



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Generación de cuestionarios  
sobre algoritmos de  
ordenación  
Documentación Técnica**



Presentado por Mario García Martínez  
en Universidad de Burgos — 10 de junio  
de 2024

Tutor: Dr. Juan José Rodríguez Díez



---

# 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 . . . . .	2
A.3. Estudio de viabilidad . . . . .	13
<b>Apéndice B Especificación de Requisitos</b>	<b>17</b>
B.1. Introducción . . . . .	17
B.2. Objetivos generales . . . . .	18
B.3. Catálogo de requisitos . . . . .	18
B.4. Especificación de requisitos . . . . .	20
<b>Apéndice C Especificación de diseño</b>	<b>33</b>
C.1. Introducción . . . . .	33
C.2. Diseño de datos . . . . .	33
C.3. Diseño procedimental . . . . .	36
C.4. Diseño arquitectónico . . . . .	38
<b>Apéndice D Documentación técnica de programación</b>	<b>41</b>
D.1. Introducción . . . . .	41
D.2. Estructura de directorios . . . . .	41
D.3. Manual del programador . . . . .	43

D.4. Compilación, instalación y ejecución del proyecto . . . . .	47
D.5. Pruebas del sistema . . . . .	48
<b>Apéndice E Documentación de usuario</b>	<b>51</b>
E.1. Introducción . . . . .	51
E.2. Requisitos de usuarios . . . . .	51
E.3. Instalación . . . . .	51
E.4. Manual del usuario . . . . .	52
<b>Apéndice F Anexo de sostenibilización curricular</b>	<b>65</b>
F.1. Introducción . . . . .	65
F.2. Implicación ética . . . . .	65
F.3. Impacto medioambiental . . . . .	66
F.4. Aspectos socioculturales . . . . .	66
F.5. Conclusión . . . . .	67
<b>Bibliografía</b>	<b>69</b>

---

# Índice de figuras

---

A.1. <i>Issues Sprint 1</i>	3
A.2. <i>Issues Sprint 2</i>	4
A.3. <i>Issues Sprint 3</i>	6
A.4. <i>Issues Sprint 4</i>	7
A.5. <i>Issues Sprint 5</i>	8
A.6. <i>Issues Sprint 6</i>	9
A.7. <i>Issues Sprint 7</i>	11
A.8. <i>Issues Sprint 8</i>	13
B.1. <i>Casos de uso.</i>	21
C.1. <i>Diagrama de secuencia.</i>	37
C.2. <i>Diagrama representativo del patrón Modelo-Vista-Controlador.</i>	38
C.3. <i>Diagrama representativo del patrón de diseño fachada.</i>	40
D.1. <i>Descarga del repositorio desde la web de GitHub.</i>	44
D.2. <i>Lanzamiento en local.</i>	45
D.3. <i>Creando una app desde Koyeb.</i>	46
D.4. <i>Configuración del despliegue de la web en Koyeb.</i>	47
D.5. <i>Selección del repositorio desde el equipo en Visual Studio Code.</i>	48
D.6. <i>Revisión de pruebas en Katalon Recorder.</i>	50
E.1. <i>Página principal de la web.</i>	53
E.2. <i>Página sobre el algoritmo Mergesort.</i>	54
E.3. <i>Página sobre el algoritmo Quicksort.</i>	54
E.4. <i>Página sobre el algoritmo Heapsort.</i>	55
E.5. <i>Elementos a rellenar en el formulario (Algoritmo Mergesort).</i>	56
E.6. <i>Seleccionar preguntas desde el formulario (Algoritmo Mergesort).</i>	57
E.7. <i>Descarga del banco de preguntas (Algoritmo Mergesort).</i>	58

E.8. <i>Página de más información.</i> . . . . .	59
E.9. <i>Inicio de sesión en Moodle.</i> . . . . .	60
E.10. <i>Añadimos la actividad del cuestionario en Moodle.</i> . . . . .	61
E.11. <i>Importación de banco de preguntas en Moodle.</i> . . . . .	62
E.12. <i>Implementación de las preguntas del banco en Moodle.</i> . . . . .	63
E.13. <i>Representación del banco de preguntas en Moodle.</i> . . . . .	63

---

# Índice de tablas

---

A.1. Lista de materiales utilizados con un valor aproximado y su amortización. . . . .	14
A.2. Costes de amortización de los materiales. . . . .	14
A.3. Desglose del coste de personal. . . . .	15
A.4. Licencias correspondientes a las herramientas utilizadas. . . . .	16
B.1. CU-1 Caso de uso 1. . . . .	22
B.2. CU-2 Caso de uso 2. . . . .	23
B.3. CU-3 Caso de uso 3. . . . .	24
B.4. CU-4 Caso de uso 4. . . . .	25
B.5. CU-5 Caso de uso 5. . . . .	26
B.6. CU-6 Caso de uso 6. . . . .	27
B.7. CU-7 Caso de uso 7. . . . .	28
B.8. CU-8 Caso de uso 8. . . . .	29
B.9. CU-9 Caso de uso 9. . . . .	30
B.10. CU-10 Caso de uso 10. . . . .	31
B.11. CU-11 Caso de uso 11. . . . .	32
D.1. Licencias correspondientes a las herramientas utilizadas. . . . .	44





## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este apartado introductorio se tratará de concretar cómo se ha realizado la estructura de objetivos a completar durante el desarrollo de la planificación del proyecto y el uso de *sprints* durante el mismo.

Para el planteamiento del desarrollo del proyecto se ha optado por usar la herramienta Github. Esta herramienta aporta los elementos necesarios para desarrollar en un formato sencillo e intuitivo la planificación del proyecto.

Una vez comenzado el proyecto y tras tratar las cuestiones correspondientes a los objetivos que se plantearon para la finalización del mismo se optó por emplear la metodología SCRUM. Método que nos permite llevar a cabo las tareas que se irían asignado conforme avance el desarrollo del trabajo pese a ser una metodología planteada para la realización de trabajos en equipo donde se fomenta el trabajo de forma organizada y colaborativa entre los miembros del proyecto.

Para adaptar de la forma más precisa posible los conceptos en los que se basa la metodología SCRUM en el desarrollo de proyectos, teniendo en cuenta que se realiza por medio de una única persona, se han tenido en cuenta los siguientes puntos para la planificación de este trabajo.

- El diseño y planificación del trabajo se han realizado de forma incremental.
- Asistencia a reuniones cada dos semanas excepto en los casos en los que se requiera más o menos tiempo.

- El diseño y planificación se del trabajo se han realizado de forma incremental.
- Seleccionar y realizar aquellas tareas que requieran mayor prioridad.
- Realizar la toma de decisiones sobre el tiempo necesario para el cumplimiento de cada tarea.
- Tras la finalización de cada Sprint, concretar una reunión para comprobar si se han cumplido con las expectativas establecidas o si es necesario una revisión más concreta de parte del desarrollo realizado

## A.2. Planificación temporal

En esta sección se detallarán los objetivos que se plantearon y se utilizaron durante todo el proceso de desarrollo. Se explicará a su vez cuáles eran los objetivos iniciales en cada Sprint y cuáles son aquellas tareas que finalmente se llevaron a cabo durante el mismo.

### Sprint 1: (1/02/2024 - 14/02/2024)

En este primer Sprint se planificaron los primeros pasos a seguir para el desarrollo del proyecto tras la primera reunión. Se comenzó con la realización de la estructura del proyecto con el fin de tener una idea clara de como se podría plantear el inicio del trabajo. Posteriormente, se realizó un pequeño ejemplo en el repositorio para hacerse a la idea de cómo manejarse en el entorno de trabajo disponible.

Inicialmente se estudiaron los conceptos teóricos que se tratarían en un principio para poder aplicarlos correctamente. Se estudiaron los conceptos correspondientes a los algoritmos de ordenación comúnmente utilizados en el ámbito profesional.

A su vez también se buscó información acerca del formato de archivos *XML Moodle* con el fin de conocer su funcionamiento para poder plantear preguntas bien estructuradas. Con ello se pretendía establecer el formato de programación que se deseaba utilizar para el proyecto. Se optó por generar archivos XML en el formato estandarizado de Moodle para su posterior exportación y uso en las respectivas plataformas docentes.

Finalmente se realizó un esquema sencillo de preguntas de ejemplo de dos de los algoritmos de ordenación de vectores más comunes. Concretamente el Mergesort u ordenación por mezcla y Quicksort u ordenación rápida.

Aquí se incluye el enlace al primer Sprint: **Sprint 1**. A continuación se muestra la lista de *Issues* correspondiente al primer Sprint.






<input type="checkbox"/>	<input checked="" type="checkbox"/> Implementación de método para la ordenación de particiones del algoritmo Quicksort unidireccional. <b>enhancement</b> <small>#5 by mgm1027 was closed on Feb 8</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Implementación de método para la ordenación de particiones del algoritmo Quicksort bidireccional. <b>enhancement</b> <small>#4 by mgm1027 was closed on Feb 8</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Creación de un método para general una pregunta sobre el funcionamiento del algoritmo Quicksort <b>enhancement</b> <small>#3 by mgm1027 was closed on Feb 8</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Método Mergesort para corregir correctamente las preguntas que correspondan <b>enhancement</b> <small>#2 by mgm1027 was closed on Feb 8</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Primer método sobre la creación de archivo formato XML (Mergesort) <b>good first issue</b> <small>#1 by mgm1027 was closed on Feb 8</small>	

Figura A.1: *Issues Sprint 1*

## Sprint 2: (15/02/2024 - 22/02/2024)

Durante el segundo Sprint se trató de implementar en las cuestiones ya replanteadas anteriormente la posibilidad de generar las preguntas con datos aleatorios. Todo ello con el fin de ofrecer distintas variaciones de problemas para resolver por el examinado durante la realización del cuestionario.

A su vez, también se reformuló la estructura de los métodos realizados durante el anterior Sprint, con el nuevo objetivo de poder generar cuestionarios completos de cierto número de preguntas seleccionado por el usuario. Todo esto con el fin de automatizar la generación de los cuestionarios de forma aleatoria.

También se abrió la posibilidad de aportar aleatoriedad no solo a los valores de los datos aportados en cada pregunta sino también a la cantidad

de datos aportado. Concretamente al tamaño de los vectores que se ofrecen para su posterior resolución.

Finalmente, se comenzó a estudiar la posibilidad de crear y aportar aleatoriedad a otro tipo de formato de preguntas para los algoritmos de ordenación ya establecidos en el proyecto, lo cual se trataría de llevar a cabo en el próximo Sprint.

Aquí se incluye el enlace al segundo Sprint: [Sprint 2](#)

A continuación se muestra la lista de *Issues* correspondiente al segundo Sprint.





<input type="checkbox"/>	<input checked="" type="checkbox"/> Añadir vectores aleatorios a las diferentes preguntas del cuestionario (Quicksort) <b>enhancement</b>	
	<small>#8 by mgm1027 was closed on Feb 20</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Generar cuestionarios con varias preguntas incluidas (Quicksort). <b>enhancement</b>	
	<small>#9 by mgm1027 was closed on Feb 20</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Generar cuestionarios con varias preguntas incluidas (Mergesort). <b>enhancement</b>	
	<small>#6 by mgm1027 was closed on Feb 15</small>	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Añadir vectores aleatorios a las diferentes preguntas del cuestionario (Mergesort) <b>enhancement</b>	
	<small>#7 by mgm1027 was closed on Feb 15</small>	

Figura A.2: *Issues Sprint 2*

### Sprint 3: (23/02/2024 - 07/03/2024)

Para la finalización de tercer Sprint se buscó inicialmente información acerca de los diferentes formatos de preguntas ofrecidos por el formato *XML Moodle*. Todo ello con el fin de seleccionar cuáles eran las mejores opciones para poder implementar como nuevas preguntas para los algoritmos *Mergesort* y *Quicksort*.

Una vez revisado se optó por desarrollar nuevas preguntas de tipo de selección múltiple para ambos algoritmos. Para ello se revisó la posibilidad

de aportar tres opciones distintas como posibles soluciones a las preguntas donde dos de las cuales serían opciones erróneas pero similares a la opción real. Además también se tuvo que estudiar la posibilidad de reordenar las opciones posibles para que la solución correcta no estuviera siempre en la misma posición.

Por otra parte, para el caso del algoritmo de ordenación rápida *Quicksort* se optó por generar dos métodos para las preguntas de selección múltiple, pero teniendo en cuenta los dos distintos métodos que existen para la correcta realización del algoritmo. El método unidireccional partiendo desde el inicio vector, y el método bidireccional partiendo desde ambos lados del vector.

Para concluir el Sprint, se decidió implementar la solución correcta en la corrección de cada pregunta del cuestionario de manera automática. Con el objetivo de que los examinados pudieran saber tras realizar la prueba cuál fue la respuesta correcta y por tanto ayudarles en su comprensión del tema.

Una vez más tras la implementación de las cuestiones de selección múltiple, se estudió la posibilidad de implementar nuevas preguntas con otros enfoques. Todo ello con el fin de aprender sobre los algoritmos que se llevarían a cabo en un futuro Sprint.

Aquí se incluye el enlace al tercer Sprint: [Sprint 3](#)

A continuación se muestra la lista de *Issues* correspondiente al tercer Sprint.





<input type="checkbox"/>	<input checked="" type="radio"/> Generación de preguntas adicionales en el cuestionario aleatorio mergesort <b>enhancement</b> #13 by mgm1027 was closed on May 8	
<input type="checkbox"/>	<input checked="" type="radio"/> Generación de preguntas adicionales en el cuestionario aleatorio quicksort <b>enhancement</b> #14 by mgm1027 was closed on May 8	
<input type="checkbox"/>	<input checked="" type="radio"/> Implementación de pregunta opción múltiple quicksort unidireccional <b>enhancement</b> #11 by mgm1027 was closed on Mar 5	
<input type="checkbox"/>	<input checked="" type="radio"/> Implementación de pregunta de opción múltiple quicksort bidireccional <b>enhancement</b> #12 by mgm1027 was closed on Mar 5	
<input type="checkbox"/>	<input checked="" type="radio"/> Implementación de pregunta de opción múltiple mergesort <b>enhancement</b> #10 by mgm1027 was closed on Mar 1	

Figura A.3: *Issues Sprint 3*

## Sprint 4: (14/03/2024 - 09/04/2024)

El objetivo del cuarto Sprint se basó en dar los primeros pasos a la creación de un entorno web con el que poder mostrar el funcionamiento del proyecto de una forma más sencilla e intuitiva para el usuario. Todo por medio de un servidor Flask y una página web diseñada para poder recoger las proposiciones para la generación de cuestionarios del usuario.

