



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

eLearningQA
Documentación Técnica



Presentado por Roberto Arasti Blanco
en Universidad de Burgos — 10 de junio
de 2022

Tutor: Raúl Marticorena Sánchez y Carlos
López Nozal

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	11
B.1. Introducción	11
B.2. Objetivos generales	11
B.3. Catalogo de requisitos	11
B.4. Especificación de requisitos	12
Apéndice C Especificación de diseño	15
C.1. Introducción	15
C.2. Diseño de datos	15
C.3. Diseño procedimental	16
C.4. Diseño arquitectónico	18
Apéndice D Documentación técnica de programación	19
D.1. Introducción	19
D.2. Estructura de directorios	19
D.3. Manual del programador	21

D.4. Compilación, instalación y ejecución del proyecto	26
D.5. Pruebas del sistema	28
Apéndice E Documentación de usuario	31
E.1. Introducción	31
E.2. Requisitos de usuarios	31
E.3. Instalación	31
E.4. Manual del usuario	32
Bibliografía	39

Índice de figuras

B.1. Diagrama de casos de uso	12
C.1. Modelo de datos base	16
C.2. Boceto de la interfaz	17
C.3. Diagrama de secuencia de informe específico	17
C.4. Diagrama de interacción con la fachada	18
D.1. Desplegable de Git en IDEA	22
D.2. Trabajos de integración continua	24
D.3. Página del proyecto en SonarCloud en https://sonarcloud.io/summary/overall?id=RobertoArastiBlanco_MoodleQA	25
D.4. Apartado de problemas de código	25
D.5. Ejecución de la clase principal	26
D.6. Ejecución en local	27
D.7. Configuración necesaria para el despliegue	28
D.8. Configuración del despliegue automático	28
E.1. Página de login	32
E.2. Página principal	33
E.3. Página de evolución del rendimiento	34
E.4. Página de informe de fases	34

Índice de tablas

A.1. Costes de personal	6
A.2. Costes de hardware	6
A.3. Costes varios	7
A.4. Costes totales	7
A.5. Licencias del software utilizado	8
B.1. Caso de uso CU-01	13
B.2. Caso de uso CU-02	13
B.3. Caso de uso CU-03	13
B.4. Caso de uso CU-04	14
B.5. Caso de uso CU-05	14

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apéndice se trata sobre los costes y viabilidad del proyecto. Contiene una planificación temporal que hablará de la cantidad de tiempo estimada que va a costar realizar el proyecto. También un estudio de viabilidad dividido en dos partes: un estudio de viabilidad económica que tratará los costes y beneficios teóricos de la realización del proyecto en un escenario real a pesar de que el fin de este trabajo sea educativo, y un estudio de viabilidad legal que discutirá si hay o no problemas derivados de las licencias de las herramientas usadas y/o infracciones de copyright.

A.2. Planificación temporal

En esta sección se habla de la cantidad de tiempo estimada necesaria al principio del proyecto y la cantidad de tiempo que ha acabado llevando, detallando lo ocurrido durante el proceso.

La planificación del proyecto se ha llevado a cabo en sprints de una duración de dos semanas. Normalmente se realizó una reunión entre sprint y sprint en la que se discutían los resultados del sprint anterior y los objetivos para el siguiente. En GitHub, los sprints están definidos como milestones (hitos) y los objetivos del sprint como issues (problemas, asuntos). Para cada sprint he creado un nuevo milestone y he incluido los issues respectivos a los objetivos del sprint a dicho milestone.

Al comienzo estimé que para completar el proyecto necesitaría 400 horas de trabajo a lo largo de 16 semanas, lo que conllevaría 25 horas de trabajo a la semana.

Ahora paso a resumir los objetivos y sucesos de cada sprint:

Sprint 0 (7/10/2021 - 21/10/2021)

En este sprint los objetivos propuestos eran instalar la plantilla de la memoria en LaTeX proporcionada por la UBU, redactar una introducción y definir los objetivos del proyecto, y sintetizar una lista de comprobaciones a realizar por parte de la aplicación a partir de unos marcos de calidad. En la reunión de cierre del sprint los tutores se mostraron disconformes con mi forma de redactar excesivamente técnica y breve y se propuso el realizar reuniones a mitad de sprint cuando yo lo viera conveniente.

Sprint 1 (21/10/2021 - 4/11/2021)

En este sprint se definieron como objetivos incluir en la introducción un resumen de la situación actual respecto del control de calidad en el e-learning y añadir las dimensiones de roles y perspectivas a la lista de comprobaciones y añadir la lista a la memoria. En la reunión de mitad de sprint se remarcó la falta de referencias bibliográficas y se me pidió que expandiera el apartado de conceptos teóricos. En la reunión de cierre del sprint los tutores se mostraron bastante más conformes con mi trabajo.

Sprint 2 (4/11/2021 - 18/11/2021)

En este sprint se definieron como objetivos añadir definiciones al apartado de conceptos teóricos, mostrar la lista de consultas definida en forma de tabla, y decantarme por alguna de los tres frameworks para el desarrollo de la aplicación (Vaadin, JavaFX, o Spring). En la reunión de mitad de sprint se me dieron indicaciones sobre aspectos a mejorar que realicé antes de acabar el sprint. Elegí Spring para crear la aplicación web.

Sprint 3 (18/11/2021 - 2/12/2021)

En este sprint el objetivo principal fue la creación de un prototipo. Creé un prototipo muy simple que mostraba la lista de cursos recientemente accedidos en la plataforma de demostración Mount Orange School tras introducir usuario y contraseña. En la reunión de final de sprint se me

propuso utilizar el patrón de diseño fachada para separar la responsabilidad de acceder a los servicios web y generar el contenido para mostrarlo en HTML.

Sprint 4 (2/12/2021 - 16/12/2021)

En este sprint se definieron como objetivos comenzar con el diseño de la aplicación, crear un pipeline de integración continua/despliegue continuo con GitHub Actions y Heroku, y añadir funcionalidad al prototipo. En la primera semana del sprint creé un boceto de la interfaz de la aplicación y un diagrama de clases preliminar, además de implementar la integración continua con GitHub Actions y despliegue continuo en Heroku. En la reunión de mitad de sprint mostré los avances que había conseguido a Raúl Marticorena (Carlos López estaba ocupado ese día) y me instó a añadir funciones al prototipo. Durante la segunda semana añadí una versión reducida del informe específico que mostraba la comprobación de si un curso tenía grupos definidos o no. En la reunión de final de sprint mostré los avances que había conseguido a Carlos López (Raúl Marticorena estaba ocupado ese día) y establecimos que no íbamos a poder reunirnos hasta la vuelta de las vacaciones y me motivó diciendo que a partir de ese punto el resultado del proyecto dependía de la cantidad de trabajo que invirtiera en él y que gracias al trabajo que había realizado hasta ese punto el proyecto estaba encarrilado.

Sprint 5 (16/12/2021 - 30/12/2021)

En este sprint se definieron como objetivos refactorizar la fachada de forma que se puedan realizar tests sobre ella, añadir tests, y seguir implementando consultas sobre los cursos Moodle en el prototipo. Al final de este sprint no hubo reunión por motivo de las fiestas navideñas.

Sprint 6 (30/12/2021 - 13/1/2022)

En este sprint decidí implementar las comprobaciones restantes, rellenar algunos apartados de la memoria y cambiar el estilo de la página del informe para que hubiera un código de cinco colores para distintos niveles de gravedad. En la reunión, a pesar de haber hecho un gran avance, decidí retrasar la entrega de este trabajo al segundo semestre.

Sprint 7 (13/1/2022 - 27/1/2022)

En este sprint refactoricé los test unitarios para que carguen los objetos necesarios desde una serie de archivos json en vez de conectarse a la página de demo de Moodle que es accesible por cualquiera y está sujeta a cambios. También se refactorizó la fachada para no tener métodos extremadamente largos y cambios varios. A la reunión de final de sprint solo pudo acudir Carlos López, y propuso añadir cinco consultas que comprobaban aspectos de los cuestionarios.

