

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

MQGenerator
Documentación Técnica



Presentado por Álvaro Hoyuelos Martín
en Universidad de Burgos — 4 de julio de 2023
Tutores: Dr. José Manuel Galán Ordax - Dra.
Virginia Ahedo García

Índice general

Índice general.....	I
Índice de figuras	III
Índice de tablas	V
Plan de Proyecto Software	8
A.1. Introducción	8
A.2. Planificación temporal	9
A.3. Estudio de viabilidad	23
Especificación de Requisitos	26
B.1. Introducción	26
B.2. Objetivos generales	26
B.3. Catálogo de requisitos	27
B.4. Especificación de requisitos	30
Especificación de diseño	39
C.1. Introducción	39
C.2. Diseño de datos	39
C.3. Diseño procedimental	41
C.4. Diseño arquitectónico	43
C.5. Diseño de interfaces	45
Documentación técnica de programación	49

D.1. Introducción	49
D.2. Estructura de directorios	49
D.3. Manual del programador	51
D.4. Instalación de las herramientas	54
Documentación de usuario	56
E.1. Introducción	56
E.2. Requisitos de usuarios	56
E.3. Instalación	57
E.4. Manual del usuario	57
Bibliografía.....	70

Índice de figuras

Figura A.1: <i>Burndown report sprint 1</i>	10
Figura A.2: <i>Issues sprint 1</i>	11
Figura A.4: <i>Issues sprint 2</i>	13
Figura A.6: <i>Issues sprint 3</i>	15
Figura A.8: <i>Issues sprint 4</i>	17
Tabla A.5: <i>Tareas sprint 4</i>	17
Figura A.10: <i>Issues sprint 5</i>	18
Figura A.12: <i>Issues sprint 6</i>	20
Figura A.14: <i>Issues sprint 7</i>	22
Figura B.1: <i>Casos de uso</i>	31
Figura D.1: <i>Descarga del repositorio mediante el entorno web.</i>	51
Figura D.2: <i>Ejecución en remoto</i>	52
Figura D.3: <i>Nueva aplicación en Heroku</i>	53
Figura D.4: <i>Configuración en Heroku</i>	54
Figura D.5: <i>Resultado del despliegue</i>	54

Figura D.6: Descargar Microsoft Visual Studio Code.....	55
Figura E.1: Página principal de la aplicación web.....	58
Figura E.2: Página de Más información	58
Figura E.3: Página de Registro	59
Figura E.4: Página de Inicio sesión	60
Figura E.5: Selección de preguntas	60
Figura E.6: Preguntas Tabla	61
Figura E.7: Descargar fichero Tabla	62
Figura E.8: Preguntas Valor ganado.....	62
Figura E.9: Descargar fichero Valor ganado.....	63
Figura E.10: Preguntas Programación ganada	64
Figura E.11: Descargar fichero Programación Ganada.....	64
Figura E.12: Registro en Moodle	65
Figura E.13: Configuración Moodle	65
Figura E.14: Crear una actividad Cuestionario	66
Figura E.15: Configurar un Cuestionario	67
Figura E.15: Importar un banco de preguntas	67
Figura E.16: Agregar preguntas	68
Figura E.17: Visualización de las preguntas resueltas	69

Índice de tablas

Tabla A.1: Estimaciones de tiempo.....	9
Tabla A.2: Tareas <i>sprint 1</i>	12
Tabla A.3: Tareas <i>sprint 2</i>	14
Tabla A.4: Tareas <i>sprint 3</i>	15
Tabla A.6: Tareas <i>sprint 5</i>	19
Tabla A.7: Tareas <i>sprint 6</i>	21
Tabla A.7: Tareas <i>sprint 6</i>	23
Tabla A.8: Listado con los materiales, su valor y su tiempo de amortización	23
Tabla A.9: Coste amortizado para el material.....	24
Tabla A.10: Desglose del coste de personal.....	25
Tabla A.11: Legalidad de herramientas y librerías	25
Tabla B.1: Nivel de frecuencia de los casos de uso	30
Tabla B.2: Caso de uso 1 – Registro de usuarios	32
Tabla B.3: Caso de uso 2 – Login de usuarios.....	32
Tabla B.4: Caso de uso 3 – Información general	33

Tabla B.5: Caso de uso 4 – Preguntas frecuentes	33
Tabla B.6: Caso de uso 5 – Ayuda al usuario.....	34
Tabla B.7: Caso de uso 6 – Cambio de idioma	34
Tabla B.8: Caso de uso 7 – Seleccionar tipo de preguntas	35
Tabla B.9: Caso de uso 8 – Generar preguntas Tabla.....	35
Tabla B.10: Caso de uso 9 – Generar preguntas Valor Ganado	36
Tabla B.11: Caso de uso 10 – Generar preguntas Programación Ganada .	36
Tabla B.12: Caso de uso 11 – Exportación del fichero.....	37
Tabla B.13: Caso de uso 12 – Generar nuevo fichero	37
Tabla D.1: Versión de los componentes	52

Plan de Proyecto Software

A.1. Introducción

En este apartado se va a destinar a explicar cómo se ha distribuido el trabajo durante los diferentes sprints y el proceso de planificación del desarrollo del proyecto. En concreto, me centraré en la planificación temporal, económica y legal del mismo.

Como ya se comentó en la memoria, para la planificación del proyecto se ha utilizado la herramienta de ZenHub, ya que es una extensión de GitHub que añade diferentes funcionalidades interesantes como la gestión de forma sencilla y eficiente las diferentes tareas a realizar en el proyecto.

Durante la primera reunión, cuando ya teníamos el tema del trabajo final, se realizó una planificación de los diferentes objetivos que tendría este proyecto. Se editaron y crearon nuevas etiquetas, de las ya disponibles en GitHub debido a que las etiquetas por defecto no eran suficientes, para una clasificación más intuitiva de las diferentes tareas que se tenían que completar a lo largo del proyecto.

También, al emplear una metodología SCRUM, es necesario establecer una estimación numérica de tiempo para la realización de cada tarea. Aunque en la creación de una tarea se especificara un valor de tiempo específico, este podría cambiar si se necesitase más tiempo o por el contrario se completase antes de lo esperado. Para conseguir una mejor visión sobre este punto se estableció para cada Story point una estimación con un valor de tiempo real indicada en la Tabla A.1.

Story point	Estimación temporal
1	1 hora
2	2 horas
3	3 horas
5	5 horas
8	8 horas
13	13 horas
24	1 día

Tabla A.1: Estimaciones de tiempo

Como se ha mencionado en la memoria, la ejecución de proyecto al no ser un grupo de trabajo sino ejecutada por solo una persona no se ha podido realizar la metodología SCRUM en su totalidad. Aun así, se ha intentado adaptar lo máximo posible.

- El diseño del proyecto se ha realizado de manera incremental.
- Realizar un nuevo Sprint por cada reunión que se ha tenido con los tutores para implementar las nuevas tareas.
- Realización de reuniones cada dos o tres semanas salvo para casos especiales.
- Toma de decisiones sobre la duración de cada tarea y la prioridad de cada una de ellas.
- Realizar una reunión cuando finalizaban un Sprint para comprobar si estaban bien implementadas las indicaciones propuestas por los tutores o si eran necesarias implementar nuevas funcionalidades.
- Utilización de ZenHub para el control de las tareas, manejando cuales estaban en proceso, cuales estaban terminadas y cuantas tareas quedaban por terminar de ese Sprint.

A.2. Planificación temporal

En este apartado se va a describir el programa temporal que se diseñó y utilizó durante todo el desarrollo del proyecto. Explicaré cada uno de los sprints que se han creado durante la realización del proyecto, con cada tarea y el gráfico *Burndown report* proporcionado por la herramienta de ZenHub.

Sprint 0: hasta el 15 de febrero

Esto no es un Sprint como tal, sino que abarca el periodo inicial de la creación de un proyecto completamente nuevo que se adaptara a temas que me quisiese especializar en un futuro como es la educación.

Sprint 1 (15/02/2023 – 02/03/2023)

En el primer *sprint* del proyecto se definieron todos los objetivos del proyecto a cumplir y cuales serían los pasos para completarlos. Se ha configurado el repositorio del proyecto, se ha elegido un entorno de desarrollo y editor de texto para la documentación. Una vez comparado todas las opciones disponibles de IDE se ha configurado para trabajar sobre él.

Se ha realizado un trabajo de investigación sobre el tema a tratar y otros estudios relacionados para saber que herramientas serían interesantes para implementar en mi proyecto. Se ha instalado Zotero para las referencias.

Al ser el primer *sprint*, la prioridad era buscar información sobre los diferentes servidores para alojar una aplicación web, que en este caso se utilizó Flask. Una vez seleccionado, se ha implementado dicho servidor.

Enlace al primer *sprint*: [Sprint 1](#).

A continuación, se va a mostrar el gráfico *Burndown report* del primer *sprint*, donde se visualiza el avance de cada tarea.

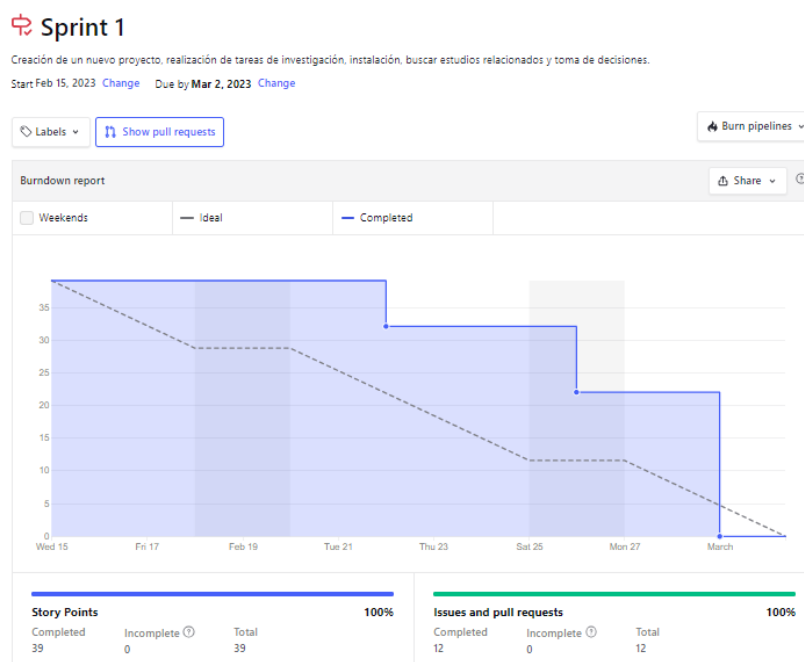


Figura A.1: *Burndown report sprint 1*

La línea gris discontinua indica como debería haber sido el proceso de ejecución optima a lo largo de todo el sprint. Como podemos apreciar es bastante difícil que se asemeje a la realidad debido a los factores externos, mostrando que se ha avanzado en ocasiones mucho y otras poco en menos tiempo.

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Completed issues and pull requests			Story points
✓ Creación del repositorio de códigos	configuración	decisión	1
MQGenerator #1 000 Closed Sprint 1			
✓ Elegir herramienta para la gestión de tareas	investigación	decisión	1
MQGenerator #2 000 Closed Sprint 1			
✓ Elegir editor de texto	investigación	decisión	1
MQGenerator #3 000 Closed Sprint 1			
✓ Elegir herramienta para la gestión de referencias	investigación	decisión	1
MQGenerator #4 000 Closed Sprint 1			
✓ Instalación de Zotero	configuración		1
MQGenerator #5 000 Closed Sprint 1			
✓ Estudio relacionados	investigación		13
MQGenerator #6 000 Closed Sprint 1			
✓ Tratamiento de licencias	documentación	investigación	1
MQGenerator #7 000 Closed Sprint 1			
✓ Versión 1: Parte externa del repositorio	documentación		1
MQGenerator #8 000 Closed Sprint 1			
✓ Elegir servidor	investigación	decisión	1
MQGenerator #9 000 Closed Sprint 1			
✓ Instalar y configurar Flask	configuración		5
MQGenerator #10 000 Closed Sprint 1			
✓ Montar servidor Flask sencillo	programación		5
MQGenerator #11 000 Closed Sprint 1			
✓ Documentación del primer Sprint	documentación		8
MQGenerator #12 000 Closed Sprint 1			

Figura A.2: Issues sprint 1

Tareas completadas	Etiquetas		Estimación
Creación del repositorio de códigos	configuración	decisión	1
Elegir herramienta para la gestión de tareas	investigación	decisión	1
Elegir editor de texto	investigación	decisión	1
Elegir herramienta para la gestión de referencias	investigación	decisión	1
Instalación de Zotero	configuración		1
Estudios relacionados	investigación		13
Tratamiento de licencias	documentación	investigación	1
Versión 1: Parte externa del repositorio	documentación		1

Elegir servidor	investigación	decisión	1
Instalar y configurar Flask	configuración		5
Montar servidor Flask sencillo	programación		5
Documentación del primer Sprint	documentación		8
TOTAL			39

Tabla A.2: Tareas *sprint 1*

Sprint 2 (02/03/2023 – 15/03/2023)

Para el segundo *sprint* se fijaron los objetivos de añadir los conceptos teóricos vistos en el anterior *sprint* a la memoria, incluyendo también en el anexo los *sprint* realizados hasta el momento.

Se ha comparado y elegido una herramienta para la realización de prototipos, principalmente para realizar un primer diseño de la aplicación web. Para este primer diseño se ha decidido que funcionalidades iba a tener una posible aplicación web final.

Una vez diseñado, se ha añadido la funcionalidad de poder registrarse en la aplicación.

Finalmente se ha realizado un estudio de los cuestionarios de Moodle y sobre el seguimiento y control de proyectos, para tener más conocimientos a la hora de su funcionamiento práctico.

Enlace al segundo *sprint*: [Sprint 2](#).

A continuación, se va a mostrar el gráfico *Burndown report* del segundo *sprint*, donde se visualiza el avance de cada tarea.

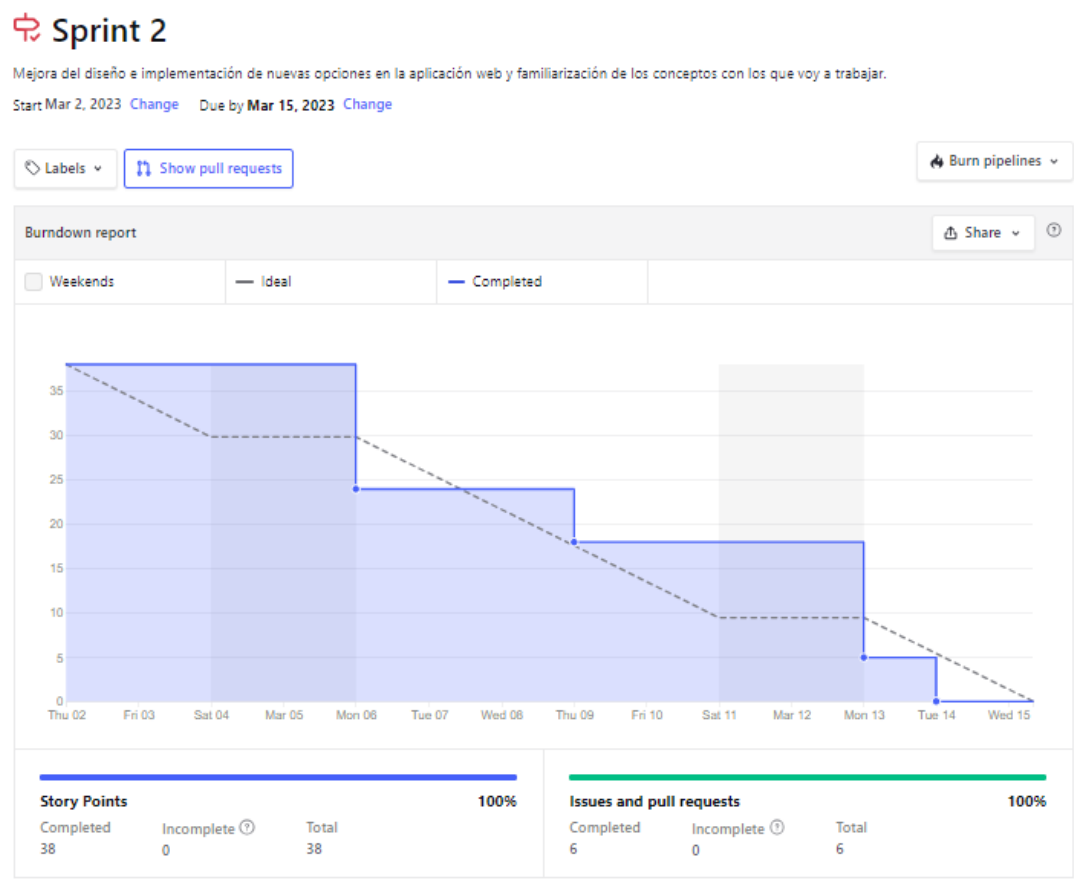


Figura A.3: Burndown report sprint 2

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Completed issues and pull requests			Story points
✓ Elegir herramienta de prototipado	investigación decisión		1
MQGenerator #13 000 Closed Sprint 2			
✓ Diseño inicial web	diseño decisión		5
MQGenerator #14 000 Closed Sprint 2			
✓ Implementación de registro	programación		5
MQGenerator #15 000 Closed Sprint 2			
✓ Configuración de Canva	configuración		1
MQGenerator #16 000 Closed Sprint 2			
✓ Estudio de conceptos teóricos	documentación investigación		13
MQGenerator #17 000 Closed Sprint 2			
✓ Estudio de la teoría de cuestionarios moodle	documentación investigación		13
MQGenerator #18 000 Closed Sprint 2			

Figura A.4: Issues sprint 2

Tareas completadas	Etiquetas	Estimación
Elegir herramienta de prototipado	investigación decisión	1
Diseño inicial web	diseño decisión	5

Implementación de registro	programación	5	
Configuración de Canva	configuración	1	
Estudio de conceptos teóricos	documentación	investigación	13
Estudio de la teoría de cuestionarios Moodle	documentación	investigación	13
<hr/>			
TOTAL			38

Tabla A.3: Tareas *sprint 2*

Sprint 3 (16/03/2023 – 03/04/2023)

Para el tercer *sprint* se fijaron los objetivos de la elección del lenguaje de programación para crear los diferentes algoritmos para la creación de preguntas. También se ha elegido la extensión estandarizada por Moodle que deben tener los ficheros creados por el algoritmo, para su futura importación y exportación a esta plataforma de aprendizaje.

