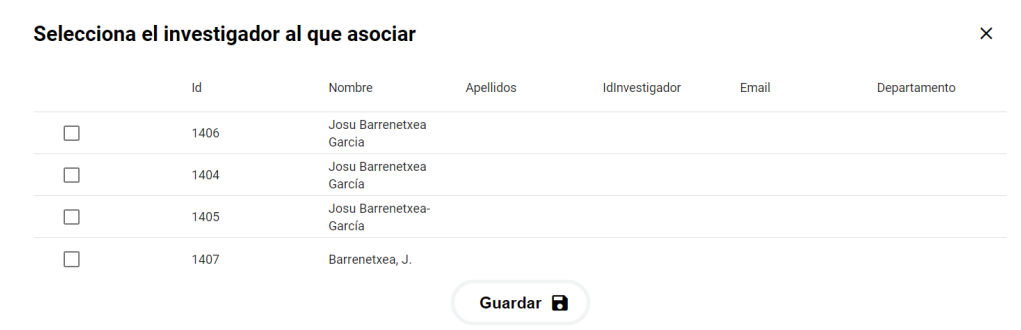


Figura E.7: Captura ventana modal fusión investigadores

Visualización publicaciones

En el listado de investigadores, presionamos en el botón de la flecha sobre el registro del que se quieran ver las publicaciones. Se abrirá una ventana modal que muestra un listado con las publicaciones. En los registros aparecerá la información general de las publicaciones, junto con un botón que redirige a la publicación en el portal de investigación, y un botón para abrir los indicadores de la publicación.

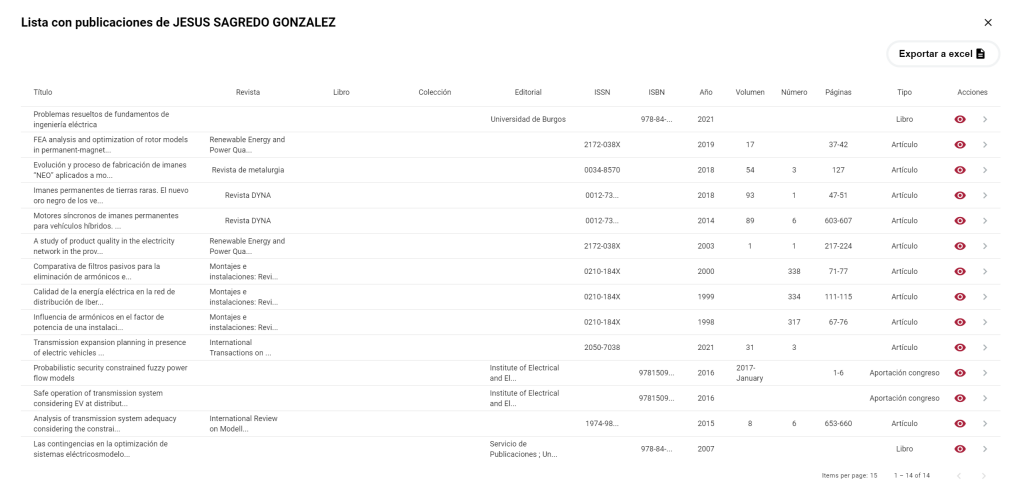


Figura E.8: Captura listado de publicaciones.

Exportación publicaciones

Dentro del listado de publicaciones anterior, hay un botón de exportar, que al presionar, se generará un documento *Excel* con la información de las publicaciones del investigador.

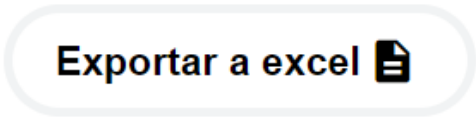


Figura E.9: Botón de exportación de publicaciones.

Visualización indicadores

En el listado de publicaciones, al presionar en el botón de la flecha de la publicación que se precise, se abrirá una nueva ventana modal que mostrará los indicadores asociados a la publicación.