Debido al desconocimiento acerca del desarrollo de páginas web y el uso de Flask, durante este Sprint se requirió realizar una prueba de contacto con dichas herramientas. Todo ello con el objetivo de hacerse una idea de que funcionalidades ofrecían y posteriormente como aplicar dichas herramientas al proyecto en curso.

Una vez realizadas las pruebas, se optó por facilitar la información acerca de los elementos deseados para la completa generación de cuestionarios por medio de un formulario web. El usuario podría completar el formulario para así obtener un archivo *XML Moodle* a su gusto, concretamente el número de preguntas en un comienzo.

Finalmente fue necesario enfocarse en la estructuración de la página web y el diseño para tener un ligero ejemplo. En un futuro se mejoraría para ofrecer una buena interfaz al usuario que fuera intuitiva y accesible.

Aquí se incluye el enlace al cuarto Sprint: [Sprint 4](#)

A continuación se muestra la lista de *Issues* correspondiente al cuarto Sprint.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementar las descargas de los bancos de preguntas. <span>enhancement</span>	
#18 by mgm1027 was closed on Apr 17			
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementar llamadas POST para la generación de banco de preguntas <span>enhancement</span>	
#17 by mgm1027 was closed on Apr 17			
<input type="checkbox"/>	<input checked="" type="checkbox"/>	implementación de flask y página base <span>enhancement</span>	
#16 by mgm1027 was closed on Apr 17			
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Creación de entorno virtual <span>enhancement</span>	
#15 by mgm1027 was closed on Apr 2			

Figura A.4: *Issues Sprint 4*

## Sprint 5: (10/04/2024 - 24/04/2024)

Para la realización del quinto Sprint se estudió las diferentes formas de mejorar la interfaz de usuario. Todo esto debido a que el diseño y la estructura de la página web correspondiente al proyecto todavía se encontraba en un estado inicial y bastante básico que solo facilitaba el funcionamiento de los métodos para la generación de cuestionarios.

Durante este Sprint se realizó un enfoque más visual con el objetivo de aportar más información acerca de los algoritmos de ordenación utilizados para este proyecto; esto debido a que se tiene en cuenta el enfoque didáctico en el que está enfocado este trabajo. Para ello, se recabó diferentes datos para la comprensión de los algoritmos y se facilitó enlaces para la obtención de mayor información en forma de imagen.

Finalmente, se optó por incluir los enlaces al repositorio del proyecto para cualquier usuario interesado en el desarrollo del mismo. También se

incluyeron enlaces a la página oficial de la universidad. Además se implementó un nuevo logo para representar al proyecto de forma visual y más profesional.

Aquí se incluye el enlace al quinto Sprint: **Sprint 5**

A continuación se muestra la lista de *Issues* correspondiente al quinto Sprint.

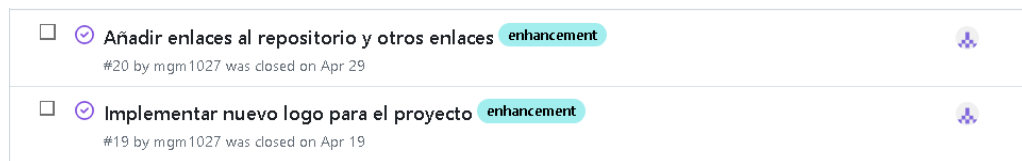


Figura A.5: *Issues Sprint 5*

## Sprint 6: (25/04/2024 - 09/05/2024)

Durante el sexto sprint se tuvo en cuenta la necesidad de añadir un nuevo algoritmo de ordenación. La razón principal era para aportar mayor variedad de nuevas preguntas además de ofrecer otros diferentes recursos de aprendizaje para aquellos que se examinaran.

Como nuevo algoritmo, se decidió incluir el *Heapsort* u ordenación por montículos, el cual permite realizar un enfoque más visual para el alumno. Todo ello teniendo en cuenta que este método se puede representar en formato de árbol o gráfico.

Además de la inclusión del nuevo algoritmo, fue necesario recabar información para poder añadir variedad a los métodos anteriores y por consiguiente generar nuevas preguntas que se diferencien de las anteriores. Por esta razón se incluyó una nueva pregunta sobre el número de comparaciones que produce el algoritmo *Mergesort*, y otra sobre la posición del pivote correspondiente al algoritmo *Quicksort*.

Para concluir con el Sprint, se barajó la idea de implementar imágenes en los cuestionarios generadas a partir de los datos aleatorios producidos en cada pregunta. Dichas imágenes se implementaron en forma de gráfico de barras. El objetivo era visualizar de forma sencilla como iban variando los datos de los vectores a medida que cada algoritmo los iba modificando.



Gracias a esta nueva implementación se pudo aportar nuevas preguntas para los algoritmos ya estructurados previamente.

Aquí se incluye el enlace al sexto Sprint: **Sprint 6**

A continuación se muestra la lista de *Issues* correspondiente al sexto Sprint.








<input type="checkbox"/>	<input checked="" type="checkbox"/> Generar pregunta con imágenes generadas (quicksort). <b>enhancement</b>	
	#27 by mgm1027 was closed on May 8	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Generar pregunta con imágenes generadas (mergesort). <b>enhancement</b>	
	#26 by mgm1027 was closed on May 8	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Incluir pregunta sobre la posición del índice (Quicksort). <b>enhancement</b>	
	#25 by mgm1027 was closed on May 3	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Implementar pregunta Mergesort sobre el número de comparaciones producido. <b>enhancement</b>	
	#24 by mgm1027 was closed on May 3	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Implementar el algoritmo heapsort a la interfaz de usuario y frontend (página web) <b>enhancement</b>	
	#23 by mgm1027 was closed on May 1	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Añadir las preguntas aleatorias base para el algoritmo heapsort <b>enhancement</b>	
	#22 by mgm1027 was closed on May 1	
<input type="checkbox"/>	<input checked="" type="checkbox"/> Implementar la ejecución del algoritmo heapsort <b>enhancement</b>	
	#21 by mgm1027 was closed on May 1	

Figura A.6: *Issues Sprint 6*

## Sprint 7: (10/05/2024 - 22/05/2024)

Para el séptimo Sprint, fue necesario revisar y volver a implementar en nuevo algoritmo incluido en el anterior Sprint. Había que tener en cuenta que dicho método de ordenación ya estaba incluido en la librería del lenguaje que se estaba utilizando para el desarrollo del proyecto.











Para poder volver a implementar el algoritmo, se observó el nuevo método y se optó por añadir nuevas preguntas a partir de dos formas similares de actuación del algoritmo en la nueva librería incluida. Puesto que estas dos formas se diferenciaban en que variaban ligeramente la ordenación de los vectores, fue necesario incluir nuevos métodos para la generación de imágenes en las nuevas preguntas que visualizaran correctamente los árboles que se construyen al utilizar el algoritmo. Puesto que la librería para la generación de imágenes no incluía métodos sencillos para la creación de imágenes de árboles binarios, se tuvo que implementar métodos que rediseñaran la imagen con el fin de visualizar un árbol de forma correcta.

Posteriormente se revisó el contenido de la página web y se optó por ofrecer más opciones de edición al usuario en los formularios. Todo ello con el fin de obtener los datos necesarios para la generación de los cuestionarios aleatorios. Por tanto, se implementó la posibilidad de alterar los rangos de longitud de los vectores que se podrían generar en las preguntas. Además, se dejó a elección del usuario la posibilidad de seleccionar el tipo de cuestiones que deseara para su cuestionario.

Finalmente, se requirió revisar una serie de errores de compilación de las imágenes que se estaban generando en los cuestionarios a la hora de generarlas desde la página web. Por otra parte, debido a que el formulario se había extendido se necesitó reajustar el *footer* de la página web puesto que se encontraba en una posición errónea. Por último, se optó por actualizar el diseño de la barra de navegación incluida en la página web para la selección de los distintos algoritmos.

Aquí se incluye el enlace al séptimo Sprint: [Sprint 7](#)

A continuación se muestra la lista de *Issues* correspondiente al séptimo Sprint.

<input type="checkbox"/>	<input checked="" type="radio"/>	Induir la posibilidad de elegir las preguntas que queremos incluir en el cuestionario desde el formulario. <b>enhancement</b>	
		#37 by mgm1027 was closed 3 weeks ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	Induir en los formularios la posibilidad de elegir rango de los vectores. <b>enhancement</b>	
		#36 by mgm1027 was closed 3 weeks ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	Recolocar el footer de la página web. <b>bug</b> <b>enhancement</b>	
		#35 by mgm1027 was closed 3 weeks ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	Solucionar error sobre la carga de imágenes de matplotlib desde el backend en la página web. <b>bug</b>	
		#33 by mgm1027 was closed 3 weeks ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	Actualizar la barra de navegación del front a una desplegable. <b>enhancement</b>	
		#34 by mgm1027 was closed 3 weeks ago	
<input type="checkbox"/>	<input checked="" type="radio"/>	Corregir el uso de heapify a la biblioteca de python. <b>bug</b> <b>enhancement</b>	
		#30 by mgm1027 was closed last month	
<input type="checkbox"/>	<input checked="" type="radio"/>	Corregir preguntas sobre la ordenación por montículos para la implementación de la librería heapq. <b>bug</b> <b>enhancement</b>	
		#31 by mgm1027 was closed last month	
<input type="checkbox"/>	<input checked="" type="radio"/>	Implementar cuestión sobre los diferentes métodos de ordenación por montículos (heapify y heappush). <b>enhancement</b>	
		#32 by mgm1027 was closed last month	
<input type="checkbox"/>	<input checked="" type="radio"/>	Implementación método para generar imágenes de árboles binarios. <b>enhancement</b>	
		#28 by mgm1027 was closed last month	
<input type="checkbox"/>	<input checked="" type="radio"/>	Implementar pregunta sobre imágenes de árboles binarios para la representación del algoritmo de ordenación por montículos (heapsort). <b>enhancement</b>	
		#29 by mgm1027 was closed last month	

[Sign in now](#) to use Zenhub

Figura A.7: *Issues Sprint 7*

## Sprint 8: (23/05/2024 - 6/06/2024)

Para este último Sprint, se trató principalmente de incluir los últimos retoques de la aplicación y de solucionar alguno de los fallos que habían surgido.

Se incluyeron nuevas preguntas para el algoritmo *Heapsort* con el fin de abarcar los dos pasos que son necesarios para realizar la ordenación. Se incluyó una pregunta sobre la construcción del montículo de máximos, y otra sobre la extracción del valor máximo y reconstrucción del montículo.

Se solucionó varios fallos con respecto a los datos utilizados durante las preguntas. Principalmente se revisó la posibilidad de incluir valores repetidos en los vectores, ya que suponía un problema para desarrollar las preguntas de forma intuitiva.

Finalizando con el diseño del software, se añadió una nueva página de información sobre el proyecto en la aplicación web, y se actualizaron algunos iconos e imágenes.

Para concluir, se realizó una refactorización y se incluyó el archivo *Dockerfile* para poder desplegar la aplicación en un servidor sin requerir otro tipo de implementaciones.

Aquí se incluye el enlace al Octavo Sprint: [Sprint 8](#)

A continuación se muestra la lista de *Issues* correspondiente al octavo Sprint.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Refactorizar el código y añadir comentarios	documentation	
		enhancement		
		#45 by mgm1027 was closed 3 days ago		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementación de un favicon para la aplicación web.	enhancement	
		#44 by mgm1027 was closed 3 days ago		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Actualizar más información.	enhancement	
		#43 by mgm1027 was closed last week		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementación de Dockerfile.	enhancement	
		#41 by mgm1027 was closed last week		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementación de página con más información del proyecto.	bug	
		enhancement		
		#42 by mgm1027 was closed last week		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Generar vectores aleatorios con valores no repetidos	bug	
		enhancement		
		#40 by mgm1027 was closed last week		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementar pregunta tras extraer el máximo de un montículo (Heapsort).	enhancement	
		#39 by mgm1027 was closed 3 weeks ago		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implementar pregunta de creación de montículos máximos.	bug	
		enhancement		
		#38 by mgm1027 was closed 3 weeks ago		

Figura A.8: Issues Sprint 8

## A.3. Estudio de viabilidad

### Viabilidad económica

Este apartado está destinado a realizar una revisión de los beneficios que se podrían obtener en un caso hipotético respecto a los costes que requeriría el desarrollo de este proyecto. Todo esto a tener en cuenta desde una perspectiva empresarial valorando los costes tanto de personal como de material utilizado.

En cuanto a los **costes materiales**, se deben tener en cuenta todos aquellos elementos utilizados para la finalización del proyecto. Siempre desde

el ámbito empresarial y obviando aquellos productos como el software libre que se puede utilizar de forma gratuita.

A continuación se van a tener en cuenta los gastos variables que se podrían haber estado acarreando durante los 5 meses de trabajo (desde febrero hasta junio). Podemos encontrar gastos de Internet, alquiler de bienes de empresa (como el alquiler del local o edificio, material de oficina, etc). También se deben considerar como gastos variables las amortizaciones consideradas en la [A.1](#) Tabla A.1, por ejemplo para el caso del ordenador teniendo en cuenta que se amortiza en 6 años entonces:

Materiales	Valor	Amortización en
Ordenador portátil + periféricos	1.100,00€	6 años
Costes indirectos	90,00€	1 año
Microsoft Windows 10 Home Plus	117,99€	6 años

Tabla A.1: Lista de materiales utilizados con un valor aproximado y su amortización.

