



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

MoodleQA



Presentado por Roberto Arasti Blanco
en Universidad de Burgos — 9 de enero
de 2022

Tutores: Raúl Marticorena Sánchez y Carlos
López Nozal



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Raúl Marticorena Sánchez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

D. Carlos López Nozal, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Roberto Arasti Blanco, con DNI 71307060E, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de enero de 2022

Vº. Bº. del Tutor:

Vº. Bº. del tutor:

D. Raúl Marticorena Sánchez

D. Carlos López Nozal

Resumen

El e-learning ha tomado un nivel de relevancia pasmoso debido a la pandemia de COVID-19 que ha obligado a las instituciones educativas tradicionales a adaptarse a la situación. La calidad del e-learning es un aspecto fundamental para la enseñanza hoy en día. Existen frameworks para valorar la calidad de un curso en línea desde distintas perspectivas (pedagógica, tecnológica y estratégica), identificando roles en el curso (diseñador, facilitador y proveedor) y en las diferentes fases (análisis, diseño, implementación y evaluación). A pesar de esto, el control de la calidad en este ámbito todavía no cuenta con herramientas capaces de ayudar a la toma de decisiones de forma automática. Bajo esta necesidad de valorar la calidad de los cursos en línea y marco teórico de calidad, este proyecto tiene como objetivo principal adaptar un framework de calidad definiendo métricas obtenidas con consultas sobre los cursos del entorno de aprendizaje Moodle. Además se desarrolla una aplicación software que dado un curso o cursos Moodle genere una evaluación de calidad utilizando como fuentes de consultas, registros de proporcionados por Moodle, servicios Web y técnicas de Web scraping.

Descriptores

Aplicación web, diseño instruccional, métricas, calidad, e-learning, revisiones automáticas, calidad en cursos en línea, aplicación software.

Abstract

E-learning has achieved an astonishing level of relevance due to the COVID-19 pandemic that has forced traditional learning institutions to adapt to the situation. The quality of e-learning is a fundamental aspect of teaching nowadays. There are frameworks to assess the quality of an online course from different perspectives (pedagogical, technological and strategic), identifying roles in the course (designer, facilitator and provider) and in the different phases (analysis, design, implementation and evaluation). Despite this, quality assurance in this area is unprovided of tools able to help in the decision making process in an automatic way. Under this necessity of assessing online course quality and theoretical framework of quality, this project has as a main objective to adapt a quality framework defining metrics obtained with queries about courses in the Moodle learning environment. In addition, a software application is developed that, given a Moodle course or courses, generates a quality evaluation using as query sources, records provided by Moodle, Web services and Web scraping techniques.

Keywords

Web application, instructional design, metrics, quality, e-learning, automatic revisions, online course quality, software application.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Definiciones básicas	5
3.2. Marco de referencia de calidad de MOOQ	5
3.3. Consultas	7
Técnicas y herramientas	11
4.1. Técnicas	11
4.2. Entorno de desarrollo integrado: IntelliJ IDEA	11
4.3. Patrón de diseño: Fachada	12
4.4. Hospedaje del repositorio: GitHub	12
4.5. Control de versiones: GitHub Desktop	12
4.6. Hospedaje de la aplicación: Heroku	12
4.7. Ejecución de tests: JUnit	12
4.8. Construcción automática del proyecto: Maven	13
4.9. Generación de tablas: TablesGenerator.com	13
4.10. Paso de JSON a POJO: Json2CSharp.com	13
4.11. Redacción en HTML: HtmlNotepad	13

4.12. Generación de diagramas UML: ArgoUML	13
4.13. Redacción de memoria y anexos: TeXstudio	14
4.14. Herramientas para acceder a la información	14
4.15. Framework de desarrollo web	14
Aspectos relevantes del desarrollo del proyecto	17
5.1. Ciclo de vida	17
5.2. Proceso de obtención de llamadas a los servicios web	18
5.3. Integración continua y despliegue continuo	20
Trabajos relacionados	21
Conclusiones y Líneas de trabajo futuras	25
Bibliografía	27

