



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

eLearningQA - Control de  
calidad



Presentado por Roberto Arasti Blanco  
en Universidad de Burgos — 9 de junio  
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 *eLearningQA - Control de calidad*.

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 junio de 2022

Vº. Bº. del Tutor:

Vº. Bº. del tutor:

D. Raúl Marticorena Sánchez

D. Carlos López Nozal





## Resumen

El proceso de enseñanza y aprendizaje en línea, conocido como e-learning, se ha implantado de manera acelerada en muchas instituciones académicas, debido en gran medida por la situación de confinamiento provocada por la pandemia COVID-19. Las instituciones educativas tradicionales se han visto obligadas a impartir sus cursos en plataformas virtuales de aprendizaje, como Moodle o similares. 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 de un LMS. Además se desarrolla una aplicación software que dado un curso o cursos genere una evaluación de calidad por medio de servicios web y otras herramientas si es necesario.

## Descriptores

Aplicación web, framework de calidad en e-learning, revisiones automáticas, calidad en cursos en línea, Learning Management System (LMS).

### **Abstract**

The online teaching and learning process, known as e-learning, has been implanted in a hurrisome way in many academic institutions, due in great measure to the situation of confinement triggered by the COVID-19 pandemic. Traditional educational institutions have been forced to impart their courses on virtual learning platforms, such as Moodle o similar. 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 an LMS. In addition, a software application is developed that, given a course or courses, generates a quality evaluation via web services and other tools if needed.

### **Keywords**

Web application, e-learning quality framework, automatic revisions, online course quality, Learning Management System (LMS).



---

# Índice general

---

<b>Índice general</b>	<b>iii</b>
<b>Índice de figuras</b>	<b>v</b>
<b>Índice de tablas</b>	<b>vi</b>
<b>Introducción</b>	<b>1</b>
<b>Objetivos del proyecto</b>	<b>3</b>
<b>Conceptos teóricos</b>	<b>5</b>
3.1. Definiciones básicas . . . . .	5
3.2. Marco de referencia de calidad de MOOQ . . . . .	6
3.3. Buenas prácticas de la docencia online . . . . .	8
3.4. Plan de calidad para cursos e-learning . . . . .	10
<b>Técnicas y herramientas</b>	<b>15</b>
4.1. Desarrollo ágil . . . . .	15
4.2. Herramientas de desarrollo . . . . .	19
4.3. Herramientas de documentación . . . . .	20
4.4. Patrón de diseño: Fachada . . . . .	21
4.5. Herramientas para acceder a la información . . . . .	21
4.6. Framework de desarrollo web . . . . .	22
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>25</b>
5.1. Ciclo de vida . . . . .	25
5.2. Proceso de obtención de llamadas a los servicios web . . . . .	26
5.3. Integración continua y despliegue continuo . . . . .	28

5.4. Fallo de seguridad Heroku/TravisCI/Github . . . . .	29
5.5. Uso del nombre Moodle . . . . .	29
5.6. Informes: pestañas vs breadcrumbs . . . . .	30
5.7. SonarCloud . . . . .	30
5.8. Diseño del logo de la aplicación . . . . .	31
5.9. Almacenamiento de registros de informes . . . . .	32
5.10. Redefinición de regla de comentarios del profesor . . . . .	32
<b>Trabajos relacionados</b>	<b>33</b>
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>37</b>
7.1. Conclusiones . . . . .	37
7.2. Líneas de trabajo futuras . . . . .	39
<b>Bibliografía</b>	<b>41</b>

---

## Índice de figuras

---

3.1. Ciclo de fases según el marco de MOOQ [20] . . . . .	7
4.1. Desarrollo incremental por sprints . . . . .	16
4.2. Versiones publicadas de la publicación . . . . .	17
4.3. Diagrama de secuencia generado mediante ArgoUML . . . . .	21
4.4. Comparativa en popularidad de Spring y Vaadin . . . . .	23
5.1. Obtención del token . . . . .	27
5.2. Obtención del JSON . . . . .	27
5.3. Uso del conversor . . . . .	28
5.4. Quality gate usada en el proyecto . . . . .	29
5.5. Gráfico del número de problemas a lo largo del tiempo . . . . .	31
5.6. Logo de la aplicación . . . . .	32

---

# Índice de tablas

---

3.1.	Consultas	de	diseño	
	Leyenda:			
	Responsabilidad:	R=Responsable,X=Involucrado		
	Perspectivas:	P=Pedagógica, T=Tecnológica, E=Estratégica	. .	11
3.2.	Consultas	de	implementación	
	Leyenda:			
	Responsabilidad:	R=Responsable,X=Involucrado		
	Perspectivas:	P=Pedagógica, T=Tecnológica, E=Estratégica	. .	12
3.3.	Consultas	de	realización	
	Leyenda:			
	Responsabilidad:	R=Responsable,X=Involucrado		
	Perspectivas:	P=Pedagógica, T=Tecnológica, E=Estratégica	. .	13
3.4.	Consultas	de	evaluación	
	Leyenda:			
	Responsabilidad:	R=Responsable,X=Involucrado		
	Perspectivas:	P=Pedagógica, T=Tecnológica, E=Estratégica	. .	13

---

# Introducción

---

El proceso de enseñanza y aprendizaje en línea, también conocido como 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 [18].

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 [11]. 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 [18]. 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 [4] y otros que analizan los datos de forma menos automática para aplicar mejoras en el e-learning a posteriori [21]. 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 [12]. 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 plataformas de e-learning. Como punto de partida se pretende integrar con Moodle (no estoy asociado a esta plataforma), que de hecho es utilizada por la Universidad de Burgos. A partir de la dirección de un curso Moodle se muestran los resultados de la revisión automática de calidad. El informe de evaluación se desglosa en indicadores cuantitativos de cobertura e indicadores

cualitativos que ayudan a identificar los problemas encontrados en el diseño, implementación y realización del curso en línea. Con esta información los docentes responsables del curso en línea podrán gestionar un proceso de mejora continua de la calidad del proceso de enseñanza y aprendizaje en sus cursos en línea.

