



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**



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



## Resumen

El presente proyecto trata sobre el desarrollo de una aplicación *back-end* y de una aplicación web, que permite la obtención de toda la información útil de la página web del portal de la investigación de la UBU, para posteriormente, agruparla y realizar una visualización personalizada.

Con ello se pretende ayudar a los directores de tesis a obtener la información sobre sus doctorandos y realizar una gestión de los datos de una forma más rápida y visual. Otro de los objetivos abordados, ha sido proporcionar ayuda al coordinador del programa de doctorado para la realización de informes mediante la información obtenida.

La aplicación *back-end* genera, mediante una migración, una base de datos relacional en la que se guarda el contenido del portal de la investigación.

Para la obtención de la información del portal de la investigación, desde la aplicación *back-end*, se lanza un proceso de *WebScraping* que recorre las páginas de dicho portal, y guarda el contenido recogido en la base de datos creada.

La aplicación web se encarga de actuar como interfaz entre usuario y la aplicación *back-end*, que contendrá la funcionalidad. Tanto del lanzado del *WebScraping* como la visualización y gestión de la información.

Las herramientas de la aplicación web incluyen: generación de tablas, exportación a fichero *.xlsx* y visualización individual de los datos.

## Descriptores

Aplicación *back-end*, aplicación web, *WebScraping*, base de datos, *Portal de la investigación*, visualización, exportación

## Abstract

This project is about the development of a back-end application and a web application, which allows to obtain all the useful information from the UBU research portal website, to later group it and make a personalised visualisation.

This is intended to help the thesis supervisors to obtain information about their doctoral students and manage the data in a faster and more visual way. Another of the objectives has been to help the coordinator of the doctoral programme to produce reports using the information obtained.

The back-end application generates, by a migration, a relational database in which the content of the research portal website is stored.

To obtain the information from the research portal website, a WebScraping process is launched from the back-end application, that runs through the pages of the portal and saves the content collected in the database created.

The web application acts as interface between the user and the back-end application, which will contain the functionality. It handles the launch of WebScraping and the visualisation and management of the information.

The web application tools include: table generation, export to `.xlsx` file and individual visualisation of the data.

## Keywords

Back-end application, web application, *WebScraping*, database, *Research portal website*, visualisation, export

---

# Índice general

---

<b>Índice general</b>	<b>iii</b>
<b>Índice de figuras</b>	<b>v</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos del proyecto</b>	<b>3</b>
2.1. Objetivos funcionales . . . . .	3
2.2. Objetivos técnicos . . . . .	4
2.3. Objetivos personales . . . . .	4
<b>3. Conceptos teóricos</b>	<b>7</b>
3.1. <i>Web Scraping</i> . . . . .	7
3.2. <i>Back-end y Front-end</i> . . . . .	11
<b>4. Técnicas y herramientas</b>	<b>13</b>
4.1. Técnicas . . . . .	13
4.2. Herramientas . . . . .	15
<b>5. Aspectos relevantes del desarrollo del proyecto</b>	<b>21</b>
5.1. Selección del proyecto . . . . .	21
5.2. Inicio de proyecto . . . . .	21
5.3. Búsqueda de implementación . . . . .	23
5.4. Migraciones . . . . .	23
5.5. <i>Web Scraping</i> . . . . .	24
5.6. Aplicación web . . . . .	28
5.7. Problemas en el desarrollo . . . . .	30

<b>6. Trabajos relacionados</b>	<b>33</b>
6.1. Introducción . . . . .	33
6.2. Universidades . . . . .	33
<b>7. Conclusiones y Líneas de trabajo futuras</b>	<b>35</b>
7.1. Conclusiones . . . . .	35
7.2. Líneas de trabajo futuras . . . . .	36
<b>Bibliografía</b>	<b>39</b>

---

## Índice de figuras

---

3.1. Ejemplo de <i>Web Scraping</i> en <i>Google Shopping</i> [8] . . . . .	9
3.2. Ejemplo de comparación de precios en <i>Camelcamelcamel</i> [4] . .	10
5.1. Captura portal de investigación . . . . .	22
5.2. Captura pestaña detalle . . . . .	25
5.3. Captura pestaña publicaciones . . . . .	26
5.4. Captura pestaña tesis . . . . .	27
5.5. Captura pestaña patentes . . . . .	27
5.6. Listado de investigadores . . . . .	28
5.8. Listado de indicadores de publicación . . . . .	29
5.7. Listado de publicaciones del investigador . . . . .	29
5.9. Ventana modal para fusionar investigadores . . . . .	32
6.1. Captura portal de investigación Universidad de La Rioja . . . .	34





---

# 1. Introducción

---

El [portal de investigación de la Universidad de Burgos](https://investigacion.ubu.es/)<sup>1</sup> (UBU) es una plataforma pública en la que se alberga la actividad académica de los profesores de la UBU. En esta plataforma se encuentran las publicaciones científicas derivadas de su trabajo, las tesis doctorales, tanto defendidas como dirigidas, las patentes registradas y las colaboraciones realizadas junto a instituciones nacionales e internacionales.

La plataforma gira en torno a la información referente a los investigadores de Universidad de Burgos. Se encuentran los departamentos, centros de investigación, áreas del conocimiento y programas de doctorado. Dentro de cada una de estas unidades, aparece una breve descripción y se encuentran los investigadores que pertenecen al grupo.

La producción científica de los investigadores de la UBU, en gran medida, es fruto de colaboraciones con otros investigadores. Pero si el investigador que ha colaborado, no está registrado dentro del portal de investigación de la UBU, no se existe una forma de obtener la información referente a este investigador.

Con la realización de este proyecto se pretende unificar la producción científica de los investigadores que no están registrados en el portal de investigación, con el fin de tener las publicaciones y sus indicadores disponibles para su visualización. También facilitar la elaboración de informes al coordinador de doctorado, gracias a esta agrupación de la información del portal.

---

<sup>1</sup><https://investigacion.ubu.es/>



---

## 2. Objetivos del proyecto

---

En este apartado se van a desglosar los objetivos que han motivado la realización de este proyecto. Lo dividiremos en objetivos funcionales, objetivos técnicos y objetivos personales.

### 2.1. Objetivos funcionales

En este apartado se enumeran los objetivos funcionales que se acordaron dar una solución con la creación de la herramienta:

- Creación de una aplicación web para mejorar la gestión de la información del portal de investigación.
- Facilitar la búsqueda de datos referentes a los investigadores del portal, mediante un filtrado por unos campos determinados.
- Ayudar a los profesores a localizar la información referente a los investigadores que no están incluidos dentro del portal de investigación.
- Permitir la unificación de las publicaciones de varios investigadores en uno concreto.
- Mejorar la visualización de los indicadores referentes a las publicaciones de los investigadores.
- Permitir la exportación a *Excel* de los indicadores de todas las publicaciones de cada investigador.
- Realizar un contenedor *Docker* para la publicación, tanto de la aplicación Web como de la aplicación *back-end* y facilitar su implementación en distintos equipos.

## 2.2. Objetivos técnicos

Los objetivos técnicos que se han abordado durante la realización del proyecto son:

- Crear una aplicación web en Angular 17 (última versión antes de la salida de Angular 18 el 22 de mayo de 2024) en la que se usa TypeScript, HTML y SCSS.
- Crear una aplicación *back-end* en .NET 8 (última versión) en la que se usa C#.
- Configurar una base de datos SQL Server conectada con el proyecto .NET.
- Usar la librería de Selenium para obtener los datos mediante *Web Scraping*.
- Utilizar una migración configurada en los modelos del proyecto .NET, para la creación automática de las tablas y relaciones de la base de datos.
- Uso de las librerías asociadas a *EntityFramework* para la gestión de los registros de la base de datos desde el proyecto .NET.
- Utilizar las librerías de *Angular Material* y *FlexLayout* para la maquetación de la aplicación web.
- Uso de la librería *HttpClient* para realizar la conexión entre la aplicación Angular y la aplicación .NET.
- Utilizar *GitHub* para la gestión de versiones y en concreto la aplicación *Fork* para la realización de los *commits*, de ambas aplicaciones.

## 2.3. Objetivos personales

Los objetivos personales que me he marcado personalmente como finalidad del proyecto son:

- Ayudar al coordinador de doctorado en la gestión de los indicadores de los doctorandos, para su posterior elaboración de informes.
- Facilitar la generación de documentación de los doctorandos.

