



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

eLearningQA Versión 2
Cuestionarios y Foros
Documentación Técnica



Presentado por Alberto Díaz Álvarez
en Universidad de Burgos — 4 de julio de 2023
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	2
A.3. Estudio de viabilidad	5
Apéndice B Especificación de Requisitos	9
B.1. Introducción	9
B.2. Objetivos generales	9
B.3. Catálogo de requisitos	9
B.4. Especificación de requisitos	10
Apéndice C Especificación de diseño	13
C.1. Introducción	13
C.2. Diseño de datos	13
C.3. Diseño procedimental	14
Apéndice D Documentación técnica de programación	17
D.1. Introducción	17
D.2. Estructura de directorios	17
D.3. Manual del programador	19
D.4. Compilación, instalación y ejecución del proyecto	23

D.5. Pruebas del sistema	24
Apéndice E Documentación de usuario	25
E.1. Introducción	25
E.2. Requisitos de usuarios	25
E.3. Instalación	25
E.4. Manual del usuario	26
Bibliografía	37

Índice de figuras

B.1. Diagrama de casos de uso	10
C.1. Diagrama de clases de los modelos utilizados en la segunda versión	14
C.2. Boceto de la interfaz	15
C.3. Diagrama de interacción con la fachada	15
D.1. Manejo de Git con Visual Studio Code	20
D.2. Trabajos de integración continua	22
D.3. Página del proyecto en SonarCloud en https://sonarcloud.io/ project/overview?id=ada1012_eLearningQA	23
D.4. Ejecución en local	24
E.1. Página de login	27
E.2. Página principal	28
E.3. Página de evolución del rendimiento	29
E.4. Página de informe de fases	30
E.5. Estadísticas de un cuestionario	31
E.6. Estadísticas de un foro	31

Índice de tablas

A.1. Costes de personal	6
A.2. Costes de hardware	6
A.3. Costes varios	6
A.4. Costes totales	7
A.5. Licencias del software utilizado	7
B.1. Caso de uso CU-01	11
B.2. Caso de uso CU-02	11
B.3. Caso de uso CU-03	11
B.4. Caso de uso CU-04	11

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Con el fin de abordar el plan del proyecto, desarrollaremos una guía detallada que incluirá los puntos clave y los avances logrados en cada sprint. Además, se registrarán las herramientas seleccionadas, las horas invertidas y otros aspectos relevantes. En cada sprint, se especificarán las fechas de inicio y finalización, los objetivos propuestos y los objetivos cumplidos, junto con los problemas encontrados durante el proceso, lo que nos permitirá mostrar la evolución que hemos logrado hasta el momento.

Además, dentro de la guía del plan de proyecto, abordaremos los requisitos detallados del mismo. Esto incluirá un análisis exhaustivo de las funcionalidades y características que la aplicación debe cumplir para satisfacer las necesidades del cliente y los usuarios finales.

También se contemplará el diseño del proyecto, donde se describirán los aspectos técnicos y estéticos que guiarán el desarrollo.

Además, se proporcionará un manual específico destinado al programador, donde se detallarán las pautas y las mejores prácticas para la codificación de la aplicación. Este manual incluirá la estructura de los directorios identificando cada parte de la aplicación y permitiendo identificar la parte a modificar del código de forma rápida y precisa.

Asimismo, se creará un manual orientado al usuario final. Este manual contendrá instrucciones claras y concisas sobre cómo utilizar la aplicación o proyecto una vez finalizado. Se incluirán descripciones detalladas de las funcionalidades, los flujos de trabajo recomendados, las configuraciones

disponibles y cualquier otra información relevante para que los usuarios finales puedan aprovechar al máximo la aplicación o proyecto.

A.2. Planificación temporal

En esta sección se estimará el tiempo de trabajo en el TFG tanto en la elaboración de la memoria y anexo como en la del propio programa, detallando los acontecimientos a lo largo del proceso.

La planificación del proyecto se ha llevado a cabo en sprints de una duración de dos semanas. Con una reunión al final de cada sprint para comentar los objetivos del último sprint y los nuevos objetivos, en varias ocasiones fue necesario reducir la duración de dichos sprints a una semana. Para el control de versiones tanto de las memorias como del programa se ha utilizado github con un seguimiento a través de commits dejando un histórico detallado. Además de realizar una release cada vez que se tuviese un prototipo funcional de la aplicación.

A continuación se mostrará un resumen de dichos sprints:

Sprint 0 (23/02/2023 - 08/03/2023)

En esta primera reunión se introdujo el proyecto anterior, posibles herramientas con las que trabajar, recursos necesarios y disponibilidad tanto del alumno como de los tutores. Como objetivos se marcaron: inspeccionar el código ya existente, poner en funcionamiento el anterior proyecto, realizar pequeños cambios en el código, inspeccionar el Webservice de Moodle, instalación y creación de Moodle en localhost para empezar a trabajar en local y crear el esqueleto de los documentos en LaTeX. También se comenta en la reunión de este sprint que el profesor Carlos López se encontraba de baja por lo que esta y las posteriores reuniones, hasta su incorporación, las haría con el tutor Raúl Marticorena.

Sprint 1 (08/03/2023 - 22/03/2023)

Durante el segundo sprint tuve varios problemas, tanto para entender cómo funcionaba el código, como para hacerlo funcionar ya que Moodle había cambiado a la versión 4.0 con cambios en sus respuesta en las llamadas de los web services. Un ejemplo es el paso de campos como *isVisible* de un entero a un booleano. Dados estos problemas y mi falta de tiempo, el profesor Raúl Marticorena y yo indicamos como objetivos seguir con la

refactorización, enviar el repositorio de github, probar la aplicación contra mountorange e investigar la integración con Heroku aunque este último punto era el menos prioritario para nosotros.

Sprint 2 (22/03/2023 - 31/03/2023)

Este sprint se decidió hacer de una semana porque a las dos semanas era semana santa y vimos más prioritario realizar la reunión antes y no aplazar la reunión una semana más dejando el sprint de tres o cuatro semanas. Durante esta semana conseguí que la aplicación se adaptase completamente a la nueva versión de Moodle. Para el siguiente sprint seguimos con los mismos objetivos ya que no había avanzado mucho.

Sprint 3 (31/03/2023 - 17/04/2023)

En la revisión de este sprint mostré la incorporación de los modelos necesarios para poder trabajar con cuestionarios en la aplicación y descubrí que Heroku se había vuelto de pago incluso para las aplicaciones de prueba por lo que dejamos el tema de buscar un servidor online de lado. Se mantienen objetivos para el siguiente sprint y se deja la próxima reunión para dentro de una semana.

Sprint 4 (17/04/2023 - 27/04/2023)

Para esta reunión se había incorporado un campo en los informes que mostraba si el curso seleccionado tenía cuestionarios o no. También se añadió un registro que mostraba el porcentaje de alumnos que realizaban los cuestionarios del curso. Esta es la primera vez del trabajo que se realiza una release ya que teníamos algo avanzada la aplicación. Para el siguiente sprint decidimos mantener la reunión semanal y pusimos como objetivo realizar un resumen del cuestionario al hacer click en dicho cuestionario junto con algún gráfico.