Se ha realizado en Python el primer tipo de preguntas que se pueden crear en la aplicación web, que en este caso son las preguntas tipo Tabla. Debido a su menor complejidad y que trata sobre los conceptos principales del tema.

Finalmente se ha añadido una versión inicial de los requerimientos actuales que se necesitan para el correcto funcionamiento de la aplicación.

Enlace al tercer *sprint*: [Sprint 3](#).

A continuación, se va a mostrar el gráfico *Burndown report* del tercer *sprint*, donde se visualiza el avance de cada tarea.

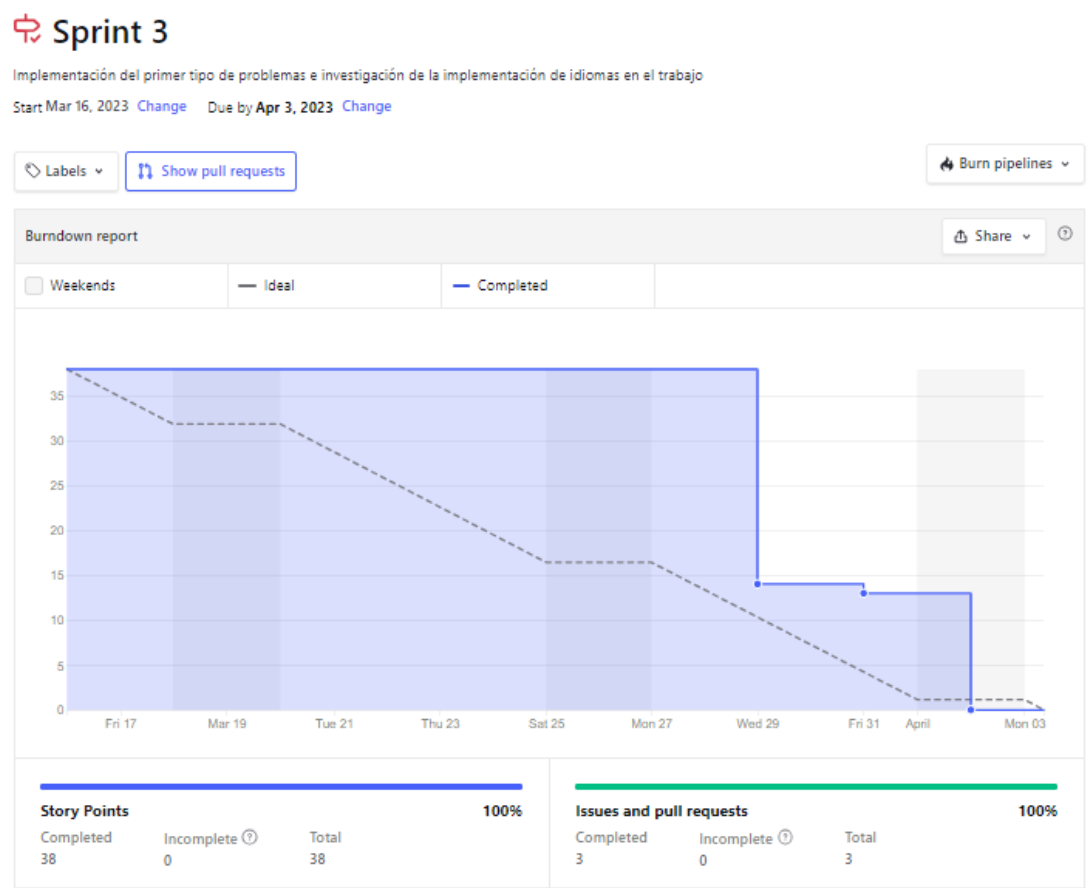


Figura A.5: Burndown report sprint 3

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Completed issues and pull requests				Story points
✓	Creación del primer tipo de preguntas	programación		24
	MQGenerator #19 000 Closed Sprint 3			
✓	Creación del archivo XML	programación		13
	MQGenerator #20 000 Closed Sprint 3			
✓	Creación fichero de requerimientos	documentación	configuración	1
	MQGenerator #21 000 Closed Sprint 3			

Figura A.6: Issues sprint 3

Tareas completadas	Etiquetas	Estimación
Creación del primer tipo de preguntas	programación	24
Creación del archivo XML	programación	13
Creación fichero de requerimientos	documentación configuración	1
TOTAL		38

Tabla A.4: Tareas sprint 3

Sprint 4 (26/04/2023 – 04/05/2023)

Para el cuarto sprint se fijó el objetivo de arreglar el código realizado en el anterior *sprint* para que cumpla con las nuevas indicaciones de los tutores.

Se han vuelto a realizar una investigación sobre como debería estructurarse el fichero .XML. Una vez sabiendo como tendrían que crearse, se ha corregido algunos errores del programa y se ha arreglado el código para que al crear estos ficheros cumplan con la estandarización de Moodle en su estructura.

Enlace al cuarto *sprint*: [Sprint 4](#).

A continuación, se va a mostrar el gráfico *Burndown report* del cuarto *sprint*, donde se visualiza el avance de la única tarea.

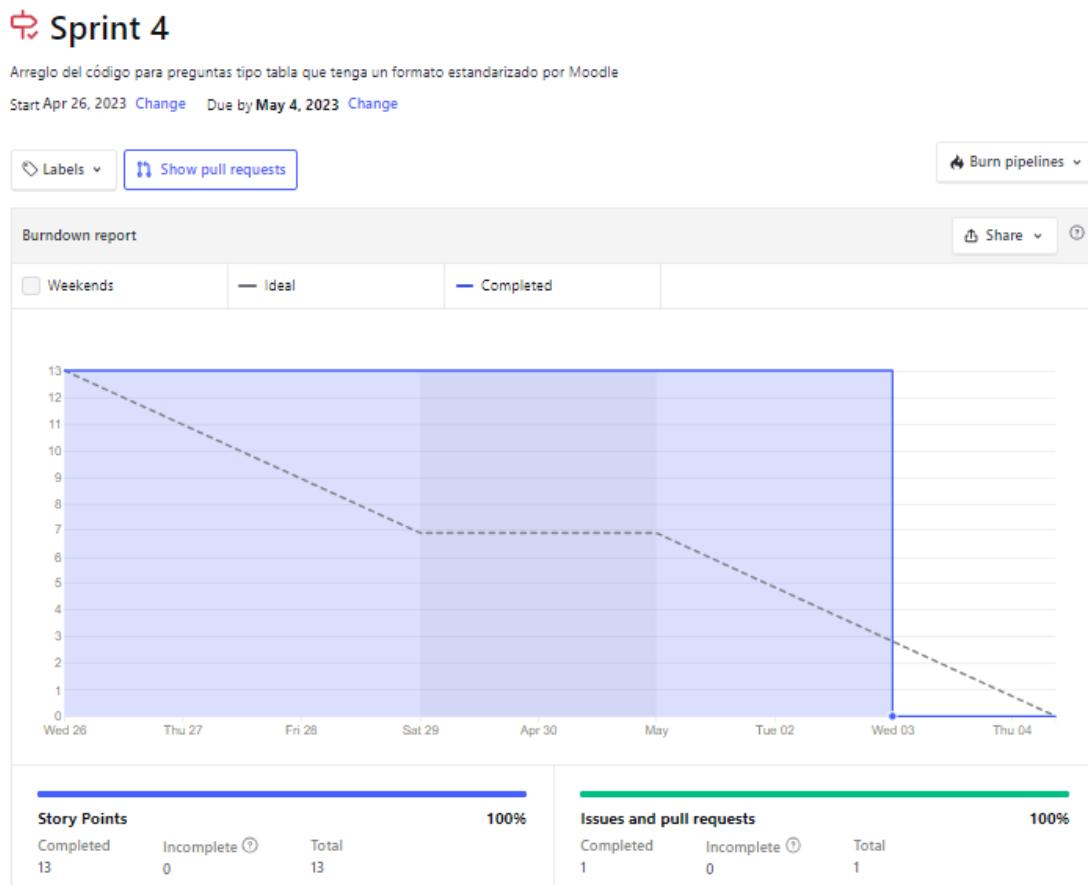
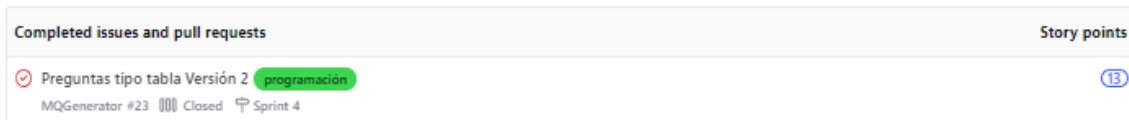


Figura A.7: *Burndown report sprint 4*

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Figura A.8: *Issues sprint 4*

Tareas completadas	Etiquetas	Estimación
Implementación de registro	programación	13
TOTAL		13

Tabla A.5: Tareas *sprint 4*

Sprint 5 (05/05/2023 – 18/05/2023)

Para el quinto *sprint* se fijaron los objetivos de investigar los conceptos teóricos sobre el responsive e internacionalización de una aplicación web, decidiendo que herramientas se pueden aplicar a mi proyecto que mejor se adapten a los requisitos.

Al ya conocer los requisitos y la estructura que debía tener el algoritmo de las preguntas tipo tabla, se procede a realizar las preguntas restantes. Se ha realizado una investigación sobre los gráficos dinámicos para poder realizar las preguntas del método valor ganado y programación ganada.

Se ha realizado un nuevo diseño de la aplicación web en Canva añadiendo las nuevas funciones que implementaré como la elección del tipo de pregunta y poder descargarse el archivo creado para exportarlo a Moodle.

Finalmente se ha configurado la herramienta para la documentación del proyecto Microsoft Word, para que tenga un diseño igual a LaTeX en la memoria y anexos. Se ha realizado la documentación los *sprints* anteriores.

Enlace al quinto *sprint*: [Sprint 5](#).

A continuación, se va a mostrar el gráfico *Burndown report* del quinto *sprint*, donde se visualiza el avance de cada tarea.

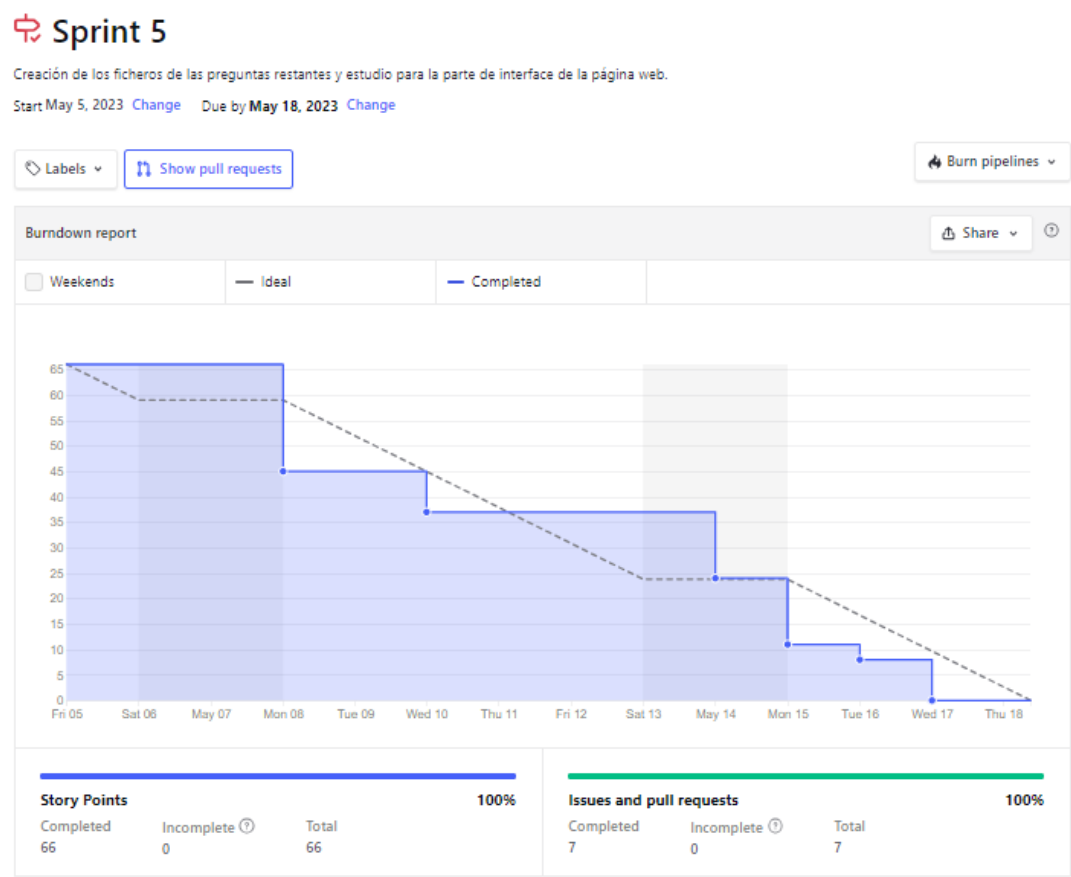


Figura A.9: Burndown report sprint 5

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Completed issues and pull requests				Story points
✓	Investigación sobre responsive del diseño web	documentación	investigación	8
MQGenerator #24 000 Closed Sprint 5				
✓	Estudio de la internacionalización de la web	documentación	investigación	13
MQGenerator #25 000 Closed Sprint 5				
✓	Creación preguntas tipo valor ganada	programación		13
MQGenerator #26 000 Closed Sprint 5				
✓	Nuevo diseño de la web	diseño		8
MQGenerator #27 000 Closed Sprint 5				
✓	Creación preguntas tipo programación ganada	programación		13
MQGenerator #28 000 Closed Sprint 5				
✓	Configuración del Word	documentación	configuración	3
MQGenerator #29 000 Closed Sprint 5				
✓	Realización de documentación	documentación		8
MQGenerator #30 000 Closed Sprint 5				

Figura A.10: Issues sprint 5

Tareas completadas	Etiquetas	Estimación
Investigación sobre responsive del	documentacióninvestigación	8

diseño web		
Estudio de la internacionalización de la web	documentación	investigación 13
Creación preguntas tipo valor ganado	programación	13
Nuevo diseño de la web	diseño	8
Creación preguntas tipo programación ganada	programación	13
Configuración del Word	documentación	configuración 3
Realización de documentación	documentación	8
TOTAL		66

Tabla A.6: Tareas *sprint* 5

Sprint 6 (21/05/2023 – 09/06/2023)

Para el sexto *sprint* se fijaron los objetivos de añadir las nuevas indicaciones de los tutores para los programas y la creación de la aplicación web.

Con las nuevas indicaciones se va a cambiar el paso por terminal del programa tabla y ser más visual las gráficas para las otras preguntas ya que los valores se dibujaban en las curvas dificultando su lectura. También se ha implementado la correcta codificación de las gráficas para que Moodle pueda leerlas correctamente, cumpliendo con el formato.

Una vez realizado todos los programas y diseño de la aplicación web en Canva, se va a montar una aplicación web desde 0 donde tenga las funcionalidades de registro y la elección de los tres tipos de preguntas diferentes.

Finalmente se ha realizado una investigación sobre diferentes servidores que pueden desplegar mi aplicación web y se ha optado por Heroku. Para terminar, se ha realizado un primer despliegue de la aplicación.

Enlace al sexto *sprint*: [Sprint 6](#).

A continuación, se va a mostrar el gráfico *Burndown report* del sexto *sprint*, donde se visualiza el avance de cada tarea.

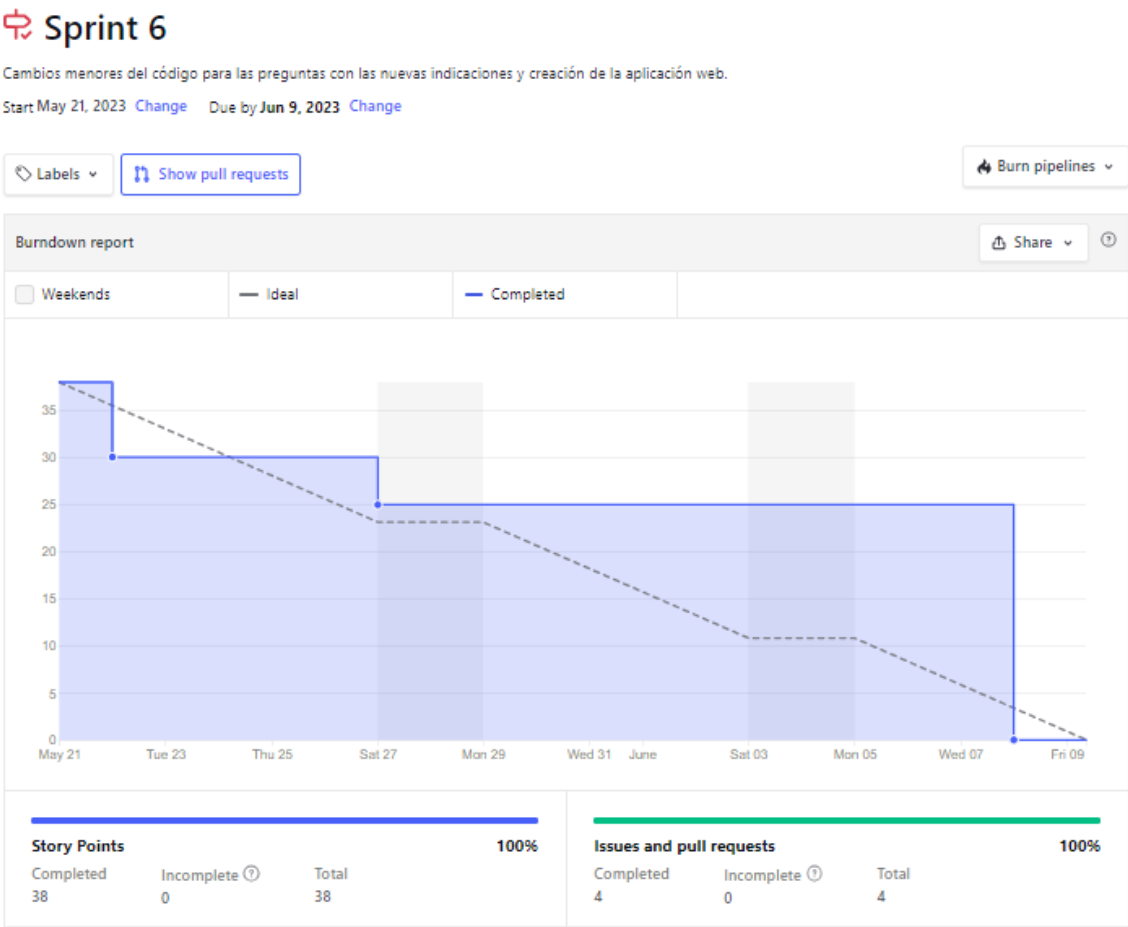


Figura A.11: Burndown report sprint 6

Por último, se va a mostrar la tabla de clasificación de cada tarea con su etiqueta correspondiente.