**1100€ / 6 años = 183,34€/año** Teniendo en cuenta que es un proyecto de 5 meses la amortización finalmente será de (ver Tabla A.2):

**183,34€/año x 1 año/12 meses x 5 meses = 76,39€** En cuanto al coste de personal, se debe tener en cuenta las personas que han formado parte del proyecto. Para ello, sabiendo que el proyecto se ha llevado a cabo por un solo ingeniero informático, con un hipotético salario en bruto de aproximadamente de 27.000€ anuales [\[3\]](#). Si se han invertido durante los 5 meses de trabajo una media de 220 horas podemos suponer que las horas cubiertas en cada semana son:

Materiales	Coste Amortizado
Ordenador portátil + periféricos	76,39€
Costes indirectos	37,50€
Microsoft Windows 10 Home Plus	8,20€
<b>TOTAL</b>	<b>122,09€</b>

Tabla A.2: Costes de amortización de los materiales.

**220 h / (4 semanas/mes x 5 meses) = 11 h/semana** Si se supone que el salario bruto que se percibe es de 13,85€/hora calculamos:

**13,85€/h x 11 h/semana x 4 semanas/mes = 609,4€/mes** En cuanto a los costes que le supone el salario del personal a la empresa se deben tener en cuenta:

- Seguridad social [4]. 31,15 % del salario neto.
  - 23,60 % contingencias comunes.
  - 5,50 % desempleo (caso de tipo general).
  - 1,25 % cotización de accidentes de trabajo y enfermedades profesionales.
  - 0,60 % formación profesional.
  - 0,20 % FOGASA.
- IRPF [5]. 15,06 % del salario neto.

**609,4€/mes / (1 - 0,1506 - 0,3115) = 1.132,92€/mes** Por lo tanto, teniendo en cuenta los costes materiales y de personal, el coste total del proyecto ascendería a un total de:

Concepto	Costes
Salario anual en bruto	27.000,00€
IRPF	4065,80€
Seguridad Social	8.410,50€
<b>TOTAL (Durante los 5 meses)</b>	<b>5.664,60€</b>

Tabla A.3: Desglose del coste de personal.

**5.664,60€ + 122,09€ = 5.786,69€** Con respecto a los beneficios, no se dispone de ningún beneficio puesto que la Universidad de Burgos proporciona este software de manera gratuita a sus miembros. Además, se tiene en cuenta que este proyecto está destinado a facilitar la docencia y facilitar un ambiente didáctico a sus alumnos.

## Viabilidad legal

Este apartado tratará de mostrar el proceso de desarrollo del proyecto desde un marco legal para su correcta distribución, modificación y uso. Hay que resaltar que se han utilizado principalmente licencias de uso libre de terceros cuya finalidad no es comercial sino destinadas para el desarrollo de proyectos o para el uso de estudiantes.

Es el caso de la licencia correspondiente a **Microsoft Visual Studio Code**, ya que en el contexto de haber sido utilizada por una única persona entra en el ámbito de licencia de uso libre. Sin embargo, si se hubiera tratado de una empresa se tendría que desarrollar según la iniciativa *Open Source* para poder continuar con la licencia de uso gratuito. En el caso contrario se recurriría a la licencia de pago de **formato empresarial**.

Aquí se muestran a su vez algunas de las licencias de uso de otras herramientas utilizadas durante el desarrollo( **A.4**ver Tabla A.4). Donde se puede observar que las licencias de uso que utilizan nos ofrecen un uso libre y sin restricciones para el desarrollo de nuestro proyecto.

Herramientas	Versión	Licencia
Flask	3.0.2	BSD
Matplotlib	3.8.4	PSF
Numpy	1.26.4	BSD
Pillow	10.3.0	PIL

Tabla A.4: Licencias correspondientes a las herramientas utilizadas.



## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

Para la especificación de requisitos, se van a definir los objetivos que se han requerido para la construcción del proyecto así como los casos de uso que posteriormente se han realizado para cumplir con dichos objetivos.

Para definir los requisitos se utilizará el estándar IEEE 830-1998 donde se define que la Especificación de Requisitos del Software (SRS) debería ser:

- **Correcta.** Puesto que todos los requisitos declarados deben haberse cumplido en el desarrollo del software.
- **Completa.** Siempre y cuando todos los requisitos estén vinculados con la aplicación y todos los datos aportados estén referenciados.
- **Inequívoca.** De forma que no haya una interpretación errónea de los requisitos y estén declarados de forma clara y concisa.
- **Consistente.** Por lo que no deben contradecirse con otros requerimientos ya establecidos de mayor importancia.
- **Trazable.** Haciendo alusión a que se debe aportar identificadores que permitan determinar los requisitos que se han empleado en el software a través de la documentación.
- **Modificable.** De forma que conserven siempre la estructura a pesar de rediseñar cualquier cambio en la especificación de requisitos.

- **Verificable.** Se debe poder confirmar que todos y cada uno de los requisitos están establecidos.
- **Identificable.** Todos los requisitos deben poder identificarse de forma clara y aportando indicadores que lo identifiquen para que otros puedan revisarlo en futuros proyectos.

## B.2. Objetivos generales

Como se adjuntó anteriormente en la memoria, los objetivos principales del proyecto son los siguientes:

1. Desarrollar una aplicación web que permita la generación de archivos en formato XML Moodle.
2. Generar preguntas en formato XML Moodle para la inclusión de cuestionarios sobre algoritmos de ordenación.
3. Aportar datos aleatorios para las diferentes preguntas que se desean elaborar e incluir durante la generación de los cuestionarios.
4. Implementar diferentes algoritmos de ordenación para procesar correctamente los datos aleatorios que se generen para las diferentes preguntas de los cuestionarios.

## B.3. Catálogo de requisitos

En este apartado se van a especificar los requisitos que se han estructurado a partir de los objetivos principales.

### Requisitos funcionales:

- **RF-1.** El usuario debe ser capaz de elegir entre crear un cuestionario con preguntas para el algoritmo Mergesort, Quicksort o Heapsort.
- **RF-2.** El usuario debe ser capaz de elegir el número de preguntas, el rango de longitud de los vectores aleatorios y las diferentes preguntas desea incluir en su cuestionario.
- **RF-3.** La aplicación debe poder generar diferentes preguntas sobre el algoritmo de ordenación Mergesort.

- **RF-3.1.** La aplicación debe ser capaz de realizar correctamente el algoritmo Mergesort.
- **RF-3.2.** La aplicación debe ser capaz de generar preguntas sobre la primera llamada del algoritmo Mergesort antes del último proceso de mezcla.
- **RF-3.3.** La aplicación debe ser capaz de generar preguntas sobre el algoritmo Mergesort en formato de selección múltiple.
- **RF-3.4.** La aplicación debe ser capaz de generar preguntas sobre el número de comparaciones que realiza el algoritmo Mergesort sobre un determinado vector.
- **RF-3.5.** La aplicación debe ser capaz de generar imágenes gráficas para cada vector de las preguntas sobre el algoritmo Mergesort.
- **RF-4.** La aplicación debe poder generar diferentes preguntas sobre el algoritmo de ordenación Quicksort.
  - **RF-4.1.** La aplicación debe ser capaz de realizar correctamente el algoritmo Quicksort de forma unidireccional.
  - **RF-4.2.** La aplicación debe ser capaz de realizar correctamente el algoritmo Quicksort de forma bidireccional.
  - **RF-4.3.** La aplicación debe ser capaz de generar preguntas sobre el primer proceso de partición del Quicksort.
  - **RF-4.4.** La aplicación debe ser capaz de crear preguntas sobre la posición del pivote tras realizar el primer proceso de partición del Quicksort.
  - **RF-4.5.** La aplicación debe ser capaz de generar imágenes gráficas para cada vector de las preguntas sobre el algoritmo Quicksort.
- **RF-5.** La aplicación debe poder generar diferentes preguntas sobre el algoritmo de ordenación Heapsort.
  - **RF-5.1.** La aplicación debe ser capaz de generar imágenes de árboles binarios para cada vector de las preguntas sobre el algoritmo Heapsort.
  - **RF-5.2.** La aplicación debe poder generar preguntas sobre el método de ordenación del algoritmo Heapsort denominado heapify.
  - **RF-5.3.** La aplicación debe poder generar preguntas sobre el método de ordenación del algoritmo Heapsort denominado heap-push.

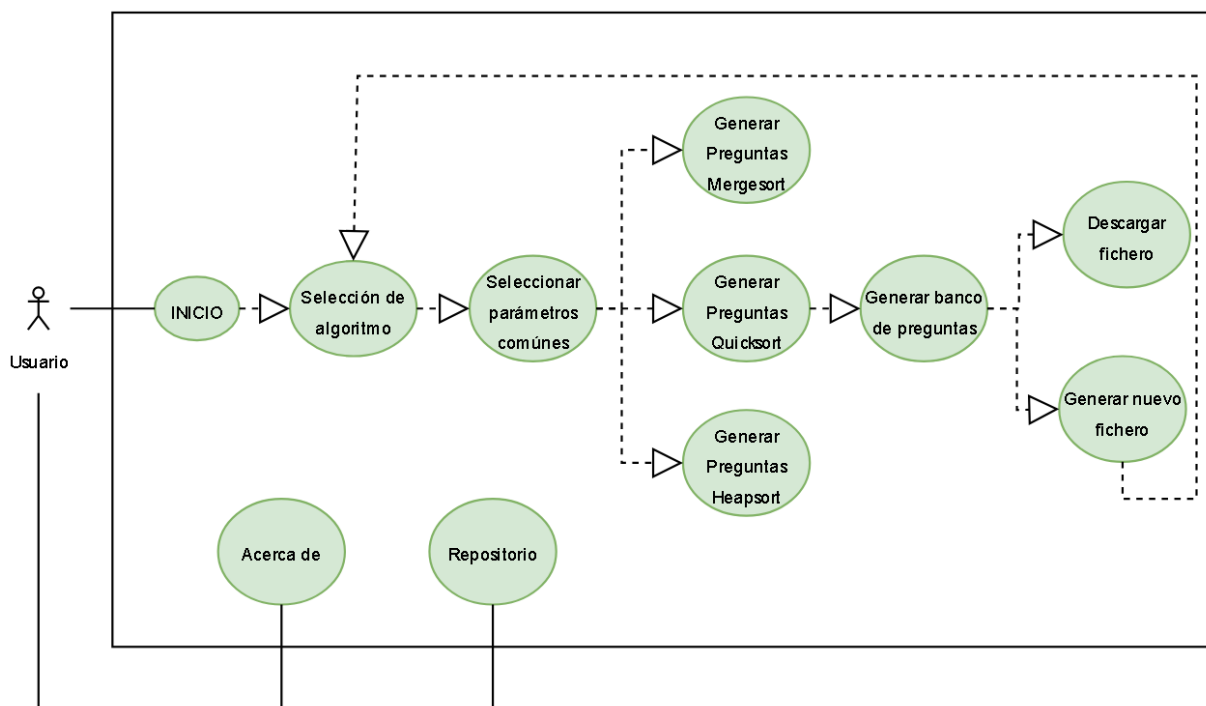
- **RF-5.4.** La aplicación debe poder crear preguntas sobre el primer proceso de creación de montículos máximos del algoritmo Heapsort.
- **RF-5.5.** La aplicación debe poder crear preguntas sobre la extracción valores máximos de los montículos del algoritmo Heapsort.
- **RF-6.** La aplicación debe generar un nuevo fichero de preguntas con los parámetros seleccionados.
- **RF-7.** El usuario debe poder descargar el fichero generado en formato a XML para su posterior visualización en Moodle.
- **RF-8.** El usuario debe poder generar un nuevo fichero distinto al ya generado.
- **RF-9.** El usuario debe poder acceder en cualquier momento a la página inicial de la aplicación web.
- **RF-10.** El usuario debe poder acceder al repositorio de GitHub desde la página web.
- **RF-11.** La aplicación debe mantener la información sobre el proyecto accesible desde cualquier página de la web.

### Requisitos no funcionales:

- **RNF-1.** La aplicación debe ser intuitiva y fácilmente accesible.
- **RNF-2.** La aplicación debe consumir pocos recursos.
- **RNF-3.** La aplicación debe ser capaz de corregir correctamente cada banco de preguntas.
- **RNF-4.** La aplicación debe ser capaz de crecer e incluir nuevas opciones sin provocar errores.
- **RNF-5.** La aplicación debe estar disponible para todos aquellos usuarios con acceso a internet.

## B.4. Especificación de requisitos

A continuación se va a proceder a asociar los requisitos definidos en el apartado anterior con los casos de uso correspondientes a la aplicación web (B.1Figura B.1).

Figura B.1: *Casos de uso.*

CU-1	Acceso a la página principal
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-9
<b>Descripción</b>	Permitir al usuario acceder a la página principal desde cualquier punto la aplicación web.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe acceder a cualquier otra página y pulsar el icono principal de la página para volver al inicio.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe acceder a cualquier otra página de la web (ya sea 'Acerca de', 'Merge-sort', 'Quicksort' o 'Heapsort').</li> <li>2. El usuario debe pulsar sobre el logo principal de la página.</li> </ol>
<b>Postcondición</b>	Se accede a la página principal de la aplicación web.
<b>Excepciones</b>	No se puede acceder a la página principal o a ninguna de las otras páginas.
<b>Importancia</b>	Baja

Tabla B.1: CU-1 Caso de uso 1.

CU-2	Seleccionar tipo de algoritmo
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-1
<b>Descripción</b>	Ofrecer al usuario la posibilidad de seleccionar el tipo de algoritmo de ordenación del que desea realizar un banco de preguntas.
<b>Precondición</b>	Haber accedido a la página inicial de la aplicación web.
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. El usuario debe pulsar sobre la barra de navegación disponible.</li><li>2. Al desplegarse la barra, el usuario debe pulsar sobre uno de los algoritmos a elegir (Mergesort, Quicksort, Heapsort) redirigiéndole a la página correspondiente.</li></ol>
<b>Postcondición</b>	Se accede a la página correspondiente del algoritmo seleccionado.
<b>Excepciones</b>	La barra de navegación no se despliega.
<b>Importancia</b>	Media

Tabla B.2: CU-2 Caso de uso 2.