---

## Objetivos del proyecto

---

El objetivo principal de este trabajo es desarrollar una aplicación web que permita al profesor evaluar las distintas fases de diseño instruccional de un curso de Moodle (diseño, implementación, realización, evaluación), tal como recomiendan algunos frameworks internacionales de calidad en e-learning [20].

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

1. Definir las entidades y atributos de nuestro modelo de calidad a partir el modelo de datos de los cursos en Moodle: Usuarios, Roles, Cursos, Grupos, Actividades, Tareas, Recursos, Surveys, Feedbacks, Foros, Debates, y Comentarios.
2. Conocer los servicios Web proporcionados por Moodle para interactuar con las entidades y atributos de nuestro modelo de calidad.
3. Definir un plan de calidad para aplicar a las entidades del modelo de cursos de Moodle a partir de frameworks internacionales de calidad en e-learning y recomendaciones del Centro de Enseñanza Virtual de la Universidad de Burgos (UBUCEV).
4. Definir evaluaciones del plan de calidad personalizadas para distintos tipos de cursos: docencia reglada, tfg y comunidades y grupos.
5. Diseñar indicadores cualitativos y cuantitativos de calidad de cada fase de diseño instruccional del curso en línea (diseño, implementación, realización y evaluación)





---

## Conceptos teóricos

---

### 3.1. Definiciones básicas

- **Administración de la calidad:** es el conjunto de actividades para definir un proceso de mejora continua: aseguramiento de calidad, planeación de calidad y control de calidad [3] [19].
- **Aseguramiento de calidad:** establecimiento de un marco de trabajo de procedimientos y estándares que conducen a curso de alta calidad
- **Planeación de calidad:** selección de procedimientos y estándares del marco de trabajo y adaptación para un curso concreto software específico
- **Control de calidad:** los procedimientos y estándares que son seguidos por el equipo de desarrollo.
- **Medición:** el proceso por el cuál se asignan números o símbolos a los atributos de las entidades del curso en línea, de tal forma que los caracteriza de manera clara a través de reglas o consultas sobre LMS.
- **E-learning:** es el proceso de enseñanza y aprendizaje impartido por medios electrónicos como internet, plataformas virtuales, medios audiovisuales, etc.
- **LMS:** Learning Management Systems, o sistemas de gestión del aprendizaje son plataformas online que permiten gestionar el aprendizaje de grupos de personas en un entorno virtual. El aprendizaje en este tipo de plataformas no se limita al campo académico, sino que también se

pueden usar los LMS dentro de empresas para instruir a los empleados. Entre los más populares se encuentran Moodle, Edmodo y Blackboard.

- **Moodle:** es una plataforma de aprendizaje LMS 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 [5].

### 3.2. Marco de referencia de calidad de MOOQ

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.

Ahora pasamos a explicar las tres dimensiones del marco de referencia de calidad en e-learning de MOOQ [20]. 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.

#### 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: diseños asistidos por el profesor basados en comunicaciones síncronas o asíncronas, diseños sin supervisión del profesor basados en recursos y actividades de aprendizajes, o mixtos.
- **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.

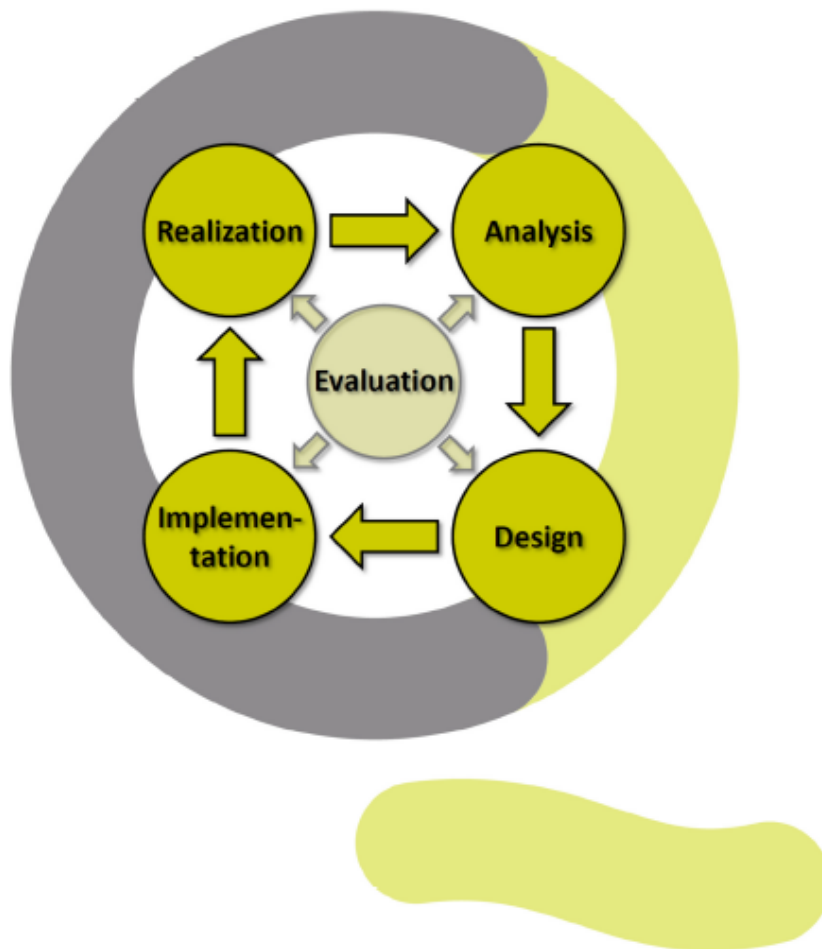


Figura 3.1: Ciclo de fases según el marco de MOOQ [20]

## 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. Buenas prácticas de la docencia online

A continuación presentamos una serie de buenas prácticas que la aplicación pretende fomentar. También nos basamos en la experiencia que se ha tenido como alumno en la plataforma UBUVirtual. Estas cuestiones no llegan a cubrir por completo lo que sería una especie de guía para gestionar cursos virtuales, es un complemento al framework de calidad.

### Retroalimentación a tiempo

Es importante que el docente responda a las dudas a tiempo, y también que el alumno conozca sus resultados en un tiempo razonable, y esto es por dos razones: la primera, que el alumno no pierda la motivación si tiene

buenos resultados, y la segunda, que se esfuerce más si tiene resultados mejorables.