Sprint 8 (27/1/2022 - 10/2/2022)

En este sprint integré la herramienta de calidad de código SonarCloud al repositorio, revisé la API de servicios web de Moodle para ver si eran factibles las consultas propuestas en la reunión anterior, creé un archivo html a modo de boceto para trabajar en el diseño del informe e hice un apartado de buenas prácticas en la docencia online. En la reunión mostré el boceto html, con el que se mostraron medianamente conformes. En la reunión también se desestimó el implementar las consultas propuestas y dejarlo como una línea futura de trabajo.

Sprint 9 (10/2/2022 - 24/2/2022)

En este sprint empecé a solucionar los errores descubiertos por SonarCloud y modifiqué el estilo de la mayoría de mis páginas con el framework Bootstrap. En la reunión, hablando del logo acacabamos descubriendo que al ser “Moodle” una marca registrada debíamos hacer cambios como por ejemplo el nombre de dominio del despliegue en Heroku.

Sprint 10 (24/2/2022 - 10/3/2022)

En este sprint cambié con Bootstrap el estilo de la página de informe, cambié todos los nombres problemáticos y solucioné casi todos los errores de código. En la reunión se decidió seguir trabajando en el estilo del informe e implementar una forma de generar alertas para los errores detectados por las consultas.

Sprint 11 (10/3/2022 - 24/3/2022)

En este sprint añadí las alertas y diseñé el logo de la aplicación. En la reunión se comentaron errores del informe y se decidió añadir gráficos a la

página de informe y hacer perfiles de configuración intercambiables para distintos tipos de curso.

Sprint 12 (24/3/2022 - 7/4/2022)

En este sprint arreglé algunas de las alertas que no funcionaban bien y añadí los perfiles de configuración, pero aplacé añadir los gráficos.

Sprint 13 (7/4/2022 - 21/4/2022)

En este sprint arregle un error encontrado por los profesores, añadí los gráficos y una página de error, pero el cambio necesario para hacer las comprobaciones flexibles tenía demasiada complejidad. Solo Carlos López pudo acudir a la reunión, así que se decidió convocar otra la semana siguiente.

Sprint 14 (21/4/2022 - 28/4/2022)

En este sprint se arreglaron problemas estéticos y pequeños bugs. En la reunión dimos un repaso a todos los errores y mejoras pendientes de la página de informe. También se decidió empezar a centrarse en la memoria y anexos.

Sprint 15 (28/4/2022 - 12/5/2022)

En este sprint se arreglaron más errores y se hicieron correcciones en memoria y anexos. En la reunión se comentaron más errores y se decidió continuar con las mismas tareas.

Sprint 16 (12/5/2022 - 26/5/2022)

En este sprint se siguió haciendo lo mismo que en el sprint anterior, arreglar errores y añadir contenido a la memoria. En la reunión se decidió pasar a hacer sprints de una semana debido a la cercanía de la fecha límite y centrarse en completar los anexos.

Sprint 17 (26/5/2022 - 2/6/2022)

En este sprint se solucionó alguno de los problemas restantes de la aplicación y se rellenaron algunos apartados restantes de los anexos. A la reunión solo pudo acudir Carlos López, y en esta se decidió dejar de arreglar errores de código y centrarse en terminar memoria y anexos.

Sprint 18 (2/6/2022 - 9/6/2022)

En este sprint se revisó la memoria a partir de los comentarios recibidos de los tutores. No hubo reunión porque se había decidido comunicarse mediante correos a partir de este punto.

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se estiman los costes que tendría la realización del proyecto en un caso real.

A pesar de haber llevado 8 meses el desarrollo del proyecto, la cantidad de trabajo ronda las 800 horas, que equivaldría a 5 meses a jornada completa que es en lo que nos basaremos para el cálculo.

Tabla A.1: Costes de personal

Concepto	Coste
Sueldo mensual neto	1.000,00€
Retención IRPF (19 %)	254,52€
Seguridad Social (6,35 %)	85,06€
Sueldo mensual bruto	1.339,58€
Total 5 meses	6.697,90€

Las cuotas a la seguridad social se componen de un 4,70 % de contingencias comunes, un 1,55 % por desempleo de tipo general, y un 0,10 % de formación profesional.

Se supone que el ordenador portátil se amortiza en cinco años y ha sido usado cinco meses.

Tabla A.2: Costes de hardware

Concepto	Coste	Coste amortizado
Ordenador portátil	550€	45,83€
Total	550€	45,83€

Tabla A.3: Costes varios

Concepto	Coste
Internet	150,00€
Electricidad	175,00€
Total	325,00€

A partir de estos costes obtenemos el coste total del proyecto:

Tabla A.4: Costes totales

Concepto	Coste
Personal	6.697,90€
Hardware	45,83€
Varios	325,00€
Total 5 meses	7.068,73€

Para hacer rentable el desarrollo de la aplicación teniendo en cuenta que es de código abierto se podría adoptar un modelo SaaS (Software as a Service) en el que los clientes paguen una suscripción al dueño del software y este a su vez se encargue del hosting y mantenimiento de la aplicación. Se podrían ofrecer suscripciones a varios niveles ofreciendo distintos niveles de funcionalidad y servicios a distintos precios.

Viabilidad legal

Licencias de software

Para determinar la licencia software que va a utilizar la aplicación hay que tener en cuenta las licencias utilizadas por las dependencias que utiliza la aplicación.

Tabla A.5: Licencias del software utilizado

Software	Descripción	Licencia
Spring Framework	Framework para aplicaciones web	Apache 2.0
Tomcat Embed Jasper	Implementación de Tomcat que incluye Jasper, el parser de JSP de Tomcat	Apache 2.0
JUnit	Framework para tests unitarios en Java	EPL
Apache Commons IO	Librería de utilidades varias (usado en traducción de imágenes a arrays de bytes)	Apache 2.0
Apache Log4j	Librería para registro de logs	Apache 2.0
Bootstrap	Librerías CSS y JavaScript para páginas web	MIT
Plotly.js	Librería JavaScript de generación de gráficos	MIT

La licencia pública de Eclipse (EPL) es compatible con estas licencias, a continuación se mencionan sus posibilidades y obligaciones.

- **Permite:** uso, reproducción, distribución, modificación, uso comercial y uso de patentes.
- **Obliga a:** revelar la fuente y el autor, mantener la misma licencia al redistribuir el software, distribuir el software libre de regalías.
- **No permite:** responsabilizar al autor o contribuidores por posibles daños, utilizar marcas propiedad del autor para promoción o publicidad.

Uso del nombre “Moodle” en la aplicación

Debido a que Moodle es una marca registrada y no soy un partner registrado de Moodle, tengo que atenerme a una serie de restricciones con respecto al uso de la palabra “Moodle” [1]:

- No puedo usar logos de “Moodle” sin consentimiento escrito de Moodle.
- No puedo usar “Moodle” en el nombre de mi software, ni en el de mi dominio ni en el de mi empresa.
- No puedo usar “Moodle” en palabras clave relacionadas con la publicidad.

- No puedo usar “Moodle” para describir servicios alrededor de Moodle de forma que la gente piense que estoy asociado a Moodle cuando no es así.

Al enterarme de esto a mitad del desarrollo he tenido que cambiar a “eLearningQA” el nombre del proyecto, el del repositorio, el del despliegue en Heroku para que el dominio deje de contener “Moodle”, y he aclarado en el readme del proyecto y en la memoria que no estoy asociado con Moodle.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se obtienen los requisitos funcionales y no funcionales del software en cuestión a partir de los objetivos generales y expectativas que tenemos del proyecto. Un documento de especificación de requisitos sirve como un medio de comunicación entre todas las partes interesadas en el desarrollo del software.

B.2. Objetivos generales

Los objetivos generales de esta aplicación son los que se listan a continuación:

- Poder realizar una serie de comprobaciones sobre los cursos de Moodle que faciliten la evaluación de la calidad de estos.
- Poder aplicar un framework de calidad del e-learning para evaluar los distintos aspectos que influyen en la calidad de los cursos.
- Poder generar informes con información concisa que permitan responder a los desafíos que empobrecen la calidad de los cursos.