<b>CU-3</b>	<b>Seleccionar parámetros comunes.</b>
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-2
<b>Descripción</b>	Ofrecer al usuario la posibilidad de seleccionar número de preguntas y el rango de las longitudes aleatorias de los vectores
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe haber elegido entre uno de los algoritmos a elegir previamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe pulsar sobre generar un banco de preguntas en la página correspondiente al algoritmo.</li> <li>2. El usuario debe poder rellenar los parámetros: número de preguntas, rango mínimo de longitud del vector y rango máximo de longitud del vector</li> </ol>
<b>Postcondición</b>	La aplicación no debe permitir que el número de preguntas sea un número menor que 1, que el rango mínimo de longitud del vector sea menor que 2 y mayor de 21, ni que el rango máximo de longitud del vector sea inferior al mínimo y superior a 22
<b>Excepciones</b>	No se puede acceder al formulario o no aparecen los campos del formulario.
<b>Importancia</b>	Alta

Tabla B.3: CU-3 Caso de uso 3.



CU-4	Generar Preguntas Mergesort
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-3
<b>Descripción</b>	Ofrecer la posibilidad al usuario de seleccionar y generar un banco de preguntas aleatorio cuyas cuestiones traten sobre el algoritmo de ordenación Mergesort.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe haber seleccionado generar un banco de preguntas del algoritmo Mergesort además de haber rellenado correctamente los parámetros comunes y de haber marcado al menos una de las preguntas a elegir.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe pulsar sobre generar un banco de preguntas en la página sobre el algoritmo Mergesort.</li> <li>2. El usuario debe rellenar correctamente los parámetros comunes de todos los algoritmos.</li> <li>3. El usuario debe seleccionar al menos una de las preguntas que se pueden visualizar en el formulario.</li> </ol>
<b>Postcondición</b>	El formulario debe asegurarse que se ha marcado al menos una pregunta a incluir.
<b>Excepciones</b>	No se visualizan las imágenes de las preguntas.
<b>Importancia</b>	Alta

Tabla B.4: CU-4 Caso de uso 4.

CU-5	Generar Preguntas Quicksort
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-4
<b>Descripción</b>	Ofrecer la posibilidad al usuario de seleccionar y generar un banco de preguntas aleatorio cuyas cuestiones traten sobre el algoritmo de ordenación Quicksort.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe haber seleccionado generar un banco de preguntas del algoritmo Quicksort además de haber rellenado correctamente los parámetros comunes y de haber marcado al menos una de las preguntas a elegir.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe pulsar sobre generar un banco de preguntas en la página sobre el algoritmo Quicksort.</li> <li>2. El usuario debe rellenar correctamente los parámetros comunes de todos los algoritmos.</li> <li>3. El usuario debe seleccionar al menos una de las preguntas que se pueden visualizar en el formulario.</li> </ol>
<b>Postcondición</b>	El formulario debe asegurarse que se ha marcado al menos una pregunta a incluir.
<b>Excepciones</b>	No se visualizan las imágenes de las preguntas.
<b>Importancia</b>	Alta

Tabla B.5: CU-5 Caso de uso 5.

CU-6	Generar Preguntas Heapsort
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	R5
<b>Descripción</b>	Ofrecer la posibilidad al usuario de seleccionar y generar un banco de preguntas aleatorio cuyas cuestiones traten sobre el algoritmo de ordenación Heapsort.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe haber seleccionado generar un banco de preguntas del algoritmo Heapsort además de haber rellenado correctamente los parámetros comunes y de haber marcado al menos una de las preguntas a elegir.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe pulsar sobre generar un banco de preguntas en la página sobre el algoritmo Heapsort.</li> <li>2. El usuario debe rellenar correctamente los parámetros comunes de todos los algoritmos.</li> <li>3. El usuario debe seleccionar al menos una de las preguntas que se pueden visualizar en el formulario.</li> </ol>
<b>Postcondición</b>	El formulario debe asegurarse que se ha marcado al menos una pregunta a incluir.
<b>Excepciones</b>	No se visualizan las imágenes de las preguntas.
<b>Importancia</b>	Alta

Tabla B.6: CU-6 Caso de uso 6.

<b>CU-7</b>	<b>Generar banco de preguntas</b>
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-6
<b>Descripción</b>	La aplicación generará un fichero en formato XML con datos aleatorios siguiendo las pautas determinadas en el formulario.
<b>Precondición</b>	El usuario debe haber accedido a una de las páginas correspondientes de un algoritmo de ordenación y debe de haber rellenado correctamente el formulario del mismo.)
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe completar correctamente el formulario del algoritmo correspondiente.</li> <li>2. El usuario debe pulsar sobre la opción 'Generar fichero.</li> </ol>
<b>Postcondición</b>	Se debe generar correctamente el fichero XML del algoritmo concreto con los datos determinados en el formulario.
<b>Excepciones</b>	No se genera el fichero XML
<b>Importancia</b>	Alta

Tabla B.7: CU-7 Caso de uso 7.

CU-8	Descargar fichero
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-7
<b>Descripción</b>	La aplicación descargará en el equipo del usuario el fichero XML generado previamente desde el navegador web.
<b>Precondición</b>	El usuario debe haber completado el formulario correctamente y debe haber generado el fichero previamente.
<b>Acciones</b>	<ol style="list-style-type: none"><li>1. El usuario debe haber generado un nuevo fichero tras completar el formulario.</li><li>2. El usuario debe pulsar sobre la opción 'descargar banco de preguntas'.</li></ol>
<b>Postcondición</b>	El fichero XML correspondiente al algoritmo debe de haberse descargado correctamente en el equipo.
<b>Excepciones</b>	No se ha generado el archivo o no se ha descargado.
<b>Importancia</b>	Alta

Tabla B.8: CU-8 Caso de uso 8.

<b>CU-9</b>	<b>Generar nuevo fichero</b>
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-8
<b>Descripción</b>	Ofrecer la posibilidad al usuario de rehacer el fichero generado previamente.
<b>Precondición</b>	El usuario debe haber completado el formulario correctamente y debe haber generado el fichero previamente.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe haber generado un nuevo fichero tras completar el formulario.</li> <li>2. El usuario debe pulsar sobre la opción 'generar nuevo banco de preguntas'.</li> </ol>
<b>Postcondición</b>	Tras pulsar sobre la opción debe redirigirse a la página del formulario correspondiente al algoritmo.
<b>Excepciones</b>	No se redirige a la página del formulario.
<b>Importancia</b>	Baja

Tabla B.9: CU-9 Caso de uso 9.

CU-10	Acceder al repositorio
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-10
<b>Descripción</b>	Ofrecer la posibilidad al usuario de acceder al repositorio de GitHub relativo al proyecto desde cualquier página de la aplicación web.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe acceder a cualquier otra página y pulsar el icono de acceso al repositorio GitHub.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe acceder a cualquier otra página de la web (ya sea 'Acerca de', 'Merge-sort', 'Quicksort' o 'Heapsort').</li> <li>2. El usuario debe pulsar sobre el icono de acceso al repositorio GitHub.</li> </ol>
<b>Postcondición</b>	La aplicación debe redirigir al usuario a la página del repositorio de GitHub del proyecto.
<b>Excepciones</b>	No se puede acceder al repositorio o a ninguna de las otras páginas.
<b>Importancia</b>	Media

Tabla B.10: CU-10 Caso de uso 10.

<b>CU-11</b>	<b>Acceder a más información del proyecto</b>
<b>Versión</b>	1.0
<b>Autor</b>	Mario García Martínez
<b>Requisitos asociados</b>	RF-11
<b>Descripción</b>	Ofrecer la posibilidad al usuario de acceder a la página con más informacion acerca del proyecto.
<b>Precondición</b>	La aplicación debe haberse iniciado y el usuario debe acceder a cualquier otra página y pulsar el icono de acceso a más información
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe acceder a cualquier otra página de la web (ya sea 'Acerca de', 'Merge-sort', 'Quicksort' o 'Heapsort'.</li> <li>2. El usuario debe pulsar sobre el icono 'Acerca de'.</li> </ol>
<b>Postcondición</b>	La aplicación debe redirigir al usuario a la página sobre más información del proyecto
<b>Excepciones</b>	No se puede acceder a la página de más información o a ninguna de las otras páginas.
<b>Importancia</b>	Baja

Tabla B.11: CU-11 Caso de uso 11.



## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En este nuevo apartado se tratarán todos aquellos aspectos relacionados con el diseño del proyecto, el tratamiento de datos con respecto al diseño y su implementación así como su diseño arquitectónico.

### C.2. Diseño de datos

El tratamiento de datos se ha llevado en común para todas las clases principalmente para los siguientes campos:

- **El número de preguntas:** Valor que designa el número de preguntas que se van a generar aleatoriamente de cada tipo de algoritmo.
- **El rango mínimo de la longitud de los vectores:** Corresponde al valor mínimo que pueda adquirir la longitud de un vector que se genere aleatoriamente para una pregunta.
- **El rango máximo de la longitud de los vectores:** Corresponde al valor máximo que pueda adquirir la longitud de un vector que se genere aleatoriamente para una pregunta.

Posteriormente para cada clase de algoritmo se implementa un vector de valores booleanos. Con ello podemos incluir a gusto del usuario las preguntas correspondientes para cada uno de los cuestionarios:

## Clase *Mergesort*

La clase *Mergesort* corresponde con la generación de preguntas sobre el algoritmo *Mergesort* donde cabe destacar los siguientes cuestiones disponibles para incluir en cada cuestionario:

- **Incluir pregunta de seleccionar el vector correcto antes del ultimo proceso de mezcla:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir el vector correcto tras ordenar con *Mergesort* antes del último proceso.
- **Incluir pregunta de completar el vector antes del ultimo proceso de mezcla:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre completar el vector tras ordenar con *Mergesort* antes del último proceso.
- **Incluir pregunta sobre el número de comparaciones del algoritmo:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre el número de comparaciones que se han producido para cada vector durante el uso del algoritmo.
- **Incluir pregunta sobre seleccionar la imagen correspondiente al algoritmo *Mergesort*:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir la imagen correspondiente al vector tras ordenar con *Mergesort* antes del último proceso.

## Clase *Quicksort*

La clase *Quicksort* corresponde con la generación de cuestiones sobre el algoritmo *Quicksort* donde hay que destacar las siguientes preguntas que pueden añadirse a cada cuestionario:

- **Incluir pregunta de seleccionar el vector correcto tras el proceso de partición de forma unidireccional:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir el vector correcto tras ordenar la primera partición del *Quicksort* de forma unidireccional.
- **Incluir pregunta de seleccionar el vector correcto tras el proceso de partición de forma bidireccional:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir

el vector correcto tras ordenar la primera partición del *Quicksort* de forma bidireccional.

- **Incluir pregunta de completar el vector tras el proceso de partición:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre completar el vector tras ordenar la primera partición del *Quicksort* tanto de forma unidireccional como bidireccional.
- **Incluir pregunta sobre la posición del pivote:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre la posición que adquiere el pivote tras producirse la primera partición del *Quicksort*.
- **Incluir pregunta de seleccionar la imagen correspondiente al vector tras el proceso de partición de forma unidireccional:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir la imagen correspondiente al vector tras ordenar la primera partición del *Quicksort* de forma unidireccional.
- **Incluir pregunta de seleccionar la imagen correspondiente al vector tras el proceso de partición de forma bidireccional:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre elegir la imagen correspondiente al vector tras ordenar la primera partición del *Quicksort* de forma bidireccional.

## Clase Heapsort

La clase *Heapsort* es la que determina la generación de cuestiones sobre el algoritmo *Heapsort*, donde hay que destacar las siguientes preguntas que pueden implementarse a cada uno de los cuestionarios:

- **Incluir pregunta sobre seleccionar la correcta implementación del algoritmo *Heapify* en el montículo:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre seleccionar el orden correcto del montículo tras usar el método *Heapify*.
- **Incluir pregunta sobre seleccionar la correcta implementación del algoritmo *Heappush* en el montículo:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre seleccionar el orden correcto del montículo tras usar el método *Heappush*.

- **Incluir pregunta sobre completar la implementación correcta en el montículo tras usar la implementación *Heapify*:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre completar el orden correcto del montículo tras usar el método *Heapify*.
  
- **Incluir pregunta sobre completar la implementación correcta en el montículo tras usar la implementación *Heappush*:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre completar el orden correcto del montículo tras usar el método *Heapify*.
  
- **Incluir pregunta sobre construir el montículo de máximos del algoritmo:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre construir el montículo de máximos completo de forma correcta.
  
- **Incluir pregunta sobre la construcción del montículo de máximos tras extraer el primer máximo:** Permite seleccionar la posibilidad de que se incluya aleatoriamente la pregunta sobre construir el montículo de máximos completo tras haber extraído el primer valor máximo.

### C.3. Diseño procedimental

Apartado destinado a representar la ejecución de los procesos destacables de la aplicación web. En la siguiente figura se muestra el diagrama de secuencias correspondiente a las distintas interacciones que se pueden producir entre el usuario y el sistema para generar un determinado banco de preguntas.