- Aplicar conocimientos obtenidos durante el grado de informática y mi estancia en una empresa de software, a un proyecto propio.
- Cerrar mi ciclo formativo con la creación de una herramienta que sea útil y aporte funcionalidad a un propósito.
- Aprender el uso de tecnologías no trabajadas durante el grado de informática y poder emplear los conocimientos adquiridos en proyectos futuros. En concreto: *.NET*, *Angular*, *Docker* y  $\text{\LaTeX}$



---

## 3. Conceptos teóricos

---

En este apartado se van a tratar los conceptos teóricos menos conocidos habitualmente y que son necesarios para comprender alguna de las partes que se han abordado durante la realización del proyecto.

### 3.1. *Web Scraping*

El *Web Scraping* es una técnica por la que se puede extraer la información estructurada de un sitio web, para después gestionarla. Para ello, se realiza una simulación de la navegación hacia las página/s web como si lo realizara un humano, y mediante algoritmos y herramientas se automatiza la obtención de la información requerida [5].

#### Técnicas

Para la realización del *Web Scraping* el primer proceso a realizar es navegar al sitio web y obtener el HTML. Posteriormente para la extracción de los datos a partir de este HTML, presentamos distintas técnicas [16]:

- Uso de expresiones regulares que analizan el texto, y conforman patrones de búsqueda que se basan principalmente, en la búsqueda de cadenas de caracteres concretos para así dar con la información que se requiere.
- Simulando el lanzamiento de peticiones HTTP al API dedicado durante el proceso de *Web Scraping*, y así obtener los registros de las llamadas obtenidas del sitio web.

- Uso de *parsers* de HTML que convierten el propio HTML en un árbol de modelos de objetos del documento (DOM), para después obtener los datos deseados a partir de este DOM.
- Uso de algoritmos también usados en minería de datos. Esta técnica se emplea en páginas web generadas dinámicamente con información proveniente de una base de datos, en la que un registro de un tipo y un nivel se encontrara en otra página similar en el mismo nivel. Mediante estos algoritmos se pueden definir unas plantillas donde se encontrara la información requerida.
- Aplicaciones externas de *Web Scraping* ya creadas, que utilizan algoritmos integrados, inteligencia artificial y procesos robóticos que tienen una interfaz propia para configurar la obtención de la información de una forma sencilla, aunque esta solución es de pago.

Para el caso concreto de este proyecto se han utilizado algunas las técnicas anteriores mezcladas. El flujo del procedimiento utilizado es el uso de algoritmos que navegan por las páginas del portal de investigación UBU y descargan el HTML. Posteriormente se han programado una serie de plantillas de cómo se encontraba la información distribuida por la web, y mediante el uso de herramientas de obtención de contenedores, conseguimos los datos concretos que requerimos. Para esta labor, se ha utilizado *Selenium*, que permite obtener el contenedor HTML que se requiere accediendo mediante su clase CSS, y una vez se tenga el contenedor, se extrae la información necesaria.

## Usabilidad

El uso del *Web Scraping* es más familiar de lo que podemos pensar y se encuentra en herramientas web que alguna vez hemos utilizado. Se utiliza habitualmente en el comercio electrónico (*e-commerce*), para la comparación de precios entre distintas páginas. a la hora de elegir el vuelo más barato entre las distintas compañías. En las páginas dedicadas a la comparación de precios, el *Web Scraping* navega sobre las webs y recoge todas las alternativas para después mostrarlas y poder elegir la más conveniente.

Existen páginas web que se dedican a vender sus servicios de *Web Scraping*, para obtener información de la competencia en tiempo real. Algunos ejemplos son [Datstrats](https://datstrats.com/)<sup>2</sup> o [Alphascape](https://alphascrape.com/es/)<sup>3</sup>

---

<sup>2</sup><https://datstrats.com/>

<sup>3</sup><https://alphascrape.com/es/>



## Ejemplos

Alguno de los ejemplos de uso del *Web Scraping* en el comercio electrónico son [Google Shopping](https://shopping.google.es/)<sup>4</sup> o [Camelcamelcamel](https://es.camelcamelcamel.com/)<sup>5</sup>

### Google Shopping

Es una plataforma ofrecida por Google que sirve para la comparación de precios de productos dentro de internet. En ella se encuentra un buscador para filtrar el producto que se desea, la herramienta obtiene los precios de los distintos vendedores disponibles y facilita la conexión con el propio vendedor.

Realizando un ejemplo en la web, buscaremos el nuevo *iPhone*. Una vez seleccionado, nos devolverá un listado con los vendedores que cumplen las características seleccionadas.



The image shows a screenshot of Google Shopping search results for an iPhone. The results are presented in a list format with three items. Each item displays the price in Euros, shipping information, the seller's name, and a button to visit the seller's website. At the bottom, there is a link to compare prices from 50 or more stores.

Opciones de compra	
<b>1339,00 €</b> Envío gratuito MediaMarkt	Visitar sitio
<b>1469,00 €</b> Envío gratuito Apple	Visitar sitio
<b>1289,00 €</b> Envío gratuito K-tuin, tiendas Apple	Visitar sitio

[Comparar precios de 50 o más tiendas](#)

Figura 3.1: Ejemplo de *Web Scraping* en *Google Shopping* [8]

<sup>4</sup><https://shopping.google.es/>

<sup>5</sup><https://es.camelcamelcamel.com/>

### Camelcamelcamel

Se trata de un rastreador de precios de la página web de *Amazon*. Su funcionamiento es proporcionar la URL de un producto de *Amazon*, y la web te devuelve el precio actual, lo compara con los registros que tiene en su base de datos sobre ese producto, y genera un histórico del precio.

Con ello se puede comparar si el precio que tiene actualmente es más alto o más bajo que lo que ha estado en meses anteriores, y con ello decidir si es un buen momento para la compra. También consta con una herramienta en la que se puede seleccionar un rango de precio para ese producto, y cuando el precio se encuentre entre ese rango, te notifica por correo.

Cogiendo el mismo producto que con la herramienta anterior vamos a buscar el histórico que tiene *Amazon* del nuevo *iPhone*.

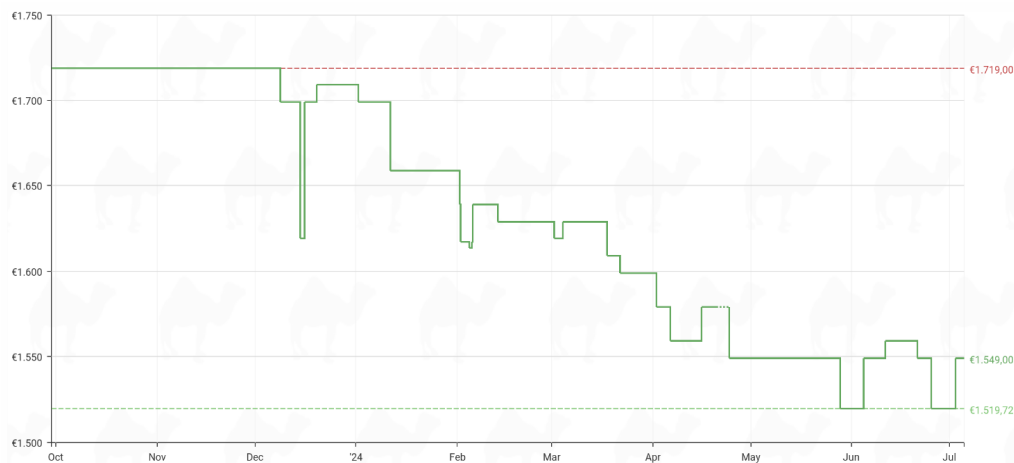


Figura 3.2: Ejemplo de comparación de precios en *Camelcamelcamel* [4]

Se puede apreciar que el precio actual se encuentra cerca del mínimo histórico, por lo que sería un buen momento para realizar la compra.

## 3.2. *Back-end y Front-end*

### *Back-end*

El *back-end* dentro de un proyecto se puede denominar como el servidor, y es la parte de la aplicación en donde se encuentra la funcionalidad que no es visible desde el lado del usuario. En el *back-end* se configura la lógica de la aplicación, la gestión del almacenamiento de datos y los requisitos de seguridad.