Completed issues and pull requests	Story points
<div><div><div>✓</div><div>Cambios en el código</div><div>programación</div></div><div>MQGenerator #31 000 Closed Sprint 6</div></div>	8
<div><div><div>✓</div><div>Creación aplicación web</div><div>diseño programación</div></div><div>MQGenerator #32 000 Closed Sprint 6</div></div>	24
<div><div><div>✓</div><div>Elección posible servidor para el despliegue</div><div>investigación decisión</div></div><div>MQGenerator #33 000 Closed Sprint 6</div></div>	5
<div><div><div>✓</div><div>Primer despliegue de la aplicación</div><div>despliegue</div></div><div>MQGenerator #34 000 Closed Sprint 6</div></div>	1

Figura A.12: Issues sprint 6

Tareas completadas	Etiquetas	Estimación
Cambios en el código	programación	8
Creación aplicación web	diseño programación	24

Elección posible servidor para el despliegue	investigación	decisión	5
Primer despliegue de la aplicación	despliegue		1
TOTAL			38

Tabla A.7: Tareas *sprint* 6

Sprint 7 (09/06/2023 – 25/06/2023)

Para el *sprint* final se fijaron los objetivos de solucionar los errores de compatibilidad al usar Flask y tener una aplicación web funcional que realice todos objetivos del proyecto.

Cada vez que se ejecutaba la aplicación salta el aviso de estar intentando mostrar una gráfica de Matplotlib desde un hilo secundario que no es el hilo principal. Finalmente se ha conseguido solucionar configurando el backend de Matplotlib a uno que no requiere una interfaz gráfica para generar las gráficas.

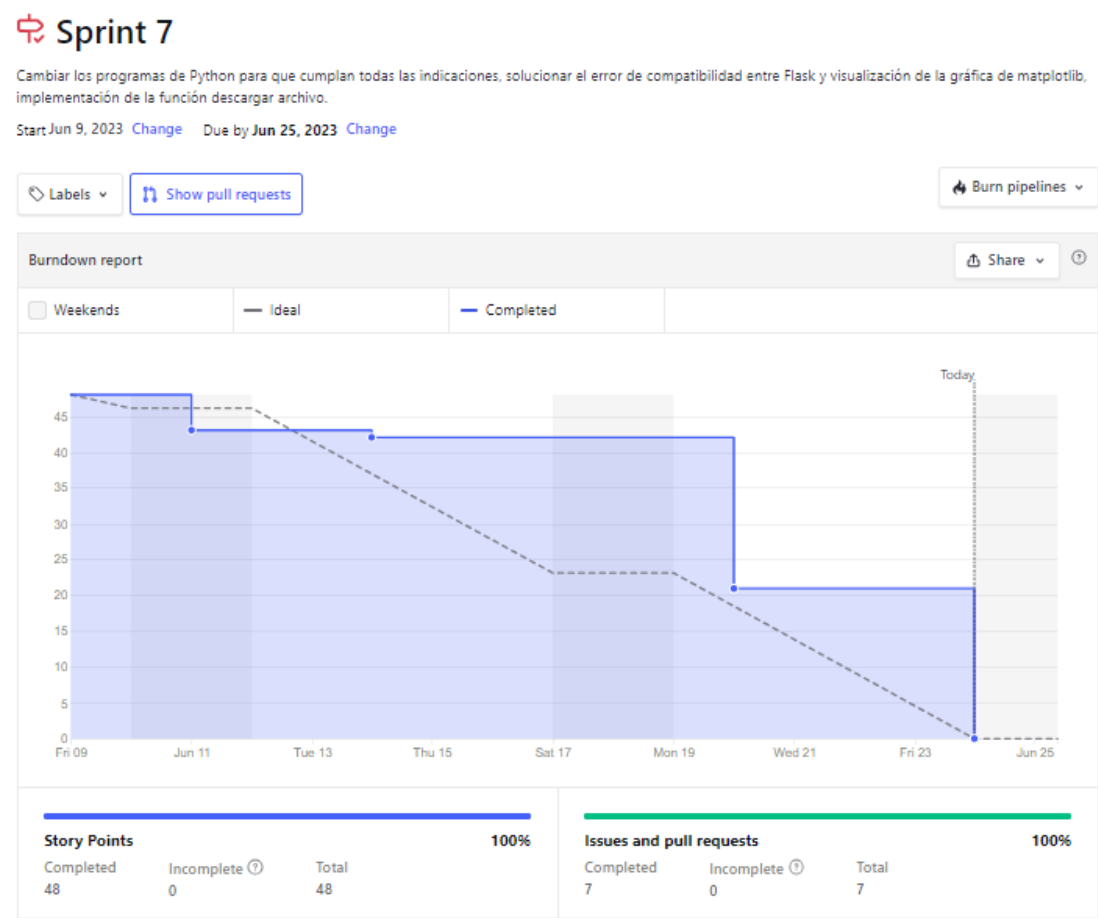
Con las nuevas indicaciones se va a cambiar el código de Python para que el usuario introduzca el número de preguntas que quiere generar y que cada pregunta sea diferente entre ellas. También incluyendo la opción de descargar el fichero creado a partir de los datos introducidos por el usuario, dando la opción de que esté en español o en inglés.

Por último, se ha implementado la internacionalización de los diferentes algoritmos y toda la aplicación web usando JSON en vez de Babel debido a problemas de compatibilidad.

Una vez implementado todas las anteriores funcionalidades se ha procedido al nuevo despliegue en Heroku añadiendo los ficheros configurativos para el correcto funcionamiento y cambiar el código, debido a que se usaba el paquete *tkinter*, que es una biblioteca de GUI (interfaz gráfica de usuario) para Python que se utiliza principalmente en entornos de escritorio y no en entornos de servidor como Heroku.

Enlace al séptimo *sprint*: [Sprint 7](#).

A continuación, se va a mostrar el gráfico *Burndown report* del séptimo *sprint*, donde se visualiza el avance de cada tarea.



Nuevos cambios en el código	programación	8
Descargar fichero	programación	5
Realización de Documentación	documentación	8
2 despliegue en Heroku	despliegue	1
Implementación de la globalización	configuración programación	13
Implementación de globalización en los ficheros Python	programación	8
TOTAL		48

Tabla A.7: Tareas *sprint* 6

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se realizará una comparación costes - beneficios del proyecto desarrollado, en el hipotético caso que se hubiese realizado en un ambiente empresarial, teniendo en cuenta los costes materiales y de personal.

Empezando con los **costes materiales**, tanto hardware como software (ver Tabla A.8). Se tendrán en cuenta los productos que se han empleado en el desarrollo, pero en su versión destinada al ámbito profesional. Además, no se mencionarán los materiales gratuitos empleados, para conseguir una simplificación del estudio.

Material	Valor	Amortizado en
Ordenador portátil y periféricos	1.000,00€	5 años
Costes indirectos	80,00€	1 año
<u>Microsoft Windows 11 Pro Edition</u>	259,00€	5 años

Tabla A.8: Listado con los materiales, su valor y su tiempo de amortización

En la Tabla A.9 podemos observar los costes que han supuesto estos materiales al proyecto, teniendo en cuenta que el periodo de desarrollo del proyecto ha sido de aproximadamente cinco meses (febrero 2023 - junio 2023). También se han añadido costes indirectos, que incluye el gasto de una tarifa a internet que es de 80 €/año y no se han contado los gastos de luz y alquiler de un edificio al desconocer estos datos.

Por ejemplo, para el ordenador portátil y periféricos si costó 1000€ y se amortizará en 5 años, entonces:

$$1000€ / 5 \text{ años} = 200€/año$$

Al ser un proyecto de 5 meses la amortización será de (ver Tabla A.9):

$$200\text{€}/\text{año} \times 1\text{año}/12\text{meses} \times 5\text{meses} = 83,34\text{€}$$

Material	Coste amortizado
Ordenador portátil y periféricos	83,34€
Costes indirectos	33,34€
Microsoft Windows 11 Pro Edition	21,59€
TOTAL	138,27€

Tabla A.9: Coste amortizado para el material

Para los **costes de personal**, se tomará como base que el desarrollo del proyecto lo ha realizado un Ingeniero Informático recién egresado, a tiempo parcial, con un salario bruto de 20.400,00€ anuales [1]. Se ha invertido un total de 280 horas repartidas en 5 meses. Las horas trabajadas a la semana son:

$$280\text{h} / (5\text{meses} \times 4\text{semanas}/\text{mes}) = 14\text{h}/\text{semana}$$

Si se hace una estimación del salario bruto que va a recibir el personal de 20€/hora, será:

$$14\text{h}/\text{semana} \times 20\text{€/h} \times 4\text{semanas}/\text{mes} = 1120\text{€/mes}$$

A continuación, se mostrarán los cálculos de estos costes para los cinco meses que tiene que pagar la empresa [2].

- Seguridad Social. 29,60%.
 - 23,60% contingencias comunes.
 - 4,70% desempleo (tipo general).
 - 0,50% cotización de accidentes de trabajo y enfermedades profesionales.
 - 0,60% formación profesional.
 - 0,20% FOGASA.

$$1120\text{€/mes} / (1 - 0,296) = 1.590,91\text{€/mes}$$

Además, se dispone de dos tutores de apoyo al alumno, con especializados en el tema, por tanto, es lógico que el pago por sus servicios sea mayor (35€/hora). Los tutores, van a guiar al alumno a lo largo del proyecto trabajando un total de 2 horas/semana. Por tanto:

$$2\text{h}/\text{semana} \times 35\text{€/h} \times 4\text{semanas}/\text{mes} = 280\text{€/mes}$$

Son entonces, 560€ brutos entre los dos profesores que habrá que sumar también los impuestos:

$$560\text{€/mes} / (1 - 0,296) = 795,46\text{€/mes entre los 2 tutores}$$

Por lo tanto, finalmente, la empresa tendrá que pagar un coste mensual (ver Tabla A.10).

Concepto	Coste
Salario mensual en bruto trabajador	1.590,91€
Salario mensual bruto 2 tutores	795,46€
TOTAL (para 5 meses)	11.931,85€

Tabla A.10: Desglose del coste de personal.

Para los **costes de software** no se ha tenido que pagar por ningún software, por lo que no habrá costes de software.

Por lo tanto, si sumamos todos los costes, obtenemos que el coste total de la realización del proyecto asciende a:

$$11.931,85\text{€} + 138,27\text{€} = 12.070,12\text{€}$$

En cuanto a los **beneficios**, no se dispone de beneficio alguno, pues la Universidad de Burgos proporcionará este software a sus alumnos de manera gratuita.

Viabilidad legal

En este apartado se tratará el marco legal que envuelve el proceso de desarrollo del proyecto, su distribución, modificación y utilización. Se va a realizar un estudio de las herramientas utilizadas (ver Tabla A.11):

Herramienta/Librería	Versión	Licencia
FLASK	2.3.2	BSD
NUMPY	1.24.3	BSD
MATPLOTLIB	3.7.1	PSF
LXML	4.9.2	BSD
PILLOW	3.5.0	PIL

Tabla A.11: Legalidad de herramientas y librerías

Como se puede apreciar, todas las herramientas y librerías que se han empleado son de uso libre sin ninguna restricción. Por lo tanto, nuestro proyecto necesita una licencia MIT, permitiendo el uso de la aplicación para todos los niveles posibles.

Especificación de Requisitos

B.1. Introducción

En este apartado se describen los objetivos del proyecto que fueron definidos y haciendo un catálogo de cada requisito necesario para conseguir lograr esos objetivos.

Se va a seguir el estándar IEEE 830-1998, el cual define que la Especificación de Requisitos Software (ERS) debe ser [\[3\]](#):

- **Completa:** Todos los requisitos deben estar reflejados en ella y todas las referencias deben estar definidas.
- **Consistente:** Debe ser coherente con los propios requerimientos y también con otros documentos de especificación.
- **Inequívoca:** La redacción debe ser clara de modo que no se pueda mal interpretar.
- **Correcta:** El software debe cumplir con los requisitos de la especificación.
- **Trazable:** Se refiere a la posibilidad de verificar la historia, ubicación o aplicación de un ítem a través de su identificación almacenada y documentada.
- **Priorizable:** Los requisitos deben poder organizarse jerárquicamente según su relevancia para el negocio y clasificándolos en esenciales, condicionales y opcionales.
- **Modificable:** Aunque todo requerimiento es modificable, se refiere a que debe ser fácilmente modificable.
- **Verificable:** Debe existir un método finito sin costo para poder probarlo.
- **Clara:** Debe estar en un lenguaje claro y entendible para quien lo va a atender.

B.2. Objetivos generales

Como ya se comentó en la memoria, los objetivos generales de este proyecto han sido los siguientes:

1. Crear una aplicación web que genere archivos importables en un formato adecuado.
2. Desarrollo de una interfaz en forma de aplicación web mediante la cual los usuarios puedan registrarse.
3. Implementación de las diferentes funcionalidades que tiene la aplicación, como la creación de preguntas para su futura exportación o el tipo de idioma.

B.3. Catálogo de requisitos

A continuación, se van a definir todos los requisitos derivados de los objetivos principales del proyecto.

Requisitos funcionales:

- **R.F-1 Selección tipo preguntas:** La aplicación debe ser capaz de permitir al usuario elegir entre crear preguntas tipo Tabla, Valor Ganado o Programación Ganada.
- **R.F-2 Generación de banco de preguntas:** La aplicación debe ser capaz de generar un banco de preguntas sobre Seguimiento y control de proyectos para la asignatura de Gestión de Proyectos. Un banco de preguntas contiene preguntas que, a su vez, tienen un título, enunciado y cuestiones.
 - **R.F-2.1 Generar preguntas Tabla:** La aplicación debe ser capaz de generar un banco de preguntas relacionadas con los Datos básicos de seguimiento y control de proyectos.
 - **R.F-2.1.1 Definir número de tareas:** el usuario debe ser capaz de introducir un número máximo de tareas superior a seis y menor que veintisiete que formaran parte de cada tabla aleatoria creada.
 - **R.F-2.1.2 Definir rango de costes:** el usuario debe ser capaz de introducir un rango de dos números separados por comas para que el coste de cada tarea valga un número aleatorio entre ese rango.
 - **R.F-2.1.3 Definir número de preguntas a generar:** el usuario debe ser capaz de introducir el número total de preguntas que quiere generar sobre este tipo.
 - **R.F-2.1.4 Definir idioma del fichero:** el usuario debe ser capaz de seleccionar si quiere que en el fichero creado el banco de preguntas estén en inglés o por el contrario en español.

- **R.F-2.2 Generar preguntas Valor Ganado:** La aplicación debe ser capaz de generar un banco de preguntas relacionadas con la metodología del Valor Ganado en el seguimiento y control de proyectos.
 - **R.F-2.2.1 Definir rango de tiempo:** El usuario debe ser capaz de introducir un rango de dos números separados por comas para que en el gráfico de cada pregunta el tiempo total valga un número aleatorio entre ese rango.
 - **R.F-2.2.2 Definir rango de costes:** El usuario debe ser capaz de introducir un rango de dos números separados por comas para que en el gráfico de cada pregunta el coste total valga un número aleatorio entre ese rango.
 - **R.F-2.2.3 Definir número de preguntas a generar:** El usuario debe ser capaz de introducir el número total de preguntas que quiere generar sobre este tipo.
 - **R.F-2.2.4 Definir idioma del fichero:** El usuario debe ser capaz de seleccionar si quiere que en el fichero creado el banco de preguntas estén en inglés o por el contrario en español.
- **R.F-2.3 Generar preguntas Programación Ganada:** La aplicación debe ser capaz de generar un banco de preguntas relacionadas con la metodología de la Programación Ganada en el seguimiento y control de proyectos.
 - **R.F-2.3.1 Definir rango de tiempo:** El usuario debe ser capaz de introducir un rango de dos números separados por comas para que en el gráfico de cada pregunta el tiempo total valga un número aleatorio entre ese rango.
 - **R.F-2.3.2 Definir rango de costes:** El usuario debe ser capaz de introducir un rango de dos números separados por comas para que en el gráfico de cada pregunta el coste total valga un número aleatorio entre ese rango.
 - **R.F-2.3.3 Definir número de preguntas a generar:** El usuario debe ser capaz de introducir el número total de preguntas que quiere generar sobre este tipo.
 - **R.F-2.3.4 Definir idioma del fichero:** El usuario debe ser capaz de seleccionar si quiere que en el fichero creado el banco de preguntas estén en inglés o por el contrario en español.
- **R.F-3 Descargar fichero:** La aplicación debe ser capaz de descargar el fichero, con un nombre genérico y un formato .XML permitiendo su importación y estando en un formato estandarizado por Moodle.

- **R.F-4 Generar nuevo fichero:** La aplicación debe ser capaz de generar un nuevo fichero tanto para un banco de preguntas ya generadas o uno nuevo.
- **R.F-5 Registro de usuarios:** La aplicación debe permitir al usuario crearse una cuenta introduciendo sus datos.
- **R.F-6 Login de usuarios:** La aplicación debe permitir al usuario iniciar sesión introduciendo sus datos.
- **R.F-7 Botones de navegación:** La aplicación debe permitir al usuario el uso de forma correcta de los botones de navegación aportados por la aplicación.
- **R.F-8 Inicio:** La aplicación debe permitir al usuario moverse a la página de “inicio” con el fin de consultar la información general del proyecto y poder usar la aplicación.
- **R.F-9 Acerca de:** La aplicación debe permitir al usuario moverse a la página de “acerca de” con el fin de consultar más información y preguntas frecuentes que aparece allí.
- **R.F-10 Repositorio:** La aplicación debe permitir al usuario moverse a la página del “Repositorio” con el fin poder ver el repositorio en GitHub.
- **R.F-11 Cambio de idioma:** La aplicación debe permitir cambiar el idioma de la página a español o a inglés.