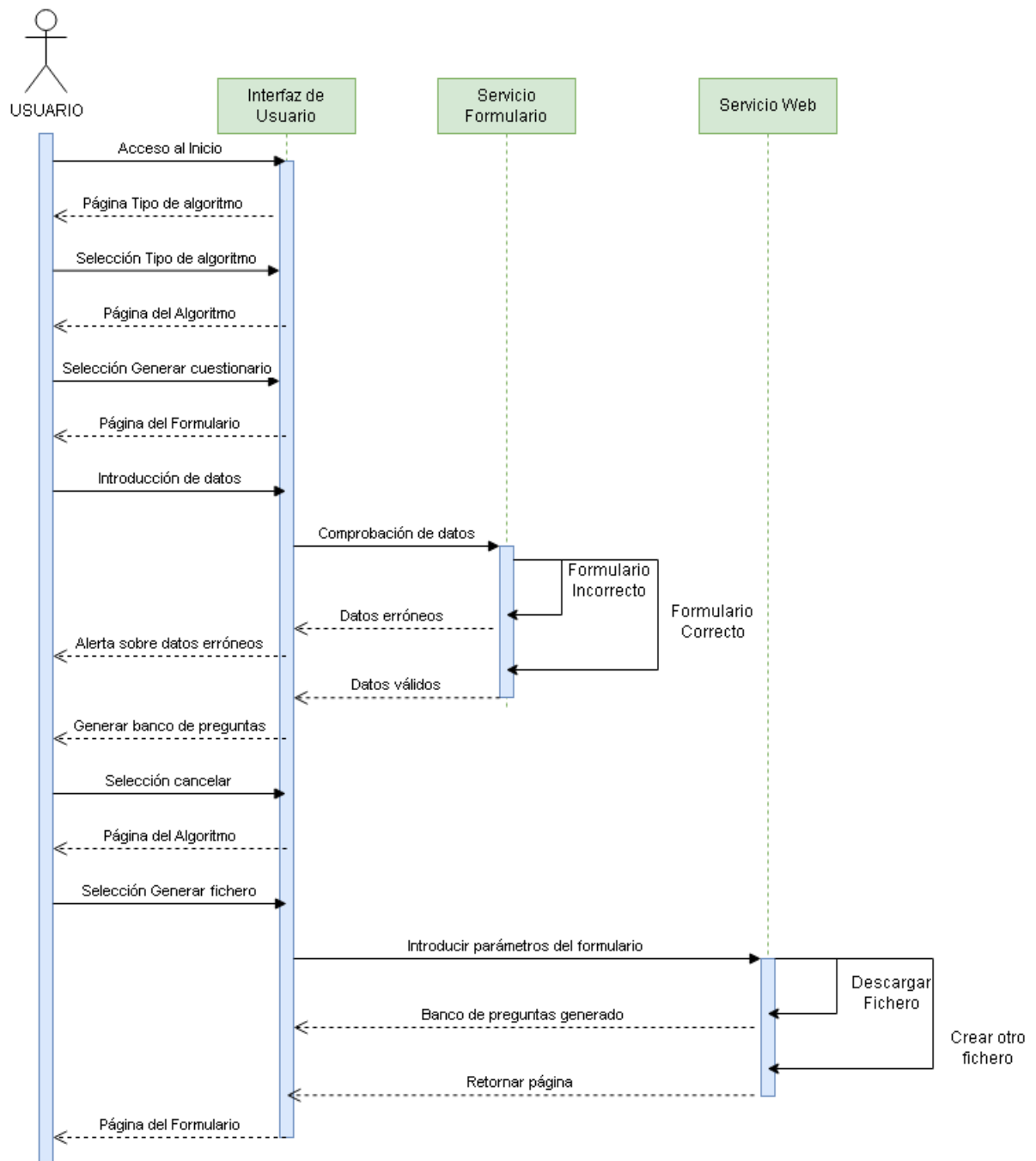


Figura C.1: Diagrama de secuencia.

## C.4. Diseño arquitectónico

En este apartado se detallan los patrones de diseño en los que se ha basado el desarrollo del proyecto para llevar una adecuada implementación del software.

### Modelo-Vista-Controlador

Modelo-Vista-Controlador o MVC es un patrón de diseño de arquitecturas software que se encarga de separar la gestión de datos y la lógica de un programa de su representación gráfica y del módulo que se encarga de gestionar la comunicación entre ambos. Este patrón está por tanto dividido en tres partes distintas y bien definidas que son el modelo, la vista y el controlador. Todo ello con el fin de facilitar la reutilización del código y separar claramente los conceptos para su posterior mantenimiento [1].

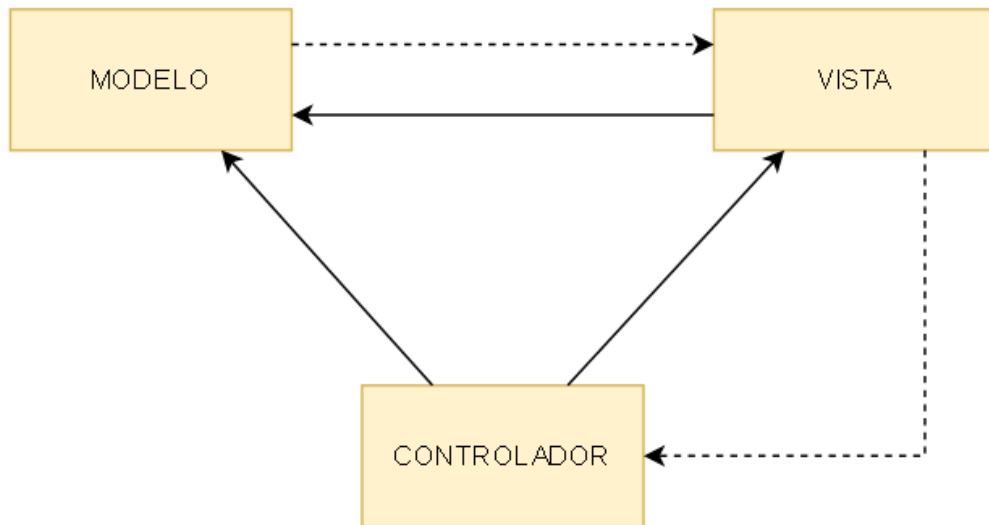


Figura C.2: *Diagrama representativo del patrón Modelo-Vista-Controlador.*

Para el caso de este proyecto cada uno de los componentes de este modelo viene dado por:

- **Modelo:** Se corresponde con la información que gestiona y se opera de la aplicación, es decir, la lógica de la misma. En nuestro caso se corresponde con la estructura de programas que generan tanto los algoritmos de ordenación como sus métodos para la generación del banco de preguntas.
  
- **Vista:** Parte visual de la aplicación correspondiente a las interfaces de usuario. En esta aplicación viene determinada por la estructura de la web como los elementos de estilo e imagen de los archivos CSS y todas las páginas HTML que la componen; elementos que se encuentran almacenados en el directorio */static* y */templates* respectivamente.
  
- **Controlador:** El elemento que se encarga de enviar y recibir las operaciones de los otros elementos del patrón, es decir, controla el tráfico de datos de los otros elementos. Para nuestro proyecto disponemos el archivo *main.py* que se compone de toda la gestión de métodos HTML para el envío y recogida de datos (métodos GET y POST) entre la interfaz de usuario y las clases de los algoritmos.

## Patrón de diseño Fachada (facade)

Este patrón estructural tiene como finalidad estructurar un entorno de programación que a su vez reduzca la complejidad de la división del sistema, todo ello para reducir las dependencias del proyecto. Este patrón se utiliza principalmente cuando se desea proporcionar una interfaz sencilla para un sistema complejo. Todo esto con el fin de facilitar al usuario la accesibilidad a la aplicación y evitar que sea participe de la gestión de operaciones que se lleve a cargo a nivel interno. [2]

Nuestra fachada viene dado por el fichero *main.py*, el cual se encarga de realizar todo el manejo de información entre los datos recogidos a través de la interfaz de usuario y la lógica del sistema.

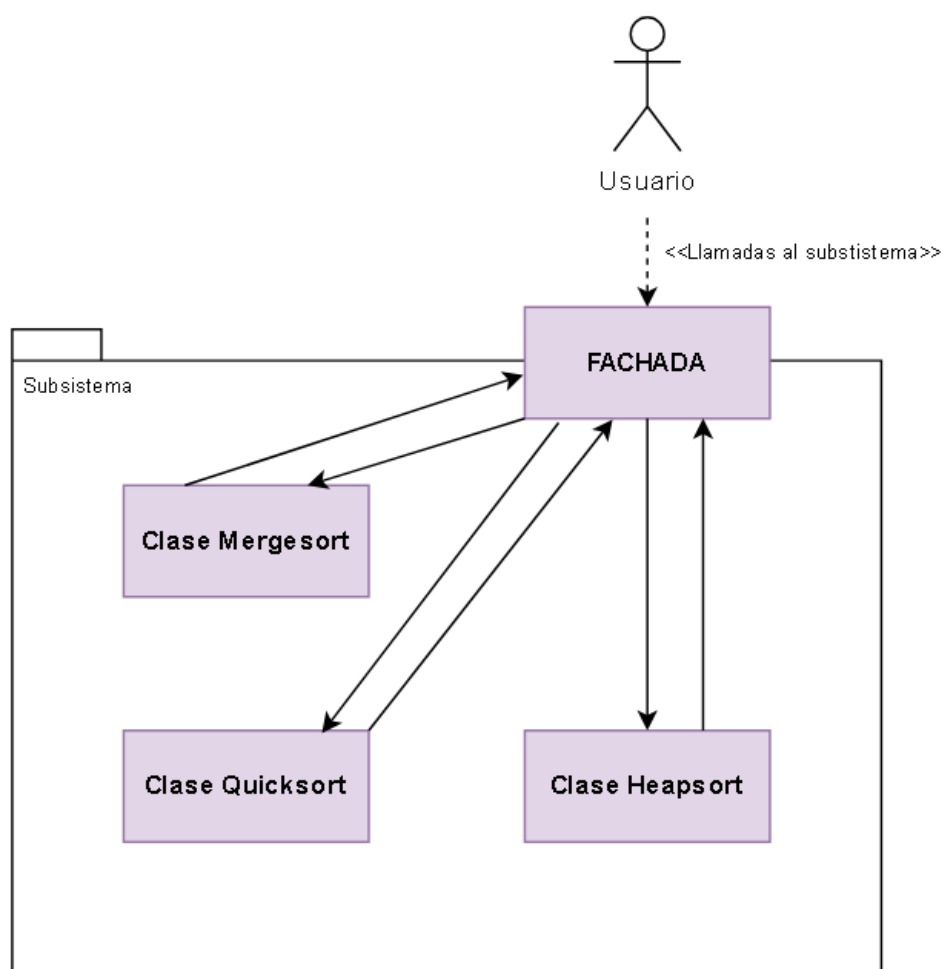


Figura C.3: *Diagrama representativo del patrón de diseño fachada.*



## Apéndice *D*

---

# Documentación técnica de programación

---

### D.1. Introducción

Este apartado está destinado a documentar todo los elementos relacionados con la estructura del código, la distribución de directorios y con cómo se realiza el proceso de descarga del directorio y el proceso de pruebas. Todo ello para facilitar el acceso al proyecto par futuros desarrolladores que quieran implementar nuevas funciones y atributos.

### D.2. Estructura de directorios

A continuación se muestra la estructura de directorios de todo el repositorio del proyecto que se puede encontrar en [GitHub](#).

- **/:** Directorio raíz del repositorio donde se encuentran todos los archivos, directorios y otros elementos del proyecto.
- **/src/main:** Directorios que contienen los archivos correspondientes a las clases para generar los bancos de preguntas que son utilizadas por el archivo `main.py` en nuestro software.
- **/src/main/mergesort.py:** Archivo encargado de la generación de los distintos tipos de preguntas disponibles para el algoritmo Mergesort.
- **/src/main/quicksort.py:** Archivo encargado de la generación de los distintos tipos de preguntas disponibles para el algoritmo Quicksort.

- **/src/main/heapsort.py:** Archivo encargado de la generación de los distintos tipos de preguntas disponibles para el algoritmo Heapsort.
- **/doc:** Directorio que contiene toda la documentación acerca del proyecto.
- **/static:** Directorio donde se alojan todos los archivos encargados de la dinámica de estilos e la aplicación web y su interfaz, y las imágenes que se utilizan en la misma.
- **/static/css:** Directorio donde se encuentra el tipo de archivo .css encargado de designar el estilo de la interfaz de usuario y de los archivos HTML de los que dispone nuestro proyecto.
- **/static/img:** Directorio donde se encuentra las imágenes que se visualizan a través de la interfaz de usuario de nuestro proyecto.
- **/static/js:** Directorio que contiene el tipo de archivo .js que permite añadir algunas funcionalidades a nuestra interfaz de usuario.
- **/static/js/script.js:** Archivo .js que ofrece a la barra de navegación donde se selecciona el tipo de algoritmo de ordenación la posibilidad de hacerla desplegable con el fin de incluir algo de dinamismo a la interfaz de usuario.
- **/templates:** Directorio que almacena los distintos archivos HTML que se mostrarán durante la ejecución de la aplicación web.
- **/LICENSE:** Fichero que muestra la licencia de uso del proyecto.
- **/README.md:** Fichero que contiene una descripción del proyecto y algunos detalles que explican como usarlo de forma correcta.
- **/pregunta\_mergesort.xml:** Se corresponde con el archivo XML que se genera después de que el usuario trate de crear un banco de preguntas para el algoritmo Mergesort por medio de la aplicación web.
- **/pregunta\_quicksort.xml:** Se corresponde con el archivo XML que se genera después de que el usuario trate de crear un banco de preguntas para el algoritmo Quicksort por medio de la aplicación web.
- **/pregunta\_heapsort.xml:** Se corresponde con el archivo XML que se genera después de que el usuario trate de crear un banco de preguntas para el algoritmo Heapsort por medio de la aplicación web.

- **/main.py:** Archivo principal que se encarga de llevar a cabo todas las llamadas al subsistema desde la interfaz de usuario y de recoger los datos obtenidos a través de la lógica del sistema por medio de las llamadas GET y POST de HTML.
- **/requirements.txt:** Archivo que permite la instalación de todos los paquetes necesarios para el correcto funcionamiento del proyecto.
- **/runtime.txt:** Fichero utilizado para desplegar en la nube nuestra aplicación web.
- **/dockerfile:** Fichero que permite generar el contenido de la aplicación y lo requerido para la instalación de la misma en una imagen de Docker que facilite posteriormente su despliegue.