Dentro del *back-end* podemos distinguir los siguiente componentes principales [7]:

- **Servidores:** Se corresponde con una máquina física o virtual que aloja, procesa y gestiona los recursos necesarios para la ejecución de una página web. Los servidores reciben la una petición desde la aplicación web, aplican la lógica necesaria y devuelven la respuesta hacia la aplicación web, normalmente mediante protocolo HTTP.
- **Lógica:** Se trata de la secuencia de operaciones que desarrollan los programadores para dar respuesta a una petición. Entre ellas se encuentran la aplicación de funciones y algoritmos para su resolución.
- *Frameworks:* Son las metodologías que se usan para estructurar el código, la lógica y otros aspectos relacionados con la arquitectura web, siguiendo unos estándares.
- **Bases de datos:** Contienen la información que utilizaran los servidores, para a través de la lógica, dar una respuesta a la aplicación web.
- **APIs:** Permiten al *back-end* establecer la conexión con otros servidores y poder acceder a distintas bases de datos dentro de un mismo proyecto.

### *Front-end*

El *front-end* dentro de un proyecto se trata de todo el desarrollo web que es visible para el usuario.

Dentro del *front-end* podemos distinguir los siguientes componentes principales [6]:

- **Estructura HTML:** HTML (*HyperText Markup Lenguaje*) es el lenguaje utilizado para la constitución de la estructura de marcas o

etiquetas que sirven para definir los contenedores que se mostraran en la página web, así como tablas, párrafos títulos...

- **Estilo y diseño:** La función de este componentes es definir el cómo de se debe mostrar el contenido HTML. Se hace uso del lenguaje CSS (*Cascading Style Sheets*) para determinar colores, tipografías, espacios, alineaciones...
- **Interacción dinámica:** En este componente se aplicara la lógica necesaria para realizar las acciones de la vista. Así como la conexión con el *back-end* y otras herramientas. En su uso más general se utiliza el lenguaje de *Javascript* para realizar esta tarea.

## Utilización *back-end* y *front-end*

En una aplicación, separar la parte de funcionalidad (*back-end*) de la parte visual del usuario (*front-end*), aporta una serie de ventajas [11]:

- **Mantenibilidad:** Al tener separadas las dos partes, resulta más sencillo mantener y actualizar el proyecto. Si se quiere actualizar funcionalidad se realizará sobre el *back-end* y si se quiere modificar lo relativo a la parte visual se realizará sobre el *front-end*.
- **Reutilización de código:** Se puede reutilizar la funcionalidad del *back-end* sobre distintas interfaces de la aplicación web.
- **Escalabilidad:** Es más sencilla la gestión de los servidores necesarios para la gestionar el rendimiento de la aplicación, pudiendo destinar los recursos de una manera más eficiente hacia la parte de la aplicación que lo necesite.
- **Colaboración:** En un grupo de trabajo con varios integrantes resulta más sencillo el reparto de tareas. También se promueve la especialización del grupo de trabajo en una de las áreas, para un desarrollo más ágil y eficiente.

---

## 4. Técnicas y herramientas

---

### 4.1. Técnicas

#### *Scrum*

Las metodologías ágiles son aquellas que permiten adaptar los objetivos y la estrategia de trabajo según las condiciones del proyecto. Con ello, se consigue flexibilidad e inmediatez sobre los próximos objetivos que abordar y conseguir así un desarrollo más eficiente.

Los requisitos que necesitaba este proyecto crearon un caldo de cultivo ideal para el uso de metodologías ágiles. El proyecto comenzó de una forma un poco ambigua y donde se tenía claro el objetivo final, pero el camino a seguir para conseguir esta meta, no estaba del todo definido. Por lo que se optó por el uso de estas metodologías en donde se iban enfrentando los objetivos a corto plazo y definiendo continuamente los siguientes.

Algunas de las metodologías ágiles más conocidas son: *Kanban*, *Extreme programming*, *Agile inception* y *Crystal*. Dentro de este espectro de metodologías ágiles, seleccionamos el uso de *Scrum* por delante de las demás.

El método de planificación *Scrum* se basa en el desarrollo del producto dividido en *sprints*. Estos *sprints* son periodos de tiempo, generalmente menores a 4 semanas, en los que un grupo de trabajo, mediante la reunión con cliente acuerdan llegar a cierta funcionalidad en un tiempo concreto. Dentro del *sprint*, en una aplicación ideal de la metodología *Scrum* a un proyecto real, distinguimos 4 tipos de reuniones principales que se llevan a cabo [12]:

1. ***Sprint Planning***: Esta reunión se realiza al inicio y sirve para seleccionar que objetivos y que funcionalidades se proponen para su realización durante el *sprint*. Habitualmente las tareas están separadas y estimadas en tiempo por el equipo de *develop*, entonces lo que hay que hacer es repartir el trabajo entre el grupo. Esta reunión tiene una segunda parte en la que se comenta con una idea general el **cómo** se van a realizar cada tarea, para intentar evitar atascos durante el desarrollo.
2. ***Daily Scrum***: Más conocida como *Daily* es una reunión que se celebra todos los días por el equipo de desarrollo y que tiene una duración estimada de 15 minutos. En ella cada miembro del equipo deberá ser capaz de contestar a estas 3 preguntas:
  - ¿Qué hice ayer?
  - ¿Qué voy a hacer hoy?
  - ¿Tengo algún problema o atasco que me impedirá llevarlo a cabo?
3. ***Sprint Review***: Esta reunión se celebra al final del *sprint* entre el equipo del proyecto, y clientes y externos. Tiene como objetivo que el equipo de desarrollo muestre los avances que se han realizado durante el *sprint* al resto de asistentes, esto puede incluir alguna demo de funcionalidad. Con los comentarios recibidos sobre el producto realizado, se idean nuevas características y se idea la prioridad de objetivos para el siguiente *sprint*.
4. ***Sprint Retrospective***: Este encuentro se realiza después del *Sprint Review* y participan solamente los miembros del equipo interno. El propósito es llegar a conclusiones de lo que ha salido mal, lo que se ha hecho bien, si se podía haber hecho alguna cosa de manera distinta... Con ello se pretende escuchar lo que ha ocurrido durante el *sprint* e intentando solventar y evitar los problemas, mejorar la metodología del equipo.

En el proyecto se ha aplicado la metodología *Scrum* pero adaptada a las necesidades y los medios habilitados. En cuanto a roles, podemos distinguir que el equipo de desarrollo está formado únicamente por mí, y los gerentes de producto y rol de cliente ha sido ejercido por mis tutores. Los *sprints* han tenido una duración inicial de dos semanas y terminaron durando una semana con el fin de aumentar la frecuencia de reunión. En cuanto al desglose de reuniones anterior, se ha realizado únicamente una reunión entre *sprints*,

en la que al comenzar se mostraban los avances de la reunión anterior, y con los comentarios e ideas de los tutores, se planificaba el siguiente *sprint* en funcionalidad y con la prioridad de requerimientos necesaria.

## 4.2. Herramientas

### Gestión de repositorio

#### GitHub

Es una plataforma que permite a los programadores crear, guardar, gestionar y compartir su código debido a su almacenamiento en la nube. Github utiliza el sistema de gestión de versiones *Git*, con el que podremos guardar el progreso alcanzado en cada rama y tenerlas bien organizadas [3, 14].

Se ha optado por el uso de GitHub debido a que su uso es gratuito y ya estaba familiarizado con la herramienta, dado que lo hemos empleado en alguna asignatura durante el grado de informática. Comparto el enlace al repositorio GitHub<sup>6</sup>.

- Herramientas consideradas: GitHub<sup>7</sup>, GitLab<sup>8</sup>

#### Fork

Es una aplicación que sirve para el manejo de los *commits* de un repositorio. Dispone de herramientas como el comparador de código entre el local y el repositorio, posibilidad de *commits* parciales dentro de un archivo y un excelente manejo entre ramas del proyecto. Cuenta con una interfaz sencilla y visual que me hizo optar por esta opción.

Aunque GitHub tiene su propia aplicación de escritorio, he optado por el uso de Fork ya que lo utilizo diariamente en el trabajo.

- Herramientas consideradas: Fork<sup>9</sup>, GitHub Desktop<sup>10</sup>

---

<sup>6</sup><https://github.com/iker-gayangos-tfg/WebScrapingTFG>

<sup>7</sup><https://github.com/>

<sup>8</sup><https://about.gitlab.com/>

<sup>9</sup><https://git-fork.com/>

<sup>10</sup><https://github.com/apps/desktop>

## Despliegue de proyecto

### Docker

Es una plataforma que permite crear, probar e implementar aplicaciones completas de una forma rápida y sencilla. Docker se encarga de empaquetar el software en contenedores que incluyen lo necesario para que el software se ejecute correctamente. Gracias a esta implementación, se asegura el correcto funcionamiento de un desarrollo software en distintos dispositivos [1, 13].

Se ha optado por el uso de Docker debido a las recomendaciones de los tutores para realizar el despliegue.