B.3. Catalogo de requisitos

La lista de requisitos funcionales es la siguiente:

- R-01: la aplicación permitirá registrarse al docente para acceder a la información sobre los cursos.
- R-02: la aplicación permitirá al profesor desconectarse.
- R-03: la aplicación mostrará un listado de los cursos en los que se encuentre el profesor.
- R-04: se deben poder generar informes que evalúen la calidad de los cursos en los que se encuentra el profesor por medio de consultas a diversos aspectos del curso como la corrección a tiempo de las tareas o la correctitud de las fechas.
- R-05: se deben poder generar informes globales que resuman los informes de todos los cursos en los que se encuentre el profesor.

B.4. Especificación de requisitos

A partir de los requisitos se obtiene el siguiente diagrama de casos de uso:

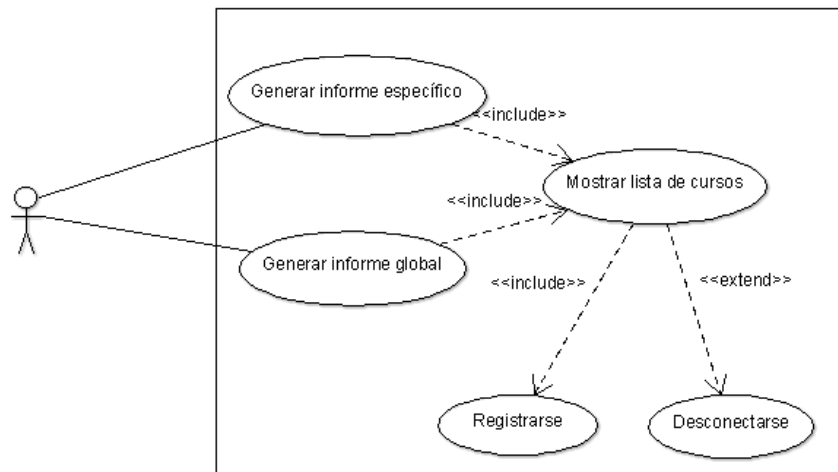


Figura B.1: Diagrama de casos de uso

Tabla B.1: Caso de uso CU-01

CU-01	Registrarse
Descripción	Permite al usuario registrarse con sus credenciales
Requisitos asociados	R-01
Precondiciones	El usuario posee una cuenta valida 1.El usuario introduce su nombre de usuario 2.El usuario introduce su contraseña 3.El usuario modifica el dominio al del servidor al que pretende acceder
Ejecución	4.El usuario selecciona en el menú desplegable el tipo de curso que pretende analizar 5.El usuario hace clic en “Entrar” 6.El usuario es llevado a la lista de cursos
Postcondiciones	
Excepciones	Las credenciales no son válidas (se indica al usuario)

Tabla B.2: Caso de uso CU-02

CU-02	Desconectarse
Descripción	Permite al usuario cerrar la sesión
Requisitos asociados	R-02
Precondiciones	El usuario está logueado en la aplicación 1.El usuario hace clic en “Desconectar” desde la lista de cursos
Ejecución	2.Se invalida la sesión del usuario 3.El usuario es llevado a la página de login
Postcondiciones	La sesión se ha cerrado correctamente
Excepciones	

Tabla B.3: Caso de uso CU-03

CU-03	Mostrar lista de cursos
Descripción	Muestra al usuario el listado de los cursos en los que está
Requisitos asociados	R-03
Precondiciones	El usuario está logueado en la aplicación 1.Se utilizan las credenciales para obtener el token de sesión 2.El token se almacena en la sesión
Ejecución	3.Se obtiene el listado de los cursos del usuario utilizando el token 4.La página muestra el listado
Postcondiciones	
Excepciones	

Tabla B.4: Caso de uso CU-04

CU-04	Generar informe específico
Descripción	Genera un informe del curso correspondiente
Requisitos asociados	R-04
Precondiciones	El usuario está logueado en la aplicación 1.El usuario hace clic en el curso correspondiente 2.Se utiliza el token para obtener información sobre el curso
Ejecución	3.Se calculan los porcentajes de desempeño 4.Se añaden los resultados a un archivo csv de registro 5.Se genera el gráfico de evolución a partir del contenido del csv correspondiente 6.Se muestran los resultados y el gráfico en la página
Postcondiciones	Se ha generado o ampliado un archivo csv para ese servidor, profesor y curso
Excepciones	

Tabla B.5: Caso de uso CU-05

CU-05	Generar informe global
Descripción	Genera un informe a partir del estado general de los cursos
Requisitos asociados	R-05
Precondiciones	El usuario está logueado en la aplicación 1.El usuario hace clic en “Informe general” 2.Se utiliza el token para obtener información sobre todos los curso
Ejecución	3.Se calculan los porcentajes de desempeño medios 4.Se muestran los resultados en la página
Postcondiciones	
Excepciones	

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se detalla en qué manera se pretende abordar el conjunto de objetivos y requisitos especificados en el apéndice **B**. Se definen los datos que utiliza la aplicación, su arquitectura, y su diseño procedimental.

C.2. Diseño de datos

En la siguiente tabla se recogen las entidades asociadas a cada una de las consultas para determinar parte de las clases necesarias para el proyecto.

Consulta	Entidades
Las opciones de progreso del estudiante están activadas	Curso
Se proporcionan contenidos en diferentes formatos	Curso, Recurso
El curso tiene grupos	Grupo, Curso
El curso tiene actividades grupales	Actividad, Curso
Los estudiantes pueden ver las condiciones necesarias para completar una actividad	Curso
Todas las actividades tienen la misma nota máxima en el calificador	Actividad, Curso
Los recursos están actualizados	Curso, Recurso
Fechas de apertura y cierre de tareas son correctas	Actividad, Curso
Se detallan los criterios de evaluación (rúbricas, ejemplos)	Curso
El calificador no tiene demasiado anidamiento	Curso
Los alumnos están divididos en grupos	Alumno, Grupo
El profesor responde en los foros dentro del límite de 48 horas lectivas desde que se plantea la duda	Curso, Foro
Se ofrece retroalimentación de las tareas	Tarea, Curso
Las tareas están calificadas	Tarea, Curso
El calificador muestra cómo ponderan las diferentes tareas	Curso
La mayoría de alumnos responden a los feedbacks	Feedback, Curso
Se utilizan encuestas de opinión	Survey, Curso

La siguiente imagen muestra las relaciones entre entidades que utiliza la aplicación.

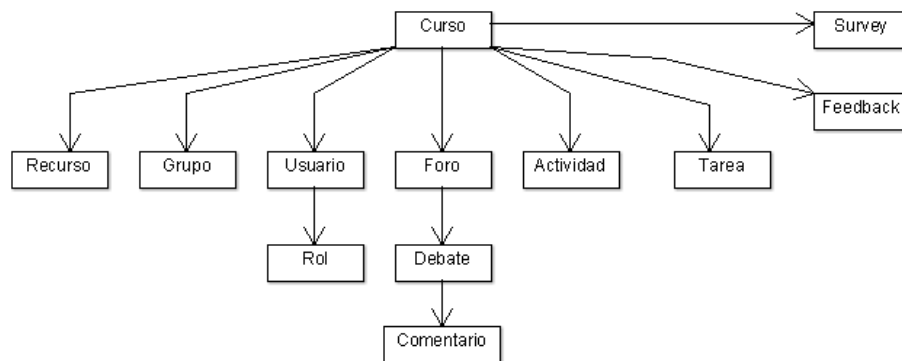


Figura C.1: Modelo de datos base

C.3. Diseño procedimental

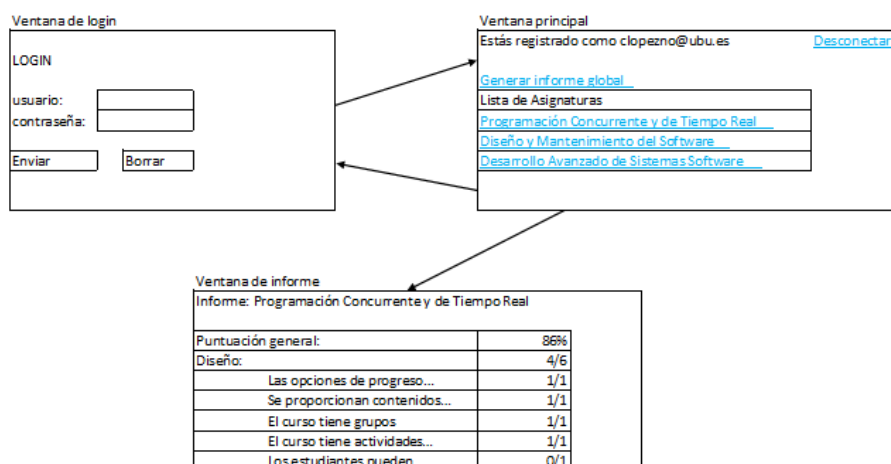


Figura C.2: Boceto de la interfaz

El siguiente diagrama de secuencia muestra el proceso de generación de informe específico. La llamada “descarga de datos” es una simplificación de más de 10 llamadas que descargan la información necesaria sobre el curso. La llamada “comprobaciones” es también una simplificación de las 17 llamadas correspondientes a las comprobaciones.