## D.3. Manual del programador

A continuación se explicará el proceso de descarga y e instalación para futuras actualizaciones que se quieran realizar sobre el proyecto. Se va a describir tanto el proceso en formato local para su edición como el despliegue que se ha realizado por medio de la plataforma **Koyeb**.

### Formato Local

Para el despliegue de la aplicación web en local se dispone del **repositorio de GitHub**, desde donde se puede descargar o bien desde la web(ver Figura D.1), o mediante la herramienta Git ejecutando el comando *git clone https://github.com/mgm1027/TFG-Sorting-Algorithm-Quiz-Generation*.

Siempre se pueden realizar cambios para una nueva versión del proyecto realizando un *fork* desde la web de GitHub para llevar un mejor control de las diferentes versiones que vaya adquiriendo el trabajo.

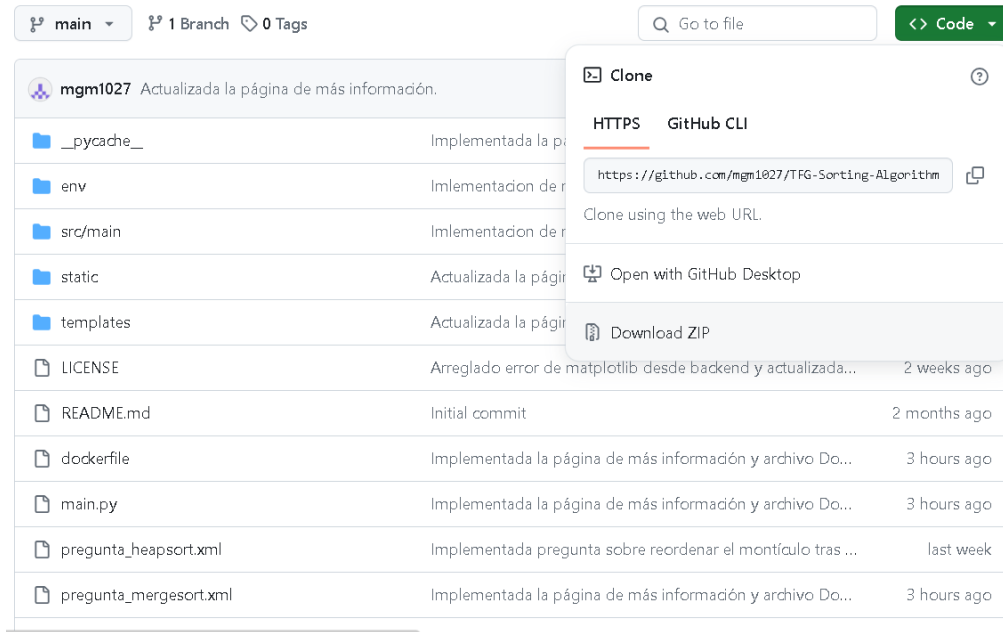


Figura D.1: Descarga del repositorio desde la web de GitHub.

Tras descargar todo el repositorio y una vez abierto desde nuestro editor de proyectos es necesario instalar las librerías para que la aplicación se despliegue correctamente. Para ello esta disponible el fichero *requirements.txt* que facilita la instalación de todas las herramientas. Desde la consola nos situamos en el directorio del proyecto y ejecutamos el siguiente comando:

**pip install -r requirements.txt**

Los componentes que se van instalar para el proyecto son los siguientes:

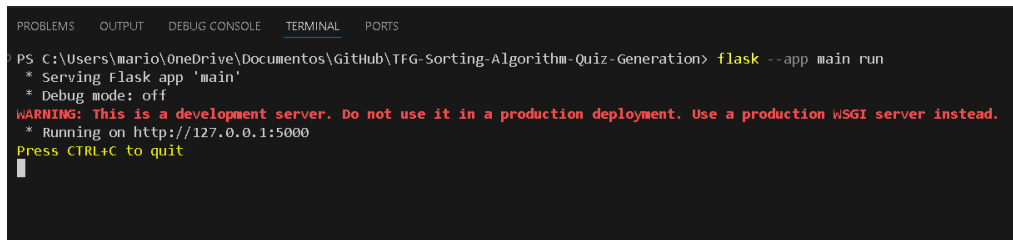
Tras ejecutar el comando podremos lanzar la aplicación web desde el ámbito local. Para ello tendremos que ejecutar en la terminal desde el directorio del proyecto el siguiente comando:

Librerías	Version
Flask	3.0.2
Matplotlib	3.8.4
Numpy	1.26.4
Pillow	10.3.0

Tabla D.1: Licencias correspondientes a las herramientas utilizadas.

### **flask --app main run**

Posteriormente la terminal mostrará la dirección URL donde se ha desplegado el proyecto, la cual debería ser <http://127.0.0.1:5000>. Abrimos el navegador e introducimos dicha url y ya podremos acceder a la web.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mario\OneDrive\Documentos\GitHub\TFG-Sorting-Algorithm-Quiz-Generation> flask --app main run
* Serving Flask app 'main'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Figura D.2: *Lanzamiento en local.*

## Despliegue en Koyeb

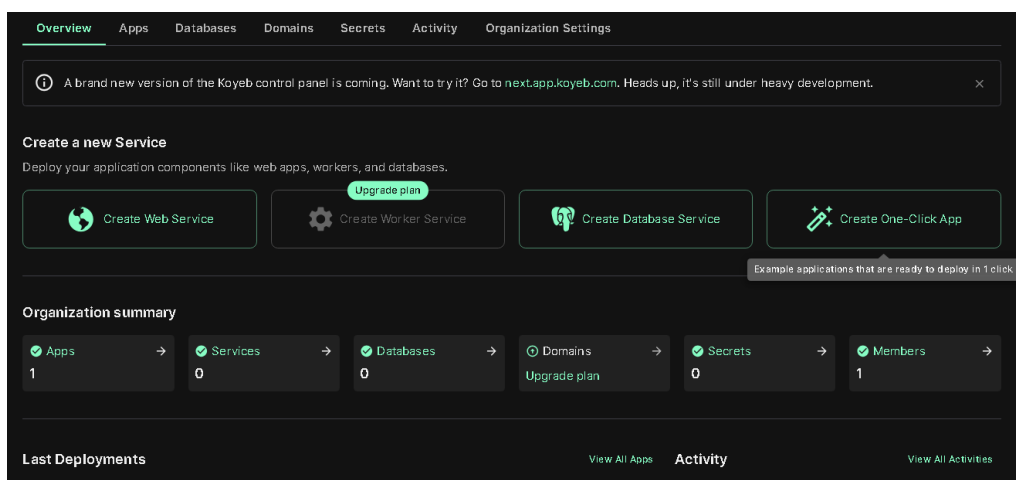
Como se comenta en la memoria, para el despliegue se ha creado una imagen Docker que facilita la implementación desde cualquier equipo. Además se ha utilizado la plataforma Koyeb para realizar el despliegue de la web en un servidor. Para realizar el despliegue es necesario disponer de los archivos:

- **Dockerfile:** Archivo que permite la creación de una imagen docker con nuestro proyecto como base.
- **Requirements.txt:** Necesario para la instalación de las librerías necesarias.

Con estos dos ficheros ya solo será necesario registrarse en la web de [Koyeb](#) y seguir una serie de pasos para el despliegue.

Tras registrarse en la web, procederemos a crear una app seleccionando la opción de crear una app de un solo click. De entre las opciones seleccionamos Flask que es el framework que está implementado en nuestro software.

En la siguiente ventana se seleccionarán los siguientes parámetros:

Figura D.3: *Creando una app desde Koyeb.*

- En el tipo de servicio se dejará el servicio web por defecto.
- En la fuente se copiará URL del repositorio de GitHub y se seleccionará la rama principal por defecto.
- En el constructor se seleccionara la opción Dockerfile para que el servidor pueda generar un contenedor de nuestra imagen Docker.
- En las variables de entorno se dejarán sin configurar puesto que no es necesario.
- En la regiones es recomendable seleccionar el servidor que esté mas cercano pero no es necesario configurarlo.
- En el escalado no es necesario modificar la información.
- En el puerto expuesto es importante indicar que nuestro puerto se trata del 5000 puesto que así lo tenemos expuesto en nuestro software.
- En el nombre del servicio quedará en libre elección para quien lo desee lanzar.

Finalmente solo será necesario esperar a que se despliegue correctamente y acceder a la dirección concreta que nos ofrece Koyeb. En nuestro caso, la dirección generada es: <https://saqg-ubu-aeee3c86.koyeb.app/>

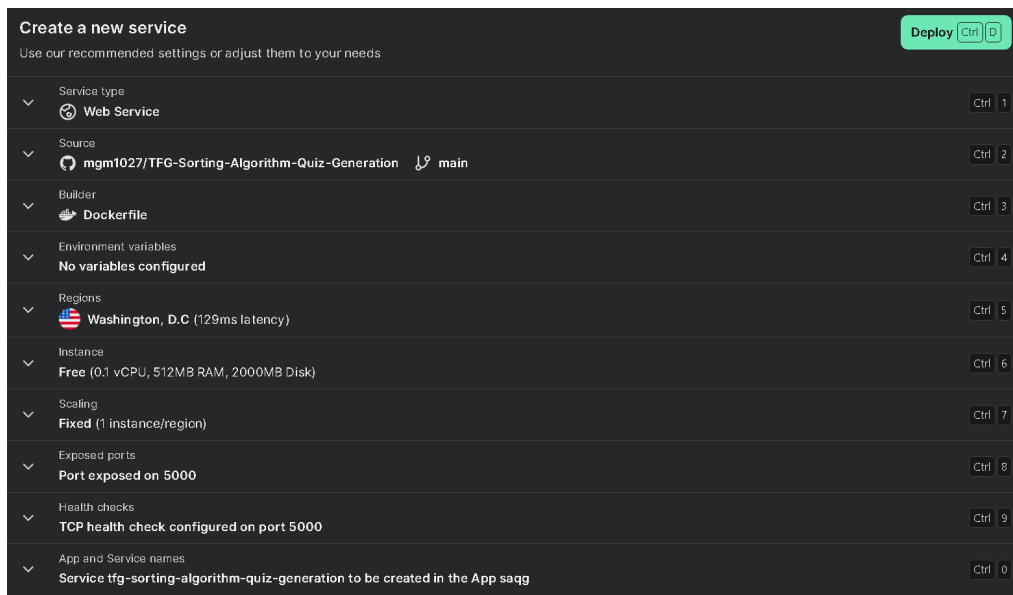


Figura D.4: Configuración del despliegue de la web en Koyeb.

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

*Microsoft Visual Studio* se trata de un editor de software multiplataforma desarrollado por *Microsoft* el cual nos permite instalar extensiones que nos facilitan el desarrollo de código. Para descargar el editor es necesario acceder a la [página oficial](#) donde podrás seleccionar la versión que deseas utilizar para el proyecto.

Posteriormente, solo será necesario abrir el ejecutable y seguir los pasos deseados. Cuando finalice podremos iniciar *Microsoft Visual Studio* y abrir el repositorio desde visual. Para ello, es recomendable descargar el repositorio y seleccionar la opción desde visual de abrir una zona de trabajo desde un archivo. Seleccionamos el archivo raíz de nuestro directorio y finalmente podremos trabajar en el proyecto.

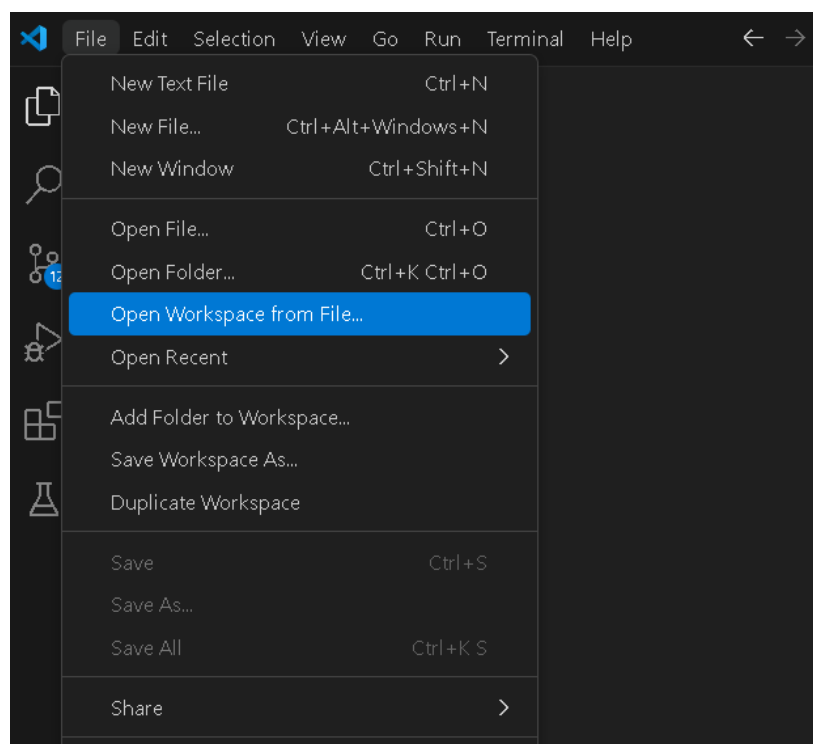


Figura D.5: Selección del repositorio desde el equipo en Visual Studio Code.

## D.5. Pruebas del sistema

En este apartado se hará hincapié en realizar una serie de pruebas. Todo ello con el fin de comprobar que los datos que se introduzcan para generar un cuestionario no conlleven a generar errores en la aplicación.

Se ha definido una serie de pruebas que, por medio de la extensión de navegador de *Katalon Recorder* se han revisado desde la aplicación web del proyecto. Esta extensión permite repetir acciones que ya se han realizado de forma automática, comprobando que dichas acciones se verifican y realizan correctamente.