- Herramientas consideradas: [Docker](https://www.docker.com/)<sup>11</sup>, [Containerd](https://containerd.io/)<sup>12</sup>

## Editores de código

### Visual Studio 2022

Es el entorno de desarrollo integrado (IDE) de Microsoft. IDE de Visual Studio es una aplicación que facilita a los programadores el desarrollo de software de una manera más eficiente. Se ocupa de realizar tareas como editar, depurar o compilar código con una interfaz clara y fácil de utilizar. Es el mejor IDE para programar en .NET y c++ en Windows [9].

Se ha elegido Visual Studio debido a las facilidades que ofrece en la creación y desarrollo de los proyectos de .NET.

- Herramientas consideradas: [Visual Studio 2022](https://visualstudio.microsoft.com/es/vs/)<sup>13</sup>, [Eclipse](https://eclipseide.org/)<sup>14</sup>

### Visual Studio Code

Es un editor de código fuente que a pesar de ser ligero en espacio, es muy potente en cuanto a posibilidades. Es el editor de código fuente desarrollado por Microsoft y es compatible con Windows, Linux y MacOS. Se trata del editor más utilizado en el desarrollo de aplicaciones web, debido al gran número de extensiones que permiten admitir prácticamente todos los lenguajes de programación [9].

---

<sup>11</sup><https://www.docker.com/>

<sup>12</sup><https://containerd.io/>

<sup>13</sup><https://visualstudio.microsoft.com/es/vs/>

<sup>14</sup><https://eclipseide.org/>



Se ha optado por esta herramienta debido a que la utilizo diariamente en el trabajo y domino su uso.

- Herramientas consideradas: [Visual Studio Code](#)<sup>15</sup>, [Sublime Text](#)<sup>16</sup>, [Atom](#)<sup>17</sup>

## Bases de datos

### SQL Server Management Studio 20

Es una aplicación de software desarrollada por *Microsoft* usada para la configuración y administración de bases de datos.

Se optó por esta opción debido a su compatibilidad con *.NET* y sencilla implementación con este *framework* [15].

- Herramientas consideradas: [SQL Server Management Studio](#)<sup>18</sup>, [HeidiSQL](#)<sup>19</sup>,

## Documentación

### L<sup>A</sup>T<sub>E</sub>X

Es una sistema de realización de documentos de alta calidad, cuyo uso más frecuente es en el ámbito científico para la realización de las publicaciones.

Se optó por esta opción debido a la limpieza y profesionalidad que le aporta a la memoria, y mediante la plantilla proporcionada por la UBU, la implementación no es complicada.

- Herramientas consideradas: [L<sup>A</sup>T<sub>E</sub>X](#)<sup>20</sup>, [Microsoft Office Word](#)<sup>21</sup>, [Open Office](#)<sup>22</sup>

---

<sup>15</sup><https://code.visualstudio.com/>

<sup>16</sup><https://www.sublimetext.com/>

<sup>17</sup><https://atom-editor.cc/>

<sup>18</sup><https://learn.microsoft.com/es-es/sql/ssms>

<sup>19</sup><https://www.heidisql.com/>

<sup>20</sup><https://www.latex-project.org/>

<sup>21</sup><https://www.office.com/>

<sup>22</sup><https://www.openoffice.org/es/>

## Editores de texto

### Overleaf

Es un editor de L<sup>A</sup>T<sub>E</sub>X online sencillo de utilizar, que permite compilar los cambios en tiempo real y visualizar el documento compilado.

Se eligió esta herramienta debido a que tiene opciones de edición y revisión en línea, y a la recomendación de uso por compañeros anteriores.

- Herramientas consideradas: [Overleaf](#)<sup>23</sup>, [Visual Studio Code](#)<sup>24</sup>

## Back-end

### .NET

Es una plataforma de código abierto que cuenta con una gran cantidad de librerías con alta escalabilidad y funcionalidad. Utiliza el lenguaje de programación C#, que está fuertemente tipado y sigue los estándares del sector [10].

Opté por esta opción debido al conocimiento previo que tenía sobre .NET, la posibilidad de probar las funciones en local gracias a *Swagger*, y el amplio catálogo de librerías que me permitían la realización del *Web Scraping* y la conexión con base de datos y la aplicación web.

- Herramientas consideradas: [.NET](#)<sup>25</sup>, [Python](#)<sup>26</sup>

## Librerías utilizadas

- **Entity Framework**<sup>27</sup>: Es un asignador de objetos que crea una capa de acceso a datos mediante .NET. Tiene compatibilidad con múltiples bases de datos como: *SQL database*, *SQLite*, *PostgreSQL*, *MySQL* o *Azure*. Se ha utilizado en el proyecto para la configuración de la migración y la gestión de los registros de la base de datos.
- **Selenium**<sup>28</sup>: Se trata de la librería que permite gestionar la automatización de las pruebas automáticas en el desarrollo web. En el proyecto

---

<sup>23</sup><https://es.overleaf.com/>

<sup>24</sup><https://code.visualstudio.com/>

<sup>25</sup><https://dotnet.microsoft.com/es-es/>

<sup>26</sup><https://es.python.org/>

<sup>27</sup><https://learn.microsoft.com/es-es/ef/>

<sup>28</sup><https://www.selenium.dev/>

se ha utilizado para simular la interacción humana en el sitio web y obtener toda la información necesaria (*Web Scraping*).

## ***Front-end***

### **Angular**

Es un *framework* de creación de aplicaciones del lado del cliente utilizando HTML y TypeScript [2]. Se basa en un módulo general que contiene a su vez módulos mas pequeños con una funcionalidad concreta. Dentro de cada modulo nos vamos a encontrar 4 archivos fundamentales:

- Archivo con extensión **.html** que funciona como vista.
- Archivo con extensión **.ts** que utiliza lenguaje TypeScript, que es una ampliación de JavaScript y donde se albergara la lógica.
- Archivo con extensión **.css** o **.scss** donde se incluirán los estilos de la vista.
- Archivo de tipo *module* que incluirá la lógica de rutas y la importación de librerías generales al proyecto.

Se ha optado por elegir este *framework* para la realización del proyecto debido a que se trata de una de las plataformas más utilizadas recientemente dentro del desarrollo *front-end* actual, y a que he trabajado anteriormente con este *framework*.

- Herramientas consideradas: **Angular**<sup>29</sup>, **React**<sup>30</sup>

### **Librerías utilizadas**

- **Angular Material**<sup>31</sup>: Es la librería oficial de componentes de Angular, ofrece una gran cantidad de componentes prediseñados (ej: tablas, botones, inputs...), tanto en funcionalidad como en su interfaz. En el proyecto los componentes insertados en las vistas han girado en torno a los ofrecidos por esta librería para una visualización mejorada, sin incluir costes de tiempo adicionales.

---

<sup>29</sup><https://angular.dev/>

<sup>30</sup><https://es.react.dev/>

<sup>31</sup><https://material.angular.io/>

- **Flex layout**<sup>32</sup>: Se trata de una herramienta para la maquetación de aplicaciones web. Aplicando unas propiedades propias de la librería se consigue gestionar el tamaño, alineación, tanto interna como externa, y la disposición en el plano.

---

<sup>32</sup><https://www.npmjs.com/package/@angular/flex-layout>

---

## 5. Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger las fases por las que ha pasado el proyecto, los problemas que han surgido y la toma de decisiones que se ha afrontado a lo largo del desarrollo para sacar adelante el proyecto.

### 5.1. Selección del proyecto

Dentro de los proyectos disponibles a realizar, había uno que trataba de la realización de una página web, y dado que había estado realizando las prácticas curriculares, posteriormente extracurriculares para finalmente quedarme a trabajar en una empresa de desarrollo software, vi la posibilidad perfecta de aplicar lo aprendido en la universidad y en mi estancia en dicha empresa, a un proyecto personal. Por ello me decanté por elegir este proyecto por delante de los demás.

### 5.2. Inicio de proyecto

Inicialmente, este proyecto nació enfocado en aumentar funcionalidades sobre la aplicación de SIGMA. Lo que se pretendía era ampliar la información que ofrecía dicha herramienta, añadir gráficos, tablas o filtros de búsqueda para que el uso de esta herramienta por el personal de UBU sería más rápido, directo y visual.

La idea inicial era que mediante la base de datos que alimenta la aplicación de SIGMA, realizaría estas mejoras mediante la aplicación web, pero debido a problemas con la nueva ley de protección de datos, no fue posible este

procedimiento. Ocurrió debido a la imposibilidad de proporcionar a un tercero una base de datos con información real de los miembros del ámbito de la universidad.