Sprint 5 (27/04/2023 - 03/05/2023)

Durante este sprint avancé en la lógica de obtención de datos de dicho resumen pero al llegar la reunión no tenía ninguna implementación definitiva por lo que no se mostraban avances en la interfaz. Para la semana siguiente debería tener implementado ya dicho resumen.

Sprint 6 (03/05/2023 - 11/05/2023)

Para este sprint ya aparecía un resumen en el cuestionario mostrando datos básicos a falta del gráfico. El profesor Raúl Marticorena vió importante meter alguna estadística como el coeficiente de curtosis (con este valor podríamos saber si hay valores atípicos en las notas) y la asimetría. Para la siguiente semana dejamos como objetivo implementar estos datos y el gráfico.

Sprint 7 (11/05/2023 - 18/05/2023)

Durante la semana, se completaron los datos restantes del resumen. Sin embargo, se detectaron errores en los cálculos según lo mencionado por Raúl Marticorena. Para la próxima semana, se planea corregir la forma en que se realizan dichos cálculos. En cuanto al gráfico, se logró implementar la lógica, pero hubo dificultades en la transferencia de información de Java a Javascript. Como resultado, se establece como objetivo para la próxima reunión finalizar el gráfico, iniciar el desarrollo de la memoria y dar inicio a la sección de foros.

Sprint 8 (18/05/2023 - 25/05/2023)

En esta reunión se incorporó el profesor Carlos López por lo que los primeros minutos fueron para ver el progreso en el proyecto, después se aportaron una serie de mejoras y fallos que no habíamos contemplado. Los cambios que se implementaron durante este sprint fueron principalmente arreglar los métodos que realizaban cálculos sobre las estadísticas de los cuestionarios, se consiguió implementar correctamente el gráfico (a falta de normalizar el valor de cada pregunta ya que se llegó a la conclusión de que habría preguntas de diferente puntuación) y se generó una release con el tema de los cuestionarios terminado. La parte de los foros se aplazó al siguiente sprint por falta de tiempo ya que quedaban apenas dos semanas para los exámenes finales.

Sprint 9 (25/05/2023 - 02/06/2023)

Durante este sprint se pulieron algunos fallos detectados por los profesores como la normalización de las notas, el número de intentos y agregar el formato de notas en español (se usan comas en vez de puntos). Luego se agregaron los porcentajes de participación en los foros, se redujo el número de llamadas a la API de Moodle. También se avanzó en la redacción de los anexos. En esta

reunión se decidió que si queríamos llegar a tiempo tendría que formalizarse un borrador de la memoria, de los anexos, generar una release totalmente funcional con todos los cambios e implementar los actions de github como es el de ejecutar SonarCloud cada vez que se haga un push a la rama develop.

Sprint 10 (02/06/2023 - 08/06/2023)

Esta reunión era clave para ver si se hacía entrega del trabajo en primera convocatoria o en segunda. Al ver que se había desarrollado una memoria y unos anexos que estaban poco elaborados y sin estar revisados, además de comprobar que la aplicación daba algunos fallos, se decidió aplazar a segunda convocatoria. También se decidió poner un poco de organización y aunque tarde pero se empezó a hacer uso de las issues y milestones. De ahora en adelante el objetivo era corregir errores, redactar una memoria de calidad e intentar incluir unas últimas mejoras que harían de este un trabajo más completo.

Sprint 11 (08/06/2023 - 26/06/2023)

Durante este sprint, que fue un poco más largo, se corrigieron errores en los cálculos de las estadísticas tanto en los cuestionarios como en los foros, se introdujo una página de carga previa a la muestra del informe ya que el tiempo de carga cada vez era mayor. En la reunión se habló de aconsejar trabajar la asincronía de la aplicación en futuras versiones. Con la aplicación funcionando correctamente comenzó el enfoque en la memoria para el siguiente sprint y en la incorporación de análisis de sentimientos en los foros.

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se estiman los costes que tendría la realización del proyecto en un caso real. Se tendrán en cuenta costes personales, de hardware y costes varios mostrando dos tablas que detallarán en qué gastos se divide cada uno.

La primera versión del proyecto supuso 8 meses de desarrollo con unas 800 horas de trabajo, esto equivaldría a cinco meses. La segunda versión que detalla el tema de los cuestionarios y foros supone en torno a unas 320 horas que equivale a dos meses de trabajo.

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 2 meses	2.679,16€

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. Como se aprecia en la tabla anterior solo en gastos personales se invertirían 2.679,16€.

Se supone que el ordenador portátil se amortiza en cinco años, en la primera versión se utilizó durante cinco meses y en esta segunda versión se ha utilizado durante dos meses aunque separaremos los gastos de versiones y se mostrarán datos únicamente de esta segunda versión.

Tabla A.2: Costes de hardware

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

En la siguiente tabla se desglosarán los costes varios como son el internet y la electricidad, a cada uno de estos le asignamos 30,00€ y 35,00€ respectivamente como coste mensual. Al haber usado coste mensual en la descripción, en la tabla se hará la multiplicación por dos meses mostrando el resultado final.

Tabla A.3: Costes varios

Concepto	Coste
Internet	60,00€
Electricidad	70,00€
Total	130,00€

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

Tabla A.4: Costes totales

Concepto	Coste
Personal	2.679,16€
Hardware	45,83€
Varios	130,00€
Total 2 meses	2.854,99€

Para rentabilizar la aplicación web se podrían tener en cuenta varias posibles soluciones como ofrecer un sistema freenium dejando parte del contenido de forma gratuita mientras que otras partes serían premium ocasionando un coste como es el caso de Spotify. Otra posible solución es la que comenta mi compañero Roberto Arasti en la versión previa mencionando una suscripción mensual los que implicaría unos ingresos continuos [1].

Viabilidad legal

Licencias de software

Respecto a la viabilidad legal en esta segunda versión, solo se ha incorporado GitHub Copilot el cual no ofrece ningún problema legal ya que sus suscripción mensual permite el libre manejo del código generado. A continuación se mostrarán los frameworks y librerías con sus respectivas licencias.

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.

GitHub Copilot es una herramienta de programación desarrollada por GitHub y OpenAI. Utiliza la tecnología de inteligencia artificial de GPT-3 para ofrecer sugerencias de código en tiempo real mientras escribes. Copilot analiza el contexto del código que estás escribiendo y genera automáticamente fragmentos de código relevantes y útiles para completar tus líneas de código. Tiene dos tarifas:

- 10 dólares al mes para uso personal.
- 19 dólares al mes por persona para uso empresarial.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En esta sección se recopilan los requisitos tanto funcionales como no funcionales del software en consideración, basados en los objetivos generales y las expectativas que se han establecido para el proyecto. La especificación de requisitos juega un papel fundamental como canal de comunicación entre todas las entidades involucradas en el desarrollo del software.

B.2. Objetivos generales

A continuación se listarán los objetivos generales de la nueva versión del proyecto:

- Obtener a simple vista la existencia de cuestionarios y foros del curso que se solicita el informe.
- Poder conocer el porcentaje de participación a nivel global y específico de los cuestionarios y foros del curso.
- Acceder a una vista más detallada con la información más importante de dichos cuestionarios y foros.