Las pruebas a realizar han sido las siguientes:

- **El número de preguntas del cuestionario no debe ser inferior a 1.** En esta prueba se comprueba que al introducir un número de



preguntas inferior a 1, el formulario muestra un mensaje de error y no genera el archivo XML.

- **El mínimo de la longitud de los vectores no debe ser menor que 2.** En esta prueba se comprueba que al introducir el rango mínimo de la longitud de los vectores inferior a 2, el formulario muestra un mensaje de error y no genera el archivo XML.
- **El mínimo de la longitud de los vectores no debe ser mayor que 21.** En esta prueba se comprueba que al introducir el rango mínimo de la longitud de los vectores superior a 21, el formulario muestra un mensaje de error y no genera el archivo XML.
- **El máximo de la longitud de los vectores no debe ser menor que el mínimo.** En esta prueba se comprueba que al introducir el rango máximo de la longitud de los vectores inferior al mínimo previamente introducido, el formulario muestra un mensaje de error y no genera el archivo XML.
- **El máximo de la longitud de los vectores no debe ser mayor que 22.** En esta prueba se comprueba que al introducir el rango máximo de la longitud de los vectores superior a 22, el formulario muestra un mensaje de error y no genera el archivo XML.
- **Debe seleccionarse al menos una pregunta para incluir en el cuestionario.** En esta prueba se comprueba que al no seleccionar alguna de las preguntas para incluir en el cuestionario, el formulario muestra un mensaje de error y no genera el archivo XML.
- **Se debe dar la opción de descargar el banco de preguntas si se han introducido los datos correctos.** En esta prueba se comprueba que al introducir los datos del formulario correctamente, este genere el archivo XML y de la opción de descargarlo.

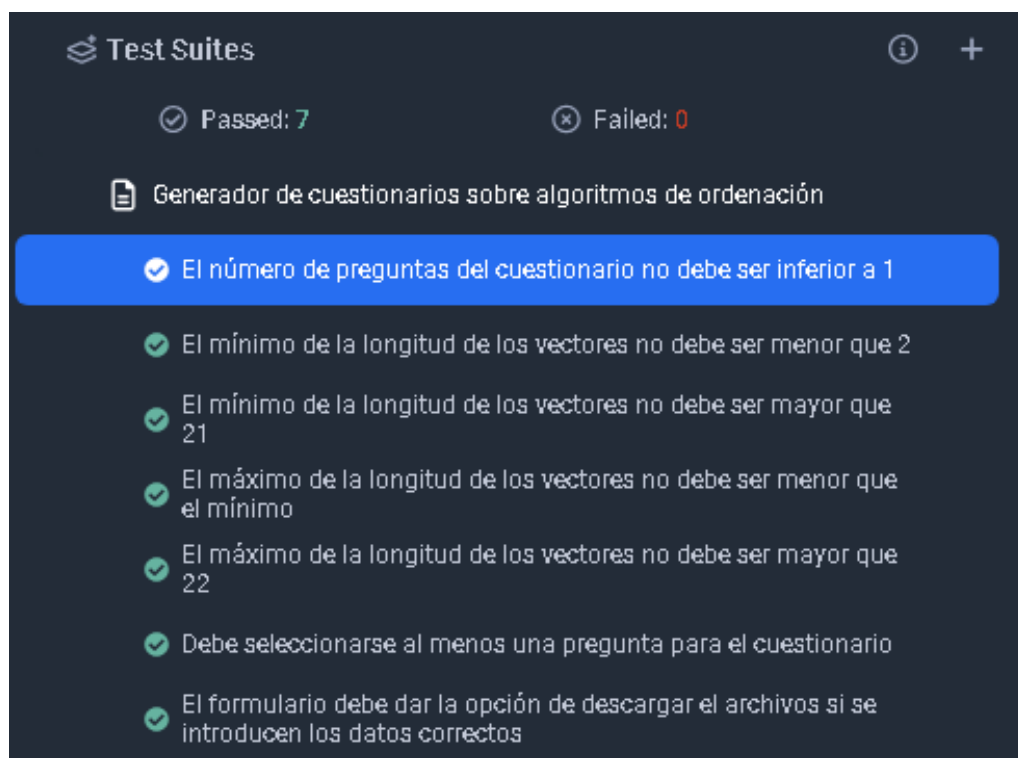


Figura D.6: *Revisión de pruebas en Katalon Recorder.*

Además de las pruebas realizadas para la aplicación web, también se llevó acabo una refactorización del código por medio de *SonarQube*. Esta herramienta es un software libre que ofrece algunas herramientas para analizar el código fuente de un software y encontrar fallos de seguridad, elementos repetidos o fallos de diseño. Gracias a esta herramienta se pudo revisar parcialmente el código y solucionar una serie de fallas para su correcta implementación.

## Apéndice *E*

---

# Documentación de usuario

---

### E.1. Introducción

Apartado destinado a tratar lo referente al ámbito del usuario. Se tratarán principalmente temas relativos a los requisitos la instalación del proyecto y el manual de uso enfocado al usuario.

### E.2. Requisitos de usuarios

Para acceder a la aplicación web, el usuario tiene disponible el acceso de forma remota desde el enlace

**<https://saqg-ubu-aeee3c86.koyeb.app>.**

Por otra parte, si desea acceder de forma local será necesario seguir los pasos de descarga del repositorio especificados en el manual del programador. En este caso, se utiliza el siguiente enlace para acceder a la web:

**<http://127.0.0.1:5000/>.**

Para el proyecto también sería conveniente disponer de un navegador para poder desplegar correctamente la aplicación web. A su vez, es recomendable disponer de un sistema operativo *Windows* puesto su desarrollo se ha enfocado específicamente para este sistema en concreto.

### E.3. Instalación

En cuanto a la instalación, no existe como tal un proceso más allá de lo mencionado anteriormente. La aplicación web ya contiene todo lo necesario

(librerías y ficheros imprescindibles) para su correcto funcionamiento gracias a la imagen de *Docker*. Esta imagen simplifica y facilita la instalación para cualquier usuario que desee acceder al proyecto en cualquier entorno.

En cuanto al formato local si que será necesaria la instalación de elementos previamente explicada en el manual del programador.

## E.4. Manual del usuario

Apartado destinado a detallar de forma comprensible y sencilla un manual para que cualquier usuario sea capaz de comprender el funcionamiento de la aplicación web.

### Página principal

Una vez iniciada la página web, se ofrecerá una breve descripción del funcionamiento del proyecto con un enlace al campo de pruebas de *Moodle* para poder probar los cuestionarios que se generen. Desde aquí se puede visualizar tanto la barra de navegación como el pie de página donde encontramos:

#### ■ Barra de navegación:

- El icono del proyecto que redirige a la página principal.
- El icono de *GitHub* que redirige al repositorio del proyecto.
- La barra desplegable que te permite acceder a las páginas de los Algoritmos (*Mergesort*, *Quicksort*, *Heapsort*).
- El enlace 'Acerca de' que redirige a la página de más información acerca del proyecto.

#### ■ Pie de página:

- El icono de la Universidad de Burgos que redirige a su página oficial.
- El icono de *Moodle* que redirige al campo de pruebas de *Moodle* para probar los bancos de preguntas.
- El icono de la Escuela Politécnica Superior que redirige a su página correspondiente de la universidad.

Figura E.1: *Página principal de la web.*

## Selección de Algoritmos

Desplegando la barra de navegación de la izquierda podremos seleccionar cualquiera de los algoritmos de lo que deseemos crear un banco de preguntas. Podremos redirigirnos a las páginas de los algoritmos *Mergesort*, *Quicksort* y *Heapsort* respectivamente donde encontraremos una breve descripción de cada algoritmo y un enlace para obtener mayor información.

## Página Mergesort



Figura E.2: *Página sobre el algoritmo Mergesort.*

## Página Quicksort



Figura E.3: *Página sobre el algoritmo Quicksort.*

## Página Heapsort



Figura E.4: Página sobre el algoritmo Heapsort.

Siguiendo desde el caso del algoritmo *Mergesort*, tendremos que pulsar sobre el botón 'Generar cuestionario *Mergesort*' donde se desplegará un formulario. Para rellenar el formulario tendremos que tener en cuenta los siguientes aspectos que se incluyen en él:

- Se debe indicar el número de preguntas que se desean incluir. Este número no debe ser inferior a 1 puesto que será necesario al menos una pregunta para generar el cuestionario.
- Se debe indicar el rango de longitud de los vectores aleatorios que se generarán en las preguntas. Hay que tener en cuenta que no puede ser menor que el rango mínimo no debe ser menor que 2 ni mayor que 21 puesto que sería demasiada carga para la aplicación. Por otra parte el rango máximo debe ser mayor al mínimo y menor que 22 por la misma razón.
- A continuación se muestran algunas imágenes de ejemplo que enseñan el formato de las preguntas que se pueden incluir en el cuestionario. Se debe seleccionar al menos una de las imágenes puesto que debe existir alguna pregunta para crear un cuestionario.



## Algoritmo de Ordenación Mergesort



 Algoritmos de ordenación Acerca de

El algoritmo de ordenación por mezcla (o **Mergesort**) es un algoritmo de ordenamiento externo desarrollado por Jhon Von Neumann basado en la técnica divide y vencerás. El método Mergesort consiste en dividir la lista desordenada de valores en dos sublistas prácticamente del mismo tamaño e ir ordenando de forma **recursiva** las dos sublistas para luego mezclar las dos sublistas en una sola lista ordenada.



**WIKIPEDIA**  
La enciclopedia libre

Número de preguntas:

Número mínimo de valores del vector:

Número máximo de valores del vector:

Figura E.5: Elementos a rellenar en el formulario (Algoritmo Mergesort).



Dado el siguiente vector:

11	65	33	7	16	100
----	----	----	---	----	-----

¿Qué nuevo vector resultaría al aplicar el método *mergesort*, en la primera llamada, una vez resueltas las llamadas recursivas, antes de realizar el último proceso de mezcla?

Select one:

☐ a. 100 11 16 33 65 7

☐ b. 11 33 65 7 16 100

☐ c. 100 16 33 7 11 65

¿Incluir este tipo de pregunta? ☐

Dado el siguiente vector:

48	98	15	54	3	82
----	----	----	----	---	----

Rellene los valores del vector resultante al aplicar el método *mergesort*, en la primera llamada, una vez resueltas las llamadas recursivas, antes de realizar el último proceso de mezcla:

--	--	--	--	--	--

¿Incluir este tipo de pregunta? ☐

Dado el siguiente vector:

11	87	78	67	85
----	----	----	----	----

¿Cuántas comparaciones se producirán usando el algoritmo *mergesort*, una vez la lista esté ordenada?

Select one:

☐ a. 9

☐ b. 8

Figura E.6: Seleccionar preguntas desde el formulario (Algoritmo Mergesort).

Finalmente será necesario pulsar sobre la opción de generar Fichero y con ello habremos finalizado la creación del banco de preguntas.

## Descargar fichero

Después de generar el fichero, la página nos dará la opción de descargarlo en nuestro equipo para posteriormente poder revisarlo en el campo de pruebas de *Moodle*.



Figura E.7: *Descarga del banco de preguntas (Algoritmo Mergesort).*

## Página de Más información

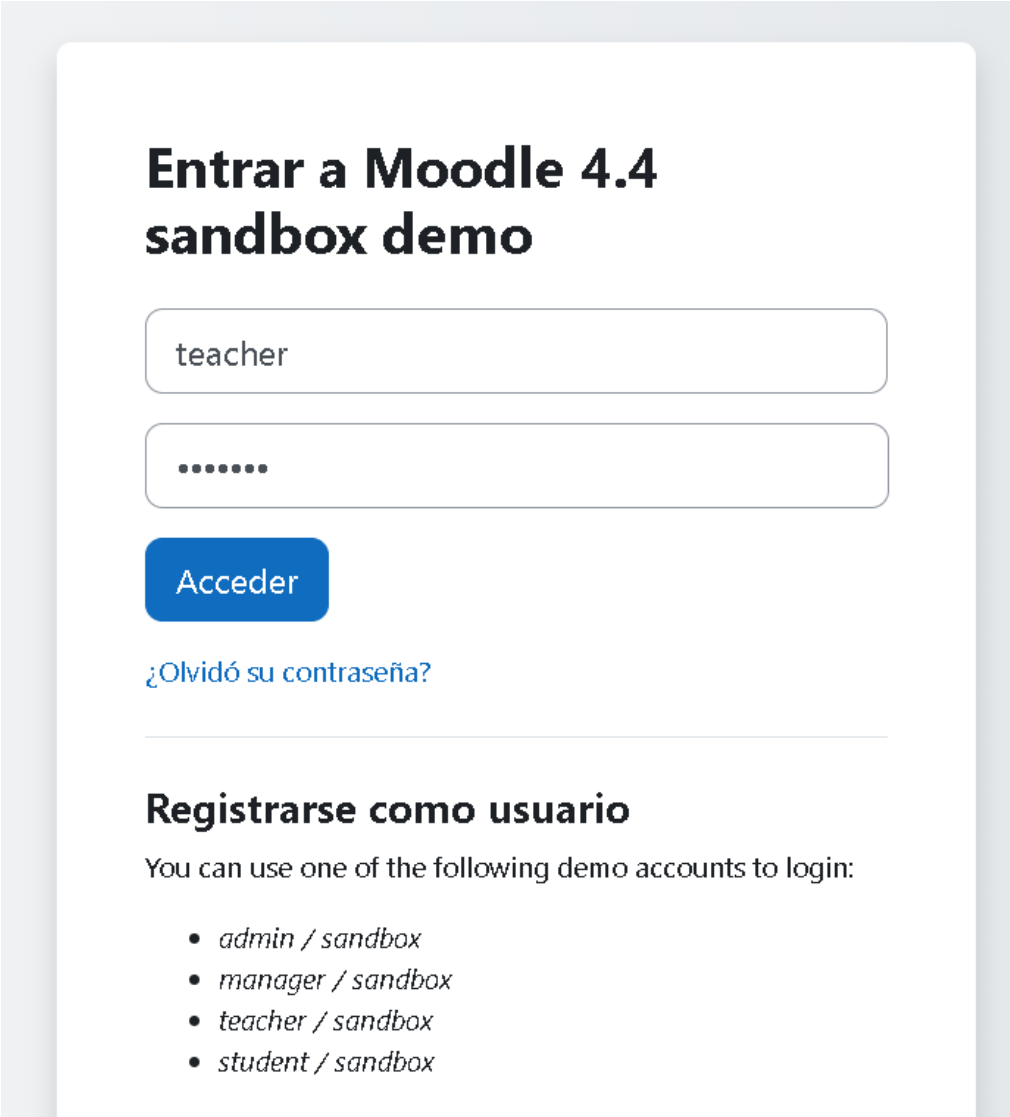
Para Acceder a la página de más información acerca del proyecto, podemos seleccionar en cualquier momento y desde cualquier parte de la aplicación web sobre la Opción de 'Acerca de' en la barra de navegación en el lado derecho para acceder a la misma. En ella encontraremos una descripción del proyecto y puntos a señalar del proyecto.