Índice de figuras

5.1. Obtención del token	19
5.2. Obtención del JSON	19
5.3. Uso del conversor	20

Índice de tablas

Introducción

El e-learning ha sido en los últimos años una forma de educar a personas con horarios poco flexibles debido a su naturaleza principalmente asíncrona o sobrepasar otra clase de limitaciones de la enseñanza tradicional, sin embargo, debido a la situación de pandemia global y consecuente confinamiento, el e-learning ha tomado un papel principal en la educación [10]. La calidad de la enseñanza desde el punto de vista del alumno es el factor que más influye en la intención de ingreso y la propensión a recomendar la institución de enseñanza [5]. Existen marcos de calidad que se podrían aplicar al e-learning, pero al no haber sido concebidos en concreto para este objetivo no son tan efectivos [10]. También existen trabajos que ponderan la factibilidad de implantar un sistema automático de evaluación de la calidad del e-learning sin entrar en detalles de los marcos de calidad a utilizar [1] y otros que analizan los datos de forma menos automática para aplicar mejoras en el e-learning a posteriori [12] o aquellos que tienen una forma de mostrar los datos para la toma de decisiones pero obtienen la información por medio de entrevistas y encuestas [6]. Además, hay montones de modelos, estándares y marcos de calidad creados con el e-learning en mente pero se encuentran a niveles de abstracción demasiado altos o utilizan comprobaciones demasiado complicadas para automatizarlas a fecha de hoy. El objetivo de este proyecto es crear una herramienta que pueda recoger información de forma automática sobre la calidad de los cursos en la plataforma de e-learning Moodle, que de hecho es utilizada por la Universidad de Burgos, para luego mostrar dicha información y así permitir a los docentes entrar en un ciclo de mejora de la calidad de sus cursos.

Objetivos del proyecto

El objetivo principal de este trabajo es desarrollar una aplicación de navegador que permita al profesor evaluar su propio diseño instruccional a través de sus asignaturas en Moodle.

Los objetivos técnicos de este trabajo para conseguir el objetivo principal son:

- Ser capaces de extraer información de la web para analizarla.
- Poder generar informes de forma automática.

Conceptos teóricos

3.1. Definiciones básicas

- **E-learning:** El e-learning es la enseñanza impartida por medios electrónicos como internet, plataformas virtuales, medios audiovisuales... etc.
- **Moodle:** Moodle es una plataforma de aprendizaje que permite a los profesores crear entornos de aprendizaje altamente personalizables. Fue creado por Martin Dougiamas que publicó su primera versión el 20 de agosto de 2002[2].
- **Calidad:** La calidad se puede definir como la capacidad de satisfacer una serie de necesidades y en el caso del e-learning se trata de las necesidades educativas del alumno, como la calidad del material educativo o la ayuda a la comprensión de este.

3.2. Marco de referencia de calidad de MOOQ

Ahora paso a explicar las tres dimensiones del marco de referencia de calidad del e-learning de MOOQ [11]. MOOQ es la alianza europea por la calidad de los MOOC (Massive Open Online Courses). Un MOOC varía respecto al caso de e-learning para el que pretendemos crear la aplicación, pero al reducir la calidad del diseño instruccional a sus fases, roles y perspectivas aplicado a un tipo de e-learning es lo suficientemente válido como para ser útil.

Fases

- **Análisis:** En esta fase se definen los objetivos, el contexto, y los recursos (docentes, tiempo, presupuesto...) para la ejecución para comprender la situación inicial.
- **Diseño:** En esta fase se define lo que se planea hacer a partir de los resultados de la fase de análisis, como por ejemplo el enfoque de enseñanza que se piensa llevar a cabo.
- **Implementación:** En esta fase se define de qué manera se van a llevar a cabo los planes descritos en la fase de diseño, como por ejemplo de qué manera se va a producir el contenido.
- **Realización:** Esta fase es en la que se interactúa con el alumno, se gestionan los problemas técnicos y las dudas de los alumnos, además de evaluar su aprendizaje.
- **Evaluación:** En esta última fase se evalúa la calidad del resto de fases mediante encuestas, entrevistas, u otros medios.

Roles

Los roles son conjuntos de responsabilidades asumidos por una o más personas. Una persona podría tomar dos o más roles dado el caso (diseñador y facilitador, por ejemplo).