B.3. Catálogo de requisitos

Después de hablar de los objetivos generales, se listarán los requisitos del sistema:

- R-01 - Comprobación de participación en foros y cuestionarios: la aplicación mostrará en el apartado de diseño una cruz o un tick por cuestionarios y otro por foros dependiendo de si estos existen en dicho curso.
- R-02 - Trazabilidad de participación en cuestionarios: la aplicación implementará en la sección de realización una fila desplegable indicando el porcentaje de alumnos que interactúa con los cuestionarios. Si se despliega se mostrará una nueva fila por cada cuestionario del curso con el porcentaje de participación de cada uno.
- R-03 - Trazabilidad de participación en foros: la aplicación implementará en la sección de realización una fila desplegable indicando el porcentaje de alumnos que interactúa con los foros. Si se despliega se mostrará una nueva fila por cada foro del curso con el porcentaje de participación de cada uno.
- R-04 - Análisis del cuestionario: al hacer click en alguno de los cuestionarios desplegados se mostrarán una serie de estadísticas junto con un grafo.
- R-05 - Análisis del foro: al hacer click en alguno de los foros desplegados se mostrarán una serie de estadísticas.

B.4. Especificación de requisitos

Para comprender estos requisitos se procederá a crear un diagrama de casos de uso de la versión 2 del proyecto:

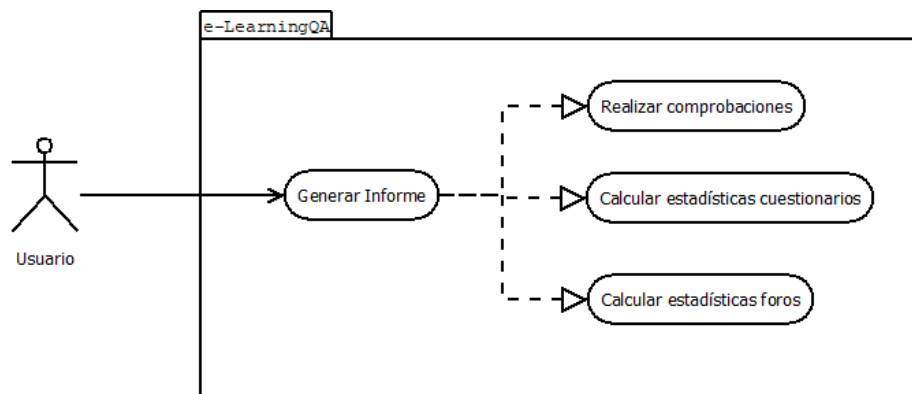


Figura B.1: Diagrama de casos de uso

Tabla B.1: Caso de uso CU-01

CU-01	Generar informe
Descripción	Implementar los cambios oportunos al informe que se generaba en la versión previa
Requisitos asociados	R-01
Precondiciones	Debe acceder con un usuario válido 1.El usuario accede a la aplicación con un usuario, contraseña, dominio y grupo existentes
Ejecución	2.El usuario elige el curso que quiere ver 3.Se mostrarán las 21 comprobaciones existentes del curso.
Postcondiciones	Se ha generado o ampliado un archivo csv para ese servidor, profesor y curso
Excepciones	

Tabla B.2: Caso de uso CU-02

CU-02	Realizar comprobaciones
Descripción	Se realizan las comprobaciones necesarias para evaluar la calidad de los cursos
Requisitos asociados	R-02
Precondiciones	El usuario está logueado en la aplicación y se ha debido generar un token 1.El usuario hace click en un curso 2.Se obtiene el token de sesión
Ejecución	3.Se obtiene la información del curso 4.Se calculan los porcentajes de desempeño 5.Se muestran los resultados en la página
Postcondiciones	
Excepciones	

Tabla B.3: Caso de uso CU-03

CU-03	Calcular estadísticas cuestionarios
Descripción	Obtiene los intentos de los cuestionarios y calcula las estadísticas
Requisitos asociados	R-03
Precondiciones	El usuario está logueado en la aplicación y se ha debido generar un token 1.El usuario accede al informe del curso. 2.Se obtienen los intentos de los cuestionarios. 3.Se calculan las estadísticas de los cuestionarios.
Ejecución	4.Se muestra una comprobación con el porcentaje de participación en los cuestionarios del curso. 5.La comprobación tendrá un botón 'Desplegar'que desplegará una fila por cada cuestionario del curso. 6.Al hacer click en un cuestionario se mostrarán las estadísticas del cuestionario correspondiente.
Postcondiciones	
Excepciones	

Tabla B.4: Caso de uso CU-04

CU-04	Calcular estadísticas foros
Descripción	Obtiene el porcentaje de participación y mensajes en los foros de un curso
Requisitos asociados	R-04
Precondiciones	El usuario está logueado en la aplicación y se ha debido generar un token 1.El usuario hace clic en el curso correspondiente 2.Se obtienen los posts de los foros. 3.Se calcula el porcentaje de participación
Ejecución	4.Se muestra una comprobación con el porcentaje de participación en los foros del curso. 5.La comprobación tendrá un botón 'Desplegar'que desplegará una fila por cada foro del curso. 6.Al hacer click en un foro se mostrarán las estadísticas del foro correspondiente.
Postcondiciones	
Excepciones	

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se proporciona una descripción detallada de cómo se tiene previsto cumplir con el conjunto de objetivos y requisitos establecidos en el apéndice B. Se incluye información sobre los datos utilizados por la aplicación, su arquitectura y su diseño de procedimientos.

C.2. Diseño de datos

En la siguiente tabla se recogen las entidades asociadas a las consultas incorporadas en la nueva versión.

Consulta	Entidades
El curso tiene al menos un cuestionario	Curso, Cuestionario
El curso tiene al menos un foro	Curso, Foro
Un mínimo de alumnos participa en los cuestionarios	Curso, Cuestionario
Un mínimo de alumnos participa en los foros	Curso, Foro

Seguidamente se mostrará una imagen del modelo de datos necesario para comprender la estructura con la que trabaja Moodle y los datos que se deberán recuperar.