### **Tener en cuenta la opinión de los alumnos**

La evaluación es una parte fundamental del ciclo del diseño instruccional. Las encuestas permiten encontrar los puntos fuertes (para no alterarlos) y los débiles (para solventarlos).

### **Revisar los cuestionarios**

Es muy útil tener en cuenta las estadísticas de los cuestionarios, aunque, son necesarios suficientes intentos por parte de los estudiantes para tener una validez razonable. Una de esas estadísticas es el índice de facilidad, que indica el porcentaje medio de la puntuación que obtienen los alumnos de una pregunta, indicando su dificultad. Otra estadística interesante es la calificación aleatoria estimada, que expresa el resultado medio de responder aleatoriamente una pregunta de tipo test; en un caso ideal este valor sería de 0%, si no, la puntuación de esa pregunta tendría un componente aleatorio. Por último, la eficiencia discriminativa[14] indica como de buena es una pregunta discriminando entre alumnos que saquen buena nota en el cuestionario y aquellos que saquen peor nota. Hay que tener en cuenta estos valores para conseguir cuestionarios más justos y efectivos.

También es importante que las preguntas tengan una retroalimentación asignada para que los alumnos aprendan del resultado en la revisión.

Es recomendable ajustar el tiempo que se permite para la realización de los cuestionarios a partir del tiempo que tomó realizarlo en anteriores ocasiones para dificultar el fraude y tampoco quedarse corto y no dejar a los alumnos terminar.

### **Trabajo en equipo**

Obtener las competencias necesarias para el trabajo en equipo es fundamental para el alumno, sobretodo en la universidad, que pretende preparar a sus estudiantes para un entorno laboral. Es muy recomendable que un curso contenga actividades con entrega por grupos.

## Claridad y corrección

La corrección y la claridad de la información en un curso ayuda a evitar que el alumno se pierda en las formas y entienda mejor lo que se pide de él. El calificador debería mostrar de forma concisa el peso de cada procedimiento, no tener demasiadas categorías dentro de otras y si las notas máximas de cada actividad coinciden, leer los resultados se vuelve mucho más sencillo.

## Apuntes actualizados

Aparte de que viene bien revisar los materiales proporcionados a los alumnos en busca de errores y para añadir novedades, el no hacerlo en mucho tiempo da al alumno una imagen de abandono por parte del profesor, y en mi experiencia como alumno he llegado a vivir esa situación.

## 3.4. Plan de calidad para cursos e-learning

En secciones previas hemos presentado un marco de calidad genérico junto con recomendaciones del Centro de Enseñanza Virtual de la Universidad de Burgos junto a la identificación de unas buenas prácticas de enseñanza en cursos en línea.

A partir de esta documentación en esta sección definiremos de manera concreta unos indicadores cualitativos y cuantitativos de calidad basados en consultas a entidades de cursos en línea.

Para facilitar la comprensión del conjunto de consultas de calidad se agrupan en las fases del marco de referencia de calidad de MOOC [20]. La Tabla 3.1 muestra el conjunto de consultas de fase de diseño, la Tabla 3.2 las de la fase de implementación, la Tabla 3.3 las de la fase de realización y la Tabla 3.4 las de la fase de evaluación. En este apartado definiremos qué información intentamos conseguir de cada fase (nos vamos a centrar en las fases de diseño, implementación, realización, y evaluación), rol y perspectiva mediante distintos métodos. La mayoría de consultas provienen de la lista de comprobación del Centro de Enseñanza Virtual de la Universidad de Burgos (UBUCEV) de asignaturas virtuales. Los procesos indicados en la última columna de las tablas se refieren a los procesos que se indican a partir de la página 10 del documento del marco de referencia de calidad de MOOC [20].

Tabla 3.1: Consultas de diseño

Leyenda:

Responsabilidad: R=Responsable, X=Involucrado

Perspectivas: P=Pedagógica, T=Tecnológica, E=Estratégica

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	
Las preguntas de los cuestionarios tienen retroalimentación	P	R	X	X	
Las preguntas de opción multiple puntúan con una calificación aleatoria estimada de cero	P	R	X	X	

Tabla 3.2: Consultas de implementación

Leyenda:

Responsabilidad: R=Responsable, X=Involucrado

Perspectivas: P=Pedagógica, T=Tecnológica, E=Estratégica

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



Tabla 3.3: Consultas de realización

Leyenda:

Responsabilidad: R=Responsable, X=Involucrado

Perspectivas: P=Pedagógica, T=Tecnológica, E=Estratégica

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
El tiempo de los cuestionarios está bien ajustado	P	R	X	X	
Los cuestionarios tienen una dificultad estimada entre unos valores umbrales	P	R	X	X	
Las preguntas de los cuestionarios son buenas discriminando <sup>[14]</sup>	P	R	X	X	

Tabla 3.4: Consultas de evaluación

Leyenda:

Responsabilidad: R=Responsable, X=Involucrado

Perspectivas: P=Pedagógica, T=Tecnológica, E=Estratégica

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. Desarrollo ágil

En este proyecto se ha llevado a cabo un desarrollo ágil, llevando a cabo diversos procesos de este como el desarrollo iterativo o la implementación continua.

#### Desarrollo iterativo

El desarrollo consiste en la revisión cíclica sobre un mismo trabajo. He dividido el trabajo en sprints de una duración de 14 días cada uno, teniendo reuniones con los tutores entre sprint y sprint, esto lo vuelve iterativo respecto al tiempo, pero también ha habido que hacer revisiones a ciertos aspectos de la aplicación como la lista de aspectos a mejorar.

#### Desarrollo incremental

El desarrollo incremental consiste en crear un prototipo funcional e ir añadiendo nuevas funcionalidades tras recibir la retroalimentación del cliente. Está fuertemente relacionado con el desarrollo iterativo y es difícil separar los dos conceptos. En bastantes de los sprints se hacen “incrementos verticales” que son mejoras que puede percibir el usuario en la aplicación.