- **Diseñadores:** Los encargados de decidir de qué forma se van a impartir el curso y generan el contenido (autores, expertos en el tema, diseñadores instruccionales).
- **Facilitadores:** Son aquellos que conocen la materia a enseñar y son capaces de explicarlo y dar feedback, además de seguir el aprendizaje de los alumnos.
- **Proveedores:** Son los encargados de proveer los medios digitales para llevar a cabo el e-learning (programadores, diseñadores y desarrolladores de software).

Perspectivas

- **Pedagógica:** El punto de vista que se centra en el contenido y en el aprendizaje por parte del alumno. Los procesos relacionados tienen que ver con el contenido, el feedback, y la evaluación de los alumnos.

- **Tecnológica:** El punto de vista que se centra en las necesidades tecnológicas del curso. La mayoría de los procesos está relacionada con esta perspectiva debido a la naturaleza del e-learning.

- **Estratégica:** El punto de vista que se centra en la consecución de los objetivos del curso dentro del tiempo y presupuesto establecidos para este. Los procesos relacionados tienen que ver con los objetivos, conceptos y el contexto en el que se enseña (presupuesto, demanda, competencia...).

3.3. Consultas

En este apartado definiré qué información intentamos conseguir de cada fase (nos vamos a centrar en las fases de diseño, implementación y realización), rol y perspectiva mediante distintos métodos. La mayoría de consultas provienen de la lista de comprobación de UBUCEV de asignaturas virtuales. Leyenda: Responsabilidad: R=Responsable, X=Involucrado Perspectivas: P=Pedagógica, T=Tecnológica, E=Estratégica

Diseño:

Consulta	Perspectivas	Diseñador	Facilitador	Proveedor	Proceso
Las opciones de progreso del estudiante están activadas	P	R	X		D-5
Se proporcionan contenidos en diferentes formatos	PT	R	X	X	D-4
El curso tiene grupos	P	R	X	X	D-3
El curso tiene actividades grupales	P	R	X	X	D-3
Los estudiantes pueden ver las condiciones necesarias para completar una actividad	P	R	X	X	
Todas las actividades tienen la misma nota máxima en el calificador	P	R	X	X	

Implementación:

Consulta	Perspectivas	Diseñador	Facilitador	Proveedor	Proceso
Los recursos están actualizados	PT	R	X	X	I-1
Fechas de apertura y cierre de tareas son correctas	PT	X	R	X	R-2
Se detallan los criterios de evaluación (rúbricas, ejemplos)	PT	R	X	X	R-3
El calificador no tiene demasiado anidamiento	PE	R	X	X	
Los alumnos están divididos en grupos	TE	X		R	I-6

Realización:

Consulta	Perspectivas	Diseñador	Facilitador	Proveedor	Proceso
El profesor responde en los foros dentro del límite de 48 horas lectivas desde que se plantea la duda	PT	X	R	X	R-2
Se ofrece retroalimentación de las tareas	PT	X	R	X	R-2
Las tareas están calificadas	PT	X	R	X	R-2
El calificador muestra cómo ponderan las diferentes tareas	PT	X	R	X	R-2

Evaluación:

Consulta	Perspectivas	Diseñador	Facilitador	Proveedor	Proceso
La mayoría de alumnos responden a los feedbacks	PTE	X	X	R	E-2
Se utilizan encuestas de opinión	PTE	X	X	R	E-2

Técnicas y herramientas

4.1. Técnicas

He dividido el trabajo en sprints de una duración de 14 días cada uno, teniendo reuniones con los tutores entre sprint y sprint. También he implementado integración continua y despliegue continuo en el proyecto. En el desarrollo del proyecto he priorizado el funcionamiento del software sobre la documentación extensiva y la respuesta al cambio sobre ceñirse a un plan. Todos estos elementos se pueden asociar a las metodologías ágiles, sin embargo, no opino que el desarrollo de este proyecto haya seguido una metodología específica.