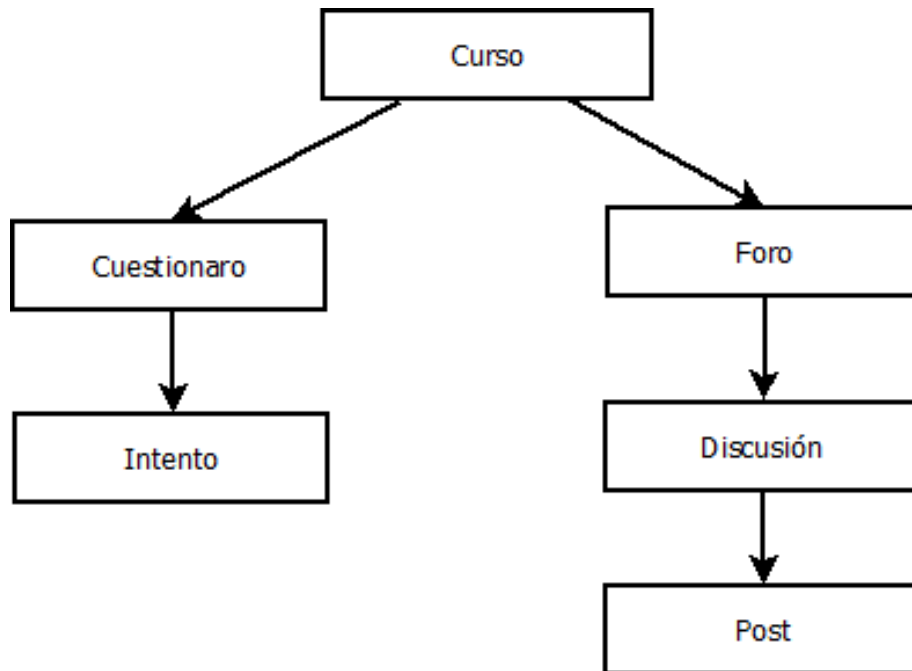


Figura C.1: Diagrama de clases de los modelos utilizados en la segunda versión

C.3. Diseño procedimental

A continuación, se mostrará un diagrama del flujo que implementa la aplicación dividiéndose en las pantallas de login, lista de cursos e informe. La dirección de las flechas indica que al hacer login se accede a la lista de cursos y al hacer logout se vuelve a la pantalla de login, dentro de la pantalla con la lista de cursos se indica que se puede acceder también a la pantalla de los informes. Estas interacciones se detallarán más adelante en el trabajo.

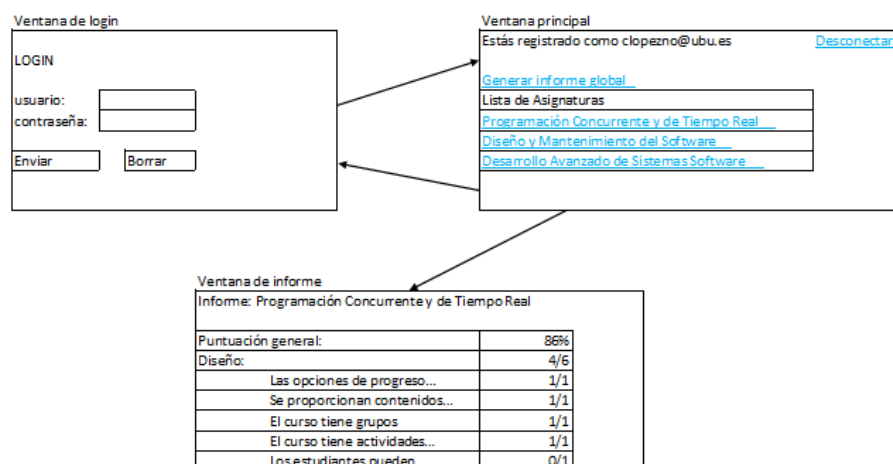


Figura C.2: Boceto de la interfaz

Patrón fachada

El patrón de diseño Fachada es un patrón estructural que proporciona una interfaz simplificada para interactuar con un sistema complejo o conjunto de clases. La idea principal detrás del patrón Fachada es proporcionar una interfaz de nivel superior que simplifique el uso del sistema subyacente. En lugar de que los clientes interactúen directamente con múltiples componentes o clases internas, pueden hacerlo a través de la fachada, que se encarga de manejar las interacciones internas y ofrecer una interfaz más simple y específica.

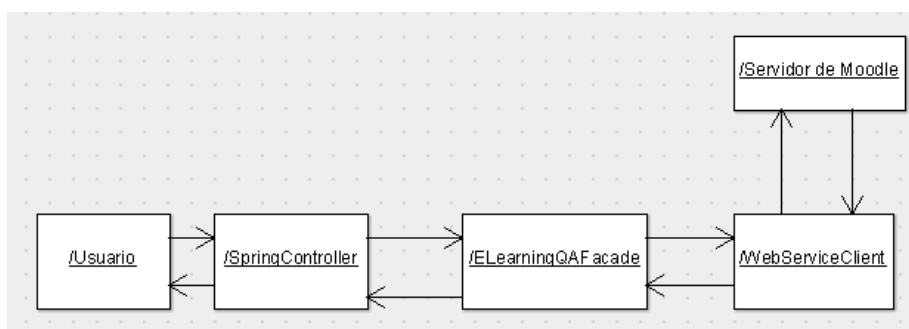


Figura C.3: Diagrama de interacción con la fachada

A continuación se mostrará la definición de Roberto Arasti de la primera versión la cuál no varía y solamente cabría destacar que en este trabajo el patrón de diseño Fachada se ha utilizado para solicitar a la clase WebServi-

ceClient las estadísticas de cuestionarios y foros para luego mostrarlos en el jsp generado. Seguidamente, se explicará como acceder a la información a través del patrón de diseño fachada: cuando un usuario intenta acceder a una página específica, como por ejemplo el informe de la aplicación, la clase SpringController se encarga de manejar la solicitud y proporcionar la respuesta correspondiente. En este caso, la respuesta es un archivo JSP, el cual debe ser compilado y ejecutado para generar una página HTML como resultado final. Al ejecutar el archivo JSP en cuestión, se desencadenan las llamadas necesarias a la fachada para realizar las comprobaciones predefinidas. Estas comprobaciones se llevan a cabo a través de la clase WebServiceClient, que accede al servidor de Moodle. Después de recibir los resultados de las comprobaciones, el archivo JSP utiliza la fachada como una clase de utilidad para procesar algunos de los resultados. Una vez finalizada la ejecución del archivo JSP, la clase SpringController devuelve la página HTML generada al usuario [1].

Apéndice *D*

Documentación técnica de programación

D.1. Introducción

En este apéndice se proporciona una guía detallada destinada a futuros programadores que utilicen el proyecto, con el objetivo de brindarles orientación y ayuda. Se incluye información sobre la estructura de directorios del proyecto, donde se explican en detalle los contenidos de cada uno. Además, se presenta un manual del programador que aborda aspectos relevantes que podrían resultar útiles. También se detallan los pasos necesarios para compilar, instalar y ejecutar el proyecto en un entorno de desarrollo integrado. Por último, se describen las pruebas del sistema que se han llevado a cabo, ofreciendo así una visión completa de las pruebas realizadas.

D.2. Estructura de directorios

La estructura del repositorio se detalla a continuación:

- `/`: contiene en archivo `README` que hace una breve explicación del funcionamiento de la aplicación y las carpetas contenedoras de la aplicación y los ficheros de la memoria.
- `/.github/workflows`: dirección de los *actions* que serán ejecutados en GitHub, en este caso será un archivo `maven.yml`. Este archivo indica que cada vez que se haga un push a la rama *develop* se realizará un análisis de los últimos cambios en SonarCloud.