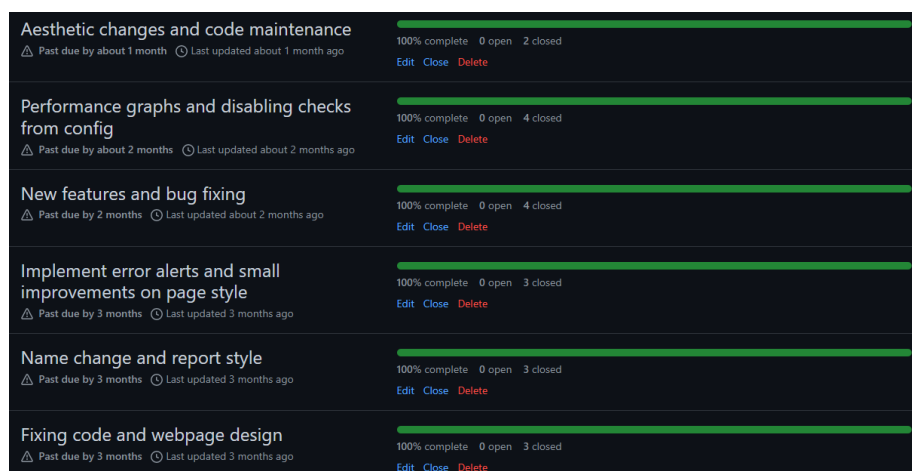


Figura 4.1: Desarrollo incremental por sprints

Los sprints se pueden ver en <https://github.com/RobertoArastiBlanco/eLearningQA/milestones>.

## Control de versiones

El control de versiones es la utilización de herramientas que registran las modificaciones en el código de un proyecto, permitiendo acceder a las diferentes versiones. Estas herramientas permiten el trabajo concurrente de varios desarrolladores y permiten la aplicación de más procesos ágiles como la implementación continua.

### 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.

### 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.

## Publicaciones frecuentes

Una publicación consiste en dejar el producto a disposición de los usuarios para recibir retroalimentación ya sean críticas o halagos. Durante el desarrollo del proyecto he publicado cuatro versiones diferentes de la aplicación, desde la v0.1 hasta la v0.4.

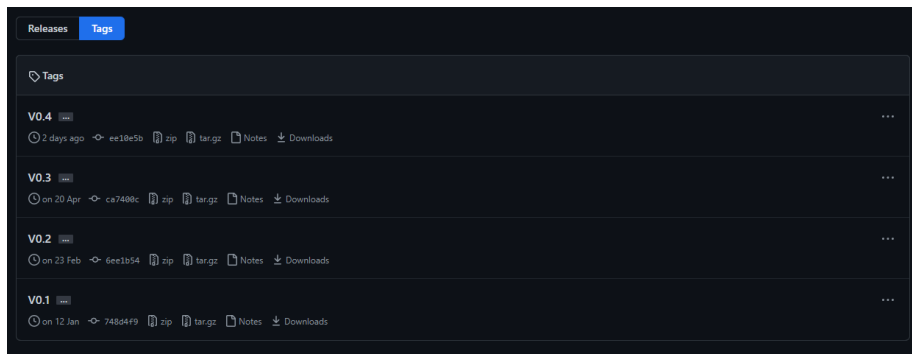


Figura 4.2: Versiones publicadas de la publicación

Las versiones se pueden ver en <https://github.com/RobertoArastiBlanco/eLearningQA/tags>.

## Refactorización

Refactorizar consiste en mejorar el código en su estructura sin alterar el comportamiento externo. He realizado dos grandes refactorizaciones de código al proyecto durante el desarrollo, la primera fue a mediados de enero para refactorizar los test y que utilicen archivos JSON, la segunda fue a principios de febrero con el objetivo de arreglar la mayoría de errores descubiertos gracias a la implantación de SonarCloud.

## Uso de test unitarios

Un test unitario es un fragmento corto de código que ejecuta parte del código fuente de un programa y comprueba el resultado devuelto para considerar si el comportamiento del programa coincide con lo esperado. He utilizado JUnit para la ejecución de los test.

## 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.

## Construcción automática

La construcción automática es el uso de herramientas como Maven o Gradle para automatizar procesos como la compilación, la ejecución de tests y el empaquetado del software.

## 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.

## Integración continua

La integración continua consiste en el componente de automatización en el repositorio de código y tiene como prerequisite la construcción automática. El proyecto contiene un archivo yml que especifica las acciones a seguir para la construcción automática del proyecto y el análisis de calidad de SonarCloud tras cada *push* al repositorio.

## Despliegue continuo

El despliegue continuo consiste en automatizar el despliegue de la aplicación aprovechando la infraestructura de integración continua ya establecida en el repositorio. Los cambios se producen en la aplicación en vivo. He desplegado mi aplicación usando Heroku. El despliegue de la aplicación se encuentra en <https://elearningqa.herokuapp.com>.

## 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.

## Herramienta de calidad de código: SonarCloud

SonarCloud es un servicio en la nube de análisis de código que detecta code smells, bugs, y vulnerabilidades de seguridad. Lo tengo integrado en el ciclo de integración continua. Permite definir un “Quality gate” para que la integración continua falle en caso de no cumplirse alguna de las condiciones definidas sobre la calidad del código. También, al ser SonarCloud una herramienta de control de calidad, ha servido de inspiración al implementar la lista de aspectos a mejorar incluida en los informes que genera la aplicación. Se puede ver la evolución de la calidad del código en la figura 5.5 y en [https://sonarcloud.io/project/activity?category=QUALITY\\_GATE&id=RobertoArastiBlanco\\_MoodleQA](https://sonarcloud.io/project/activity?category=QUALITY_GATE&id=RobertoArastiBlanco_MoodleQA).

## 4.2. Herramientas de desarrollo

### 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.

### 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 las llamadas REST tenían que ser deserializadas a objetos para poder manejarlas con facilidad.

### 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 formato de texto. Lo he utilizado para la redacción de la página del manual de usuario.

## **Framework CSS: Bootstrap**

Bootstrap es un conjunto de librerías de CSS y javascript de código abierto creado por empleados de Twitter. Ofrece cantidad de recursos que facilitan la disposición de elementos en pantalla y contiene elementos como acordeones y carruseles ya implementados.