4.2. Entorno de desarrollo integrado: IntelliJ IDEA

Para el desarrollo de la aplicación web en Java, he elegido IDEA en oposición a Eclipse. A pesar de que conozco Eclipse desde hace tres años y esta es la primera vez que utilizo IDEA, el año pasado utilicé Android Studio para desarrollar una aplicación Android, durante ese desarrollo me dí cuenta de que estaba basado en IDEA y que me sentía mucho más cómodo programando en Java en IDEs de JetBrains que en Eclipse. Facilita bastante la refactorización y escritura del código debido a funciones como el completado de código y las acciones rápidas, funcionalidades que también posee Eclipse, pero que en IDEA son más intuitivas y satisfactorias.

4.3. Patrón de diseño: Fachada

El patrón de diseño fachada consiste en crear una clase que haga de intermediario entre el cliente y uno o varios subsistemas de la aplicación con varios propósitos: Simplificar y centralizar el control, actuar como elemento de seguridad restringiendo el acceso a ciertas partes, y separar responsabilidades de los subsistemas. Un mismo sistema podría tener varias fachadas distintas que den un mismo servicio de distintas formas, por ejemplo, mi fachada se utiliza para generar parte del contenido de mi aplicación web, pero si quisiera trasladar la aplicación a una de escritorio solo tendría que crear una nueva fachada dejando los sistemas subyacentes intactos.

4.4. Hospedaje del repositorio: GitHub

Github es un sitio de hospedaje de repositorios de código gratis para repositorios públicos. GitHub Actions, integrado en GitHub, permite definir flujos de trabajo y lo he utilizado para implementar la integración continua.

4.5. Control de versiones: GitHub Desktop

GitHub Desktop es una aplicación que proporciona una interfaz gráfica para interactuar con tus repositorios en GitHub. Simplifica las acciones más comunes asociándoles botones, muestra todas las diferencias introducidas antes de realizar un commit, gestiona tus credenciales de forma automática, y permite alternar entre ramas y repositorios en dos clicks, con lo que se agiliza mucho el control de versiones.

4.6. Hospedaje de la aplicación: Heroku

Heroku es una plataforma de hospedaje y computación de aplicaciones web dinámicas. Tiene una versión para hospedar una aplicación no comercial y poder ejecutarla durante 500 horas al mes de forma gratuita.

4.7. Ejecución de tests: JUnit

JUnit es un framework para la realización de tests en Java. Es compatible con varios entornos de desarrollo integrados e incluso se puede usar mediante línea de comandos.

4.8. Construcción automática del proyecto: Maven

Maven es una herramienta para la construcción de proyectos software. Utiliza un archivo definido dentro del proyecto llamado "pom.xml" para definir la configuración necesaria para construir el proyecto como las dependencias o el formato al que compilar.

4.9. Generación de tablas: TablesGenerator.com

TablesGenerator.com es una página que permite crear una tabla con facilidad y puede convertirla a distintos formatos (LaTeX, HTML, texto simple...). La he utilizado para diseñar las tablas utilizadas en la aplicación y las tablas en memoria y anexos.

4.10. Paso de JSON a POJO: Json2CSharp.com

Json2CSharp.com es una página que permite la conversión de un objeto JSON a su POJO en Java correspondiente entre otras. La he utilizado en el proceso de implementación de las comprobaciones sobre los cursos Moodle debido a que las respuestas a mis llamadas REST tenían que ser deserializadas a objetos para poder manejarlas con facilidad.

4.11. Redacción en HTML: HtmlNotepad

HtmlNotepad es un editor HTML que facilita la redacción de textos en HTML. Contiene gran cantidad de atajos de teclado que agiliza mucho el formateo de texto. Lo he utilizado para la redacción de la página del manual de usuario.

4.12. Generación de diagramas UML: ArgoUML

ArgoUML es una herramienta de dibujo de diagramas UML. Lo he utilizado para todos los diagramas de los anexos.

4.13. Redacción de memoria y anexos: TeXstudio

TeXstudio es un editor de LaTeX. Contiene cantidad de herramientas necesarias para la creación de documentos en LaTeX y cuenta con un visor de PDF.