Requisitos no funcionales:

- **R.N.F-1 Usabilidad:** La aplicación debe ofrecer una facilidad de uso, ser intuitiva y accesible, permitiendo su navegación y aprendizaje de una forma rápida.
- **R.N.F-2 Rendimiento:** La aplicación debe optimizar el uso de recursos mejorando la velocidad del tiempo de respuesta.
- **R.N.F-3 Seguridad:** La aplicación debe gestionar todos los datos de una forma segura y sin mantener información vulnerable de forma innecesaria.
- **R.N.F-4 Fiabilidad:** La aplicación debe de ser capaz de crear cada banco de preguntas junto a su respuesta correcta sin equivocarse en los cálculos.
- **R.N.F-5 Escalabilidad:** La aplicación debe ser capaz de crecer y adaptarse a nuevas funcionalidades sin errores.
- **R.N.F-6 Mantenibilidad:** La aplicación debe ser capaz de ofrecer una estabilidad para su escalado, ofreciendo una claridad en su documentación y código.
- **R.N.F-7 Internacionalización:** La aplicación debe estar preparada para cambiar a los idiomas disponibles o para futuros idiomas a implementar.

- **R.N.F-8 Disponibilidad:** La aplicación debe poder estar disponible para todos los usuarios que dispongan de un dispositivo y acceso a internet.

B.4. Especificación de requisitos

En este apartado se va a relacionar los requisitos definidos con los distintos casos de uso de la aplicación local y de la aplicación web (Figura B.1).

A continuación, se va a especificar el nivel de uso de las frecuencias indicadas en los casos de uso.

Nivel	Descripción
BAJO	Frecuencia del caso de uso menor a un 30% de las veces que se usa la aplicación.
MEDIO	Frecuencia del caso de uso entre 30 – 60% de las veces que se usa la aplicación.
ALTO	Frecuencia del caso de uso entre 60 – 90% de las veces que se usa la aplicación.
MUY ALTO	Frecuencia del caso de uso mayor a un 90% de las veces que se usa la aplicación.

Tabla B.1: Nivel de frecuencia de los casos de uso

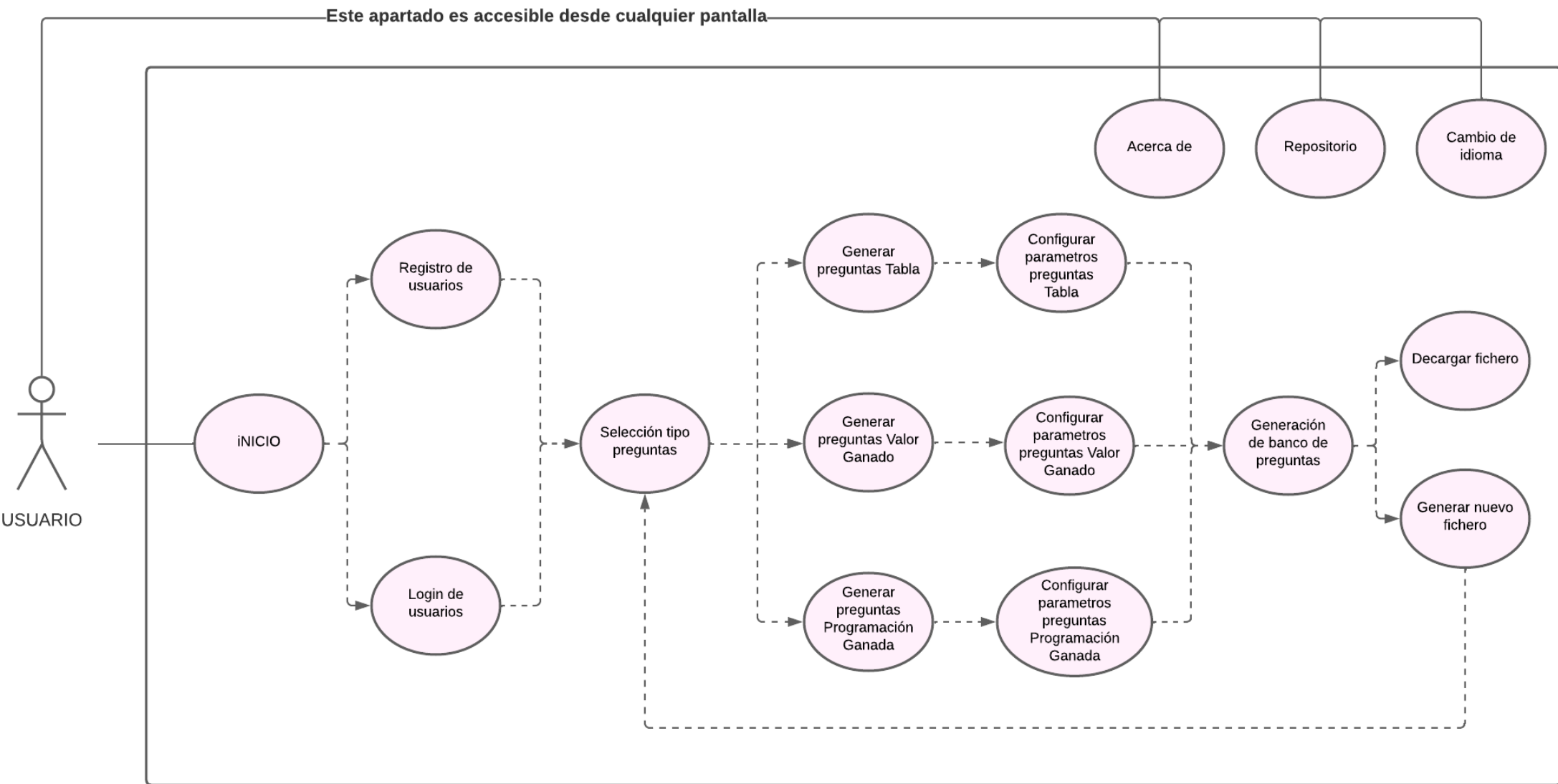


Figura B.1: Casos de uso

CU-1	Registro de usuarios
Requisitos relacionados	R.F-5
Descripción	Permite al usuario registrarse.
Precondiciones	La aplicación debe estar en ejecución, el usuario debe haber pulsado en empezar de la página de inicio, en Login el usuario debe seleccionar la opción de no tienes cuenta, registrarse.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la opción de Crear cuenta. 2. El usuario introduce su nombre, apellido, nombre de usuario y contraseña. 3. Pulsa el botón de registrarse.
Postcondiciones	El formulario de registro no debe tener fallos.
Excepciones	<ol style="list-style-type: none"> 1. Carácter incorrecto (mensaje). 2. No introducir un valor en una casilla (mensaje).
Importancia	Alta
Frecuencia	Muy alta

Tabla B.2: Caso de uso 1 – Registro de usuarios

CU-2	Login de usuarios
Requisitos relacionados	R.F-6
Descripción	Permite al usuario iniciar sesión.
Precondiciones	La aplicación debe estar en ejecución, el usuario debe haber pulsado en empezar de la página de inicio, en registro hay que seleccionar la opción de ya tengo cuenta, iniciar sesión.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la opción de Login. 2. El usuario introduce su nombre de usuario y contraseña. 3. Pulsa el botón de acceder.
Postcondiciones	El formulario de Login no debe tener fallos.
Excepciones	<ol style="list-style-type: none"> 1. Carácter incorrecto (mensaje). 2. No introducir un valor en una casilla (mensaje).
Importancia	Alta
Frecuencia	Alta

Tabla B.3: Caso de uso 2 – Login de usuarios

CU-3	Información general
Requisitos relacionados	R.F-8
Descripción	Permite al usuario acceder a la ventana inicial con la información general de la aplicación.
Precondiciones	Tener conexión a internet.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón de “Inicio”. 2. Se abre otra pestaña con la información general.
Postcondiciones	-
Excepciones	-
Importancia	Alta
Frecuencia	Muy alta

Tabla B.4: Caso de uso 3 – Información general

CU-4	Preguntas frecuentes
Requisitos relacionados	R.F-9
Descripción	Permite al usuario acceder a la ventana de “Más información” con más información sobre la aplicación.
Precondiciones	Tener conexión a internet.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón de “Más información”. 2. Se abre otra pestaña con las preguntas frecuentes.
Postcondiciones	-
Excepciones	-
Importancia	Baja
Frecuencia	Baja

Tabla B.5: Caso de uso 4 – Preguntas frecuentes

CU-5	Ayuda al usuario
Requisitos relacionados	R.F-10
Descripción	Permite al usuario acceder a la página “Repositorio” para conocer el repositorio en GitHub.
Precondiciones	Tener conexión a internet y estar registrado en GitHub.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón de “Repositorio”. 2. Se abre otra pestaña con el repositorio de la aplicación.
Postcondiciones	-
Excepciones	-
Importancia	Baja
Frecuencia	Baja

Tabla B.6: Caso de uso 5 – Ayuda al usuario

CU-6	Cambio de idioma
Requisitos relacionados	R.F-11
Descripción	Permite al usuario cambiar el idioma de toda la aplicación.
Precondiciones	Tener conexión a internet.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona la bandera del idioma que desea. 2. Se cambia el idioma en todas las páginas de la aplicación.
Postcondiciones	Se cambia el idioma
Excepciones	-
Importancia	Alta
Frecuencia	Media

Tabla B.7: Caso de uso 6 – Cambio de idioma

CU-7	Seleccionar tipo de preguntas
Requisitos relacionados	R.F-1, R.F-7
Descripción	Permite al usuario seleccionar que tipo de preguntas quiere crear para el fichero del banco de preguntas.
Precondiciones	Haberse registrado o iniciado sesión.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón Tabla, Valor Ganado o Programación Ganada. 2. Te redirecciona a la página de configuración dependiendo de que pregunta se haya elegido.
Postcondiciones	Se abre la ventana para configurar el tipo de pregunta escogido.
Excepciones	-
Importancia	Muy alta
Frecuencia	Muy alta

Tabla B.8: Caso de uso 7 – Seleccionar tipo de preguntas

CU-8	Generar preguntas Tabla
Requisitos relacionados	R.F-2, R.F-2.1, R.F-2.1.1, R.F-2.1.2, R.F-2.1.3, R.F-2.1.4, R.F-7
Descripción	Permite al usuario escribir las características que desea que tengan cada pregunta Tabla en el fichero del banco de preguntas.
Precondiciones	Haber seleccionado el tipo de preguntas Tabla.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón Tabla. 2. El usuario introduce el número de tareas. 3. El usuario introduce un rango de costes. 4. El usuario introduce el número de preguntas a generar. 5. El usuario indica el idioma del fichero a exportar. 6. El usuario pulsa el botón Generar Fichero.
Postcondiciones	Se abre la ventana para descargar el fichero o cancelar la generación.
Excepciones	No haber introducido alguno de los campos (mensaje).
Importancia	Muy alta
Frecuencia	Alta

Tabla B.9: Caso de uso 8 – Generar preguntas Tabla

CU-9	Generar preguntas Valor Ganado
Requisitos relacionados	R.F-2, R.F-2.2, R.F-2.2.1, R.F-2.2.2, R.F-2.2.3, R.F-2.2.4, R.F-7
Descripción	Permite al usuario escribir las características que desea que tengan cada pregunta de Valor Ganado en el fichero del banco de preguntas.
Precondiciones	Haber seleccionado el tipo de preguntas Valor Ganado.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón Valor Gando. 2. El usuario introduce el rango de tiempo. 3. El usuario introduce un rango de costes. 4. El usuario introduce el número de preguntas a generar. 5. El usuario indica el idioma del fichero a exportar. 6. El usuario pulsa el botón Generar Fichero.
Postcondiciones	Se abre la ventana para descargar el fichero o cancelar la generación.
Excepciones	No haber introducido alguno de los campos (mensaje).
Importancia	Muy alta
Frecuencia	Alta

Tabla B.10: Caso de uso 9 – Generar preguntas Valor Ganado

CU-10	Generar preguntas Programación Ganada
Requisitos relacionados	R.F-2, R.F-2.3, R.F-2.3.1, R.F-2.3.2, R.F-2.3.3, R.F-2.3.4, R.F-7
Descripción	Permite al usuario escribir las características que desea que tengan cada pregunta de Programación Ganada en el fichero del banco de preguntas.
Precondiciones	Haber seleccionado el tipo de preguntas Programación Ganada.
Acciones	<ol style="list-style-type: none"> 1. El usuario presiona el botón Programación Ganada. 2. El usuario introduce el rango de tiempo. 3. El usuario introduce un rango de costes. 4. El usuario introduce el número de preguntas a generar. 5. El usuario indica el idioma del fichero a exportar. 6. El usuario pulsa el botón Generar Fichero.
Postcondiciones	Se abre la ventana para descargar el fichero o cancelar la generación.
Excepciones	No haber introducido alguno de los campos (mensaje).
Importancia	Muy alta
Frecuencia	Alta

Tabla B.11: Caso de uso 10 – Generar preguntas Programación Ganada

CU-11	Exportación del fichero
--------------	--------------------------------

Requisitos relacionados	R.F-3, R.F-7
Descripción	Permite al usuario descargarse el banco de preguntas creado en el formato.XML.
Precondiciones	Tener un banco de preguntas cargado y estar en la página de descargas de la aplicación.
Acciones	<ol style="list-style-type: none"> 1. Se pulsa el botón “Descargar Fichero” 2. Se llama al fichero creado con un nombre genérico dependiendo del tipo de preguntas creado. 3. Se descarga el fichero en formato .XML.
Postcondiciones	Se queda en la misma pantalla, pero con el fichero descargado en el dispositivo del usuario.
Excepciones	-
Importancia	Alta
Frecuencia	Alta

Tabla B.12: Caso de uso 11 – Exportación del fichero

CU-12	Generar nuevo fichero
Requisitos relacionados	R.F-1, R.F-4, R.F-7
Descripción	Permite al usuario generar un nuevo banco de preguntas con un tipo de preguntas igual o diferente.
Precondiciones	Haber creado un fichero de cualquier tipo de preguntas.
Acciones	<ol style="list-style-type: none"> 1. Se pulsa en el botón “Crear otro fichero”
Postcondiciones	Se saltará a la ventana de elección del tipo de preguntas.
Excepciones	-
Importancia	Alta
Frecuencia	Media

Tabla B.13: Caso de uso 12 – Generar nuevo fichero

Especificación de diseño

C.1. Introducción

En este apartado se tratarán los aspectos relativos a la organización y el diseño de la aplicación.

C.2. Diseño de datos

En el diseño de datos, se va a definir la estructura de la base de datos utilizados en el proyecto y donde están almacenados.

Variable de sesión

Estos datos son guardados en variables de sesión.

- **Sesión de usuario:** Se guarda el nombre de usuario de las personas que se han registrado o iniciado sesión.
- **Sesión de fichero:** Se guarda el nombre del tipo de pregunta. Si se guarda el nombre de Tabla, el nombre del fichero será ese nombre más el formato .XML. Si se guarda el nombre de Valor Ganado, el nombre del fichero será ese nombre más el formato .XML. Si se guarda el nombre de Programación Ganada, el nombre del fichero será ese nombre más el formato .XML.
- **Sesión de configuración:** Se guarda la configuración de la red que se está usando.
- **Sesión de idioma:** Se guarda el idioma seleccionado.

La variable fichero servirá para dar nombre al fichero de descarga que hagamos en la aplicación (exportación). La variable del nombre de usuario servirá para saludar al usuario con mensajes de avisos.

Base de datos temporal

Estos datos son guardados de manera temporal mientras se esté usando la aplicación.

- **Id del usuario:** Será el nombre de usuario de las personas registradas o que hayan iniciado sesión.

Cuando el usuario entre en la aplicación y se registra o inicia sesión, su nombre de usuario se queda guardado. Cuando el usuario salga de la aplicación web, la sesión se borrará, por lo que también se borrarán todos los datos de sesión.

Clase Tabla

La clase Tabla, es una clase de Python donde está almacenada todo el algoritmo que realiza las preguntas de tipo Tabla. Contiene los siguientes campos:

- **El valor de tareas:** Que será el máximo número de tareas que puede tener una pregunta.
- **El rango de costes:** Que será el rango mínimo y máximo de costes que puede tener una tarea.
- **El valor de preguntas:** Que será el número de veces que se tiene que ejecutar el programa y crear ese número de preguntas.
- **El valor de idioma:** Que será el idioma que estará traducido todas las preguntas.

Clase ValorGanado

La clase ValorGanado, es una clase de Python donde está almacenada todo el algoritmo que realiza las preguntas de tipo Valor Ganado. Contiene los siguientes campos:

- **El rango de tiempo:** Que será el rango mínimo y máximo de tiempo que puede tener el gráfico del Valor Ganado.
- **El rango de costes:** Que será el rango mínimo y máximo de costes que puede tener el gráfico del Valor Ganado.
- **El valor de preguntas:** Que será el número de veces que se tiene que ejecutar el programa y crear ese número de preguntas.
- **El valor de idioma:** Que será el idioma que estará traducido todas las preguntas.

Clase ProgramacionGanada

La clase ProgramacionGanada, es una clase de Python donde está almacenada todo el algoritmo que realiza las preguntas de tipo Programación Ganada. Contiene los siguientes campos:

- **El rango de tiempo:** Que será el rango mínimo y máximo de tiempo que puede tener el gráfico de la Programación Ganada.
- **El rango de costes:** Que será el rango mínimo y máximo de costes que puede tener el gráfico de la Programación Ganada.
- **El valor de preguntas:** Que será el número de veces que se tiene que ejecutar el programa y crear ese número de preguntas.
- **El valor de idioma:** Que será el idioma que estará traducido todas las preguntas.

Clase main

La clase main, es una clase de Python donde está almacenada toda la lógica de la aplicación, es decir, donde están todos los métodos encargados de configurar la funcionalidad de todo el proyecto.

C.3. Diseño procedimental

En este apartado se va a explicar la ejecución de los diferentes procesos más interesantes de la aplicación web. En la siguiente figura se puede apreciar el diagrama de secuencias indicando las diferentes interacciones de los diferentes componentes del sistema para la tarea de realizar un banco de preguntas.

Registro e inicio de sesión

Para el primer diagrama de secuencia se va a representar los pasos que sigue el programa en el momento en el que el usuario pulsa el botón de “Empezar” y necesita registrarse o iniciar sesión para acceder a la aplicación.