- **/Memo/plantillaLatex-master**: dirección de la memoria, anexos y archivos en formato \LaTeX para la configuración de los distintos PDFs.
- **/Memo/plantillaLatex-master/img**: carpeta donde se alojan las imágenes que utilizarán la memoria y los anexos.
- **/Memo/plantillaLatex-master/tex**: ficheros en formato \LaTeX que sirven para generar los PDFs de la memoria y anexos.
- **/Project/prototipo**: carpeta que contiene la aplicación completa.
- **/Project/prototipo/configurations**: carpeta con los archivos que asignan las propiedades del proyecto.
- **/Project/prototipo/src**: código fuente de la aplicación Java.
- **/Project/prototipo/src/main**: este directorio abarca el código fuente de la aplicación, los casos de prueba y los recursos empleados en su funcionamiento.
- **/Project/prototipo/src/main/java/es/ubu/lsi**: Dirección de las clases que permiten el correcto funcionamiento de la aplicación web.
- **/Project/prototipo/src/main/java/es/ubu/lsi/model**: dentro de este directorio se encuentran las clases *Java Bean* necesarias para la deserialización de los datos obtenidos a través de servicios web. Estas clases permiten transformar los datos extraídos en objetos que pueden ser fácilmente manipulados y utilizados en el desarrollo de la aplicación.
- **/Project/prototipo/src/main/resources**: contiene el archivo de propiedades de la aplicación web, el cual contiene configuraciones y ajustes específicos para su correcto funcionamiento.
- **/Project/prototipo/src/main/resources/images**: ubicación de las imágenes desplegadas en la web.
- **/Project/prototipo/src/main/resources/json**: información de los cursos que se utilizarán en los casos de pruebas.
- **/Project/prototipo/src/main/resources/json/informe**: ubicación donde se encuentran los textos estáticos utilizados en la tabla del informe de fases, los cuales están almacenados en formato JSON.
- **/Project/prototipo/src/main/resources/static/js**: 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 casos de pruebas 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

Requisitos del proyecto:

- Java JDK 11
- Un IDE como puede ser Eclipse, NetBeans o IntelliJ IDEA

Para descargar el JDK 8 empleado en la aplicación es necesaria una cuenta de Oracle y posteriormente obtener el JDK en el siguiente enlace <https://www.oracle.com/java/technologies/downloads/#java11>.

En este caso como se ha trabajado con eclipse adjunto el enlace de la descarga <https://www.eclipse.org/downloads/>

Obtención del código fuente

Para obtener el código fuente de la aplicación web solo vamos a necesitar clonar el proyecto en nuestro equipo o en el caso de eclipse dentro del propio IDE en la versión importar proyecto Git. El enlace al proyecto es <https://github.com/ada1012/eLearningQA>. Si tenemos intención de hacer cambios en la aplicación, antes de clonar el proyecto deberemos hacer un *fork* para así tener una copia en nuestro perfil de dicho proyecto. En mi caso al manejar git con la extensión de Visual Studio Code se pueden manejar los *pull*, *push* y *commits* desde la interfaz gráfica, en la siguiente figura se puede apreciar el estado de las ramas del repositorio.

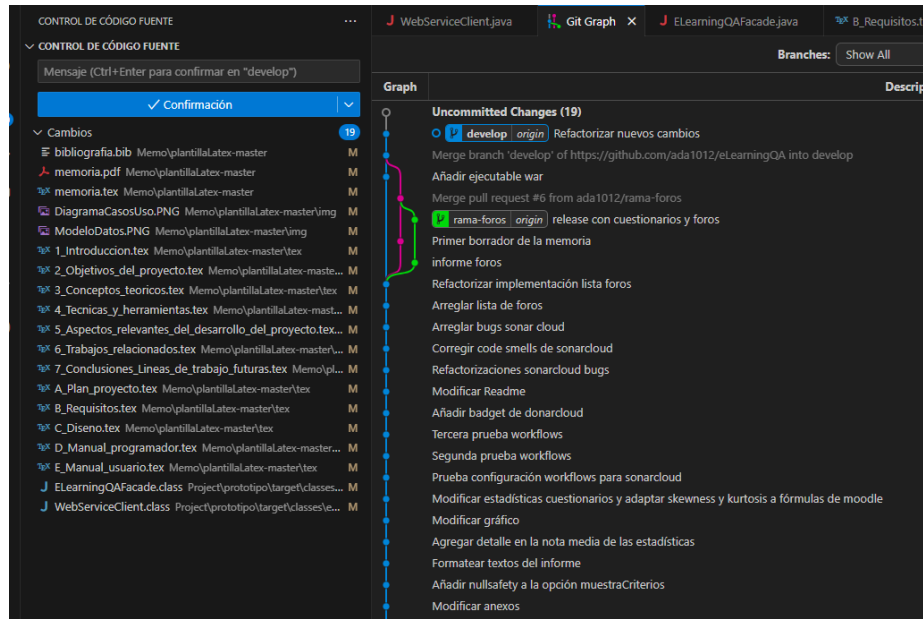


Figura D.1: Manejo de Git con Visual Studio Code

Importar el proyecto en eclipse

Para clonar el proyecto de GitHub en Eclipse debemos copiar la URL del repositorio de GitHub que contiene el proyecto. En Eclipse, seleccionaremos *File* en el menú y seguidamente la opción *Import*, en la lista de opciones, expande la carpeta *Git* y selecciona *Projects from Git*. Posteriormente, seleccionamos *Clone URI* y pegamos la URL del repositorio de GitHub. En la siguiente pantalla, seleccionamos *Import as general project* con la casilla del proyecto Maven seleccionada y ya estaría el repositorio listo en el IDE de Eclipse.

Propósito de las clases principales

- **Application:** clase principal de la aplicación desde la que se 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. También trabajará con los datos obtenidos de Moodle para adaptarlos a las estadísticas de cuestionarios y foros.
- **FacadeConfig:** almacena la configuración elegida por el usuario, junto con el host de Moodle, se almacena como atributo en la fachada para poder pasarlo a las funciones del `WebServiceClient` y así permitir que este último modifique su comportamiento de acuerdo a la configuración establecida.
- **AlertLog:** la clase `AlertLog` se encarga de generar y almacenar mensajes de alerta. Estos mensajes se pasan como parámetro a las funciones del `WebServiceClient` con el fin de que puedan ser almacenados en la lista de mejoras del informe.
- **RegistryIO:** se encarga de almacenar y acceder a la información guardada en archivos CSV que se utilizan para generar los gráficos. Además, proporciona una función que permite generar un gráfico utilizando los datos contenidos en un archivo CSV.
- **AnalysisSnapshot:** representa un conjunto de datos almacenados en cada línea de los archivos CSV. Esta clase proporciona funcionalidades que facilitan el almacenamiento y la manipulación de los datos, así como la generación de gráficos a partir de ellos.