4.14. Herramientas para acceder a la información

- **Web scraping:** Es el conjunto de técnicas utilizadas para extraer y almacenar información de la web, un programa que hace esto de forma automática se llama web crawler. Algunas de estas técnicas nos permiten obtener información que no podríamos sacar con otras herramientas como por ejemplo si una página contiene enlaces fuera de Moodle a partir de su código HTML.
- **Web services:** Es un medio de comunicación entre dos aplicaciones en ordenadores distintos dentro de una red mediante el uso de distintos protocolos. Se puede pedir cierta información o acciones al servidor por medio de llamadas a funciones. En el caso de los web services que proporciona Moodle existe un servidor que hace de intermediario entre cliente y proveedor [7]. Por ejemplo, llamando a la función `core_grades_get_grades` podríamos obtener las notas de un alumno.
- **Logs:** Son registros de actividad que Moodle crea a partir de los eventos que realizan los usuarios, como publicar en un foro o empezar un cuestionario. Se pueden descargar en forma de archivo. Un ejemplo del uso que le daríamos sería obtener la actividad de un profesor para saber si responde a las dudas de los alumnos en los foros.

4.15. Framework de desarrollo web

Respecto a qué framework utilizar para el desarrollo de la aplicación web hemos barajado estas opciones:

- **Spring:** Es un framework para la creación de aplicaciones Java. Gestiona las dependencias entre objetos de forma automática lo que permite un bajo nivel de acoplamiento y ofrece un framework para desarrollo

de aplicaciones web. Su fuerte es la escalabilidad, pero es complicado aprender a usarlo.

- **Vaadin:** Es un framework de desarrollo web con una gran librería de componentes web. Usa GWT (Google Web Toolkit) para compilar Java a JavaScript y evitar al programador usar otros lenguajes aparte de Java. Su versión de pago ofrece una herramienta de edición gráfica que acelera el proceso de creación. Su fuerte es la velocidad de creación de prototipos, pero tiene mala escalabilidad.

De entre estas dos me decanto por Spring, ya que la escalabilidad es bastante mejor y es mucho más popular que Vaadin, lo que me ayudará a encontrar documentación y tutoriales a la hora de encontrar problemas durante el desarrollo.

Aspectos relevantes del desarrollo del proyecto

5.1. Ciclo de vida

La realización de este trabajo se ha llevado a cabo en sprints de una duración de 14 días con reuniones entre sprint y sprint, y con frecuencia han habido reuniones a mitad de estos.

El proyecto empezó sintetizando una lista de aspectos a comprobar en los cursos Moodle a partir del marco de referencia de calidad del e-learning de MOOQ [11] y un documento interno de UBUCEV proporcionado por los tutores. Después tuve que decidir si crear una aplicación de escritorio o una aplicación web y qué framework utilizar para desarrollar la aplicación. A partir de ahí, se hizo un prototipo para comprobar que era capaz de acceder a los web services de Moodle desde mi aplicación. Este prototipo solo mostraba la lista de cursos accedidos recientemente a partir de unas credenciales para la página de demostración de Moodle llamada Mount Orange School. Más tarde establecí el ciclo de integración continua/despliegue continuo y lancé la aplicación en Heroku. Una semana después, al final del sprint había creado una versión del informe específico que solo realizaba una comprobación sobre los cursos. A partir de ahí, debido a las fechas (segunda quincena de diciembre) no hubo ninguna reunión con los tutores hasta la vuelta de las vacaciones de navidad. Durante ese periodo, añadí el resto de las comprobaciones al informe y me dediqué a completar partes de la memoria.

5.2. Proceso de obtención de llamadas a los servicios web

Para obtener la información necesaria para las comprobaciones sobre los cursos de Moodle he tenido que realizar llamadas REST a distintas funciones de la API de servicios web de Moodle [8]. Sin embargo, la tabla que detalla la lista de funciones de la API omite los parámetros necesarios para las llamadas a las funciones. Para averiguar qué parámetros debía utilizar en cada función tuve que acceder al repositorio de Moodle [9] y encontrar las funciones que detallaban el funcionamiento de la función de la API en la que estaba interesado cada vez. El nombre de una función se puede dividir en dos partes: el nombre del componente que posee la función, y el nombre de la función. Las funciones están escritas en PHP y aparecen dentro de los archivos llamados `.externallib.php`. Para cada función existen tres funciones asociadas: `<NOMBRE DE LA FUNCIÓN>`, `<NOMBRE DE LA FUNCIÓN>_parameters`, y `<NOMBRE DE LA FUNCIÓN>_returns`. La función a secas define el comportamiento de la función, y en su declaración se puede ver qué parámetros son opcionales y cuáles no. La función con `"parameters"` al final describe los nombres y los tipos de datos de los parámetros que espera recibir la función. La función con `"returns"` al final describe los nombres y los tipos de datos de los atributos que devuelve la función. Por ejemplo, si quiero averiguar como llamar a la función `mod_forum_get_forums_by_courses` tengo acceder al fichero `/mod/forum/externallib.php` (supuesto por el nombre del componente que contiene la función, `"mod_forum"` en este caso) y buscar las funciones `get_forums_by_courses`, `get_forums_by_courses_parameters`, y `get_forums_by_courses_returns`.