Para afrontar este problema, mis tutores presentaron una alternativa. Dado que la información que está visible en el portal de investigación es publica para todos los usuarios, plantearon la posibilidad de realizar un *Web Scraping* de todo el portal de investigación, y así poder obtener, las partes necesarias para montar la base de datos que después se utilizaría en la aplicación web.

Durante este cambio proyecto se comentó que en el portal de investigación (ver Figura 5.1) no se puede acceder en conjunto a la producción científica que realizan los doctorandos que no están dados de alta en el portal. Por los que no era posible ver su información relativa a publicaciones, e indicadores de las mismas. Surgía así un nuevo rumbo, cuyo objetivo final sería la posibilidad, mediante un correcto filtrado, de poder acceder a toda la información relativa a un investigador, este dado de alta en el portal o no.



Figura 5.1: Captura portal de investigación

### 5.3. Búsqueda de implementación

Con el nuevo objetivo definido, me puse a investigar sobre cuál era la mejor forma de llevar a cabo el *Web Scraping*. Tras el proceso de búsqueda concluí que las mejores librerías para realizar este proceso se encontraban en *Python* y *.NET*. Aunque durante mi estancia en la universidad ya habíamos trabajado con *Python*, la implementación para convertirla en una aplicación *back-end* tenía una curvatura de aprendizaje alta. Mientras que *.NET*, lo había utilizado minoritariamente en el trabajo como aplicación pura de *back-end*, por lo que ya tenía una idea general de su funcionamiento. También conocía la facilidad para la gestión de la base de datos, por lo que me decante por esta plataforma.

Tras la elección de la plataforma a usar para el *back-end*, investigué sobre las nuevas tecnologías para la realización del *front-end*. Enseguida aparecieron dos claras candidatas: *Angular* y *React*. Entre las alternativas me decanté por el uso de *Angular* para la realización de la aplicación web, dado que era el *framework* de programación que más dominaba y donde mejores resultados iba a poder aportar a la aplicación final.

Después de la elecciones anteriores, busqué información acerca de la base de datos. Para los proyectos de *.NET*, lo más recomendable era el uso de la tecnología *SQL Server*, dado la compatibilidad y su implementación con el proyecto *back-end*.

Debido a que era necesario realizar un despliegue del proyecto para su uso por los profesores de la universidad, seguí las recomendaciones de los tutores sobre la utilización de *Docker* para suplir esta necesidad. Mediante la utilización de contenedores *Docker* se permitía el correcto funcionamiento de la aplicación *back-end* conectada a una base de datos, y posteriormente el despliegue de la aplicación web. Por lo que tras realizar unas pruebas con unos proyectos de prueba pequeños, constaté que esta opción era la mejor para el despliegue.

### 5.4. Migraciones

Con las implementaciones anteriores elegidas, investigué la mejor opción para poder montar la estructura de base de datos, junto con las relaciones entre las tablas. Necesitaba que la base de datos fuera escalable para ir montándola según la programación fuese avanzando. Para no tener que hacer este proceso manualmente en la base de datos y con el fin evitar fallos entre la aplicación *.NET* y la base de datos, opté por ayudarme de las migraciones.

Así, con una configuración correcta de los modelos en el proyecto de *.NET*, podría generar toda la base de datos sin miedo a fallos. Y cuando se amplíen modelos y relaciones, durante la realización del proyecto, mediante otra migración se actualizarán las tablas en la base de datos.

## 5.5. *Web Scraping*

Comenzamos el proyecto de *.NET* en Visual Studio 2022, en mi caso lo tenía instalado en el terminal, sino habría que haber instalado este editor de texto. Creé la aplicación *.NET*, en su versión más reciente (*.NET 8*), con una plantilla ofrecida por el editor de código, y dentro del proyecto, creé un controlador donde comenzar a trabajar.

Para comenzar el proceso de *Web Scraping* en la aplicación de *.NET*. Existían varias librerías de *Web Scraping*, pero la más potente y más usada es *Selenium*. Se trata de una herramienta que nos permite visitar las URL elegidas, como si de una interacción humana se tratara, de forma automática. La funcionalidad es amplia, permitiendo incluso hacer click en botones o en redirecciones, que posteriormente serán necesaria.

Durante las primeras reuniones se definió qué partes se iban a guardar del portal de investigación inicialmente, y fue con eso con los que se empezó. Nos dimos cuenta que la estructura del portal de investigación gira en torno a los investigadores, y que a partir de ellos se conseguiría sacar toda la información que pretendíamos. Por lo que el primer paso fue obtener un listado de todos los investigadores del portal. El portal de investigación está estructurado de forma que todos los investigadores están asociados a un centro académico. Por lo que para obtener un listado completo de investigadores hay que entrar en todos los centros académicos y guardar los investigadores pertenecientes a dicha unidad. Con *Selenium* se obtiene el id único que identifica al investigador en el portal y se guarda en una lista.

La información referente a los investigadores está bien distinguida en las siguiente pestañas:

- **Detalle** (ver Figura 5.2): En esta pestaña aparecen nombre y apellidos, email y las unidades a las que pertenece:
  - Departamento
  - Centro académico
  - Unidades de investigación



- Programas de doctorado
  - Áreas
  - Grupos de investigación
- **Publicaciones** (ver Figura 5.3): Aquí aparece un listado con todas las publicaciones en las que ha participado el investigador, y dentro de cada una de ellas, la información referente a la publicación y los indicadores, que indican la importancia que ha tenido la publicación.
  - **Tesis** (ver Figura 5.4): En esta pestaña aparece la tesis del investigador, en caso de que la haya realizado, y un listado con las tesis que ha dirigido y otro listado con las tesis en las que ha sido miembro del tribunal.
  - **Patentes** (ver Figura 5.5): Aparece un listado con las patentes de las que es inventor o coinventor.



The image shows a web profile for Jesus Manuel Alegre Calderon, a university professor. It includes a circular profile picture, his name, title, and a navigation menu with tabs for Perfil, Publicaciones, Colaboración, Tesis, Patentes, and Financiación. Below the menu, his department, academic center, research unit, doctoral program, area, research group, and email are listed.

**JESUS MANUEL  
ALEGRE CALDERON**  
CATEDRÁTICO DE UNIVERSIDAD

**PERFIL** PUBLICACIONES COLABORACIÓN TESIS PATENTES FINANCIACIÓN

Departamento: **INGENIERIA CIVIL**

Centro académico: **ESCUELA POLITECNICA SUPERIOR**

Unidad de Investigación: **UNIDAD DE INVESTIGACION CONSOLIDADA 163- Integridad estructural**

Programa de doctorado: **Ingeniería y Tecnologías Industrial, Informática y Civil (UBU)**

Área: **Mecánica de Medios Continuos y Teoría de Estructuras**

Grupo de investigación: **Integridad Estructural**

Email: **jalegre@ubu.es**

Figura 5.2: Captura pestaña detalle

## Publicaciones (136)

Publicaciones de JESUS MANUEL ALEGRE CALDERON.