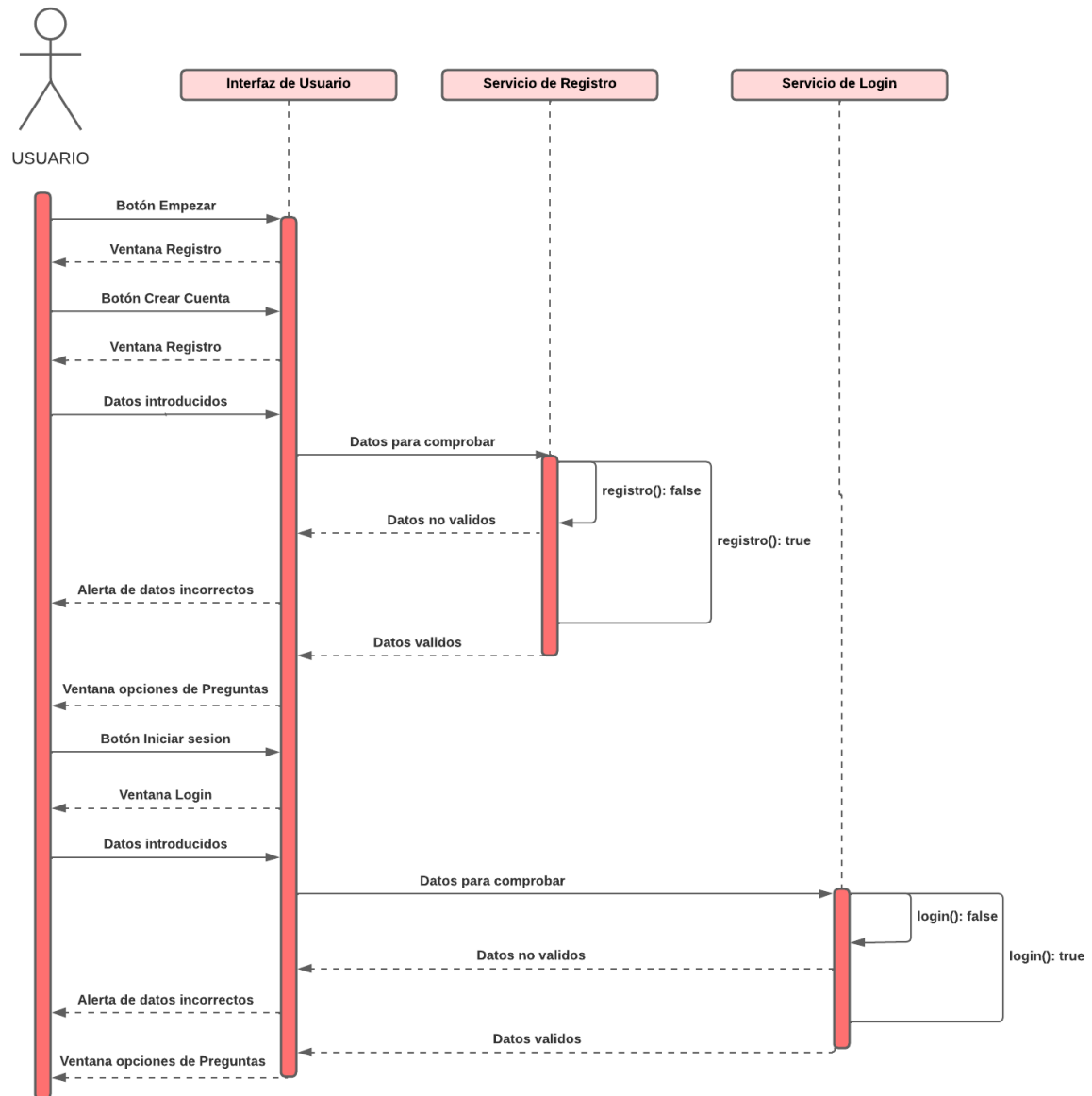


Figura C.1: Diagrama de secuencia de Registro y Login

Creación banco de preguntas

Para el siguiente diagrama se centra en como se realiza el proceso de crear la exportación del fichero. Como los tres tipos de preguntas tienen un mismo diagrama secuencial solo se mostrará uno de ellos.

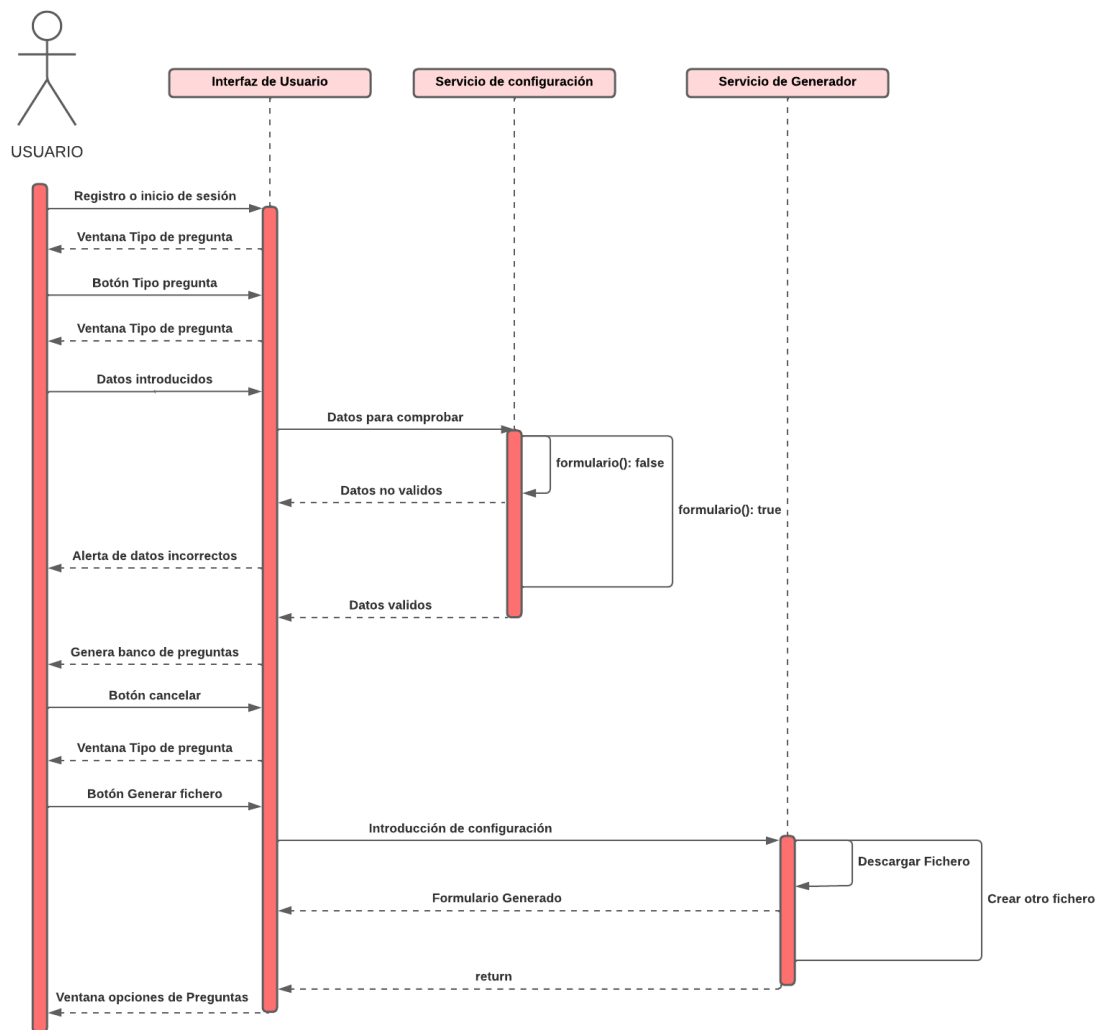


Figura C.2: Diagrama de secuencia de Creación banco de preguntas

C.4. Diseño arquitectónico

En este apartado se va a explicar el patrón de diseño que se ha utilizado para que cumpla con la mantenibilidad y un correcto diseño del software.

Modelo-Vista-Controlador

Modelo-vista-controlador (MVC), es un diseño de arquitectura de software, que separa los datos y la lógica de ejecución de la aplicación web, de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Es por ello por lo que se construye a partir de tres componentes distintos que son el modelo, la vista y el controlador. Por un lado, se define los componentes para la representación de la información, y por otro lado para la interacción del usuario. [\[4\]](#)

En nuestra aplicación cada uno de estos componentes abarca:

- **Modelo:** La lógica de la base de datos, es decir, el patrón que siguen los datos del programa.
En MQGenerator, el modelo está compuesto por la base de los datos y la estructura de los programas para extraer esos datos.
- **Vista:** La definición de todas las interfaces del usuario.
En MQGenerator, la vista está compuesto por todos los ficheros HTML que tienen la aplicación web, donde se encuentran agrupados en el directorio /templates.
- **Controlador:** Las partes del proyecto que manda realizar las operaciones, es decir, los que controlan cómo se envían los datos y como recuperarlos.
En MQGenerator, el controlador está compuesto por el archivo main.py. Además, existen diferentes algoritmos en Python incluidos en el directorio /src.
- clases .CS que hay detrás de cada página, encargadas de la gestión de eventos de usuario y llamadas a métodos.

En la Figura C.3 se puede observar un sencillo diagrama arquitectónico de este diseño:

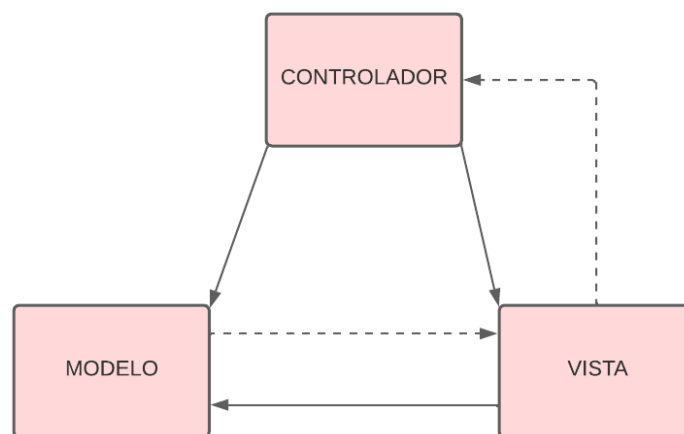


Figura C.3: Diseño arquitectónico – Modelo-Vista-Controlador

Fachada (Facade)

El patrón Fachada o "Facade" es un patrón de diseño de software estructural. Fue creado para estructurar un entorno de programación de forma sencilla y reducir su complejidad (ver Figura C.4) con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos. Por lo tanto, se aplicará cuando se necesite proporcionar una interfaz simple para un subsistema complejo, o cuando se quiera estructurar varios subsistemas en capas, ya que las fachadas serían el punto de entrada a cada nivel. [\[5\]](#)

En MQGenerator, la fachada es el módulo *main.py* ya que comunica y controla todos los módulos con el usuario, realizando la acción de intermediario.

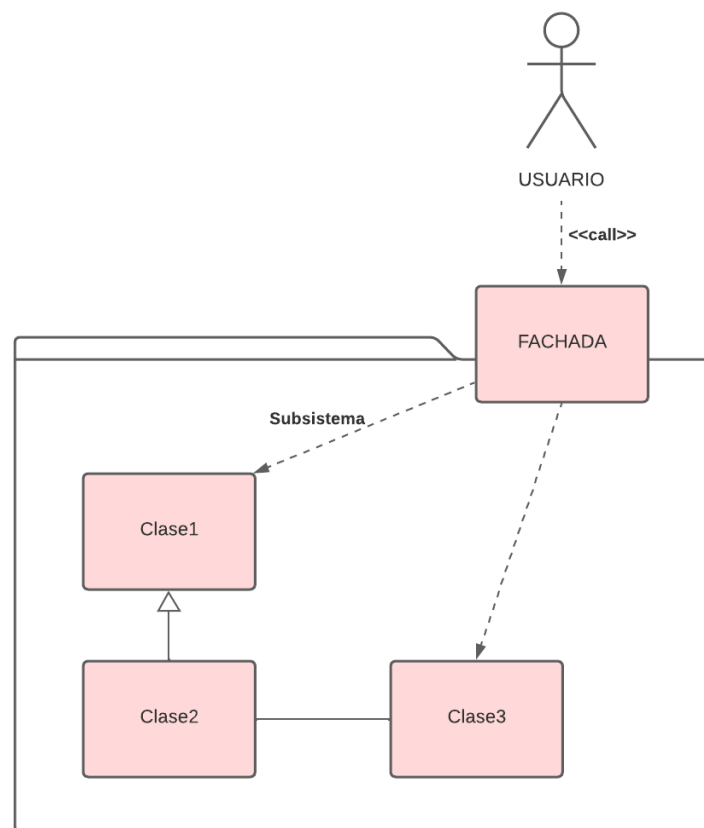
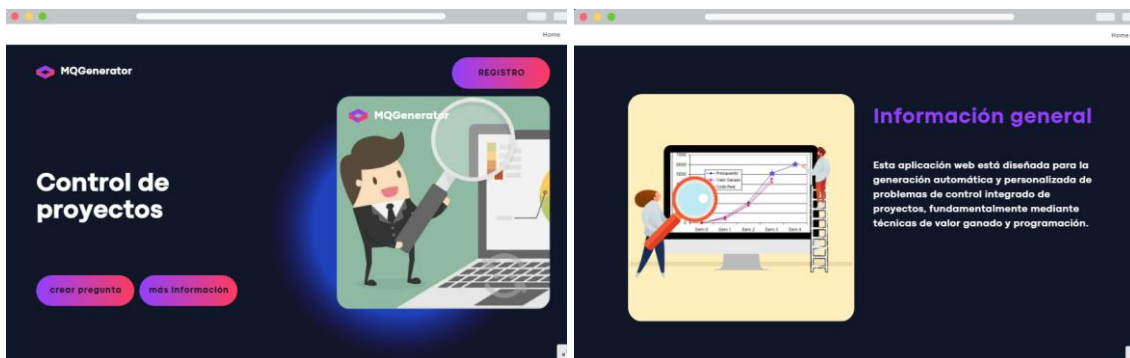


Figura C.4: Diseño arquitectónico - Fachada

C.5. Diseño de interfaces

En este apartado se va a mostrar como fue el prototipo de la interfaz de la aplicación, utilizando la herramienta Canva.



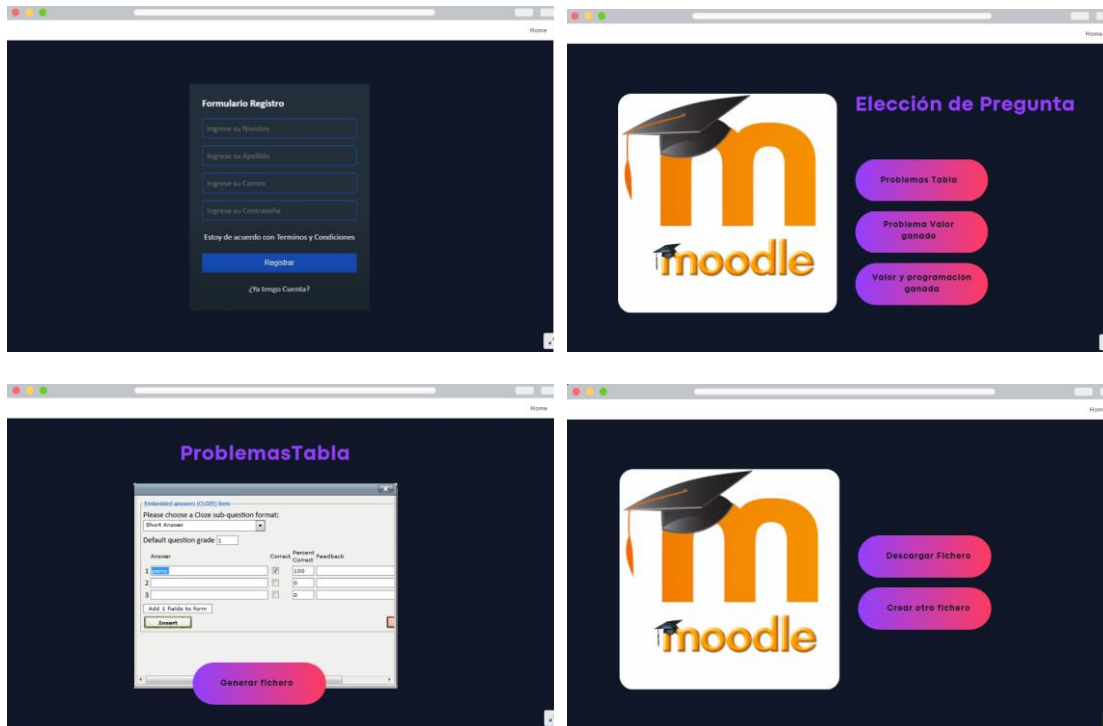





Figura C.5: Prototipo de la interfaz


A partir de este prototipo sirvió para tomar las decisiones de las funcionalidades que quería que estuviesen en este proyecto, y que visión debía tener la aplicación a nivel de usuario. Evidentemente se ha cambiado el diseño para que tuviese una interfaz más familiar y más clara para que el usuario no se canse cuando lo utiliza mucho tiempo.


Cada una de las páginas finales de la aplicación web han sido creadas desde cero, como el registro, la barra del menú, el logotipo, etcétera. El resultado final de la aplicación es la siguiente:




 Inicio Repositorio Más información

Bienvenido  

 UNIVERSIDAD DE BURGOS





INFOMACIÓN GENERAL

¿Qué es MQGenerator?

MQGenerator es una aplicación web desarrollada como TFG del grado de ingeniería informática en la Universidad de Burgos que permite generar de forma automática cuestionarios para la asignatura de Gestión de Proyectos de una forma sencilla.


¿Quién integra el proyecto?

El proyecto de MQGenerator ha sido iniciado por el alumno del grado de Ingeniería Informática Álvaro Hoyuelos Martín, tutorizado por Dña. Virginia Ahedo y D. Jose Manuel Galán.


¿Qué tipo de preguntas están disponibles?


Actualmente contamos con preguntas CLOZE donde genera una tabla o una gráfica aleatoria y calcula las soluciones a las posibles preguntas.


¿Qué herramientas se han utilizado para el proyecto?




Registrate

 name

 surname

 user

 password

Registrar

¿Ya tienes una cuenta? Iniciar Sesión

 Inicio Repositorio Más información

Bienvenido  

Tabla

Valor Ganado

Programación Ganada

MQ Inicio Repositorio Más información Bienvenido

Suponga un proyecto en el que en el momento actual y de acuerdo al plan de proyecto, las actividades desde la A hasta la E deberían estar ya completadas y el resto de actividades no comenzadas.

El informe de seguimiento del proyecto es el siguiente:

Tareas	CPTP	CRTR	%Completado
A	1900	3700	100%
B	1600	7900	100%
C	4000	2100	100%
D	7300	1900	100%
E	7500	5300	100%
F	6400	5700	30%
G	1500		
H	6900		

Calcule:

El valor planificado (PV) en el momento actual

El coste actual (AC) en el momento actual

El valor ganado (EV) en el momento actual

El CPI del proyecto (con dos decimales)

El SPI del proyecto (con dos decimales)

Introduce el número máximo de tareas, mínimo 6

Introduce el rango de costes para CPTP y CRTR separado por comas

Introduce el número de preguntas que quieres generar

Selecciona el idioma que quieres crear el fichero Español ▼

Figura C.6: Interfaz final

Cabe resaltar que con la implementación de la internacionalización dependiendo del idioma que el usuario haya seleccionado la interfaz y las imágenes de cada página pueden cambiar para que se adapten a ese idioma.