Para probar el funcionamiento de las llamadas y al mismo tiempo obtener las clases que debía definir como POJO para manejar los datos recibidos como JSON (JSON (JavaScript Object Notation), es una forma de representar un objeto en formato de texto muy usado para transmitir información en aplicaciones web), hice lo siguiente: Uso la página de demostración de Moodle llamada Mount Orange School para hacer pruebas. Hago llamadas REST de forma manual con el navegador Chrome y obtengo el token con las credenciales de profesor (usuario:"teacher" contraseña:"moodle")

19

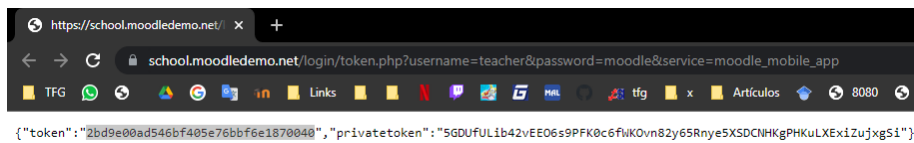


Figura 5.1: Obtención del token

. Después utilizo el token para la siguiente llamada y obtengo una respuesta en formato JSON.

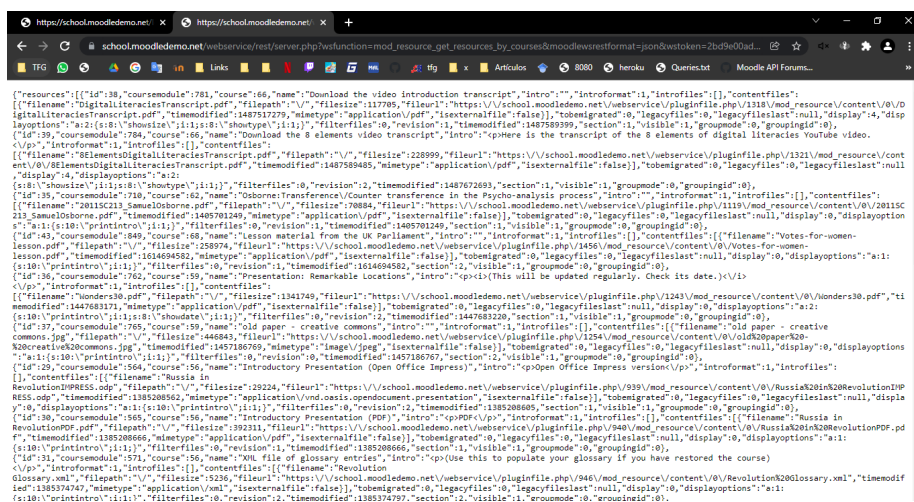


Figura 5.2: Obtención del JSON

Copio la respuesta en un conversor de JSON a POJO.