## **Librería de generación de gráficos: Plotly**

Plotly es una librería de generación de gráficos para el lenguaje de programación Python que también tiene una versión para Javascript. La he elegido por estar recomendada por W3Schools, una página web de tutoriales de desarrollo web creada en 1998, y por ser capaz de dibujar líneas independientes entre sí al usar pares de coordenadas  $x$  e  $y$  y no tablas de valores.

## **4.3. Herramientas de documentación**

### **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.

### **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.

### **Generación de diagramas UML: ArgoUML**

ArgoUML es una herramienta de dibujo de diagramas UML. Lo he utilizado para todos los diagramas de los anexos. La he utilizado para realizar diagramas de clases, de secuencia, y de casos de uso. Aceleré bastante el tiempo de modelado pero es bastante ortopédico en el sentido que hay ligeros cambios que no permite realizar.



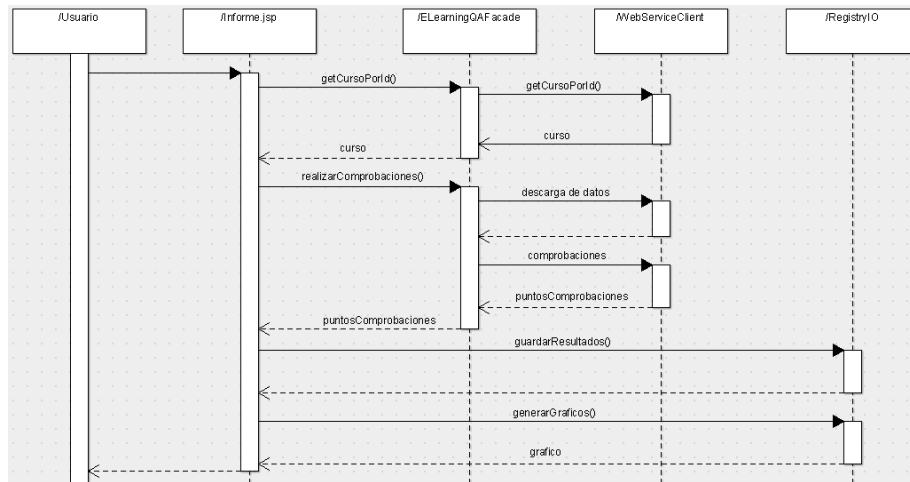


Figura 4.3: Diagrama de secuencia generado mediante ArgoUML

## 4.4. 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, en este caso, la fachada se utiliza para generar parte del contenido de la aplicación web, pero si quisiera trasladar la aplicación a una de escritorio solo se tendría que crear una nueva fachada dejando los sistemas subyacentes intactos [7]. Para más información sobre el uso concreto que se le da a este patrón de diseño, consultar el apartado C.4 de los anexos.

## 4.5. Herramientas para acceder a la información

A continuación se describen los medios que se han barajado para la extracción de la información de los cursos que queremos analizar con nuestra aplicación. Finalmente solo se han utilizado servicios web, sin embargo, es interesante ponderar sus ventajas.

- **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 externos 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 [13]. 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.

## 4.6. 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.

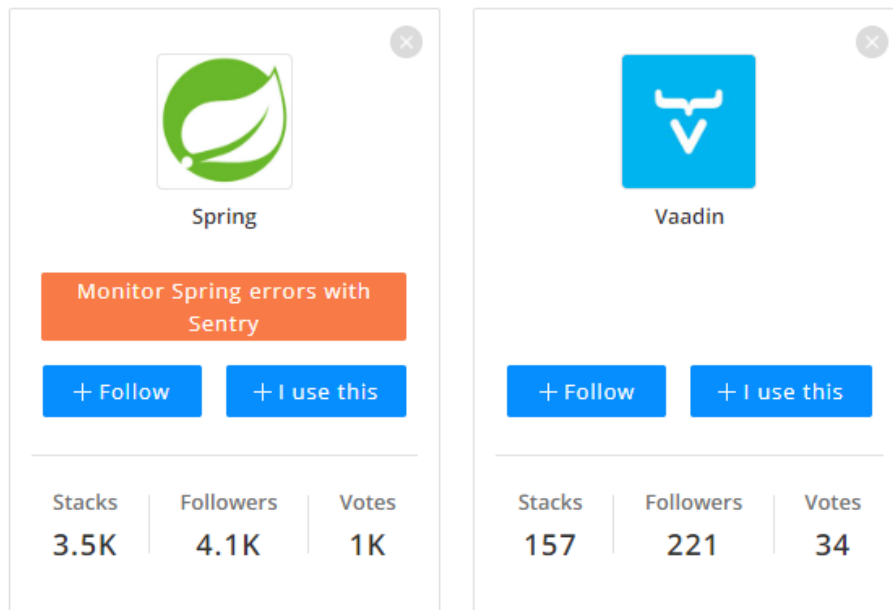


Figura 4.4: Comparativa en popularidad de Spring y Vaadin

De entre estas dos nos decantamos por Spring, ya que la escalabilidad es bastante mejor y es mucho más popular que Vaadin, lo que nos 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 [20] 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. Decidimos retrasar la entrega del trabajo al segundo semestre.

El siguiente paso fue añadir *SonarCloud* para analizar el código, y arreglar la gran mayoría de los errores encontrados. Se añadió *Bootstrap* para mejorar bastante el estilo de la página. Se añadieron varias mejoras de características no funcionales a la aplicación: un logo, configuraciones intercambiables, alertas de las causas de los fallos de las comprobaciones, página de error y gráficos de evolución de la calidad de los cursos. El resto del desarrollo se dedicó a solucionar errores pequeños.

## 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 [15]. 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 [16] 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
- <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

## 27

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 (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”).

[illegible]

Figura 5.2: Obtención del JSON

Copio la respuesta en un conversor de JSON a POJO.