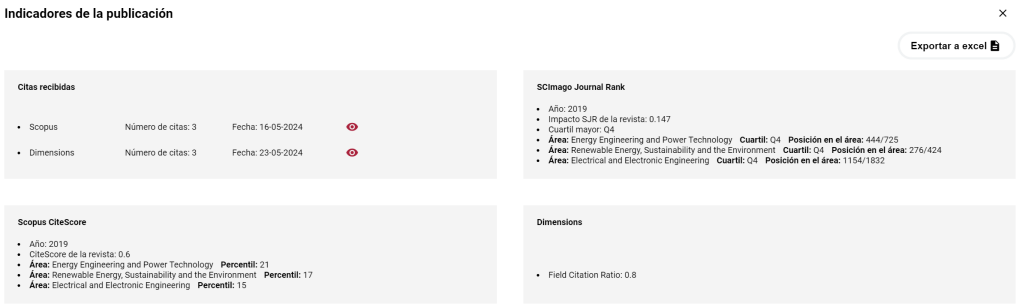


Figura E.10: Captura vista indicadores.

Exportación indicadores

Dentro de la ventana modal de indicadores anterior, hay un botón de exportar, que al presionar, se generará un documento *Excel* con los indicadores asociados a la publicación.





---

## Anexo de sostenibilización curricular

---

La creación de aplicaciones web además de las ventajas que proporciona, como la accesibilidad de datos o la optimización de procesos, también están vinculadas a ll cuidado del medio ambiente.

En la digitalización de procesos, hay que tener en cuenta, además de las características técnicas, el seguimiento de prácticas éticas en el desarrollo que conllevarán una tecnología sostenible. Debido a la situación actual en la que encuentra el planeta, debemos tener en cuenta tanto los impactos económicos, como los sociales y los ambientales. Esto promoverá que una conservación más adecuada de nuestro planeta, a la vez que ofrecemos un potencial transformador.

En el desarrollo de software comprometido con la mantenibilidad, podemos destacar 4 puntos relevantes [3, 2]:

- **Reducción de los desplazamientos físicos:** Las nuevas herramientas software nos permiten la comunicación remota, evitando desplazamientos físicos, pudiendo acceder a la distinta información sin tener que moverse.
- **Uso responsable de recursos naturales:** Mediante la digitalización reducimos mucho el consumo de papel, lo que supone una gran disminución en la tala de árboles y en gasto de agua para la su producción de este (reduciendo los residuos y fomentando el reciclaje).
- **Mejora de la eficiencia energética.**

- **Disminución de las emisiones de efecto invernadero:** Favoreciendo el uso de fuentes de energía renovables (eólica, solar o hidroeléctrica) para suministrar la infraestructura del software, conseguimos reducir la emisión de gases como el carbono, ralentizando el cambio climático.

La integración de la sostenibilidad en el desarrollo de aplicaciones web es esencial de cara al futuro, hay que tener un mayor compromiso con el medio ambiente. De aquí en adelante estoy comprometido en promover la implantación de prácticas sostenibles, priorizando no solo la parte económica, sino también los valores ambientales.

---

## Bibliografía

---

- [1] Imagina formación. ¿qué es sonarqube? herramienta clave para desarrolladores. <https://imaginaformacion.com/tutoriales/que-es-sonarqube>, 2024. [Internet; Visitado por última vez: 5-julio-2024].
- [2] Jose Luis Otero. Digitalización sostenible: hacia un mundo más verde y conectado. <https://cuervaenergia.com/es/comunidad/gestion-empresarial/digitalizacion-sostenible/>, 2023. [Internet; Visitado por última vez: 7-julio-2024].
- [3] SAP Concur Team. Ventajas de la digitalización para el medio ambiente. <https://www.concur.co/blog/article/ventajas-de-la-digitalizacion-para-el-medio-ambiente>, 2024. [Internet; Visitado por última vez: 7-julio-2024].