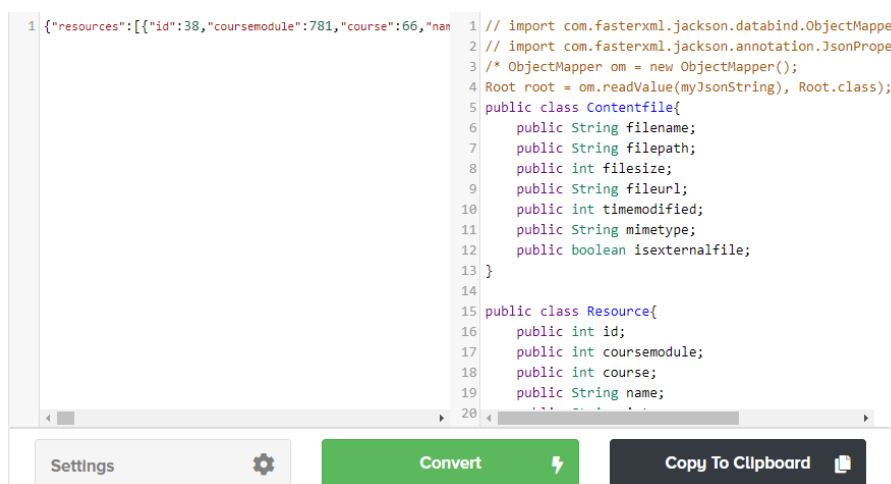


Figura 5.3: Uso del conversor

Creo las clases correspondientes, añado un constructor vacío y encapsulo los atributos.

5.3. Integración continua y despliegue continuo

La integración continua consiste en la automatización de la compilación y ejecución de pruebas cada vez que se suben cambios al repositorio. El despliegue continuo es la automatización del despliegue de un producto tras cada cambio en el repositorio. He implementado la integración continua del proyecto con GitHub Actions, primero, establecí en el archivo pom.xml que la compilación del proyecto fuera en formato WAR (Web Application Resource), luego, creé el archivo "maven.yml" en la carpeta de workflows para establecer que cada vez que se realice un push en la rama principal el proyecto se compile y se ejecuten los tests con Maven. He implementado el despliegue continuo en Heroku, la mayoría del proceso ha sido bastante intuitiva, ya que una de las opciones que ofrecía era GitHub como método de despliegue pero debido a que mi repositorio contiene por un lado la memoria y por otro el proyecto software, he tenido que añadir un buildpack (conjuntos de scripts de código abierto usados para compilar las aplicaciones en Heroku) que permite especificar una subcarpeta del repositorio para usarla como directorio raíz del proyecto software. También establecí en las opciones que el despliegue automático espere a que se supere la integración continua.

Trabajos relacionados

Este apartado resume los estudios y proyectos realizados por terceros que están relacionados con la calidad del e-learning, con la medición de esta, y con sus consecuencias. También se habla de las novedades que trae este proyecto a este ámbito.

Automated e-learning quality evaluation

Un artículo presentado por Rositsa Doneva y Silvia Gaftandzhieva en la conferencia internacional del e-learning celebrada en Berlin en septiembre de 2015. En este artículo se llevan a cabo dos experimentos para probar la factibilidad de implementar un sistema automático de evaluación de la calidad en Moodle. El primer experimento consiste en integrar UBIS-Jaspersoft, un sistema de business intelligence para universidades, con la base de datos de Moodle para analizar los resultados de las respuestas de los alumnos en un modulo de feedback de Moodle. El segundo experimento consiste en la creación de cuatro servicios web para ser utilizados en la evaluación de distintos indicadores de la calidad [1].

Perceived Service Quality and Student Loyalty in an Online University

Un artículo presentado por María-Jesús Martínez-Argüelles y Josep-Maria Batalla-Busquets en la revista IRRODL en 2016. Este artículo estudia la relación entre todos los aspectos de la enseñanza (incluida la interfaz de usuario en el e-learning) y la percepción de calidad del servicio por parte del estudiante, y entre esta última y la lealtad y la disposición a la

recomendación por parte de este. El estudio llega a la conclusión de que hay una relación directa y no solo indirecta entre estas variables [5].

Dashboard for Evaluating the Quality of Open Learning Courses

Un artículo presentado por Gina Mejía-Madrid, Faraón Llorens-Largo, y Rafael Molina-Carmona en la revista Sustainability en 2020. Este artículo presenta un modelo para la evaluación de la calidad de los cursos de Open Learning y un dashboard creado a partir de resultados de encuestas y entrevistas. La palabra "dashboard" se podría traducir como el panel de instrumentos de un coche u otro vehículo, la analogía viene de qué el tipo de dashboard al que nos estamos refiriendo es un conjunto de tablas y gráficos fáciles de leer que permiten a aquel que lo mira entender la información de forma rápida y tomar decisiones. En el artículo también se habla de forma muy breve de automatizar la obtención de datos para generar el dashboard pero sin llegar a definir si pretenden automatizar las encuestas o obtener los datos de forma automática por otros medios [6].