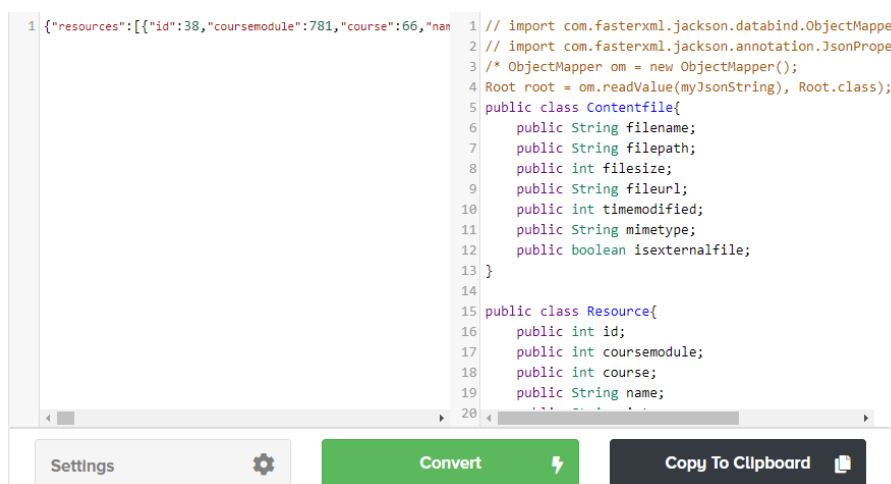


Figura 5.3: Uso del conversor

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

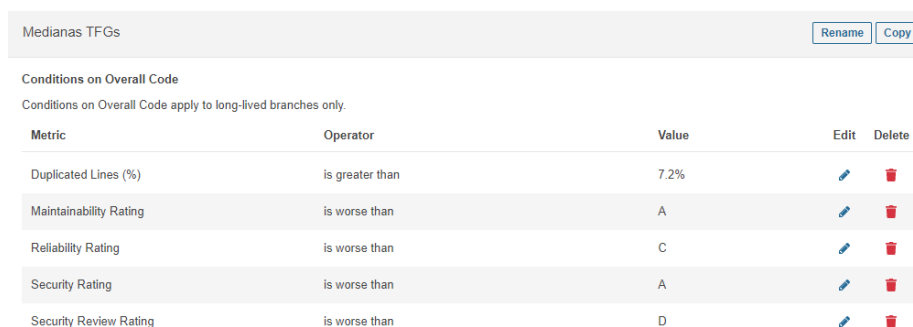
### 5.3. Integración continua y despliegue continuo

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. También, tras definir una quality gate en SonarCloud, ha pasado a formar parte de la integración continua, y, en caso de no cumplirse los estándares de la quality gate (figura 5.4), la integración continua cuenta como fallida.



The screenshot shows the 'Medianas TFGs' configuration page in SonarCloud. It includes a table of quality gate conditions. The table has columns for Metric, Operator, Value, Edit, and Delete. The conditions are: Duplicated Lines (%) is greater than 7.2%, Maintainability Rating is worse than A, Reliability Rating is worse than C, Security Rating is worse than A, and Security Review Rating is worse than D. Each row has an edit icon (pencil) and a delete icon (trash).

Metric	Operator	Value	Edit	Delete
Duplicated Lines (%)	is greater than	7.2%		
Maintainability Rating	is worse than	A		
Reliability Rating	is worse than	C		
Security Rating	is worse than	A		
Security Review Rating	is worse than	D		

Figura 5.4: Quality gate usada en el proyecto

## 5.4. Fallo de seguridad Heroku/TravisCI/Github

El 9 de abril de 2022 unos repositorios privados de Github fueron descargados de forma indebida debido a un ataque por medio de Heroku y TravisCI. El atacante robó tokens de autorización de Github almacenados por Heroku y TravisCI el día 7 y los utilizó para buscar repositorios basándose en las organizaciones a las que pertenecían el día 8 para acabar descargándolos el día 9. Github descubrió esto el 12 de abril y avisó a Heroku el día siguiente [8]. Tras tres días de investigación, Heroku revocó los tokens el 16 de abril, desabilitando el despliegue automático en el proceso y afirmó no reestablecerlo hasta cerciorarse de que hacerlo sea seguro. Heroku reestableció el servicio el 25 de mayo [10].

## 5.5. Uso del nombre Moodle

Como el objetivo inicial del proyecto es integrar Moodle en la aplicación, el nombre de tanto el proyecto como la aplicación iba a ser “MoodleQA”, pero en medio del desarrollo los tutores y yo descubrimos que al ser “Moodle” una marca registrada [17], existe cierta cantidad de restricciones sobre el uso de esta palabra. Las restricciones que nos conciernen son no poder utilizar “Moodle” en el nombre de tu software ni en el nombre de dominio ni

usarlo para describir tu aplicación de forma que los usuarios crean que estás asociado con Moodle si no lo estás. Por ello he tenido que revisar todas las menciones que hago a Moodle tanto en la memoria y anexos como en la aplicación.

## 5.6. Informes: pestañas vs breadcrumbs

A la hora de mostrar los informes se presenta la disyuntiva de si hacerlo en una sola pestaña de navegador y permitir al usuario navegar entre los distintos cursos con ayuda de una miga de pan o breadcrumb, o por el contrario generar cada informe en una nueva pestaña. Me he decantado por generar una pestaña por informe por dos razones: el rendimiento y la comparación de informes. Cada vez que se genera un informe se descarga la información necesaria para las consultas y se realizan las consultas sobre dicha información, para un servidor puede que la carga no sea mucha y se puedan generar los informes al instante, pero ejecutando la aplicación en local desde mi ordenador portátil he llegado a tener esperas de alrededor de diez segundos por un solo informe. A la hora de comparar informes es más ágil el cambiar de pestaña que retroceder a la página anterior e ir al otro informe.

## 5.7. SonarCloud

En esta sección agrupo y expongo dos sucesos interesantes ocurridos relacionados con *SonarCloud* tras la implantación de esta herramienta al repositorio.

### Definición del quality gate de SonarCloud

En SonarCloud la quality gate es una serie de condiciones sobre la calidad del código, por ejemplo, que la puntuación de seguridad no esté por debajo de “A”. Existe una quality gate predefinida llamada Sonar way, pero está definida para proyectos reales, así que en mi caso he decidido definir mi propia quality gate a partir de otros proyectos en Sonarcloud de la misma índole. Realicé una búsqueda y encontré 25 TFGs de la UBU, y calculé las medianas de dichos trabajos para obtener las condiciones de mi quality gate.