Integración continua

Dentro del proyecto se encuentra un archivo denominado `maven.yml`, ubicado en la ruta `/.github/workflows`. Este archivo contiene una serie de instrucciones que se ejecutan automáticamente cuando se realiza un push a la rama `develop` del repositorio. A continuación, proporcionaremos una breve explicación de dichas instrucciones.

```

github > workflows > ! maven.yml

You have 4 files | 2 authors (You and others)
1 name: SonarCloud
2 on:
3   push:
4     branches:
5       - develop
6   pull_request:
7     types: [opened, synchronize, reopened]
8 jobs:
9   build:
10    name: Build and analyze
11    runs-on: ubuntu-latest
12    steps:
13      - uses: actions/checkout@v3
14        with:
15          fetch-depth: 0 # Shallow clones should be disabled for a better relevancy of analysis
16      - name: Set up JDK 11
17        uses: actions/setup-java@v3
18        with:
19          java-version: 11
20          distribution: 'zulu' # Alternative distribution options are available.
21      - name: Cache SonarCloud packages
22        uses: actions/cache@v3
23        with:
24          path: ~/.sonar/cache
25          key: ${{ runner.os }}-sonar
26          restore-keys: ${{ runner.os }}-sonar
27      - name: Cache Maven packages
28        uses: actions/cache@v3
29        with:
30          path: ~/.m2
31          key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
32          restore-keys: ${{ runner.os }}-m2
33      - name: Build and analyze
34        env:
35          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }} # Needed to get PR information, if any
36          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
37      run: mvn -f Project/prototipo/pom.xml -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=ada1012_eLearningQA
38

```

Figura D.2: Trabajos de integración continua

En primer lugar, se destaca que la máquina virtual utilizada para la integración continua se basa en la versión más actualizada de Ubuntu. La máquina virtual se refiere al entorno de ejecución en el que se ejecuta el flujo de trabajo de GitHub Actions. Posteriormente, se procede a la instalación de JDK 11. Además, se realiza el almacenamiento en caché de los paquetes de SonarCloud y Maven en la memoria de la máquina virtual.

Por último, se lleva a cabo la construcción y análisis del proyecto, indicando explícitamente la ubicación del archivo `pom.xml`. Esta especificación se debe a que el directorio raíz del proyecto difiere del directorio raíz del repositorio, lo cual ha ocasionado algunos inconvenientes que abordaremos posteriormente.

Herramienta de calidad de código

Para analizar la calidad del código del proyecto se ha utilizado SonarCloud como se hizo en el trabajo anterior.

La página principal brinda un resumen conciso del estado actual del proyecto en términos de errores, defectos de código, aspectos de seguridad y la cantidad de código duplicado presente. Además, proporciona información sobre el estado del proyecto con respecto al Quality Gate, que evalúa si se cumplen una serie de condiciones predefinidas.

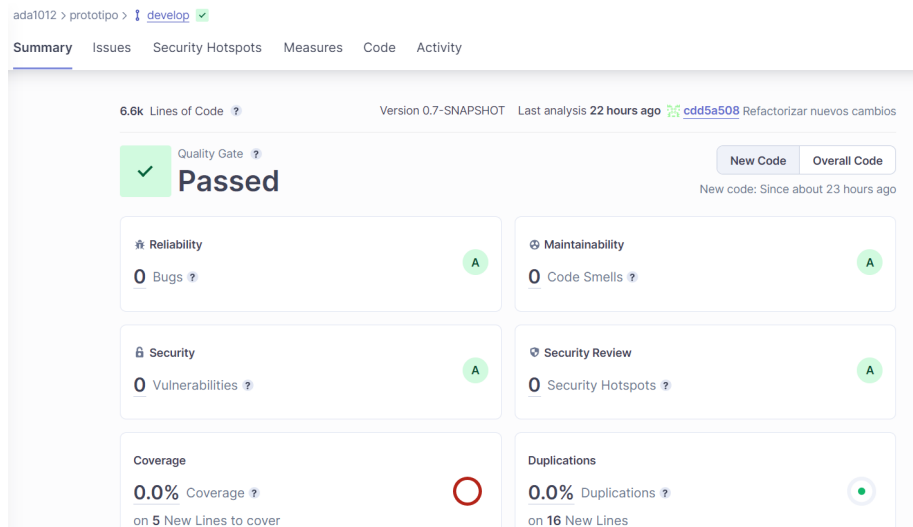


Figura D.3: Página del proyecto en SonarCloud en https://sonarcloud.io/project/overview?id=ada1012_eLearningQA

Con esta herramienta se consigue una ayuda a la refactorización inmediata ya que cada vez que se suba un fragmento de código al proyecto esta aplicación resumirá los distintos fallos encontrados, haciendo distinción entre los ya existentes y los nuevos.

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, los IDEs mencionados previamente los ejecutarán sin problema al ejecutar la opción **Run** y se podrá acceder a dicha aplicación a través de la url: <http://localhost:8080/>.

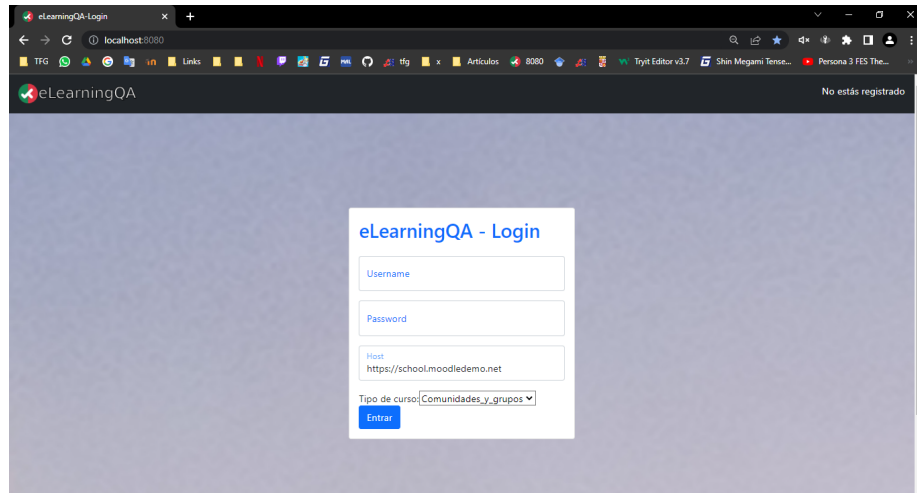


Figura D.4: Ejecución en local

D.5. Pruebas del sistema

Los test se realizan con JUnit y se han mantenido con respecto a la versión anterior ejecutándose a través del archivo `ELearningQAFacadeTest` y yendo contra unos archivos estáticos en formato JSON los cuales se encuentran en la ruta *resources/json*. Ante las nuevas funcionalidades habría que modificar el archivo `ELearningQAFacadeTest` incorporando los test de los métodos que queramos comprobar y asignar unos datos estáticos en el archivo JSON. La opción de obtener los datos de prueba de un servidor de momento es una opción poco viable por el tema del tiempo de carga y el aumento de llamadas en el futuro.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice se presentan los aspectos relevantes de la aplicación que son de interés para el usuario, tales como los requisitos necesarios para utilizar la aplicación y un manual de usuario. Además, se destaca que el manual de usuario está disponible en formato web y puede ser accedido desde la página de inicio de sesión de la aplicación.

E.2. Requisitos de usuarios

Los requisitos como usuario son:

- 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 y para poder obtener información de dichos cursos.

E.3. Instalación

Como se ha mencionado en temas anteriores con Java ya instalado el único requisito es ejecutar el archivo WAR que se genera a partir del código fuente de la aplicación.