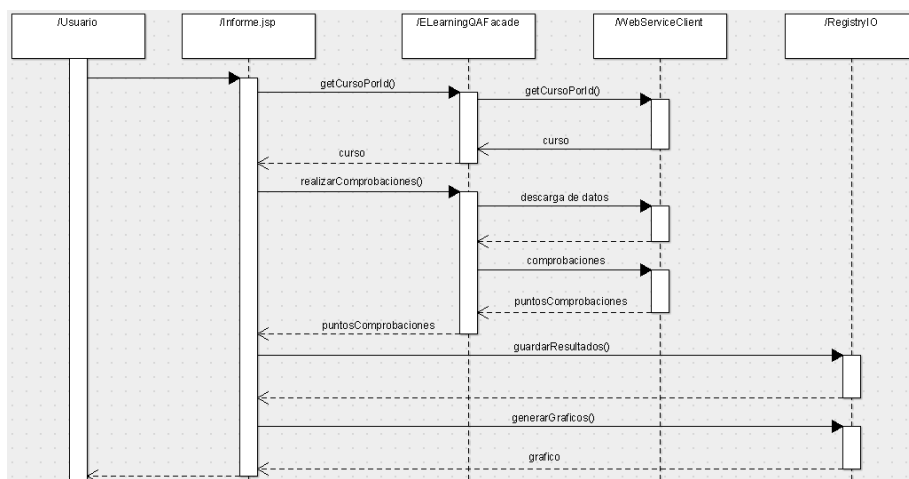


Figura C.3: Diagrama de secuencia de informe específico

C.4. Diseño arquitectónico

Patrón fachada

El patrón de diseño fachada consiste en utilizar una clase a modo de interfaz de acceso a un subsistema complejo. La principal ventaja que aprovecha la aplicación de este patrón de diseño es simplificar una serie de funciones del subsistema con un mismo fin en una sola llamada a la fachada. Aparte de esto, reduce el acoplamiento, permitiendo reutilizar partes del subsistema. Un subsistema puede tener varias fachadas con distintas funcionalidades y en caso de añadir un elemento nuevo al subsistema solo sería necesario modificar la fachada.

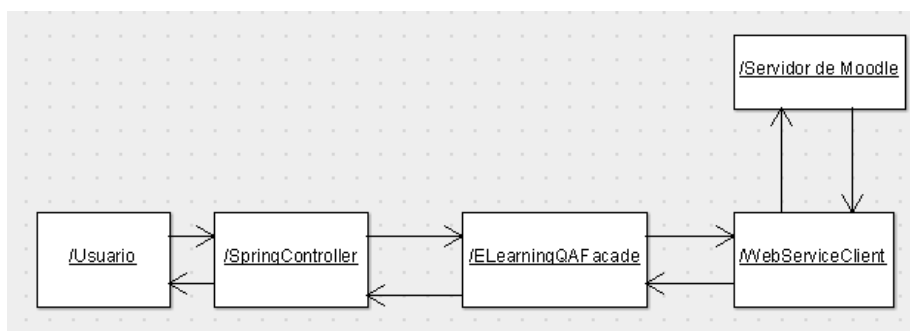


Figura C.4: Diagrama de interacción con la fachada

Vamos a explicar la forma en la que el usuario accede a información por medio de la fachada: El usuario intenta acceder, por ejemplo, a la página de informe de la aplicación. La clase SpringController se encarga de devolver las respuestas a peticiones de este tipo, siendo en este caso un archivo jsp, pero, antes debe ser compilado y ejecutado para que la respuesta sea una página html. La ejecución del jsp en cuestión desencadena las llamadas necesarias a la fachada para hacer las comprobaciones definidas, que se realizan por medio de la clase WebServiceClient que accede a su vez al servidor de Moodle. Después de recibir los resultados de las comprobaciones, el jsp utiliza la fachada como clase de utilidad para procesar algunos de los resultados. Tras terminar la ejecución del jsp, la clase SpringController devuelve la página html generada al usuario.

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este apéndice se describe una serie de detalles sobre la programación de la aplicación para que sirvan de ayuda a futuros programadores que utilicen el proyecto. Contiene la estructura de directorios del proyecto en la que se explica el contenido de cada uno, un manual del programador que explica ciertos aspectos que puedan ser de ayuda, los detalles sobre la compilación, instalación, y ejecución en un entorno de desarrollo integrado, y por último, qué pruebas del sistema se han realizado.

D.2. Estructura de directorios

El repositorio está organizado de la siguiente manera:

- `/`: Contiene el README y un `.gitignore` que evita que se suban los archivos de configuración del IDE al repositorio.
- `/.github/workflows`: Contiene el archivo que establece las tareas de integración continua. Las tareas son: instalar Java en la máquina que realiza las tareas, almacenar paquetes de SonarCloud en la caché, almacenar paquetes de Maven en la caché, montar (compilar, ejecutar tests y empaquetar) con maven y analizar con SonarCloud.
- `/Memo/plantillaLatex-master`: Contiene la memoria y los anexos.

- **/Memo/plantillaLatex-master/img**: Contiene las imágenes a usar en memoria y anexos.
- **/Memo/plantillaLatex-master/tex**: Contiene los apartados en formato L^AT_EX usados de la memoria y anexos.
- **/Project/prototipo**: Contiene el proyecto de la aplicación web. Incluye el Procfile, un archivo usado por Heroku para el despliegue continuo que indica el comando a ejecutar para arrancar la aplicación.
- **/Project/prototipo/configurations**: Contiene los archivos de propiedades.
- **/Project/prototipo/src**: Contiene el código fuente de la aplicación.
- **/Project/prototipo/src/main**: Contiene el código, los test, y los recursos que utiliza la aplicación.
- **/Project/prototipo/src/main/java/es/ubu/lsi**: Contiene las clases necesarias para el funcionamiento de la aplicación.
- **/Project/prototipo/src/main/java/es/ubu/lsi/model**: Contiene clases Java Bean para poder deserializar los datos extraídos por medio de servicios web.
- **/Project/prototipo/src/main/resources**: Contiene el archivo de propiedades de la aplicación web y otros recursos.
- **/Project/prototipo/src/main/resources/images**: Contiene las imágenes a mostrar en la aplicación web.
- **/Project/prototipo/src/main/resources/json**: Contiene información de cursos en formato json para usarla en los tests.
- **/Project/prototipo/src/main/resources/json/informe**: Contiene los textos estáticos de la tabla del informe de fases en formato json.
- **/Project/prototipo/src/main/resources/static/js**: Contiene las librerías JavaScript que utiliza la aplicación.
- **/Project/prototipo/src/main/webapp/WEB-INF/jsp**: Contiene las páginas de la aplicación en formato jsp.
- **/Project/prototipo/src/test/java/es/ubu/lsi**: Contiene los test sobre las consultas que realiza la aplicación en los cursos.

- **/Project/prototipo/target**: Es el directorio donde Maven deposita los resultados, como las clases compiladas y la aplicación compilada en formato WAR.