## SonarCloud en el ciclo de desarrollo

Desde que añadí *SonarCloud* al ciclo de integración y despliegue continuo he creado el hábito de comprobar los bugs, smells, y vulnerabilidades generados tras cada commit, esto evita que la cantidad de problemas se acumule y a su vez hace que el código sea más fácil de mantener. Además, el hecho de entender los code smells conforme los solucionaba me ha enseñado a tenerlos con menor frecuencia. A continuación se muestra un gráfico generado por SonarCloud del número de problemas a lo largo del tiempo en el que se aprecia que tras solucionar los problemas iniciales el gráfico se mantiene relativamente plano.

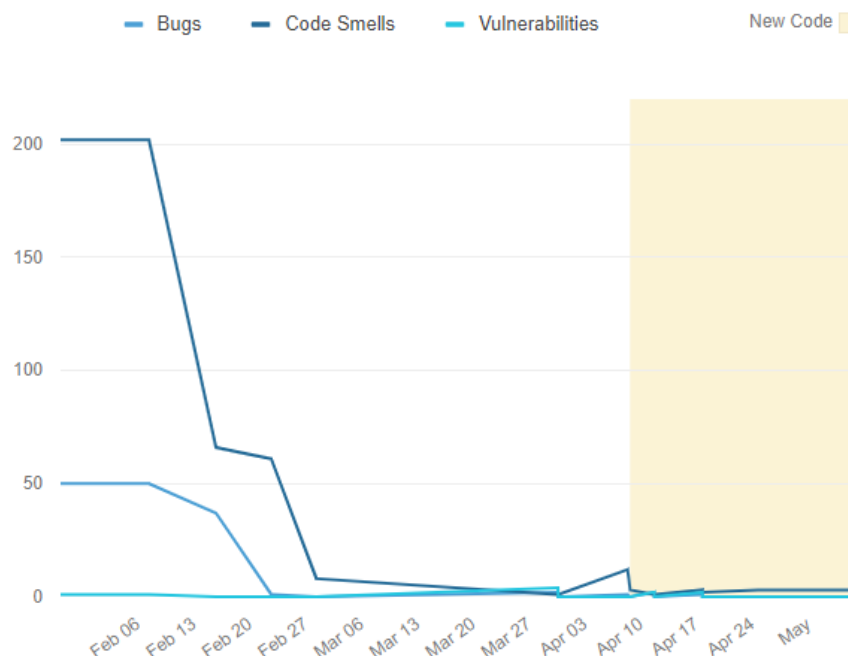


Figura 5.5: Gráfico del número de problemas a lo largo del tiempo

## 5.8. Diseño del logo de la aplicación

Debido a que la aplicación realiza una serie de consultas se me ocurrió que el logo lo formasen unas aspas y un check, pero sin el uso de colores el símbolo resultante puede resultar confuso, así que rodeé la parte del check con color verde y la parte de las aspas con color rojo para hacerlo más entendible. Los colores usados son los de Bootstrap para crear concordancia

con el resto de la página y de paso crear las imágenes de check y aspas usados en los resultados de las consultas y que de esta forma se mimeticen en el color de fondo de la celda. El texto del logo es blanco con reborde negro para hacerlo legible con independencia del color del fondo.



Figura 5.6: Logo de la aplicación

## 5.9. Almacenamiento de registros de informes

La aplicación almacena los registros de los informes generados para poder generar los gráficos de evolución. Ejecutando la aplicación en local no supone ningún problema, sin embargo, el almacenamiento en Heroku es volátil [1]. La solución a esto sería contratar un proveedor de active storage como Amazon S3 para tener un almacenamiento que funcione independiente a la aplicación [2].

## 5.10. Redefinición de regla de comentarios del profesor

Originalmente lo que se comprobaba era si el ratio entre items de calificación comentados dividido entre el número total de items comentables se encontraba por encima de cierto umbral, sin embargo, esto da falsos negativos en caso de que los estudiantes no realicen las entregas correspondientes. Si reducimos el concepto de items comentables a aquellos que tengan calificación evitaremos este problema.

También existe otro problema, actualmente, para poder realizar la consulta se comprueban los comentarios de la columna de comentarios de los calificadores de todos los alumnos, columna que el profesor puede elegir no mostrar en el calificador. Este problema causa que nuestra regla compruebe de forma implícita una decisión de diseño aparte de su objetivo principal, que es comprobar que el profesor dé retroalimentación a los alumnos al corregir (que corresponde a la fase de realización).

---

## Trabajos relacionados

---

Este apartado resume los estudios y proyectos realizados por terceros que están relacionados con la garantía y control de calidad de los procesos de enseñanza y aprendizaje en cursos en línea, 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 [4].

### **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 [11].

## **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 [12].

## **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 [18].

## **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 [21].

## 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 [6].

## Course Checks Block Plugin

Otro plugin para Moodle que permite realizar una serie de comprobaciones automáticas [9].

## Comparativa de herramientas relacionadas

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.

Característica	eLearningQA	Course Checker	Course Checks Block
Idiomas	Español	Español, inglés, alemán, y portugués	Español, inglés, portugués, y griego
Nº de comprobaciones	17	10	7
Ejecución independiente de comprobaciones	No	Sí	No
Contiene enlaces para solventar los problemas	Sí	Sí	No
Versiones de Moodle compatibles	v3.8+	v3.6+	v2.6+
Tipo	Aplicación web	Plugin	Plugin





---

# Conclusiones y Líneas de trabajo futuras

---

## 7.1. Conclusiones

En el estudio de trabajos relacionados hemos visto la importancia de controlar la calidad en los procesos de enseñanza y aprendizaje de e-learning. La calidad en este ámbito suele tomar un papel secundario, pero no debería, ya que influye directamente en la lealtad del estudiante, cuestión que está adquiriendo relevancia en los últimos años.