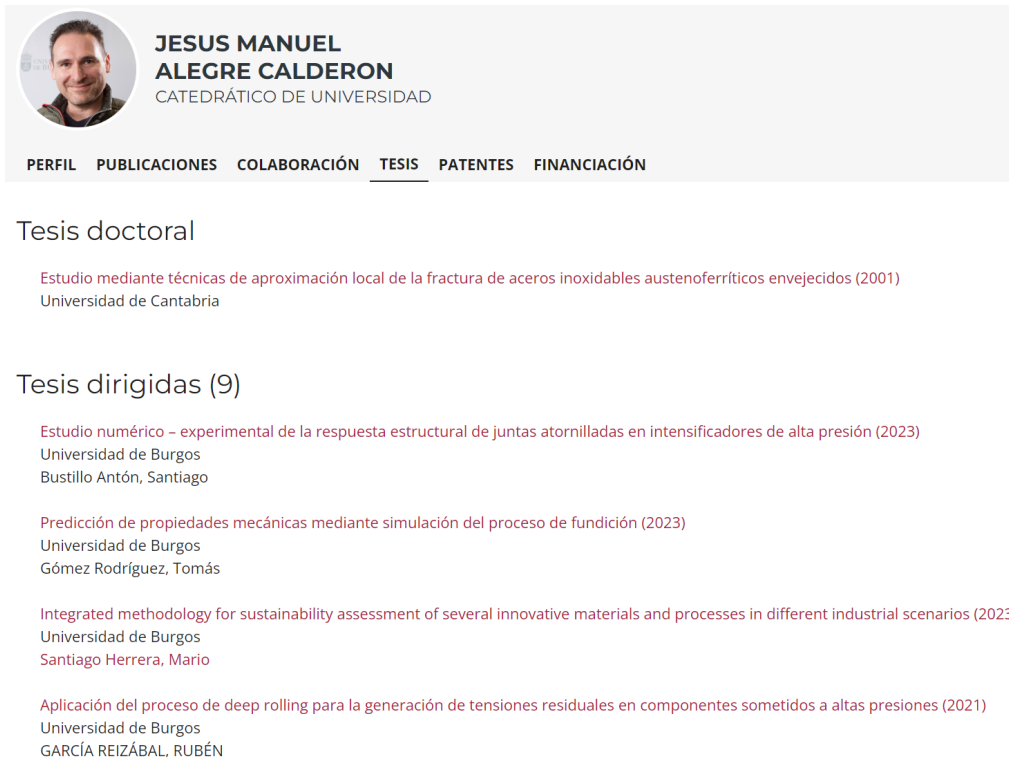
### 2024

- Adjustment of Mechanical Property Predictions for Structural Castings in the Wind Gearbox Sector  
International Journal of Metalcasting, Vol. 18, Núm. 1, pp. 717-733
- Effect of electrochemical charging on the hydrogen embrittlement susceptibility of a low-alloyed tempered martensitic steel submitted to high internal pressure  
International Journal of Hydrogen Energy, Vol. 63, pp. 657-667
- Evaluating hydrogen embrittlement susceptibility of a 2205 DSS  
Procedia Structural Integrity
- Ex-ante life cycle assessment of directed energy deposition based additive manufacturing: A comparative gearbox production case study  
Sustainable Materials and Technologies, Vol. 39
- Explicit implementation of hydrogen transport in metals  
International Journal of Mechanical Sciences, Vol. 273
- Hydrogen Embrittlement of AISI 316L steel produced by Selective Laser Melting  
Procedia Structural Integrity

### 2023

- Caracterización de la fragilización por hidrógeno mediante ensayos de fractura hidráulica a elevada presión interna en el acero 42CrMo4: efecto del tamaño de grano  
Revista española de mecánica de la fractura, Núm. 6, pp. 9-14
- Comparative Life Cycle Assessment and Cost Analysis of the Production of Ti6Al4V-TiC Metal-Matrix Composite Powder by High-Energy Ball Milling and Ti6Al4V Powder by Gas Atomization  
Sustainability (Switzerland), Vol. 15, Núm. 8

Figura 5.3: Captura pestaña publicaciones



**JESUS MANUEL ALEGRE CALDERON**  
CATEDRÁTICO DE UNIVERSIDAD

PERFIL PUBLICACIONES COLABORACIÓN TESIS PATENTES FINANCIACIÓN

### Tesis doctoral

Estudio mediante técnicas de aproximación local de la fractura de aceros inoxidables austenoferríticos envejecidos (2001)  
Universidad de Cantabria

### Tesis dirigidas (9)

Estudio numérico – experimental de la respuesta estructural de juntas atornilladas en intensificadores de alta presión (2023)  
Universidad de Burgos  
Bustillo Antón, Santiago

Predicción de propiedades mecánicas mediante simulación del proceso de fundición (2023)  
Universidad de Burgos  
Gómez Rodríguez, Tomás

Integrated methodology for sustainability assessment of several innovative materials and processes in different industrial scenarios (2023)  
Universidad de Burgos  
Santiago Herrera, Mario

Aplicación del proceso de deep rolling para la generación de tensiones residuales en componentes sometidos a altas presiones (2021)  
Universidad de Burgos  
GARCÍA REIZÁBAL, RUBÉN

Figura 5.4: Captura pestaña tesis



**JESUS MANUEL ALEGRE CALDERON**  
CATEDRÁTICO DE UNIVERSIDAD

PERFIL PUBLICACIONES COLABORACIÓN TESIS PATENTES FINANCIACIÓN

### Patentes (1)

DISPOSITIVO PARA ENSAYO DE COMPONENTES SOMETIDOS A ALTA PRESIÓN EN UNA MÁQUINA DE TRACCIÓN-COMPRESIÓN 2018

Figura 5.5: Captura pestaña patentes

El portal de investigación consta con más información referente a los investigadores pero no es relevante para la realización del *Web Scraping*.

Para acceder a la información de cada pestaña antes mencionada me di cuenta que la URL tenía la misma estructura, y cambiando el identificador del investigador, se podía acceder a cada pestaña de cada investigador.

Para ello, primero creé una estructura de modelos en el proyecto de *.NET* y lancé una migración para la creación de la base de datos relacionada.

Una vez con la base de datos configurada y después de muchas pruebas individuales, confeccioné el código para acceder a cada pestaña de cada investigador y guardar la información de los contenedores elegidos en la base de datos. Este proceso depende de las capacidades del terminal y la velocidad del ordenador, pero su tiempo estimado de funcionamiento es cercano a las 14 horas en las que guarda aproximadamente 200.000 registros en la base de datos.

## 5.6. Aplicación web

Con la base de datos terminada, me dispuse a comenzar el desarrollo de la aplicación *front-end*. Comencé instalando Visual Studio Code como editor de texto. También instalé *Angular* y generé un nuevo proyecto. Dentro del proyecto de *Angular* generé un módulo de investigadores que sería el centro de la aplicación. Desde *.NET* creé un controlador de investigadores donde estaría toda la funcionalidad, y programé funciones para mandar información de la base de datos a la aplicación web. Una vez hecha la funcionalidad en *back-end*, dispuse una tabla de investigadores con la información más relevante. Para la maquetación de la misma, instalé *Angular material* para tener estilos predefinidos y *FlexLayout* para manejar la disposición de los contenedores en página.

WebScrapingTFG							
<div> <div>Investigadores</div> <div>Investigador</div> <div>Investigadores relacionados</div> </div>							
	Id	Nombre	Apellidos	Idinvestigador	Email	Departamento	Acciones
<input type="checkbox"/>	1	VICTORIA	ABAD SAN MARTIN	34937	vaspol@ubu.es	INGENIERIA ELECTROMECHANICA	>
<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 ELECTROMECHANICA	>
<input type="checkbox"/>	4	FERNANDO	AGUILAR ROMERO	34946	fagular@ubu.es	INGENIERIA ELECTROMECHANICA	>
<input type="checkbox"/>	5	VIRGINIA	AHEDO GARCIA	34947	vaiedo@ubu.es	INGENIERIA DE ORGANIZACIÓN	>
<input type="checkbox"/>	6	LOURDES	ALAMEDA CUENCA ROMERO	34949	lalameda@ubu.es	CONSTRUCCION 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	ALLEGRE CALDERON	34954	jalegre@ubu.es	INGENIERIA CIVIL	>
<input type="checkbox"/>	10	ALVARO	ALISTE RABANO	1096523	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	jealonso@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 ELECTROMECHANICA	>

Figura 5.6: Listado de investigadores

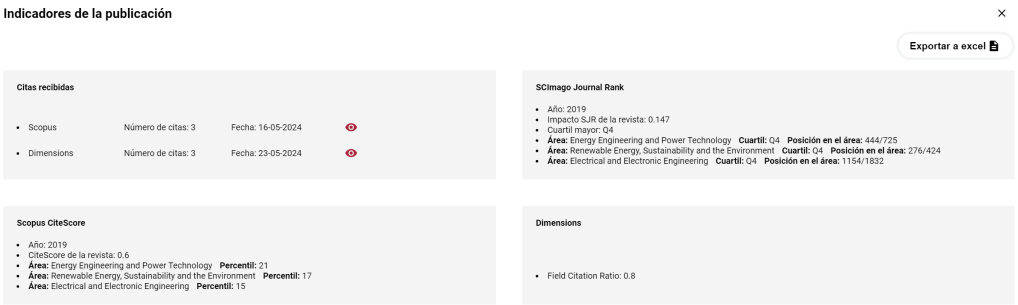


Figura 5.8: Listado de indicadores de publicación

Para cada fila de la tabla, incluí un botón donde se pueden ver listadas todas las publicaciones del investigador, también se puede realizar una exportación a *Excel* del listado de publicaciones junto a sus indicadores para un manejo de la información en local.