D.3. Manual del programador

Entorno

Para trabajar en el proyecto se necesita:

- Java JDK 8
- IntelliJ IDEA

JDK 8 se puede descargar desde <https://www.oracle.com/java/technologies/downloads/#java8> si se está en posesión de una cuenta de Oracle.

La versión gratuita de IDEA se puede descargar desde <https://www.jetbrains.com/es-es/idea/download>.

Obtención del código fuente

Primero se debería crear un *fork* del repositorio principal en <https://github.com/RobertoArastiBlanco/eLearningQA>. IDEA tiene Git integrado, lo que facilita acciones como los *pull*, *push* y *commit*, pero se puede usar la herramienta de control de versiones que se prefiera para clonar el *fork* y así tenerlo descargado en nuestro equipo.

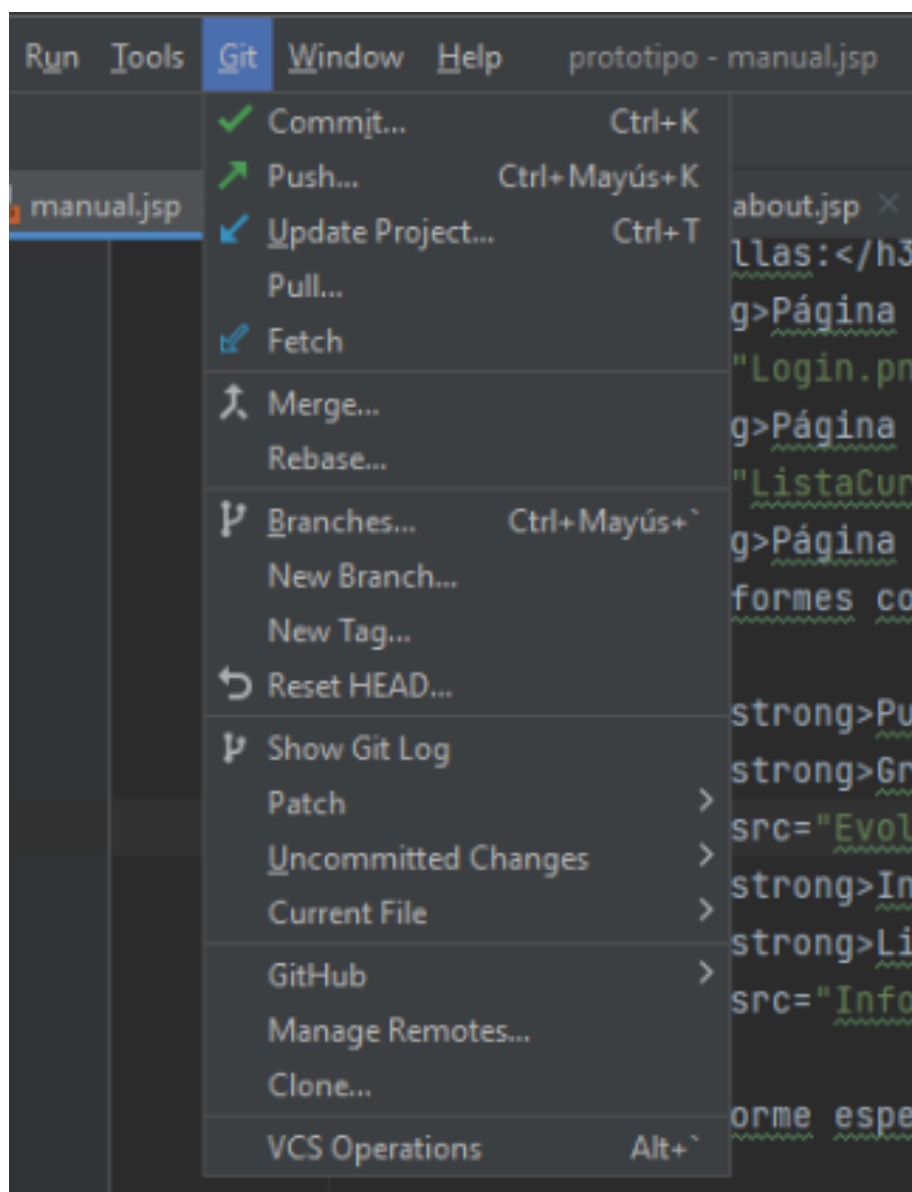


Figura D.1: Desplegable de Git en IDEA

Para trabajar en el proyecto se debe abrir desde IDEA utilizando /Project/prototipo como directorio raíz.

Propósito de las clases principales

- **Application:** es la clase principal de la aplicación, arranca la aplicación web.
- **SpringController:** indica mediante anotaciones y funciones qué devolver cuando el usuario accede a cada página. En este caso solo devuelve páginas jsp e imágenes, pero se podrían devolver otros tipos de recurso, como archivos de audio o documentos.
- **ELearningQAFacade:** hace de intermediario entre el usuario y la clase `WebServiceClient`. También contiene funciones para generar tablas y sus contenidos a partir de la información obtenida.
- **WebServiceClient:** contiene el código de las comprobaciones y realiza las llamadas REST al servidor de Moodle para obtener la información necesaria.
- **FacadeConfig:** almacena la configuración elegida por el usuario junto con el host de Moodle y se guarda como atributo de la fachada para poder pasarlo a las funciones de `WebServiceClient` y que pueda cambiar su comportamiento según la configuración.
- **AlertLog:** genera y almacena mensajes de alerta, se pasa como parámetro a las funciones de `WebServiceClient` para que puedan almacenar los mensajes para la lista de mejoras del informe.
- **RegistryIO:** se encarga de almacenar y acceder a la información guardada en csv utilizada para generar los gráficos. También contiene una función que genera un gráfico a partir de los datos de un csv.
- **AnalysisSnapshot:** la clase que representa el conjunto de datos almacenado en cada línea de los archivos csv, facilita el trabajo de almacenamiento de datos y generación de gráficos.

Integración continua

El proyecto contiene un archivo llamado `maven.yml` en `/.github/workflows` en el que se indica los pasos a seguir tras realizar un *push* al repositorio. Vamos a explicarlos brevemente.

```

11 jobs:
12   build:
13
14     runs-on: ubuntu-latest
15
16     steps:
17       - uses: actions/checkout@v2
18       - name: Set up JDK 11
19         uses: actions/setup-java@v1
20         with:
21           java-version: 11
22           distribution: 'adopt'
23           cache: maven
24       - name: Cache SonarCloud packages
25         uses: actions/cache@v1
26         with:
27           path: ~/.sonar/cache
28           key: ${{ runner.os }}-sonar
29           restore-keys: ${{ runner.os }}-sonar
30       - name: Cache Maven packages
31         uses: actions/cache@v1
32         with:
33           path: ~/.m2
34           key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
35           restore-keys: ${{ runner.os }}-m2
36       - name: Build and analyze
37         env:
38           GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }} # Needed to get PR information, if any
39           SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
40         run: mvn -f Project/prototipo/pom.xml -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=RobertoArastiblanco_MoodleQA

```

Figura D.2: Trabajos de integración continua

Lo primero que se puede apreciar es que la máquina virtual que se encarga de la integración continua utiliza la versión más reciente de Ubuntu. Después, se instala JDK 11 (antes se utilizaba JDK 8 para la integración continua, pero tras añadir SonarCloud al proceso, JDK 11 pasó a ser necesario), luego se almacenan en la memoria caché de la máquina virtual los paquetes de SonarCloud y Maven. Finalmente, se construye y analiza el proyecto, se indica la ubicación del archivo `pom.xml` debido a que el directorio raíz del proyecto no coincide con el del repositorio (después veremos otro problema causado por esto).

Herramienta de calidad de código

Para analizar la calidad del código del proyecto hemos utilizado SonarCloud. Para integrar SonarCloud en un proyecto se puede hacer desde <https://sonarcloud.io/projects/create> y se hace relativamente fácil, ya que durante el proceso se te guía paso a paso.

La página principal resume el estado del proyecto respecto a errores, *code smells*, seguridad, y cantidad de código duplicado. También muestra el estado del proyecto según el *Quality Gate*, que indica si se cumple una serie de condiciones establecidas.