Para ejecutar un archivo WAR sin tener el IDE de Eclipse, puedes seguir los siguientes pasos:

1. Asegúrate de tener instalado Java Development Kit (JDK) en tu sistema. Puedes verificarlo ejecutando el comando `java -version` en tu terminal o símbolo del sistema.
2. Abre una terminal o símbolo del sistema y navega hasta el directorio donde se encuentra el archivo WAR.
3. Ejecuta el siguiente comando para iniciar el servidor web embebido Tomcat:

```
java -jar prototipo-0.8-SNAPSHOT.war
```
4. El servidor Tomcat iniciará y desplegará la aplicación. Verás un mensaje en la terminal indicando la URL en la que puedes acceder a la aplicación (<http://localhost:8080/>). Abre un navegador web y visita esa URL para acceder a la aplicación.

E.4. Manual del usuario

Pantallas:

Página de login: la página de inicio de sesión de la aplicación incluye un formulario donde los usuarios pueden ingresar sus credenciales para acceder al sistema. Además, se encuentran disponibles enlaces adicionales para facilitar la navegación y brindar información adicional. Estos enlaces incluyen un enlace “Acerca de” que proporciona detalles sobre la aplicación, un enlace “Manual de usuario” que redirige a la versión web del manual de usuario, y un enlace de contacto para comunicarse con el equipo de soporte.

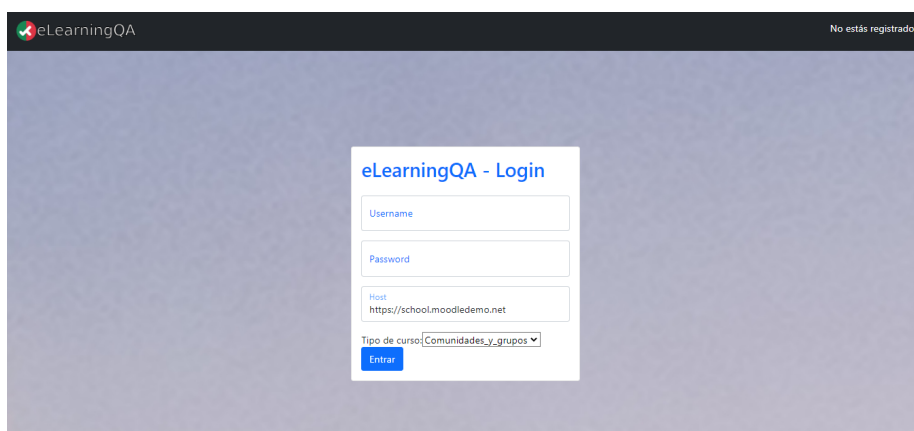


Figura E.1: Página de login

Página principal: la página principal de la aplicación presenta una cabecera que muestra el nombre de usuario registrado, ofreciendo así una experiencia personalizada. Además, incluye un botón para desconectarse y finalizar la sesión del usuario. También se encuentra disponible un enlace que permite generar un informe global que abarca todos los cursos.

En la página principal, se muestra una tabla que contiene una lista de enlaces para generar informes específicos de cada curso. Estos enlaces permiten acceder rápidamente a los informes detallados de cada curso en particular. Además, se incluye una caja de búsqueda que facilita la búsqueda y filtrado de cursos dentro de la tabla. Esto proporciona una manera conveniente de acceder a la información deseada de manera eficiente.

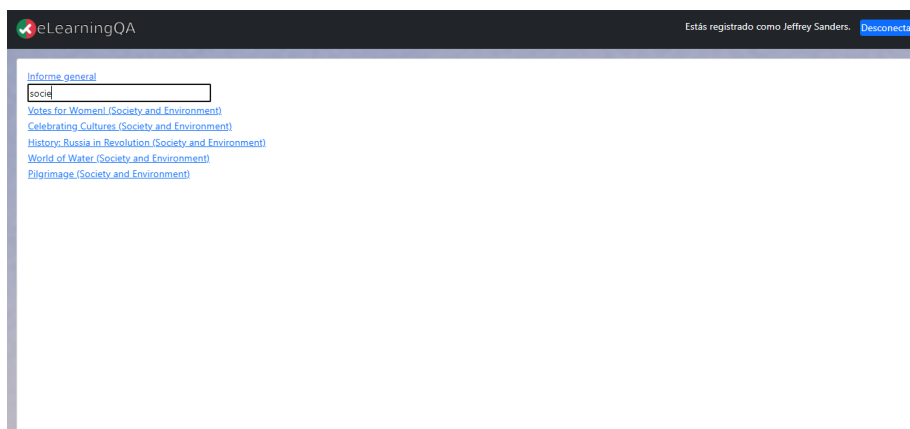


Figura E.2: Página principal

Página de informe: la página de informe variará dependiendo de si se ha hecho uso de la opción “Generar informe global” o por el contrario si se ha elegido la opción de un curso en específico.

Los informes están compuestos por los siguientes apartados:

- **Puntuaciones por rol y perspectiva:** el informe incluye una matriz que evalúa y asigna puntuaciones a cada uno de los roles (Diseñador, Facilitador y Proveedor) en relación a las tres perspectivas establecidas (Pedagógica, Tecnológica y Estratégica). Esta matriz refleja los resultados del análisis y proporciona una visión comparativa de cómo se desempeñan los roles en cada una de las perspectivas mencionadas. A través de las puntuaciones asignadas, se puede evaluar el grado de cumplimiento y efectividad de cada rol en relación a las diferentes dimensiones pedagógicas, tecnológicas y estratégicas identificadas en el informe.
- **Gráfico de evolución:** la aplicación guarda las puntuaciones cada vez que se genera un informe, permitiendo así la creación de un gráfico interactivo que muestra la evolución de estos valores a lo largo del tiempo.



Figura E.3: Página de evolución del rendimiento

- **Informe de fases:** en el informe, se incluye una tabla que presenta los resultados de cada una de las comprobaciones realizadas en el análisis. Estas comprobaciones están agrupadas en las diferentes fases del diseño instruccional, que incluyen Diseño, Implementación, Realización y Evaluación. Además, se proporciona una puntuación general que resume el desempeño global del proyecto.

En el caso específico de los cuestionarios y foros, se presenta una lista desplegable que permite desglosar los resultados de cada uno de ellos. En esta lista, se muestra el porcentaje de participación de cada cuestionario o foro, así como el porcentaje de participación general. Esta información brinda una visión detallada de la contribución de cada componente y su impacto en el conjunto del proyecto, permitiendo una evaluación más precisa y completa.

- **Lista de mejoras:** en el informe, se presenta una lista de motivos concretos que han generado resultados negativos en el análisis. Estos motivos son identificados para poder abordarlos y tenerlos en cuenta en futuras acciones. Al hacer clic en las filas de la tabla, se tiene la capacidad de filtrar la lista de mejoras, lo que permite una exploración más específica y focalizada de los aspectos que requieren atención y solución.
- **Informe de estadísticas:** finalmente, el informe presenta una sección dedicada a las estadísticas de los cuestionarios y foros. En el caso de los cuestionarios, se muestra un gráfico que representa el índice de dificultad de las preguntas, lo que proporciona una visión clara sobre la complejidad del contenido evaluado. En cuanto a los foros, se incluye

una lista que muestra los distintos hilos de discusión presentes en ellos, brindando una visión general de los temas abordados y fomentando la participación activa de los usuarios en las conversaciones relevantes. Además, en cada foro se muestra un análisis de sentimiento de los mensajes que indicarán si tienen una connotación positiva o negativa.