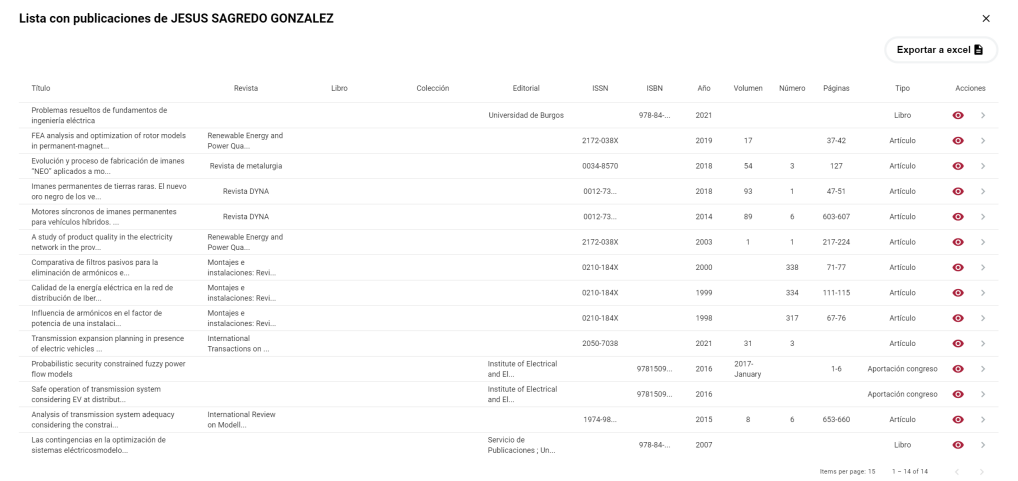


Figura 5.7: Listado de publicaciones del investigador

También, para cada publicación hay preparada una ventana modal que muestra los indicadores que tiene dicha publicación y su respectiva exportación a *Excel*.

## 5.7. Problemas en el desarrollo

En la producción de esta aplicación han aparecido una serie de problemas y dificultades, algunas debido a errores en el portal de investigación y otras debido a complicaciones en el desarrollo.

En el portal de investigación, un investigador puede pertenecer como máximo a un departamento y como máximo a un centro de investigación. Durante el proceso de pruebas de *Web Scraping*, encontré 3 investigadores que tenían múltiples departamentos y un investigador que tenía múltiples facultades. Este suceso hizo que tendría que modificar la estructura de la base de datos y añadir una tabla intermedia que almacenara esta posible multiplicidad. Gracias a encontrar este error, se enviaron los identificadores de los investigadores afectados y se pudo arreglar este error internamente en el portal de investigación.

Relacionado con esto anterior, de que un investigador solo tiene un departamento y un centro de investigación, me topé que con que había investigadores asociados a un departamento incluidos en el listado de investigadores de otro departamento, esto también pasaba con los centros de investigación. Por lo que al obtener la lista con los investigadores de todo el portal, tenía algunos repetidos debido a que habían sido incluidos desde departamentos o centros de investigación de los cuales no eran miembros.

Para la solución en código, añadí una verificación para evitar duplicados. Para ayudar a solucionar este problema dentro del portal de investigación se preparó un desarrollo que mediante *Web Scraping* obtenía el id del investigador, el departamento al que estaba asociado y los departamentos en los que aparecía como miembro sin serlo. Se consiguieron generar así 2 documentos, uno que incluía 204 registros de investigadores con departamentos erróneos y 112 registros de investigadores con centros de investigación erróneos.

Otro de los grandes problemas que me he encontrado durante la realización del proyecto es que la página web ha ido cambiando a lo largo de los meses de desarrollo, por lo que ha habido que rehacer código varias veces para una misma funcionalidad. Al depender del *Web Scraping* para la obtención de los datos, dependo a su vez de lo que haya en la página web del portal del investigador en el momento que se lance el proceso. Han cambiado textos que utilizaba para acceder a los contenedores mediante *Selenium*.

Por ejemplo, lo que en este momento son centros de investigación, hace escasos meses se denominaban facultades. También ha habido cambios en cuanto a la composición de la web, cambiando la estructura de contenedores o sustituyendo listas por tablas. Estructura a partir de la cual había progra-

mado una plantilla y con el cambio en los contenedores, se tuvo que volver a desarrollar el proceso para el correcto *Web Scraping* de los datos.

Por último, una de las grandes dificultades que he encontrado en la realización del proyecto ha sido en torno a la asignación de los investigadores que no están dados de alta en el portal, dentro de las publicaciones. Esto es debido a que dentro de las publicaciones, en caso de no estar dado de alta, los investigadores aparecen escritos con texto plano y sin ninguna referencia, por lo que, si un investigador aparece con un error en la escritura de su nombre, no tengo manera de saber que es la misma persona que ha hecho otra publicación con el nombre bien escrito y se tratan como investigadores diferentes. Lo mismo ocurre si uno aparece con tilde y otro sin tilde, similar con diéresis, guiones, espacios u otros símbolos.

Además, a veces no se utiliza el nombre completo, sino que se incluye una abreviación del nombre junto al primer apellido, al comparar este nombre abreviado con nombre completo, se interpretara también como dos investigadores distintos, siendo en realidad un único investigador.

Para comprender este apartado, un ejemplo:

**Josu Barrenetxea García** es el nombre correctamente escrito.

**Josu Barrenetxea-García** escrito con guión.

**Josu Barrenetxea Garcia** escrito sin tilde.

**Barrenetxea, J.** sería la abreviatura.

El investigador es la misma persona, pero se interpretaría como 4 investigadores diferentes.

Para solucionar este apartado, se ha hecho un desarrollo que permite fusionar toda la información de los investigadores en uno único. Desde *Angular* se ha implementado un filtrado por nombre y un selector de investigadores con los que ha colaborado para acotar la búsqueda. También se ha incluido en el listado una opción para seleccionar los investigadores que se quieren fusionar, y al presionar en un nuevo botón de fusionar implementado, se abre una ventana modal en la que se muestran los investigadores seleccionados y se elige a qué investigador se va a enlazar la información referente al resto de seleccionados.

**Selecciona el investigador al que asociar** ×

	Id	Nombre	Apellidos	IdInvestigador	Email	Departamento
<input type="checkbox"/>	1406	Josu Barrenetxea	García			
<input type="checkbox"/>	1404	Josu Barrenetxea	García			
<input type="checkbox"/>	1405	Josu Barrenetxea-	García			
<input type="checkbox"/>	1407	Barrenetxea, J.				


**Guardar** 

Figura 5.9: Ventana modal para fusionar investigadores



---

## 6. Trabajos relacionados

---

### 6.1. Introducción

La aplicación desarrollada en este proyecto consta con una funcionalidad muy concreta, y en el mercado no existen herramientas similares que cumplan los objetivos que se han abordado. Por lo que en esta sección, se pondrán a prueba los portales de investigación de otras universidades, para constatar la falta de funcionalidad que precisan los coordinadores del programa de doctorado para elaborar sus informes.

### 6.2. Universidades

Para la realización de este apartado se han tenido en cuenta los portales de investigación de las universidades de **La Rioja**<sup>33</sup>, **Murcia**<sup>34</sup> y **Oviedo**<sup>35</sup>.

En la navegación por los portales de las distintas universidades se puede apreciar que tienen un aspecto y estructura similar al **portal de investigación de la univervidad de Burgos**<sup>36</sup> portal de investigación de la univervidad de Burgos, como se puede apreciar en la imagen:

---

<sup>33</sup><https://investigacion.unirioja.es/>

<sup>34</sup><https://portalinvestigacion.um.es/>

<sup>35</sup><https://portalinvestigacion.uniovi.es/>

<sup>36</sup><https://investigacion.ubu.es/>



Figura 6.1: Captura portal de investigación Universidad de La Rioja

La investigadores siguen siendo el eje en torno al que gira la información del portal, y de igual forma que pasaba en la Universidad de Burgos, no es posible acceder a la información de los investigadores que no están dados de alta en el portal. El problema que tienen los coordinadores de doctorado para obtener la información de sus doctorandos, ocurre también en el resto de universidades.

En conclusión, el desarrollo de esta herramienta, que ha conseguido facilitar la labor del coordinador de doctorado en la gestión de la información y en la elaboración de informes, podría aplicarse al resto de universidades. Gracias a que estos portales presentan una estructura similar en cuanto a la agrupación y disposición de la información en la página. Personalizando el código de la función de *Web Scraping* en la aplicación de *.NET*, a la estructura de contenedores que disponen las distintas páginas del resto de portales, se conseguiría solucionar el objetivo abordado para distintas universidades.