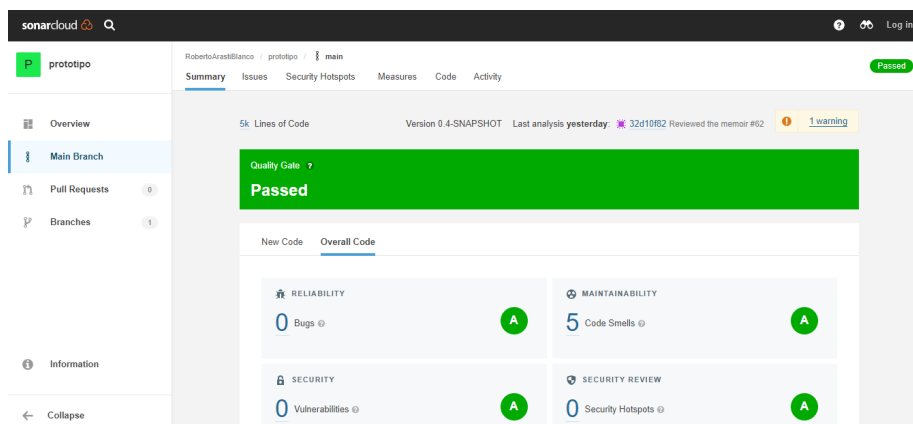


Figura D.3: Página del proyecto en SonarCloud en https://sonarcloud.io/summary/overall?id=RobertoArastiBlanco_MoodleQA

Es muy útil a la hora de arreglar los errores encontrados, ya que posee variedad de filtros para los errores, explica el motivo del error, posible causa, e incluso suele sugerir formas de solventarlo.

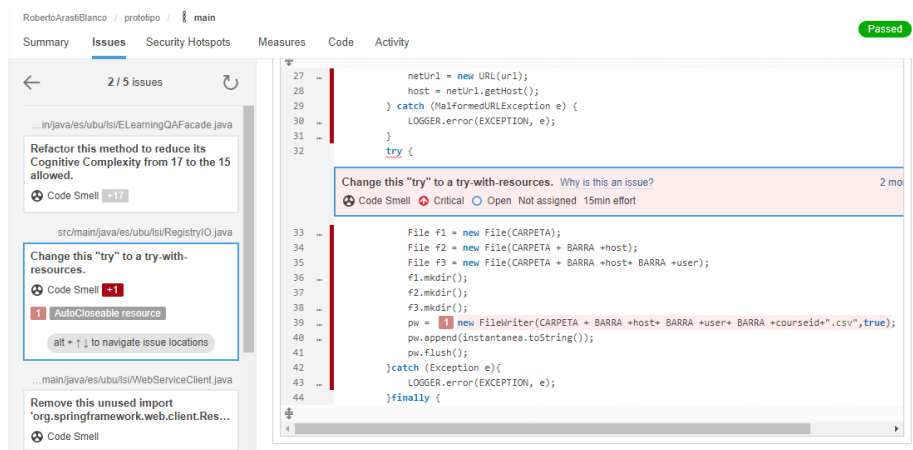


Figura D.4: Apartado de problemas de código

D.4. Compilación, instalación y ejecución del proyecto

Ejecución en local

Para ejecutar la aplicación en local solo es necesario ejecutar la clase principal: Application, IDEA se encarga de compilar las clases necesarias de forma automática. Con la aplicación en ejecución, se puede acceder a ella en <http://localhost:8080/>.

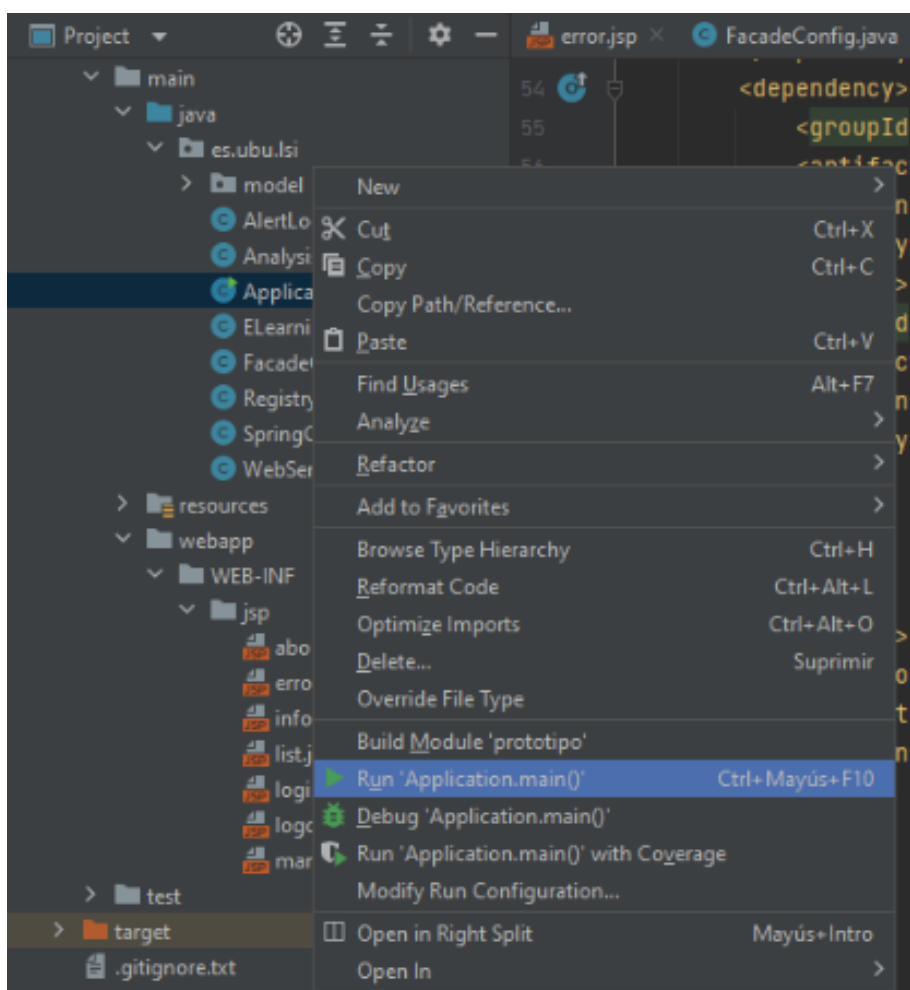


Figura D.5: Ejecución de la clase principal

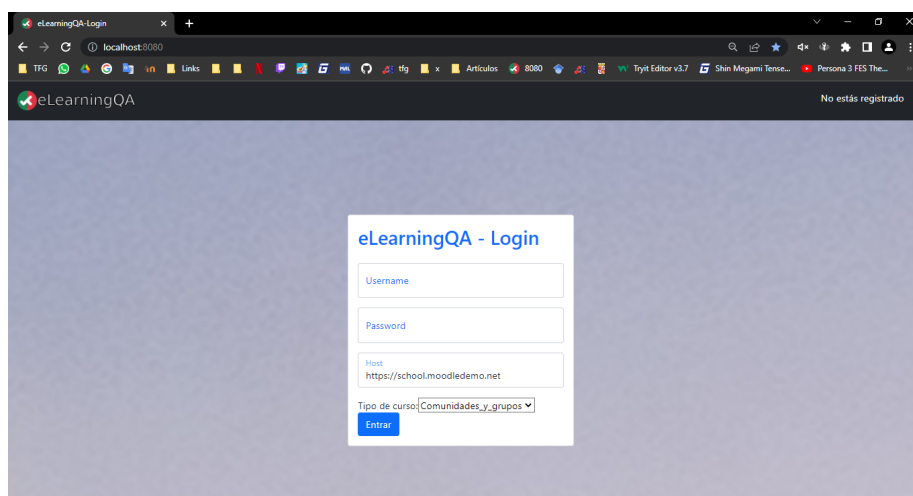


Figura D.6: Ejecución en local

Despliegue en Heroku

A la hora de desplegar la aplicación en Heroku, existen algunas consideraciones a tener en cuenta para conseguirlo.