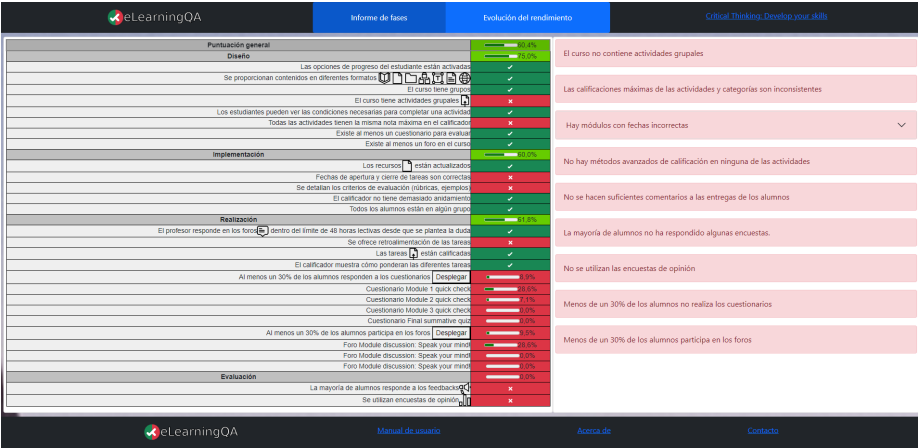


Figura E.4: Página de informe de fases

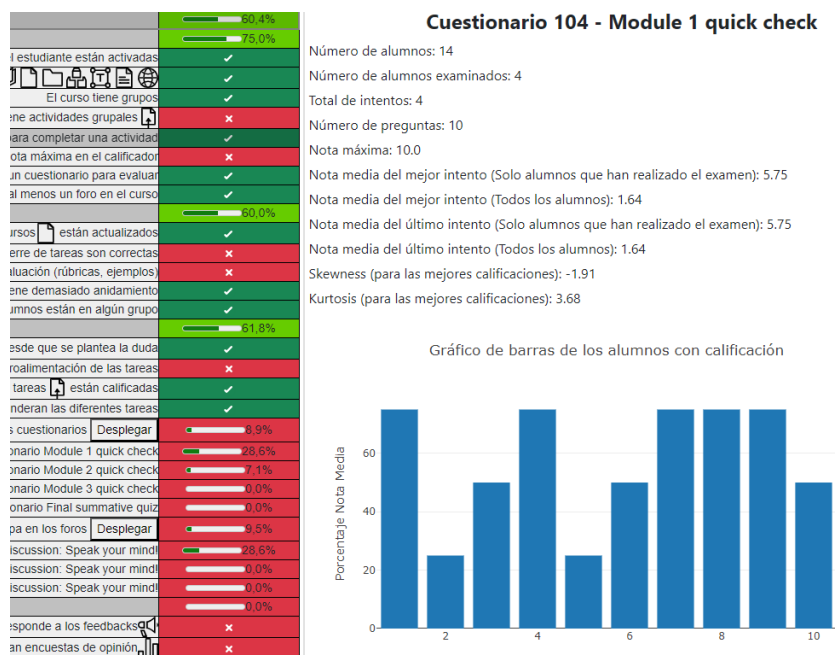


Figura E.5: Estadísticas de un cuestionario

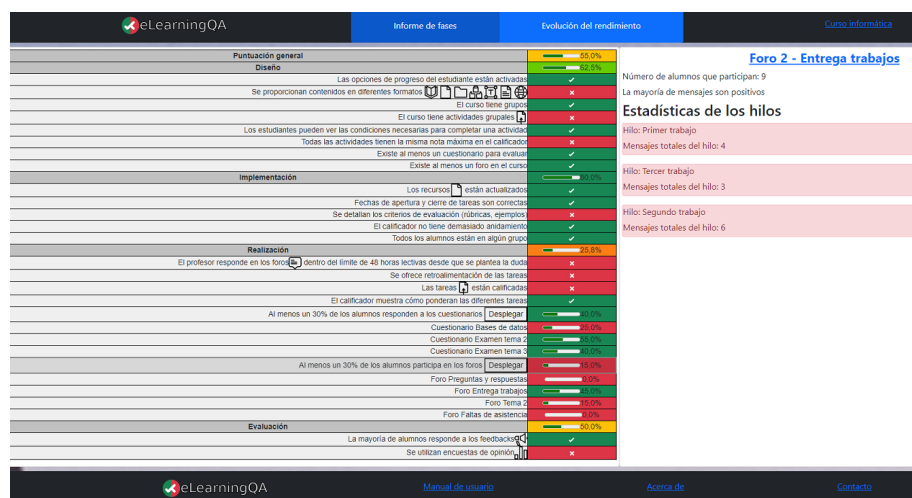


Figura E.6: Estadísticas de un foro

En el informe específico, los resultados de las comprobaciones en el informe de fases se presentan de manera absoluta, es decir, se indican simplemente si el criterio se cumple o no. Sin embargo, en el informe global, se utiliza un enfoque diferente. En lugar de mostrar resultados individuales

para cada curso, se evalúa la cantidad de cursos que cumplen cada criterio. De esta manera, se muestra un resultado diferente según el número 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
- **Segunda versión - El curso tiene al menos un cuestionario:** se comprueba que exista un cuestionario dentro de la planificación del curso. Esta comprobación es esencial para garantizar que los estudiantes tengan la oportunidad de evaluar su comprensión y aplicar los conocimientos adquiridos durante el curso. Responsable: Diseñador Involucrados: Facilitador y Proveedor Perspectivas: Pedagógica
- **Segunda versión - El curso tiene al menos un foro:** se comprueba que exista un foro dentro de la planificación del curso. Esta comprobación es fundamental para fomentar la participación activa de los estudiantes y promover la colaboración y el intercambio de ideas entre ellos. 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 items de calificación. Responsable: Facilitador Involucrados: Diseñador y Proveedor Perspectivas: Pedagógica y Tecnológica
- **Segunda versión - Al menos un X % de los alumnos responden a los cuestionarios:** se comprueba que hay un mínimo de alumnos que interactúan con los cuestionarios propuestos para el desarrollo del curso. Realizando una comprobación general media de todos los cuestionarios y una comprobación más específica por cada uno de los cuestionarios del curso. Responsable: Proveedor Involucrados: Diseñador y Facilitador Perspectivas: Pedagógica
- **Segunda versión - Al menos un X % de los alumnos participa en los foros:** se comprueba que hay un mínimo de alumnos que interactúan con los foros propuestos para el desarrollo del curso. Realizando una comprobación general media de todos los foros y una comprobación más específica por cada uno de los foros del curso. Responsable: Proveedor Involucrados: Diseñador y Facilitador Perspectivas: Pedagógica

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] Roberto Arasti Blanco. elearningqa, 2022.