---

## 7. Conclusiones y Líneas de trabajo futuras

---

### 7.1. Conclusiones

Tras el desarrollo del proyecto, se puede constatar que se ha dado solución a los problemas que han surgido durante el progreso del mismo, para dar pie al objetivo final. Se ha tenido que lidiar con el cambio de enfoque del proyecto que había inicialmente, teniendo que realizar *Web Scraping* para obtener la base de datos, añadir funcionalidad para conseguir fusionar investigadores o los filtros de las tablas.

Como resultado, se ha conseguido un entorno con una aplicación *back-end* y una aplicación *front-end*, que obtiene la información del portal del investigador mediante *Web Scraping*, y es capaz de interaccionar con una aplicación web, a la que le envía la información de la base de datos. Con esto se ha logrado dar solución al problema que tenía el coordinador del programa de doctorado de poder obtener la producción científica de todos los doctorandos, ya estén dados de alta en el portal de investigación como los que no lo están.

A nivel personal, el desarrollo del trabajo ha sido duro y tedioso, al tener que solucionar problemas de configuración de los entornos, y sobre todo, el hecho de tener que rehacer funcionalidad que ya estaba desarrollada debido a los cambios en el portal de investigación. Pero a pesar de todo, quedo satisfecho con el conjunto de funcionalidad al que se ha logrado llegar y que cumple con los requisitos propuestos al inicio.

En cuanto a lo aprendido a nivel formativo, destaco el haber conseguido desarrollar una aplicación completa, separando el *front-end* del *back-end*

y configurado con una base de datos, todo ello incluido en un contenedor *Docker* para una sencilla portabilidad. Comentar también el uso de *.NET* que aunque lo conocía por encima, gracias al desarrollo del proyecto, he conseguido profundizar mucho más en el aprendizaje, incluso siendo capaz de realizar la configuración de modelos, para, con una migración, construir la base de datos relacional.

También resaltar, el aprendizaje obtenido sobre el *Web Scraping* y el funcionamiento de *Selenium* para su implementación en el proyecto. Y finalmente el uso de *Docker*, que era una herramienta que desconocía totalmente, pero que al realizar su configuración en el proyecto, guardo todo lo aprendido para proyectos futuros.

En conclusión, estoy satisfecho de todo lo aprendido y del camino que he recorrido para llevarlo a cabo. La aplicación cumple con los requisitos del proyecto, y está preparada para que sea escalable y puedan ampliarse funcionalidades, no tanto en el *back-end* ya que considero que la parte de *Web Scraping* es muy completa, sino más enfocado en extender las posibilidades que puede alcanzar la aplicación de *front-end*.

## 7.2. Líneas de trabajo futuras

A medida que iban sucediendo las reuniones con los tutores entre los *sprints*, surgían ideas y posibles implementaciones que complementarían la aplicación y la harían más completa, a continuación se proporciona un listado con algunas de ellas:

- Como desarrollo ideal, esta aplicación está preparada para alojarse en un sitio web público al que cualquier persona interesada podría acceder. Con el fin de tener que hacer el *Web Scraping* una sola vez, debido a que tarda cerca de 12 horas en completarse, y tener una base de datos común para todos.
- Complementando el punto anterior y pensando en los permisos que tendrían los usuarios sobre la web, habría que distinguir los roles para poder realizar las acciones. Limitando así que solo los usuarios con el rol de administrador puedan lanzar el proceso de *Web Scraping* o fusionar los investigadores, que propician cambios en la base de datos, que afectarían a todos los usuarios.

El resto de personas solamente tendría la posibilidad de visualizar la información. Para ello, habría que incluir un apartado de identifica-

ción antes de acceder a la aplicación. Las alternativas para esto son gestionarlo con el uso de la base de datos, o implementar *Azure*, para poderse identificar con el email de la universidad.

- Uno de los problemas a futuro que supone haber obtenido la base de datos con *Web Scraping*, es que una vez realizado el proceso, si la página del portal de investigación sufre algún cambio en cuanto a datos (ej: publicaciones nuevas, indicadores modificados...), la base de datos que alimenta la aplicación estaría desactualizada.

Por lo que uno de los desarrollos a futuro sería implementar la posibilidad de hacer *Web Scraping* sobre partes concretas y que la información guardada en base de datos se modifique o se añada. Por ejemplo, poder lanzar el *Web Scraping* sobre un investigador en concreto, y obtener toda su información relacionada actualizada.

- Mejorar la visualización y ampliar la aplicación web con más pestañas. Actualmente, se dispone de la pestaña de investigadores, pero en base de datos, está guardada la información referente a departamentos, centros de investigación... Se pueden añadir pestañas sobre estas unidades y permitir una visualización de los investigadores miembros, con filtrado, paginación...
- Dentro de cada investigador, nos hemos centrado en la visualización de las publicaciones, ya que era el objetivo, pero se podría ampliar la información, añadiendo sus tesis y patentes realizadas.
- Durante una de las reuniones de control, se comentó que el coordinador del programa de doctorado tenía que realizar un informe llamado *Seguimiento anual de los títulos oficiales de Doctorado*, en el que se debe adjuntar la siguiente información:
  - Tesis doctorales defendidas
  - Contribuciones científicas de los estudiantes
  - Indicadores de menciones en las tesis defendidas

Gracias a la aplicación desarrollada, es posible acceder a toda la información anterior. Pero una posible mejora, sería filtrar los indicadores más relevantes y poder generar un informe personalizado que facilite la realización de la tarea. También se puede plantear la generación directa de la documentación requerida, creando una plantilla de cómo sería el documento, y que se rellene dinámicamente con la información de base de datos. Para esta generación de documentos, realicé una

búsqueda sobre posibles librerías en *.NET* que permitirían el desarrollo, y aunque llegué a realizar pruebas con alguna de estas, por falta de tiempo no se pudo llevar a cabo la implementación.

---

## Bibliografía

---

- [1] Amazon. ¿que es Docker? <https://aws.amazon.com/es/docker/>, 2023. [Internet; Visitado por última vez: 26-junio-2024].
- [2] Angular. Introducción a los conceptos de angular. <https://docs.angular.lat/guide/architecture>, 2020. [Internet; Visitado por última vez: 26-junio-2024].
- [3] Gustavo B. ¿qué es Github? <https://www.hostinger.es/tutoriales/que-es-github>, 2023. [Internet; Visitado por última vez: 26-junio-2024].
- [4] Camelcamelcamel. Comparación apple iphone 15 pro max. <https://es.camelcamelcamel.com/product/BOCHX2WQPW>, 2024. [Internet; Visitado por última vez: 27-junio-2024].
- [5] Datademia. ¿qué es web scraping? <https://datademia.es/blog/que-es-web-scraping>. [Internet; Visitado por última vez: 27-junio-2024].
- [6] Fernán García de Zúñiga. Frontend: ¿qué es y para qué se utiliza en desarrollo web? <https://www.arsys.es/blog/frontend-que-es-y-para-que-se-utiliza-en-desarrollo-web>, 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [7] Gluo. Backend: ¿qué es y para qué sirve? <https://www.gluo.mx/blog/backend-que-es-y-para-que-sirve>, 2023. [Internet; Visitado por última vez: 26-junio-2024].
- [8] Google. Comparación apple iphone 15 pro max. <https://shopping.google.es/>, 2024. [Internet; Visitado por última vez: 27-junio-2024].

- [9] Microsoft. Visual studio: IDE y editor de código para desarrolladores de software. <https://visualstudio.microsoft.com/es/>, 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [10] Microsoft. ¿qué es .net? <https://dotnet.microsoft.com/es-es/learn/dotnet/what-is-dotnet>, 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [11] Brendon V. Ridge. La importancia de separar el frontend y el backend en el desarrollo web. <https://www.mediummultimedia.com/web/por-que-separar-frontend-y-backend/>, 2023. [Internet; Visitado por última vez: 26-junio-2024].
- [12] Sandra Garrido Sotomayor. Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>, 2023. [Internet; Visitado por última vez: 26-junio-2024].
- [13] Wikipedia. Docker (software) — Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Docker\\_\(software\)](https://es.wikipedia.org/wiki/Docker_(software)), 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [14] Wikipedia. Github — Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/GitHub>, 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [15] Wikipedia. Sql server management studio — Wikipedia, la enciclopedia libre. [https://en.wikipedia.org/wiki/SQL\\_Server\\_Management\\_Studio](https://en.wikipedia.org/wiki/SQL_Server_Management_Studio), 2024. [Internet; Visitado por última vez: 26-junio-2024].
- [16] Wikipedia. Web scraping — Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Web\\_scraping](https://es.wikipedia.org/wiki/Web_scraping), 2024. [Internet; Visitado por última vez: 27-junio-2024].