Documentación técnica de programación

D.1. Introducción

En este apéndice se va a documentar los aspectos relacionados a la estructura del repositorio, cómo descargarse y compilar el repositorio, el proceso de despliegue entre otros. Todo esto con el objetivo de facilitar el trabajo a futuros colaboradores que quieran crear su propia mejorar la aplicación web incluyendo nuevas funcionalidades.

D.2. Estructura de directorios

Se explicará continuación la estructura de directorios del repositorio principal en [GitHub](#) del proyecto.

- **/:** Directorio raíz del repositorio. En él se encuentran todos los directorios del repositorio y los archivos.
- **/.idea:** La configuración del proyecto con un conjunto de archivos .XML, especifica la configuración predeterminada del proyecto recién creado.
- **/.vscode:** Directorio creado por el editor de código fuente Visual Studio Code, almacena la configuración y personalización del editor.
- **/doc:** Directorio que contiene toda la documentación del proyecto.
- **/src:** Directorio que contiene los archivos .py a los que llama la función principal *main* para realizar todas las operaciones.
- **/src/ProgramacionGanada.py:** Fichero encargado de generar el banco de preguntas tipo Programación Ganada.

- **/src/Tabla.py**: Fichero encargado de generar el banco de preguntas tipo Tabla.
- **/src/ValorGanado.py**: Fichero encargado de generar el banco de preguntas tipo Valor Ganado.
- **/static**: Directorio donde se encuentran la configuración de estilos que se van a aplicar a la aplicación web y las imágenes que se van a mostrar para cada una de las páginas de la página.
- **/static/css**: Directorio que contiene los ficheros .css para la configuración de la interfaz de la aplicación web.
- **/static/img**: Directorio donde se encuentran las imágenes que se van a utilizar en la aplicación web.
- **/static/js**: Directorio que contiene los ficheros .js para poder realizar ciertas acciones en la web.
- **/static/js/app.js**: Archivo que permite realizar acciones como saltar alertas o poder comprobar y redireccionar un botón.
- **/static/js/traductor.js**: Archivo que permite traducir toda la aplicación entera y cambiar las imágenes dependiendo de que idioma esté la página.
- **/static/lang**: Directorio que almacena las traducciones de la aplicación web.
- **/templates**: Directorio con las diferentes páginas .html que contiene la parte visual de la aplicación web.
- **/LICENCE**: Fichero con la licencia del proyecto.
- **/Procfile**: Archivo que especifica cuál es el archivo principal de la aplicación para el despliegue en Heroku.
- **/ProgramacionGanada.xml**: Archivo ejemplo de un posible banco de preguntas de Programación Ganada.
- **/README.md**: Archivo que contiene una descripción del proyecto y como utilizarlo en otro dispositivo.
- **/Tabla.xml**: Archivo ejemplo de un posible banco de preguntas de Tabla.
- **/ValorGanado.xml**: Archivo ejemplo de un posible banco de preguntas de Valor Ganado.
- **/main.py**: Archivo Python principal que realiza las llamadas a los demás archivos y configura todo el proyecto.
- **/requirements.txt**: Archivo utilizado para poder instalar todos los paquetes y herramientas necesarios para este proyecto.

- **/runtime.txt**: Archivo para que cuando se despliegue en Heroku detecte el lenguaje de programación y su versión que se emplea en la aplicación.

D.3. Manual del programador

En este apartado se tratará cómo un programador puede descargar el proyecto y desarrollar su propia versión. Se describen los pasos para obtener las herramientas necesarias y configurar el entorno de desarrollo.

Ejecución en local

Como ya se ha comentado, el código se encuentra disponible en el [repositorio de GitHub](https://github.com/xhm1001/MQGenerator). Por lo tanto, se puede descargar mediante la aplicación web (ver Figura D.1) o mediante la herramienta Git y el comando `git clone https://github.com/xhm1001/MQGenerator`.

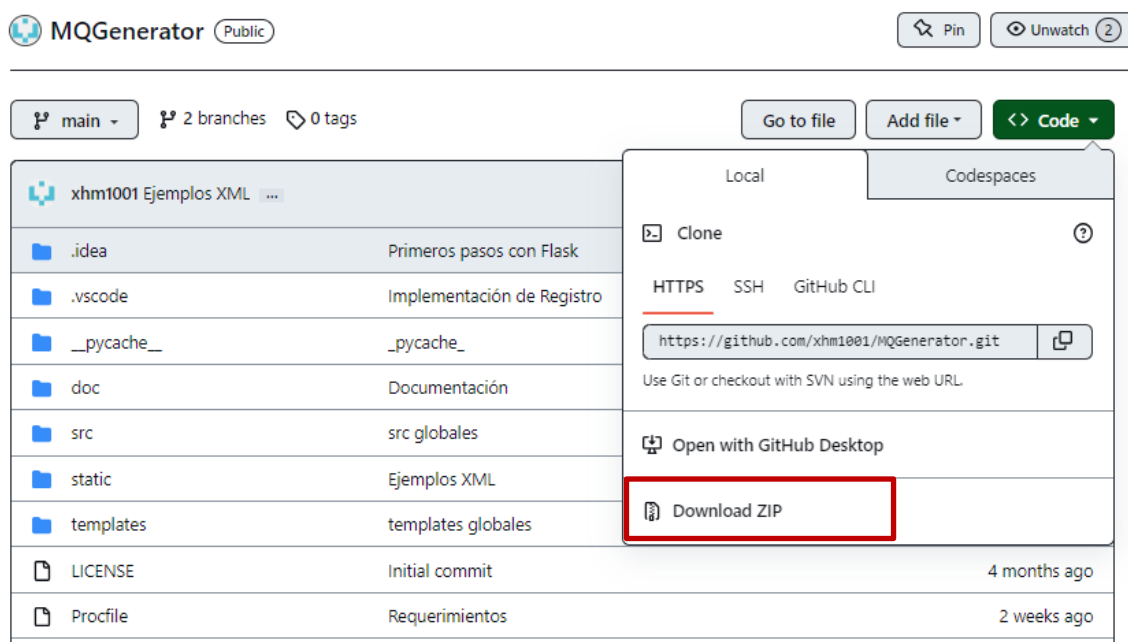


Figura D.1: Descarga del repositorio mediante el entorno web.

Si se desea hacer cambios creando una nueva versión de la aplicación, lo mejor sería realizar un *fork* en GitHub para que el control de versiones se pudiera visualizar de una mejor manera.

Componentes

Una vez descargado se deberán instalar las herramientas utilizadas para el funcionamiento del proyecto (ver Tabla D.1). Para ello debemos situarnos en la terminal de nuestro desarrollador, abrir la carpeta del proyecto y ejecutar el comando:

pip install -r requirements.txt

Los diferentes componentes que se van a descargar son:

Herramienta/Librería	Versión
FLASK	2.3.2
NUMPY	1.24.3
MATPLOTLIB	3.7.1
LXML	4.9.2
PILLOW	3.5.0

Tabla D.1: Versión de los componentes

La versión necesaria que se debe tener de Python es Python 3.9.13.

Una vez ejecutado el anterior comando, ya se podrán ejecutar la aplicación en modo local. Para ello se debe abrir la terminal del sistema, situarse en la carpeta del proyecto y ejecutar el siguiente comando:

python main.py

Y la línea de comandos deberá mostrar, con diferente ruta de directorios, una dirección(ver Fig. D.2) <http://127.0.0.1:5000> y ya podremos utilizar la aplicación en local.

```

C:\Users\alvar\Desktop\ALVARO\MQGenerator>python main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a prod
uction WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 932-713-420

```

Figura D.2: Ejecución en remoto

Despliegue

Como ya se comentó en la memoria, Heroku es una herramienta de pago para cualquier usuario que no sea estudiante. Es por ello por lo que se necesita conectar una cuenta de estudiante de GitHub con Heroku, ya que de esta manera obtienes los créditos suficientes para desplegar un proyecto durante varios meses.

Para desplegar la aplicación en Heroku se puede realizar de dos maneras diferentes, por la línea de comandos siguiendo el siguiente [tutorial](#) seleccionando el lenguaje Python, o a través de la web de Heroku que es como se ha realizado en la primera versión.

Antes de poder realizar un despliegue en Heroku es necesario configurar el proyecto con tres ficheros obligatorios para su correcto funcionamiento:

- **Procfile:** Archivo que especifica cuál es el archivo principal de la aplicación para el despliegue en Heroku.
- **Requirements.txt:** Archivo utilizado para poder instalar todos los paquetes y herramientas necesarios para este proyecto.
- **Runtime.txt:** Archivo para que cuando se despliegue en Heroku detecte el lenguaje de programación y su versión que se emplea en la aplicación.

Una vez se tengan estos tres ficheros y una cuenta en Heroku, en la barra de navegación aparecerá la opción de “*Create new app*” (ver Fig. D.3) y nos aparecerá la opción de indicar el nombre que queremos darle a nuestra aplicación.

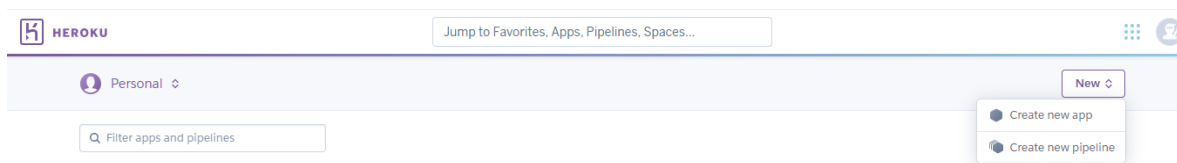
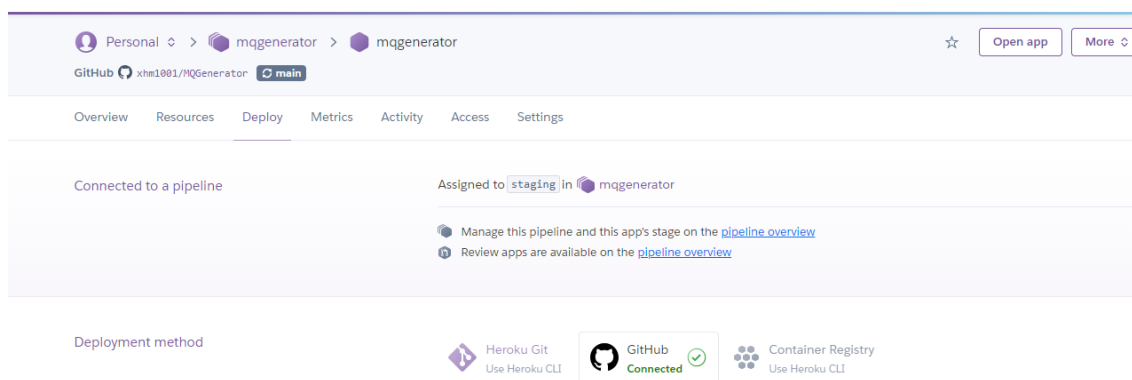


Figura D.3: Nueva aplicación en Heroku

Cuando tengamos el paso anterior debemos configurar en la pantalla *Deploy* para que se conecte al repositorio de GitHub y que se actualice de manera automática cualquier cambio que se realice en el repositorio (ver Fig. D.4).



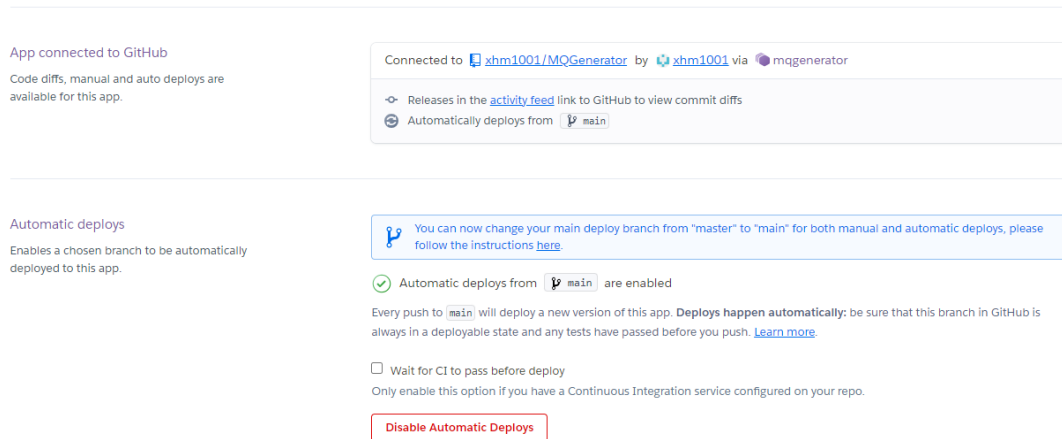


Figura D.4: Configuración en Heroku

Finalmente, solo tenemos que realizar el despliegue pulsando en el *Manual Deploy* la primera opción “*Deploy branch*” y esperar a que aparezca en pantalla lo siguiente:

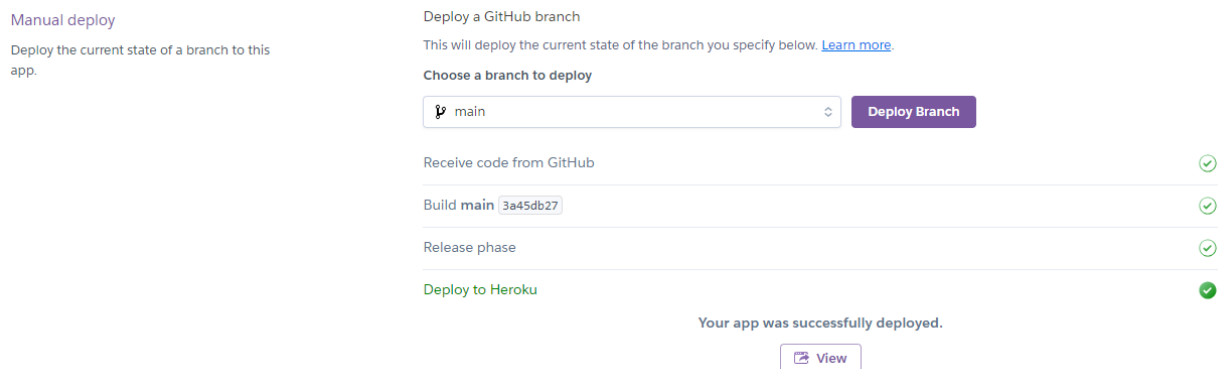


Figura D.5: Resultado del despliegue

D.4. Instalación de las herramientas

Microsoft Visual Studio Code

Microsoft Visual Studio Code es un editor de código fuente multiplataforma desarrollado por Microsoft muy versátil. Esto es debido a que se pueden instalar distintas extensiones proporcionados por la aplicación mediante un buscador, permitiendo trabajar con la mayoría de los lenguajes de programación de forma intuitiva, cómoda y rápida.

Para descargar esta herramienta se debe visitar esta página web:

<https://code.visualstudio.com/download> (ver Fig. D.6) y seleccionar la versión que utilice cada sistema.

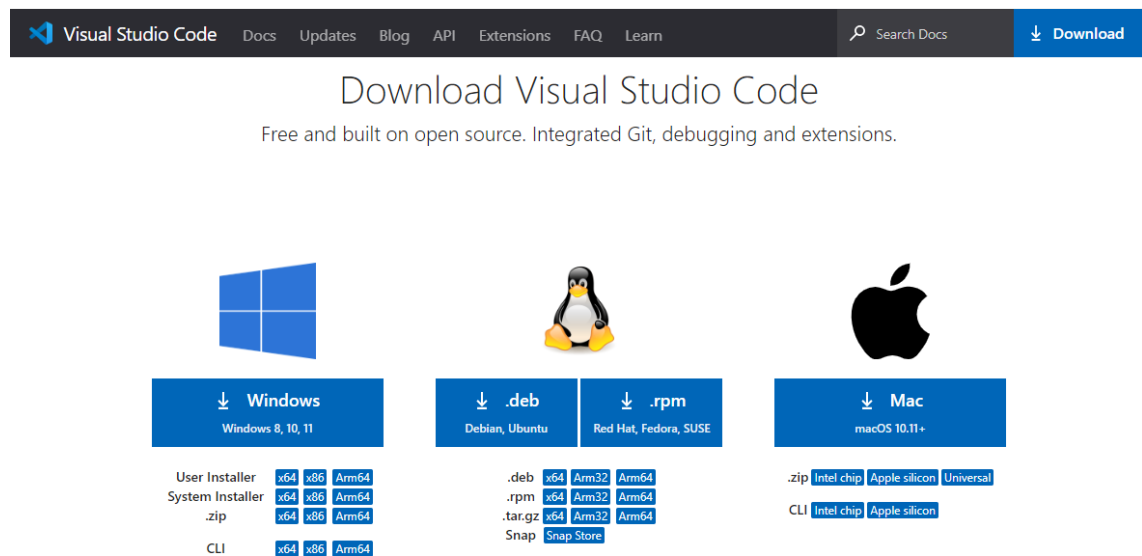


Figura D.6: Descargar Microsoft Visual Studio Code

Abrimos el archivo .exe instalado y seguimos los pasos que aparecen en pantalla. Una vez realizados el anterior paso, ya es posible empezar a trabajar con Visual Studio Code.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice se va a tratar los temas relacionado con la parte del usuario. Se tratarán los temas relativos a requisitos, instalación de la aplicación y el manual de uso del usuario para usar el proyecto de manera efectiva.

E.2. Requisitos de usuarios

Para poder acceder a la aplicación, el usuario lo puede hacer de forma remota o de forma local. Para acceder de forma remota debe abrir su navegador e introducir el siguiente enlace <https://mggenerator.herokuapp.com/> , en cambio si quiere acceder de forma local debe acceder al siguiente enlace <http://127.0.0.1:5000> y ya podrá utilizar la aplicación web sin límites.

El usuario debe cumplir con los siguientes requisitos:

- **Sistema operativo Windows:** Esto es debido a que la aplicación fue desarrollada para ser utilizada en este sistema operativo.
- **Navegador:** Es el principal requisito para poder abrir la aplicación web y acceder a sus páginas, el utilizado principalmente es Google Chrome.

E.3. Instalación

En cuanto a la instalación, no existe un proceso de ya que los ficheros y librerías necesarias ya se incluyen dentro de la propia aplicación desplegada.