Para indicar a Heroku cómo ejecutar la aplicación, el directorio raíz del proyecto debe contener un archivo llamado **Procfile** (sin extensión de archivo) con el comando “web: java -jar -Dserver.port=\$PORT target/prototipo-0.4-SNAPSHOT.war” (el nombre del ejecutable puede cambiar al cambiar la versión de la aplicación en **pom.xml**)

Debido a que el directorio raíz de la aplicación difiere del directorio raíz del repositorio, es necesario el uso de un *buildpack* específico, un script que se ejecuta en el despliegue y configura el entorno de ejecución.

Se debe especificar la ruta del directorio raíz de la aplicación asociada a la clave **PROJECT_PATH** en variables de configuración y añadir los buildpacks <https://github.com/timanovsky/subdir-heroku-buildpack.git> y **heroku/java**

The screenshot shows the Heroku configuration interface. The 'Config Vars' section on the left explains that these variables change app behavior. The main area has a 'Config Vars' header with a 'Hide Config Vars' button. Below it are input fields for 'PROJECT_PATH' (containing 'Project/prototipo') and 'KEY', and a 'VALUE' field. An 'Add' button is to the right. The 'Buildpacks' section on the left explains that buildpacks are scripts for installing dependencies. The main area has an 'Add buildpack' button. Below it, two buildpacks are listed: 'https://github.com/timanovsky/subdir-heroku-buildpack.git' and 'heroku/java', each with a remove icon (X).

Figura D.7: Configuración necesaria para el despliegue

En la pestaña “Deployment” debes elegir GitHub como método de despliegue. Al indicar el repositorio, utiliza el *fork* creado, porque se te pedirá registrarte con GitHub para obtener un *token OAuth* para tener acceso al repositorio y se necesita ser colaborador para tener acceso.

The screenshot shows the Heroku deployment configuration. The 'Deployment method' section at the top shows three options: 'Heroku Git' (selected), 'GitHub' (labeled 'Connected'), and 'Container Registry'. Below this, the 'App connected to GitHub' section shows the app is connected to 'RobertoArastiBianco/eLearningQA' by 'RobertoArastiBianco'. It includes links to 'activity feed' and 'main'. The 'Automatic deploys' section at the bottom shows that automatic deploys from the 'main' branch are enabled. A message states: 'You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.' A checkbox 'Wait for CI to pass before deploy' is checked.

Figura D.8: Configuración del despliegue automático

D.5. Pruebas del sistema

Los test se realizan con JUnit. La clase `ELearningQAFacadeTest` contiene los test unitarios de la clase `ELearningQAFacade`. Al principio se realizaban los test contra la página `school.moodledemo.net`, accesible por cualquiera para realizar pruebas en un entorno Moodle, el problema de hacer esto es que ralentizaba la ejecución de los test notablemente y el resultado de estos estaba sujeto a cambios que pudiera hacer otra persona usando la aplicación para hacer pruebas. La solución que se encontró a este problema fue refactorizar los

test para que extrajeran los datos de los cursos de prueba de archivos JSON almacenados en un directorio de recursos de la aplicación. Para generar estos archivos creé de forma temporal una función llamada “guardarComoJSON” que convertía un objeto cualquiera a JSON y lo almacenaba en el directorio “resources/json”. La clase `ELearningQAFacadeTest` contiene una función con la anotación “@BeforeAll” que indica que debe ejecutarse antes que el resto de funciones de test. La función en cuestión inicializa la fachada y carga el conjunto de datos de cursos que se encontraba almacenado en formato JSON.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice se incluyen los detalles de la aplicación que conciernen al usuario, como los requisitos para utilizar la aplicación y un manual de usuario que también existe en formato web y puede ser accedido desde la página de login de la aplicación.

E.2. Requisitos de usuarios

Para utilizar la aplicación como usuario se necesita:

- Una cuenta en un servidor de Moodle con rol de profesor en los cursos a analizar
- Conexión a internet para acceder al despliegue de la aplicación

E.3. Instalación

No se requiere instalación por parte del usuario ya que la aplicación esta desplegada en Heroku y se puede acceder a ella en <https://elearningqa.herokuapp.com>.

E.4. Manual del usuario

Pantallas:

Página de login: la página de login contiene un formulario de inicio de sesión, un link “Acerca de”, un link “Manual de usuario” que muestra la versión web de este manual, y un link de contacto.

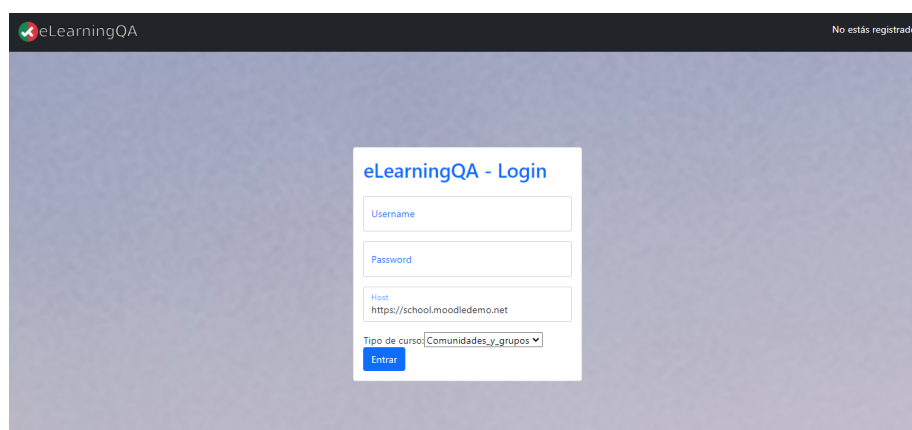


Figura E.1: Página de login

Página principal: la página principal contiene una cabecera que indica el nombre de usuario con el que se ha registrado, un botón para desconectarse, un enlace para generar un informe global de todos los cursos, y una tabla con una lista de enlaces para generar los informes específicos de cada curso con una caja de búsqueda.

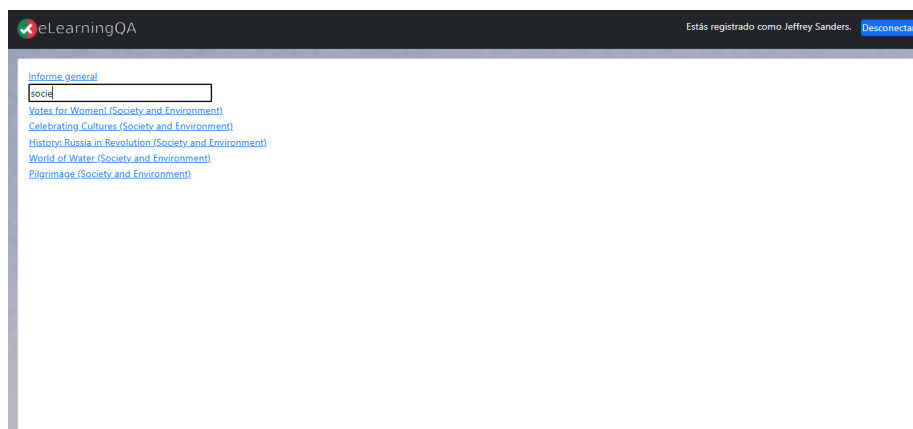


Figura E.2: Página principal

Página de informe: la página de informe genera el informe dependiendo de cómo se haya accedido a esta, ya sea mediante el enlace “Generar informe global” o por medio de los enlaces a informes de cursos específicos.

Los informes contienen los siguientes elementos:

- **Puntuaciones por rol y perspectiva:** el informe contiene una matriz de roles y perspectivas que da distintas puntuaciones a cada uno de los roles (Diseñador, Facilitador, y Proveedor) en cada una de las tres perspectivas definidas (Pedagógica, Tecnológica, y Estratégica) según los resultados del análisis.
- **Gráfico de evolución:** la aplicación almacena estas puntuaciones cada vez que se carga un informe para generar un gráfico interactivo de la evolución de estos valores.