A Hierarchical Model to Evaluate the Quality of Web-Based E-Learning Systems

Un artículo presentado por Muhammad Abdul Hafeez y otros seis autores en la revista Sustainability en 2020. Este artículo presenta un modelo para definir la calidad de los sistemas de e-learning generado a partir de una serie de encuestas que les permitieron identificar los factores clave para la calidad según su importancia. El resultado es un modelo con forma de árbol en el que los nodos hoja son los aspectos a evaluar y que se encuentran ordenados por su relevancia dentro de su nodo padre [10].

Data Analysis for Evaluation on Course Design and Improvement of “Cyberethics” Moodle Online Courses

Un artículo presentado por Motonori Nakamura y Hiroshi Ueda en la revista Procedia Computer Science en 2017. En este artículo se crea un sistema de recolección de datos con el objetivo de analizar los resultados en la recepción por parte de los estudiantes en un curso Moodle de ciberética de Japón para poder ver los efectos de los cambios realizados en el diseño del curso a lo largo de los años. El artículo concluye que los cambios conllevaron resultados concretos [12].

Moodle Course Checker Plugin

Un plugin para Moodle que permite realizar una serie de comprobaciones mediante un conjunto de comprobadores independientes que se pueden ejecutar de forma individual [3].

Course Checks Block Plugin

Otro plugin para Moodle que permite realizar una serie de comprobaciones automáticas [4].

¿En qué se diferencia mi proyecto?

Este proyecto adapta un marco de calidad de cursos de e-learning a una serie de comprobaciones sobre cursos Moodle y es capaz de generar informes que además de mostrar los resultados de dichas comprobaciones muestra la traslación de estos a las responsabilidades de cada fase, rol, y perspectiva descritos en el marco y señala qué elementos se pueden cambiar para mejorar los resultados.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Rositsa Doneva and Silvia Gaftandzhieva. Automated e-learning quality evaluation. INTERNATIONAL CONFERENCE ON E-LEARNING, 2015.
- [2] Martin Dougiamas and Peter C Taylor. Interpretive analysis of an internet-based course constructed using a new courseware tool called moodle. In *2nd conference of herdsa (the higher education research and development society of australasia)*, pages 7–10, 2002.
- [3] FFHS, Adrian Perez Rodriguez, Christoph Karlen, and Sascha Vogel. Moodle Course Checker Plugin, 5 2021.
- [4] Mike Grant, Jez h, Manoj Solanski, and Kevin Moore. Course Checks Block Plugin, 3 2018.
- [5] María-Jesús Martínez-Argüelles and Josep-Maria Batalla-Busquets. Perceived service quality and student loyalty in an online university. *International Review of Research in Open and Distributed Learning*, 17(4):264–279, 2016.
- [6] Gina Mejía-Madrid, Faraón Llorens-Largo, and Rafael Molina-Carmona. Dashboard for evaluating the quality of open learning courses. *Sustainability*, 12(9):3941, 2020.
- [7] Moodle. Web services - MoodleDocs, 10 2020.
- [8] Moodle. Web service API functions, 6 2021.
- [9] Moodle. Moodle repository on GitHub, 1 2022.

- [10] Abdul Hafeez Muhammad, Ansar Siddique, Ahmed E Youssef, Kashif Saleem, Basit Shahzad, Adnan Akram, and Al-Batool Saleh Al-Thnian. A hierarchical model to evaluate the quality of web-based e-learning systems. *Sustainability*, 12(10):4071, 2020.
- [11] Christian M Stracke, Esther Tan, António Moreira Texeira, Maria do Carmo Pinto, Bill Vassiliadis, Achilles Kameas, Cleo Sgouropoulou, and Gérard Vidal. Quality reference framework (qrf) for the quality of massive open online courses (moocs): Developed by mooq in close collaboration with all interested parties worldwide. 2018.
- [12] Hiroshi Ueda and Motonori Nakamura. Data analysis for evaluation on course design and improvement of “cyberethics” moodle online courses. *Procedia Computer Science*, 112:2345–2353, 2017.