Para su uso en local sí que son necesarios instalar la información ya explicada en el apartado “Manual del programador” de este documento.

E.4. Manual del usuario

En este apartado se va a destinar a redactar un sencillo manual que el usuario pueda emplear para entender cómo utilizar la aplicación en cada una de las páginas que contienen la aplicación.

Recomendaciones antes de utilizar la aplicación

Se recomienda no utilizar los botones de navegación para retroceder, refrescar o avanzar a otra página porque puede generar errores con las sesiones.

Barra de navegación

En cada una de las pantallas salvo en la pantalla de Registro e Inicio de sesión se dispone de una barra de navegación, que permite acceder de forma más rápida a diferentes funcionalidades.

- **Inicio:** Cuando pulsamos volvemos al inicio de la aplicación donde se encuentra la información general y poder crear el banco de preguntas.
- **Repositorio:** Cuando pulsamos nos redirecciona a la página oficial del repositorio en GitHub de este proyecto.
- **Más información:** Cuando pulsamos volvemos a la página donde se encuentra más información relevante del proyecto junto con las herramientas utilizadas.
- **Cambio de idioma:** Cuando pulsamos en cualquiera de los dos botones con una imagen de una bandera, cambia completamente el idioma de todas las páginas de la aplicación.

Página de inicio

Una vez se haya abierto la aplicación web, se nos abrirá una ventana del navegador web por defecto accediendo a la dirección principal. Esta será la página principal del proyecto donde se encuentra un conjunto de párrafos dando la bienvenida al usuario y explicando por encima la descripción del proyecto y como funciona.

Para poder empezar a utilizar la función principal de la aplicación es necesario pulsar el botón de “Empezar” el cual nos llevara a la pantalla de Registro.



Figura E.1: Página principal de la aplicación web

Página de Más información

En esta página se muestran toda la información relevante de la aplicación respondiendo a las preguntas generales que se hacen los usuarios incluyendo algunas de las herramientas junto a su dirección principal que se han empleado durante todo el proyecto.

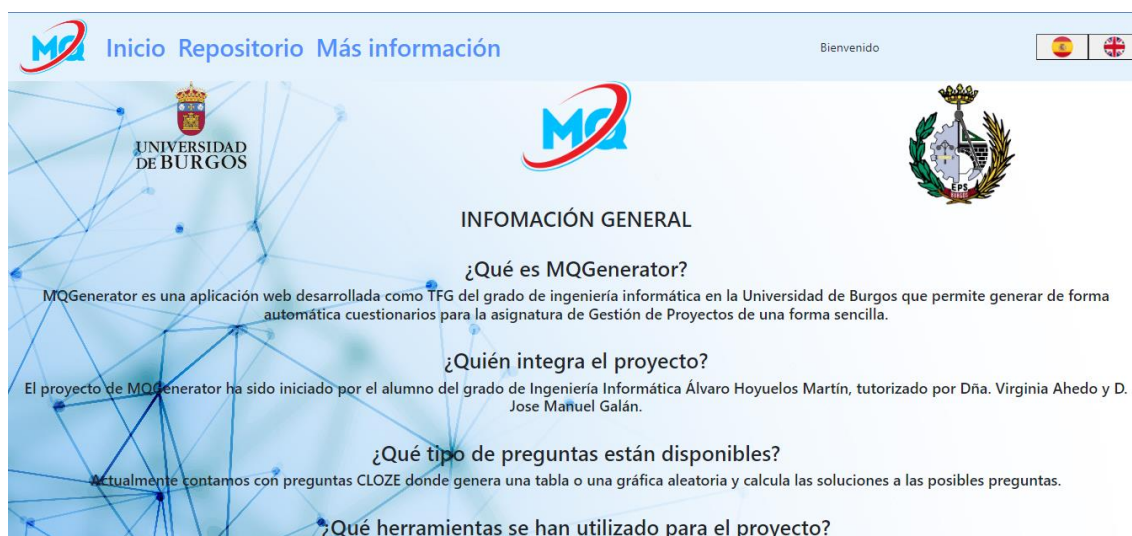


Figura E.2: Página de Más información

Página de Registro

En esta página el usuario se puede registrar para poder utilizar la aplicación. Para ello necesita introducir, sin unos caracteres no permitidos porque sino salta una alerta, el nombre, la contraseña, el nombre de usuario y una contraseña. Cuando se pulsa en el botón de “Registrar” y están todos los campos rellenados correctamente ya se podrán crear ficheros con bancos de preguntas.

Si se pulsa el botón de “Iniciar sesión” te redireccionará a la página de Iniciar sesión.

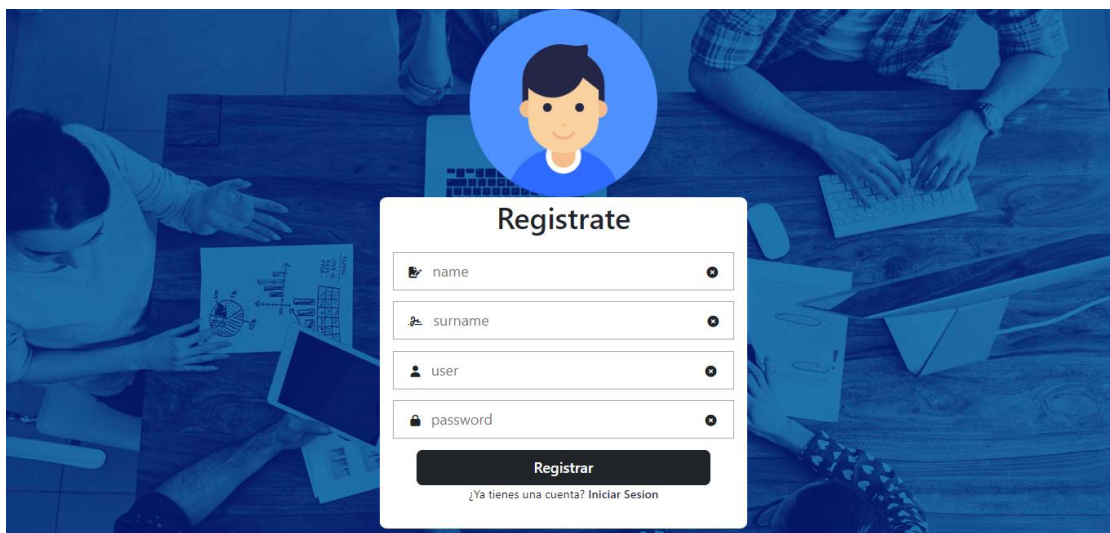


Figura E.3: Página de Registro

Página de Inicio sesión

En esta página el usuario puede Iniciar sesión para poder utilizar la aplicación. Para ello necesita introducir, sin unos caracteres no permitidos porque si no salta una alerta, el nombre de usuario y la contraseña. Cuando se pulsa en el botón de “Entrar” y están todos los campos rellenados correctamente ya se podrán crear los ficheros.

Si se pulsa el botón de “Regístrate” te redireccionará a la página de Iniciar sesión.

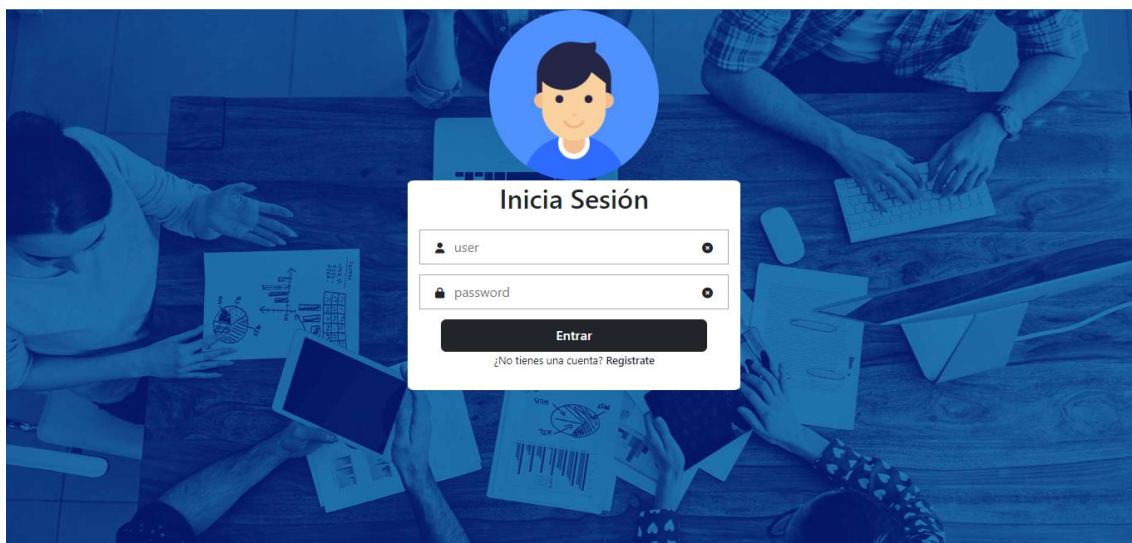


Figura E.4: Página de Inicio sesión

Selección de preguntas

En esta página el usuario se encuentra un menú de selección de los tres diferentes tipos de preguntas que el usuario puede generar un banco de preguntas.

El botón “Tabla” es el encargado de configurar las preguntas tipo Tabla.

El botón “Valor Ganado” es el encargado de configurar las preguntas tipo Valor Ganado.

El botón “Programación Ganada” es el encargado de configurar las preguntas tipo Programación Ganada.



Figura E.5: Selección de preguntas

Preguntas Tabla

Esta página se mostrará cuando el usuario haya pulsado el botón “Tabla”.

El usuario debe introducir para cada casilla un número o un rango de números separados por comas, donde viene explicado con un texto lo que hay que introducir en cada uno. También debe seleccionar el idioma entre español o inglés que quiere crear las preguntas del banco de preguntas.

Finalmente, si el usuario pulsa el botón de “Cancelar” será redirigido otra vez a la página de selección de preguntas o si pulsa en el botón de “Generar Fichero” la aplicación generará un fichero Tabla.xml que dependiendo del número de preguntas que se quieren generar tardará más o menos en cargar la siguiente página.

Suponga un proyecto en el que en el momento actual y de acuerdo al plan de proyecto las actividades desde la A hasta la E deberían estar ya completadas y el resto de actividades no comenzadas.

El informe de seguimiento del proyecto es el siguiente:

Tareas	CPTP	CRTR	%Completado
A	1900	3700	100%
B	1600	7900	100%
C	4000	2100	100%
D	7300	1900	100%
E	7500	5300	100%
F	6400	5700	30%
G	1500		
H	6900		

Calcule:

El valor planificado (PV) en el momento actual

El coste actual (AC) en el momento actual

El valor ganado (EV) en el momento actual

El CPI del proyecto (con dos decimales)

El SPI del proyecto (con dos decimales)

Introduce el número máximo de tareas, mínimo 6

Introduce el rango de costes para CPTP y CRTR separado por comas

Introduce el número de preguntas que quieres generar

Selecciona el idioma que quieres crear el fichero Español ▼

Figura E.6: Preguntas Tabla

Obtener fichero Tabla

Esta página se mostrará cuando el usuario haya pulsado el botón “Generar Fichero” en la página preguntas tabla

Si el usuario pulsa el botón de “Descargar Fichero” se descargará un fichero de nombre Tabla.xml en el dispositivo del usuario y se mantendrá en la misma página por si se quiere descargar el fichero más de una vez.

Si el usuario pulsa en el botón de “Crear otro fichero” será redirigido otra vez a la página de selección de preguntas.

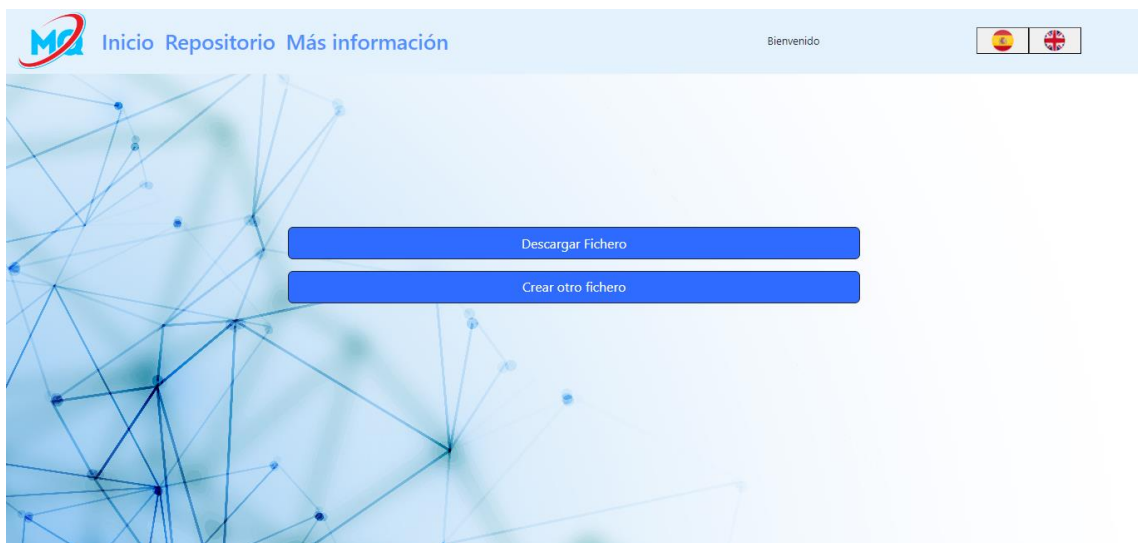


Figura E.7: Descargar fichero Tabla

Preguntas Valor Ganado

Esta página se mostrará cuando el usuario haya pulsado el botón “Valor Ganado”.

El usuario debe introducir para cada casilla un número o un rango de números separados por comas, donde viene explicado con un texto lo que hay que introducir en cada uno. También debe seleccionar el idioma entre español o inglés que quiere crear las preguntas del banco de preguntas.

Finalmente, si el usuario pulsa el botón de “Cancelar” será redirigido otra vez a la página de selección de preguntas o si pulsa en el botón de “Generar Fichero” la aplicación generará un fichero ValorGanado.xml que dependiendo del número de preguntas que se quieren generar tardará más o menos en cargar la siguiente página.

 This screenshot shows the 'Preguntas Valor Ganado' form. On the left, there is a graph titled 'Le presentan el informe de valor ganado de un proyecto en curso (ver figura)'. The graph plots 'Coste acumulado (€)' on the y-axis (0 to 7000) against 'Tiempo (meses)' on the x-axis (0 to 10). It shows three curves: PV (red), EV (green), and AC (blue). Key values are marked: 3600.00 € for AC at month 4, 5700.00 € for EV at month 6, and 6400.00 € for PV at month 6. Below the graph, there are radio button questions: 'En vista de los datos, ¿cómo va el proyecto en costes?' (Con sobrecoste, De acuerdo a lo planificado, Con ahorro en coste) and '¿Y en tiempo?' (En adelanto, De acuerdo a lo planificado en plazo, En retraso). A text input field asks for the 'última estimación de coste final para el proyecto haría? (en €)'. On the right, there are three input fields with labels: 'Introduce el rango de tiempo final (meses) separado por comas' (example: 10,20), 'Introduce el rango de coste acumulado final (€) separado por comas' (example: 2000,50000), and 'Introduce el número de preguntas que quieres generar'. At the bottom right, there is a language selection dropdown currently set to 'Español'.

Figura E.8: Preguntas Valor ganado

Obtener fichero Valor Ganado

Esta página se mostrará cuando el usuario haya pulsado el botón “Generar Fichero” en la página preguntas valor ganado.

Si el usuario pulsa el botón de “Descargar Fichero” se descargará un fichero de nombre ValorGanado.xml en el dispositivo del usuario y se mantendrá en la misma página por si se quiere descargar el fichero más de una vez.

Si el usuario pulsa en el botón de “Crear otro fichero” será redirigido otra vez a la página de selección de preguntas.

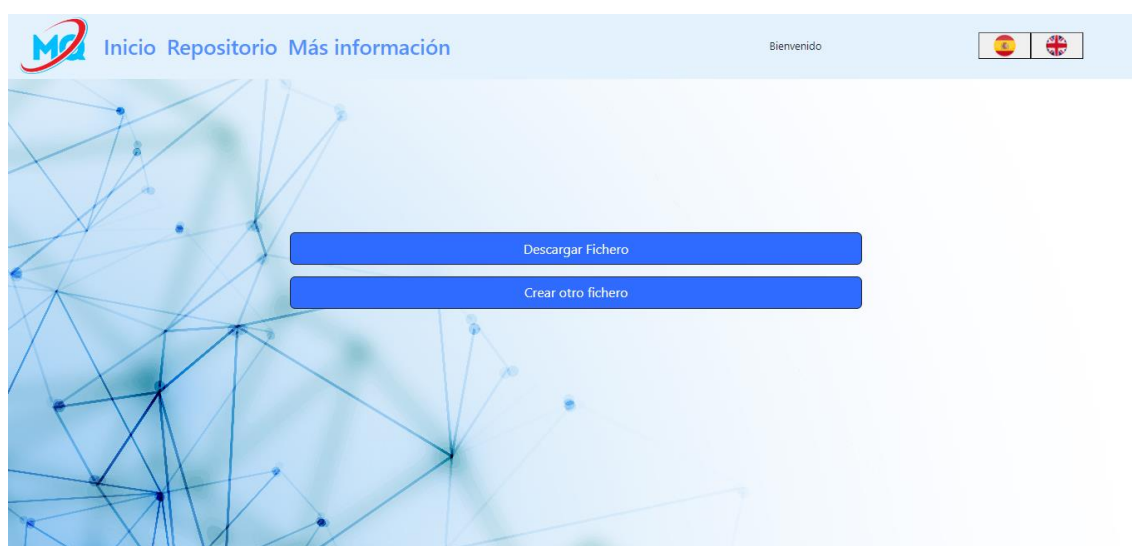


Figura E.9: Descargar fichero Valor ganado

Preguntas Programación ganada

Esta página se mostrará cuando el usuario haya pulsado el botón “Programación Ganada”.

El usuario debe introducir para cada casilla un número o un rango de números separados por comas, donde viene explicado con un texto lo que hay que introducir en cada uno. También debe seleccionar el idioma entre español o inglés que quiere crear las preguntas del banco de preguntas.

Finalmente, si el usuario pulsa el botón de “Cancelar” será redirigido otra vez a la página de selección de preguntas o si pulsa en el botón de “Generar Fichero” la aplicación generará un fichero ProgramacionGanada.xml que dependiendo del número de preguntas que se quieren generar tardará más o menos en cargar la siguiente página.