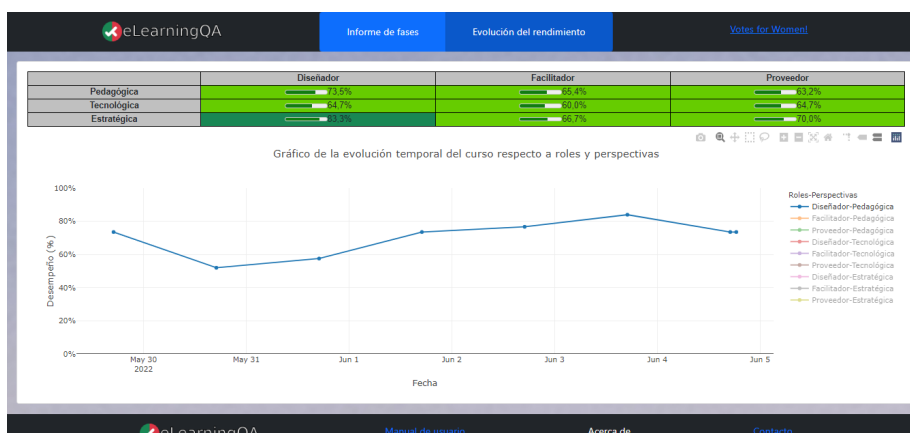


Figura E.3: Página de evolución del rendimiento

- **Informe de fases:** el informe contiene también una tabla que muestra los resultados en cada una de las comprobaciones del análisis y las agrupa en las distintas fases del diseño instruccional (Diseño, Implementación, Realización, y Evaluación) además de dar una puntuación general.
- **Lista de mejoras:** por último, el informe muestra una lista de motivos concretos que producen resultados negativos en el análisis para subsanarlos o tenerlos en cuenta de cara al futuro. Haciendo clic en las filas de la tabla se puede filtrar la lista de mejoras.

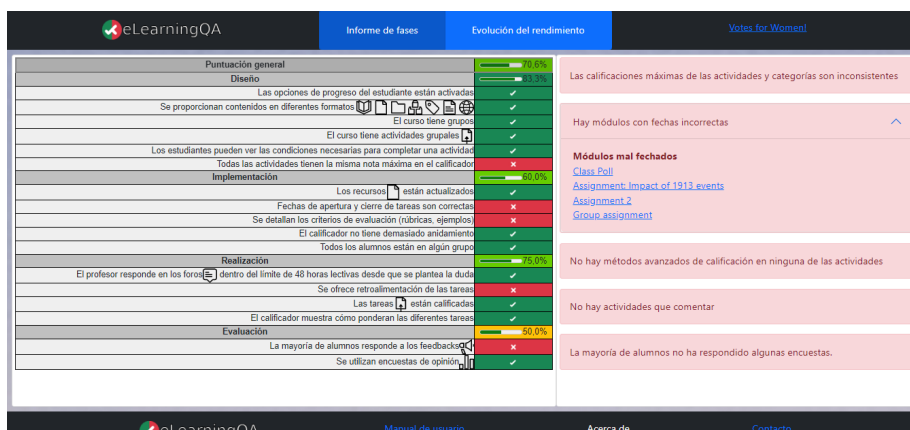


Figura E.4: Página de informe de fases

El informe específico muestra los resultados de las comprobaciones en el informe de fases de forma absoluta, es decir, que solo indica si el criterio se cumple o no, sin embargo, en el informe global, se muestra un resultado u otro dependiendo de la cantidad de cursos que satisfacen dicho criterio.

Acciones del usuario:

- **Login:** para acceder a la aplicación son necesarias las credenciales de acceso a una cuenta de la plataforma Moodle a la que accede la aplicación (en el caso del prototipo es Mount Orange School). Debe introducir su usuario y contraseña en los campos “Username” y “Password”. Si quiere vaciar los campos pulse el botón “Borrar”. Para acceder a la página principal, pulse el botón “Entrar” tras haber introducido sus credenciales.
- **Desconectar:** desde la página principal, si desea finalizar su sesión, pulse el botón “Desconectar”. Esto invalidará sus credenciales y le impedirá acceder a la aplicación hasta que se registre de nuevo con unas credenciales válidas.
- **Generar informe específico:** la página principal muestra una tabla con todos los cursos en los que se encuentra matriculado el usuario registrado en formato de enlace. Al clicar un enlace, se generará un informe en una pestaña aparte del navegador que mostrará los resultados del análisis que ha realizado la aplicación sobre el curso correspondiente.
- **Generar informe global:** en la página principal hay un enlace llamado “Generar informe global”. Al hacer clic sobre este, se generará un informe en una pestaña aparte del navegador que mostrará un resumen de los análisis de todos los cursos en los que se encuentra matriculado el profesor.

Explicación de las comprobaciones de los informes:

Las siguientes comprobaciones están relacionadas con los roles, fases, y perspectivas mencionados anteriormente. Los distintos procesos del diseño instruccional se encuentran divididos en fases, con ciertas perspectivas en mente, y son responsabilidad directa o indirecta de ciertos roles. Al estar las comprobaciones ligadas a esos procesos se muestran agrupadas por fases, y después de la explicación se indican los roles responsables e involucrados, además de las perspectivas correspondientes.

Diseño:

- **Las opciones de progreso del estudiante están activadas:** se comprueba que estén habilitadas las opciones de progreso de los estudiantes en el curso. Responsable: Diseñador Involucrados: Facilitador Perspectivas: Pedagógica
- **Se proporcionan contenidos en diferentes formatos:** se comprueba que haya variedad de formatos en los recursos del curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **El curso tiene grupos:** se comprueba que existan grupos definidos en el curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica
- **El curso tiene actividades grupales:** se comprueba que existan actividades con entrega grupal habilitada en el curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica
- **Los estudiantes pueden ver las condiciones necesarias para completar una actividad:** se comprueba que esté habilitada la opción de mostrar las condiciones para completar una actividad en el curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica
- **Todas las actividades tienen la misma nota máxima en el calificador:** se comprueba que exista una consistencia en las notas máximas de los items de calificación (tareas, entregas, cuestionarios) del curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica

Implementación:

- **Los recursos están actualizados:** se comprueba que los recursos del curso tengan una fecha de creación reciente. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica y Tecnológica

- **Fechas de apertura y cierre de tareas son correctas:** se comprueba que las fechas de apertura y cierre de tareas y cuestionarios no se solapen de forma errónea con las fechas de inicio y fin del curso. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **Se detallan los criterios de evaluación:** se comprueba que exista en al menos una actividad una rúbrica o una guía de calificación en el curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **El calificador no tiene demasiado anidamiento:** se comprueba que la estructura de las categorías del calificador no sea demasiado enrevesada. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica y Estratégica
- **Todos los alumnos están en algún grupo:** se comprueba que cada alumno pertenezca a un grupo. Responsable: Proveedor Involucrados: Diseñador Perspectivas: Tecnológica y Estratégica

Realización:

- **El profesor responde en los foros dentro del límite de 48 horas lectivas desde que se plantea la duda:** se comprueba que no hayan preguntas por parte de alumnos que estén sin responder en un tiempo razonable. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **Se ofrece retroalimentación de las tareas:** se comprueba que el profesor deje comentarios en la mayoría de calificaciones que haga. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **Las tareas están calificadas:** se comprueba que no hayan entregas de alumnos que hayan pasado una semana sin calificación. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **El calificador muestra cómo ponderan las diferentes tareas:** se comprueba que el calificador muestre los pesos de los ítems de calificación. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica

Evaluación:

- **La mayoría de alumnos responden a los feedbacks:** se comprueba que no hayan muchos alumnos que no respondan a los feedbacks. Responsable: Proveedor Involucrados: Diseñador y Facilitador Perspectivas: Pedagógica, Tecnológica, y Estratégica
- **Se utilizan encuestas de opinión:** se comprueba que el curso contenga encuestas de opinión. Responsable: Proveedor Involucrados: Diseñador y Facilitador Perspectivas: Pedagógica, Tecnológica, y Estratégica

Bibliografía

- [1] Moodle. Trademarks - Moodle, 2 2022. [Online; Accedido 26-Febrero-2022]. URL: <https://moodle.com/trademarks>.