Figura E.8: *Página de más información.*

## Visualización de preguntas

Para poder comprobar que nuestro banco de pruebas se adapta correctamente a nuestras especificaciones previas, tendremos que acceder al campo de pruebas de *Moodle*. Se puede tener acceso en nuestra aplicación web desde el pie de página.

Una vez dentro, tendremos que acceder simulando que somos un usuario editor o profesor para poder crear un *quiz* o cuestionario comprobable. El nombre de usuario y la contraseña serán *'teacher'* y *'sandbox'* respectivamente para poder realizar el acceso.



The image shows a login interface for Moodle 4.4 sandbox demo. It features a title 'Entrar a Moodle 4.4 sandbox demo', a username field with 'teacher' entered, a password field with dots, a blue 'Acceder' button, and a link '¿Olvidó su contraseña?'. Below this is a section 'Registrarse como usuario' with a text 'You can use one of the following demo accounts to login:' and a bulleted list of demo accounts: 'admin / sandbox', 'manager / sandbox', 'teacher / sandbox', and 'student / sandbox'.

## Entrar a Moodle 4.4 sandbox demo

teacher

.....

Acceder

[¿Olvidó su contraseña?](#)

---

### Registrarse como usuario

You can use one of the following demo accounts to login:

- *admin / sandbox*
- *manager / sandbox*
- *teacher / sandbox*
- *student / sandbox*

Figura E.9: *Inicio de sesión en Moodle.*

Posteriormente, podremos acceder a cualquiera de los cursos de ejemplo sin necesidad de crear uno. Seleccionaremos la opción de añadir una actividad o recurso (es recomendable tener la opción de modo edición activada en la parte superior derecha).

De las opciones disponibles, deberemos seleccionar la opción *Quiz* para poder realizar un banco de preguntas. Una vez dentro de la edición de *quiz*

no es necesario rellenar todos los campos; sólo será necesario rellenar el nombre y pulsar sobre la opción de guardar y mostrar al final del formulario.

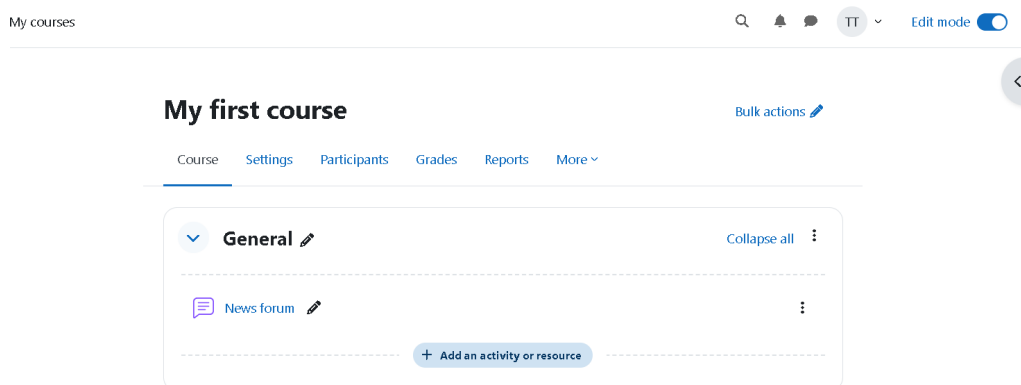


Figura E.10: *Añadimos la actividad del cuestionario en Moodle.*

Una vez creada la actividad, accederemos a la opción de banco de preguntas y desplegaremos la barra de navegación que nos muestra inicialmente y seleccionaremos la opción importar. Una vez dentro, seleccionamos el formato (*Moodle XML*) y importaremos el banco de preguntas que hemos descargado de nuestra aplicación web.

## Import questions from file?

### File format

- ☐ Aiken format ?
- ☐ Blackboard ?
- ☐ Embedded answers (Cloze) ?
- ☐ GIFT format ?
- ☐ Missing word format ?
- ☒ Moodle XML format ?

### > General

#### Import questions from file

Import

Choose a file... Maximum size for new files: 256 MB

pregunta\_quicksort (4).xml

Import

Required

Figura E.11: *Importación de banco de preguntas en Moodle.*

Finalmente, iremos al apartado de preguntas para añadir nuestras cuestiones al cuestionario. Seleccionaremos la opción de añadir desde un banco de preguntas y escogeremos aquellas que queramos visualizar de nuestro banco.

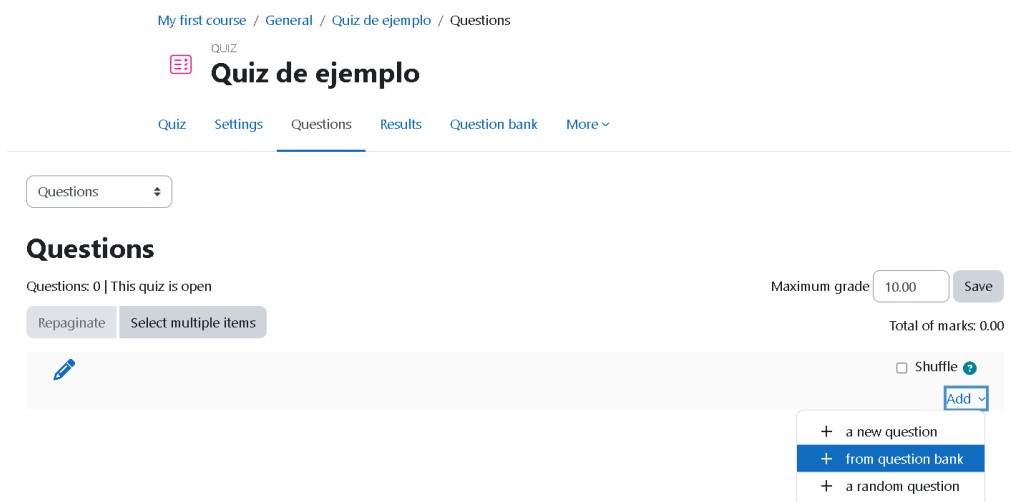


Figura E.12: Implementación de las preguntas del banco en Moodle.

Una vez incluidas, podremos previsualizar el cuestionario para probar si el banco de preguntas funciona correctamente y según nuestras especificaciones previas.

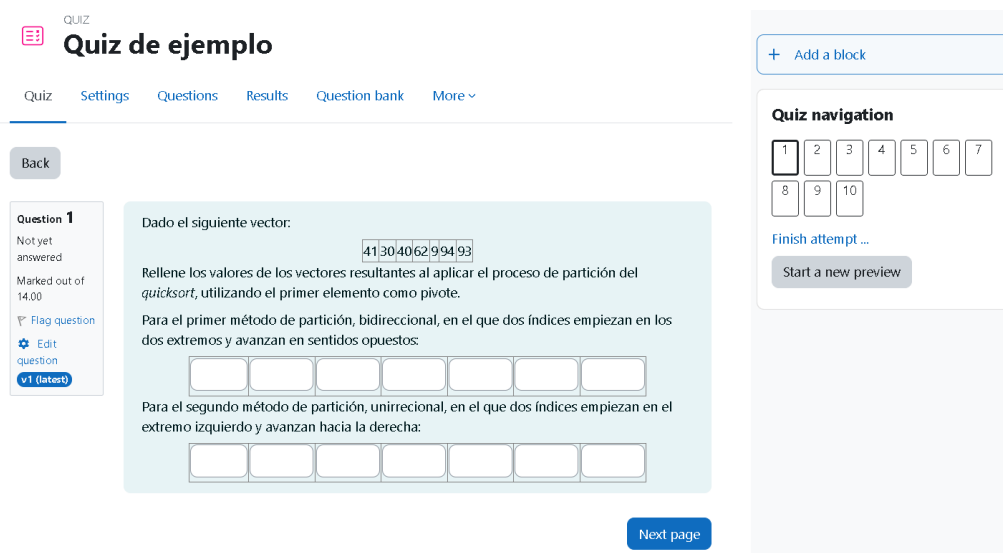


Figura E.13: Representación del banco de preguntas en Moodle.





## **Anexo de sostenibilización curricular**

---

### **F.1. Introducción**

En este apartado se incluye mi reflexión personal con respecto a los aspectos que se abordan en este proyecto sobre la sostenibilidad y su enfoque. Se detallarán las diferentes competencias que se han podido observar durante el desarrollo del proyecto que contribuyen a realizar un trabajo de forma sostenible. Todo ello con el fin de observar los diferentes ámbitos sostenibles que se han ido aplicando y aprendiendo durante el grado de ingeniería informática.

Puesto que este proyecto está principalmente ambientado para su uso en la docencia, se debe destacar la importancia que acarrea socialmente un trabajo similar. Este software está destinado para facilitar a los futuros alumnos del grado de ingeniería informática, así como para otros grados la comprensión acerca del funcionamiento y aplicación de la algoritmia. Principalmente es un proyecto que favorece a la sociedad, ya no sólo por parte de facilitar a los docentes si no para fomentar el aprendizaje de las futuras generaciones.

### **F.2. Implicación ética**

Por otra parte, me gustaría señalar a su vez la implicación ética y deontológica que acarrea el desarrollo de software y aplicaciones web. Este proyecto me ha ayudado a comprender el impacto ético que los trabajos

realizados por un ingeniero informático deben acarrear. Un ingeniero siempre es responsable del producto final que elabora y por eso debe ser capaz de ofrecer un buen trabajo que no impacte de manera negativa. Siempre respetando tanto la privacidad de los usuarios que disfruten del proyecto final como sus derechos.

### **F.3. Impacto medioambiental**

En cuanto al impacto del entorno, diría que la docencia online, ámbito en el que está basado este proyecto, favorece bastante en la reducción de gasto de material escolar y de oficina tanto a alumnos como profesores. Esto supone un menor impacto medioambiental en la sociedad y favorece la inclusión de las futuras generaciones a las nuevas tecnologías. Es por ello que desde mi punto de vista este proyecto contribuye en fomentar el uso de la docencia online, y por tanto reducir el impacto medioambiental.

### **F.4. Aspectos socioculturales**

En cuanto al impacto cultural, como futuro ingeniero informático soy consciente del gran impacto de las nuevas tecnologías y del manejo que tienen con ellas las nuevas generaciones. Pero también soy consciente de la brecha digital que a día de hoy afecta a gran parte de la sociedad más envejecida. Es por esto que creo que es importante fomentar el aprendizaje de los mayores sobre aspectos de la informática. Además, facilitando dicho aprendizaje desde la docencia online se fomentaría el uso de las nuevas tecnologías y complementaría a su educación. Con eso en mente, el desarrollo de una interfaz de usuario que ofrezca una buena accesibilidad facilitaría la comprensión del mundo digital a aquellos con complicaciones para manejarse en estas.

Con respecto al impacto social, en el grado somos conscientes de lo mucho que afectan el desarrollo de nuevas tecnologías a la sociedad. El mundo digital esta muy globalizado y como consecuencia es necesario ser más responsable y colaborar con la sociedad para fomentar un desarrollo sostenible. En cuanto a este proyecto, está destinado a mejorar los aspectos de educación y fomentar el aprendizaje. Es por esta razón que espero que el desarrollo de este trabajo pueda tener un impacto positivo en la sociedad actual.

## **F.5. Conclusión**

Como conclusión, este proyecto ha sido elaborado teniendo en cuenta un desarrollo sostenible siempre siguiendo el enfoque desde el punto de vista de un ingeniero informático. Por esta razón, me ha ayudado a comprender las competencias básicas que se deben tener en cuenta a la hora de desarrollar un software de manera sostenible. Ha fomentado mi auto aprendizaje y me ha aportado las herramientas para continuar con este enfoque durante mi futuro laboral.



---

## Bibliografía

---

- [1] Modelo vista controlador. [https://es.wikipedia.org/wiki/Modelo\\_vista\\_controlador](https://es.wikipedia.org/wiki/Modelo_vista_controlador). [Accedido 30-05-2024].
- [2] Patrón de diseño fachada. [https://es.wikipedia.org/wiki/Facade\\_\(patrón\\_de\\_diseño\)](https://es.wikipedia.org/wiki/Facade_(patrón_de_diseño)). [Accedido 30-05-2024].
- [3] Salario medio para ingeniero informático en España, 2024. <https://es.talent.com/salary?job=ingeniero+informático#:~:text=El%20salario%20ingeniero%20informático%20promedio,hasta%20a%2036.250%20al%20año>. [Accedido 27-05-2024].
- [4] Seguridad social: Cotización / recaudación de trabajadores — seg-social.es. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537>. [Accedido 27-05-2024].
- [5] Banco Santander. Calculadora irpf — bancosantander.es. <https://www.bancosantander.es/particulares/cuentas-tarjetas/cuentas-corrientes/calculadora-irpf#:~:text=De%2020.200%20a%2035.200,tipo%20impositivo%20es%20de%2045%25>. [Accedido 27-05-2024].