MQ Inicio Repositorio Más información Bienvenido

Le presentamos el informe de valor ganado de un proyecto en curso (ver figura).

Coste acumulado (€)

Tiempo (meses)

— PV — EV — AC

6700.00 €
6400.00 €
3600.00 €

En vista de los datos, ¿cómo va el proyecto en costes?

☐ Con sobrecoste
☐ Con costes de acuerdo a lo planificado
☐ Con ahorro en coste

¿Y en tiempo?

☐ En adelanto
☐ De acuerdo a lo planificado en plazo
☐ En retraso

¿qué estimación de coste final para el proyecto haría? (en €)

Introduce el rango de tiempo final (meses) separado por comas

ex: 10,20

Introduce el rango de coste acumulado final (€) separado por comas

ex: 2000,50000

Introduce el número de preguntas que quieres generar

Selecciona el idioma que quieres crear el fichero

Español

Figura E.10: Preguntas Programación ganada

Obtener fichero Programación ganada

Esta página se mostrará cuando el usuario haya pulsado el botón “Generar Fichero” en la página preguntas programación ganada.

Si el usuario pulsa el botón de “Descargar Fichero” se descargará un fichero de nombre ProgramacionGanada.xml en el dispositivo del usuario y se mantendrá en la misma página por si se quiere descargar el fichero más de una vez.

Si el usuario pulsa en el botón de “Crear otro fichero” será redirigido otra vez a la página de selección de preguntas.

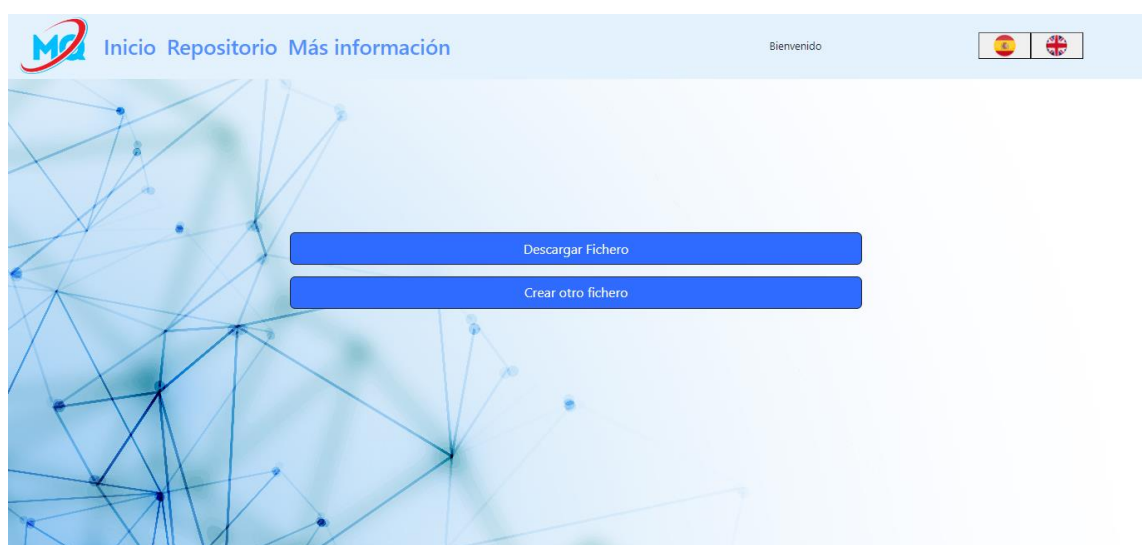


Figura E.11: Descargar fichero Programación Ganada

Visualización de las preguntas

Una vez creado y descargado los ficheros de las preguntas, podemos visualizar el resultado creando un cuestionario en Moodle. Para ello es necesario situarnos en la herramienta de [Sandbox Moodle Demo](#) e iniciar sesión pulsando en “log in”:

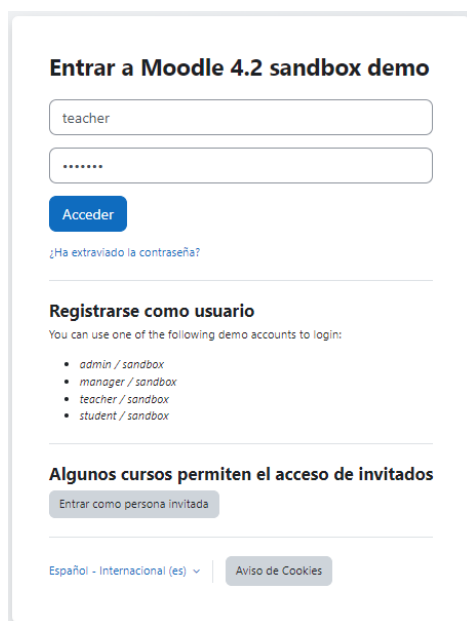


Figura E.12: Registro en Moodle

Introducimos para el nombre de usuario *teacher* y para la contraseña *sandbox* y pulsamos el botón de “Acceder”.

Dentro de nuestro curso (ver Fig. E.13) debemos activar en modo “Edit mode” y seleccionar el idioma, en este caso “Español – Internacional (es)”

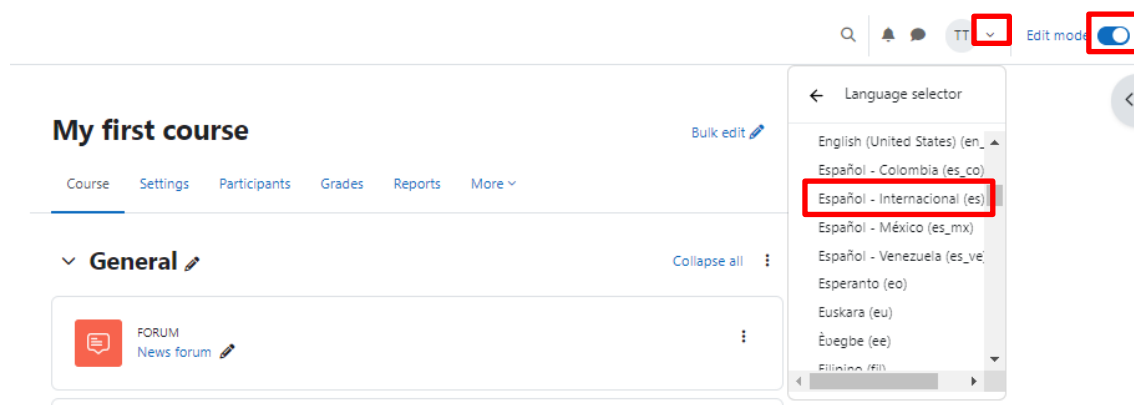


Figura E.13: Configuración Moodle

Añadimos una nueva actividad a nuestro curso y seleccionamos la opción de “Cuestionario”

My first course

[Edición masiva](#)[Curso](#) [Configuración](#) [Participantes](#) [Calificaciones](#) [Informes](#) [Más](#)

General

[Colapsar todo](#)

FORO
News forum



RECURSO
Fashion design

[Marcar como hecha](#)

CUESTIONARIO
clothing construction theory

[Marcar como hecha](#)

Throughout the process, attention to detail, precision, and understanding of garment construction principles are essential to create a well-made and visually pleasing garment.

[Añadir una actividad o un recurso](#)

Añadir una actividad o un recurso

[Todos](#) [Actividades](#) [Recursos](#)

Archivo



Área de texto y
medios



Base de datos



Carpeta



Chat



Consulta



Cuestionario



Encuesta



Encuestas
predefinidas



Foro



Glosario



H5P



Herramienta
externa



Lección



Libro



Página



Paquete de
contenido IMS



Paquete SCORM



Taller



Tarea



URL



Wiki

[O buscar contenido en moodleNet](#)

Figura E.14: Crear una actividad Cuestionario

Configuramos el nuevo cuestionario añadiendo un nuevo nombre y añadiendo las características que cada usuario quiera poner, en mi caso lo dejaré todo por defecto.

The screenshot shows the 'My first course' interface with a navigation bar containing 'Curso', 'Configuración', 'Participantes', 'Calificaciones', 'Informes', and 'Más'. The 'Configuración' tab is active. Below the navigation bar, there is a section titled 'Agregando un nuevo Cuestionario' with a checkmark icon and a help icon. A link 'Expandir todo' is visible. Under the 'General' section, there is a 'Nombre' field with a red information icon and the text 'Cuestionario Ejemplo'. Below this, there is a 'Competencias' section with a checkbox 'Enviar notificación de actualización de contenido' and a help icon. At the bottom, there are three buttons: 'Guardar cambios y regresar al curso', 'Guardar cambios y mostrar', and 'Cancelar'. A red information icon with the text 'Requerido' is also present.

Figura E.15: Configurar un Cuestionario

Pulsaremos en el botón de “Guardar cambios y mostrar” para guardar la configuración del cuestionario.

A continuación, nos debemos situar en “Banco de preguntas” (ver Fig. E.16) seleccionar la opción “Importar” , “Formato Moodle XML” y arrastrar los diferentes ficheros que se han creado en la aplicación web.

The screenshot shows the 'Banco de preguntas' (Question Bank) page with a navigation bar containing 'Cuestionario', 'Configuración', 'Preguntas', 'Resultados', 'Banco de preguntas', and 'Más'. The 'Banco de preguntas' tab is active. Below the navigation bar, there is a section titled 'Importar preguntas de un archivo' with a checkmark icon and a help icon. A link 'Expandir todo' is visible. Under the 'Formato de archivo' section, there are several radio button options: 'Blackboard', 'Formato Aiken', 'Formato de palabra ausente', 'Formato GIFT', 'Formato Moodle XML' (selected), and 'Respuestas incrustadas (Cloze)'. Below this, there is a 'General' section with a sub-section 'Importar preguntas de un archivo'. This sub-section has a label 'Importar' and a red information icon. It contains a text input field with the placeholder 'Seleccione un archivo...' and a note 'Tamaño máximo para archivos nuevos: 256 MB'. Below the input field is a large dashed box with a download icon and the text 'Puede arrastrar y soltar archivos aquí para añadirlos'. At the bottom, there is an 'Importar' button and a help icon.

Figura E.15: Importar un banco de preguntas

Por último, debemos agregar estas preguntas de forma aleatoria al cuestionario situándonos en “Preguntas y pulsando el botón de agregar del banco de preguntas:

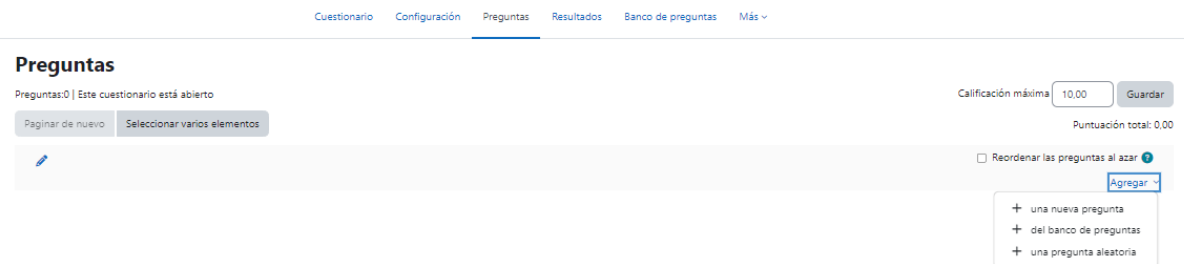
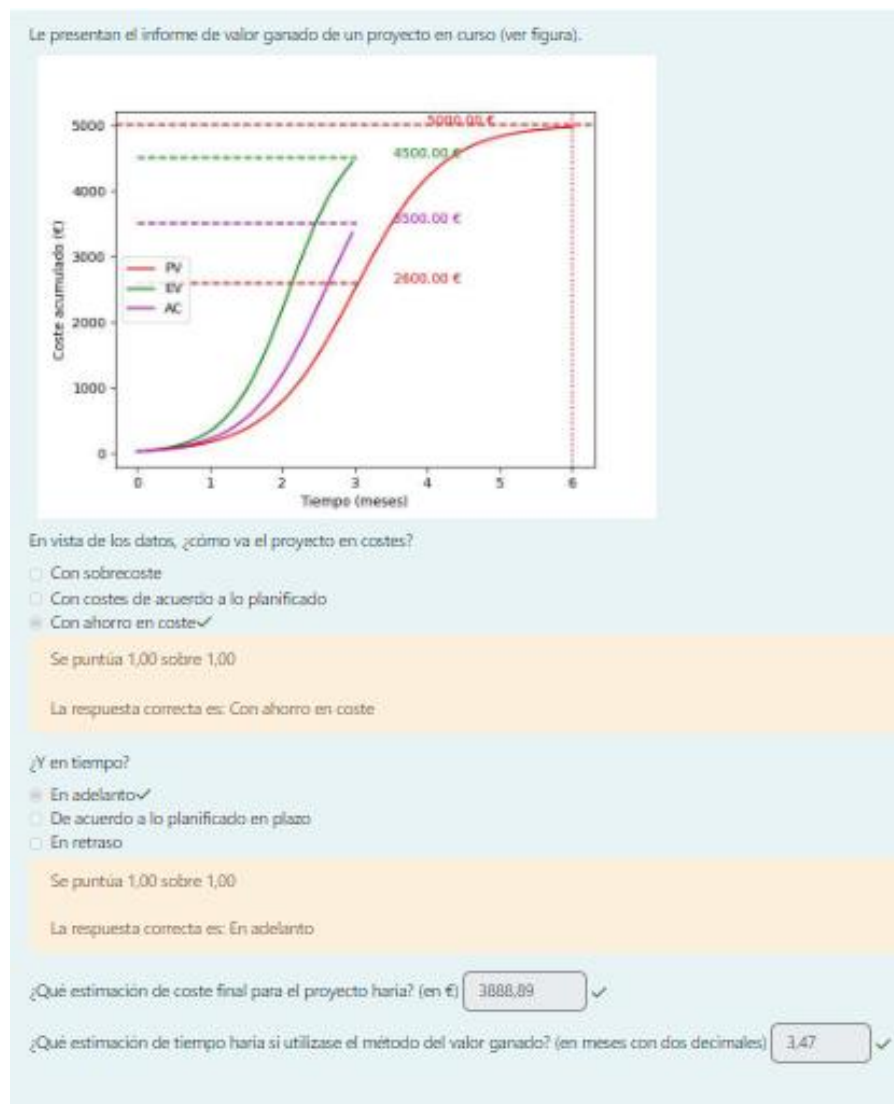


Figura E.16: Agregar preguntas

A continuación, se mostrará como será la interfaz de un cuestionario resuelto correctamente cada pregunta.



Suponga un proyecto en el que en el momento actual y de acuerdo al plan de proyecto **las actividades desde la A hasta la E** deberían estar ya completadas y el resto de actividades no comenzadas.

El informe de seguimiento del proyecto es el siguiente:

Tareas	CPTP	CRTR	%Completado
A	3500	5700	100%
B	4800	6200	100%
C	6200	3800	100%
D	2400	8000	100%
E	6100	4200	100%
F	2100	5000	80%
G	3000		
H	3900		
I	6100		

Calcule:

El valor planificado (PV) en el momento actual ✓

El coste actual (AC) en el momento actual ✓

El valor ganado (EV) en el momento actual ✓

El CPI del proyecto (con dos decimales) ✓

El SPI del proyecto (con dos decimales) ✓

Para calcular PV= Sumar el CPTP de todas las tareas completadas.

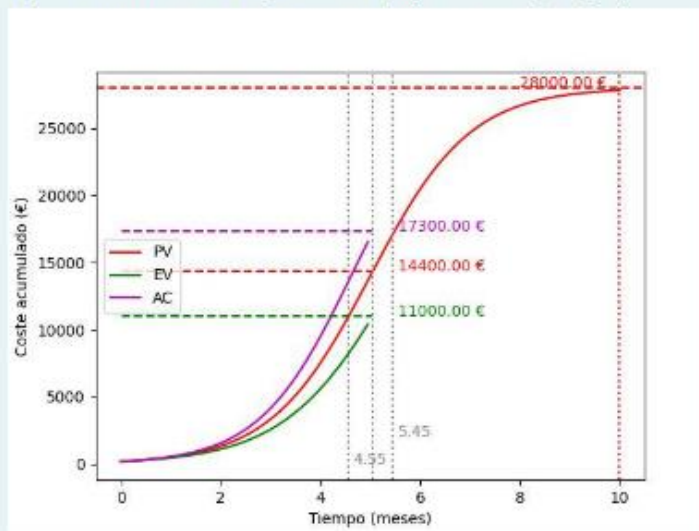
Para calcular AC= Sumar todos los CRTR.

Para calcular EV= Multiplicar los CPTP por sus respectivos porcentajes y sumarlos.

Para calcular CPI= Dividir EV por el AC.

Para calcular SPI= Dividir EV por el PV.

Le presentan el informe de valor ganado de un proyecto en curso (ver figura).



En vista de los datos,

¿Cuál es el Cost Variance del proyecto? (introducir con el signo apropiado) ✓

¿Cuál es el SPI? (con dos decimales) ✓

De acuerdo a la Programación Ganada el proyecto va:

- ☒ Retrasado ✓
☐ Adelantado
☐ De acuerdo a lo planificado

Se puntúa 1,00 sobre 1,00

La respuesta correcta es: Retrasado

¿cuánto? (introducir en meses sin signo y 2 decimales) ✓

Figura E.17: Visualización de las preguntas resueltas

Bibliografía

- [1] «¿Cuánto Cobra un Ingeniero Informático? (Sueldo 2023) | Jobted.es». <https://www.jobted.es/salario/ingeniero-inform%C3%A1tico> (accedido 24 de junio de 2023).
- [2] «Seguridad Social: Cotización / Recaudación de Trabajadores». <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537?changeLanguage=es> (accedido 24 de junio de 2023).
- [3] «IEEE830.pdf». Accedido: 24 de junio de 2023. [En línea]. Disponible en: <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
- [4] «Modelo–vista–controlador», *Wikipedia, la enciclopedia libre*. 27 de septiembre de 2021. Accedido: 26 de junio de 2023. [En línea]. Disponible en: <https://es.wikipedia.org/w/index.php?title=Modelo%E2%80%93vista%E2%80%93controlador&oldid=138615253>
- [5] «Facade (patrón de diseño)», *Wikipedia, la enciclopedia libre*. 22 de mayo de 2023. Accedido: 26 de junio de 2023. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Facade_\(patr%C3%B3n_de_dise%C3%B1o\)&oldid=151333047](https://es.wikipedia.org/w/index.php?title=Facade_(patr%C3%B3n_de_dise%C3%B1o)&oldid=151333047)