Los objetivos marcados para este proyecto fueron:

1. Definir las entidades y atributos de nuestro modelo de calidad a partir el modelo de datos de los cursos en Moodle: Usuarios, Roles, Cursos, Grupos, Actividades, Tareas, Recursos, Surveys, Feedbacks, Foros, Debates, y Comentarios.

Este objetivo se ha conseguido, se han llegado a definir las clases necesarias para recibir la información pertinente para los análisis que realizamos utilizando una conversión a partir de sus representaciones en JSON.

2. Conocer los servicios Web proporcionados por Moodle para interactuar con las entidades y atributos de nuestro modelo de calidad.

Este objetivo también se ha conseguido, con complicaciones. Ha sido necesario acceder al código fuente de Moodle desde GitHub [16] para rellenar las lagunas en la documentación oficial de su API de servicios Web [15].

3. Definir un plan de calidad para aplicar a las entidades del modelo de cursos de Moddle a partir de frameworks internacionales de calidad en e-learning y recomendaciones del Centro de Enseñanza Virtual de la Universidad de Burgos (UBUCEV).

Este objetivo se ha cumplido de forma satisfactoria, se han asociado responsabilidades, perspectivas y procesos a cada una de las comprobaciones que hemos definido en el plan de calidad (sección 3.4).

4. Definir evaluaciones del plan de calidad personalizadas para distintos tipos de cursos: docencia reglada, tfg y comunidades y grupos.

Este objetivo se ha cumplido de forma parcial, se han definido archivos de configuración para cada tipo de curso, estas configuraciones controlan cómo de estrictas o laxas son algunas de las comprobaciones del plan de calidad, pero lo ideal sería que se pudieran activar o desactivar reglas según la configuración usada para hacerlo más personalizable.

5. Diseñar indicadores cualitativos y cuantitativos de calidad de cada fase de diseño instruccional del curso en línea (diseño, implementación, realización y evaluación)

Este objetivo se ha cumplido, en el plan de calidad hemos definido implícitamente un indicador cualitativo para cada una de las comprobaciones realizadas, y, a partir de esos indicadores, otros que resumen el desempeño en cada una de las fases.

Se ha desarrollado y desplegado una aplicación open source, plenamente funcional, que puede ser utilizada por profesores para recibir una evaluación de su diseño e implementación del curso en Moodle.

Siendo este mi segundo trabajo de fin de grado, dispongo de un punto de referencia para ver qué diferencias hay entre ambos trabajos y sus consecuencias.

La primera gran diferencia es la organización del desarrollo en sprints, careciendo el primero de una organización temporal definida. Esto ha hecho mucho más fácil marcarse objetivos y cumplirlos manteniendo un ritmo de trabajo constante.

La presencia de dos tutores en oposición a uno solo permite tener más perspectivas sobre el trabajo a la hora de tener ideas sobre nuevas funcionalidades a implementar, sobretodo si están especializados en materias diferentes. También, cuando alguno de ellos no podía reunirse, el otro estaba disponible, cosa que resulto bastante conveniente a la hora de no perder el

ritmo que llevaba. Por último, a menudo me han presentado herramientas para facilitar y agilizar el desarrollo del trabajo.

La complejidad de este trabajo es bastante mayor y me ha obligado a aprender sobre algunas tecnologías con las que no estaba familiarizado como las herramientas de integración continua y despliegue continuo, cuestiones de desarrollo web (CSS, JavaScript, sesiones HTTP etc.), trabajar con una API de servicios web y adaptarse a ella. Este proceso de aprendizaje me ha resultado ameno.

## 7.2. Líneas de trabajo futuras

Aunque la aplicación desarrollada en este TFG es funcionalmente completa existen múltiples factores que hacen que pueda necesitar de futuras adaptaciones. Se considera interesante tener en cuenta factores como: la subjetividad intrínseca asociada a los procesos de calidad y las fuentes tecnológicas para definir controles de calidad automáticos, las diferencias en la distribución de responsabilidades de roles académicos, el tamaño de las instituciones académicas y las diferentes soluciones tecnológicas de implementación de cursos en línea. El análisis detallado de cada factor hace que surjan muchas líneas de trabajo futuras, a continuación se enumeran algunas que se han considerado interesantes.

- El método que utiliza la aplicación para realizar la comprobación de si el profesor responde a las dudas de los alumnos es una solución preliminar e incompleta. Se tiene como objetivo a futuro encontrar una forma más fiable de determinar qué es una duda y cuándo ha sido resuelta. El uso de modelos basados en el procesamiento del lenguaje natural puede ser un campo exploratorio que permita poder clasificar un mensaje del foro como una respuesta de dudas de un profesor. Pensamos que el diseño experimental y el posterior análisis de un clasificador con este cometido es suficientemente complejo para ser considerado un TFG por sí mismo. Otro problema son las situaciones en las que la duda del alumno ha sido respondida por otro alumno y no hace falta responderla o se resuelve la duda en un comentario independiente en el mismo foro y no se detecte como respuesta.
- De las cinco consultas definidas referentes a los cuestionarios Moodle (sección 3.4), no hay ninguna implementada en la aplicación, y esto se debe a que al menos cuatro de ellas no están soportadas por la API de servicios web de Moodle. Una posible solución a estas comprobaciones

es el uso de técnicas de web scraping, pero teniendo en cuenta que las páginas que contengan la información deseada pueden variar en función del servidor que aloje la plataforma, no es una solución trivial.

- Por el momento la aplicación solo integra cursos Moodle, pero sería conveniente que la aplicación permitiera analizar cursos online de otros LMS como Blackboard o Edmodo. Sin embargo, realizar los cambios para esto supondría adaptarse a las APIs de servicios correspondientes suponiendo que sean lo suficientemente parecidas, y en caso de no poder acceder a la información necesaria mediante servicios web, habría que implementar otras formas de acceder a la información necesaria.
- Los plugin de Moodle con los que comparo la aplicación tienen más idiomas disponibles, esto se debe a que están dispuestos de forma que cualquiera pueda aportar sus propias traducciones, sin embargo, el internacionalizar la aplicación la haría más competitiva.
- La lista de cursos que muestra la aplicación es la lista de los cursos en los que se encuentra matriculado el usuario con independencia de su rol o los permisos que tenga. Sería interesante seguir mostrando los mismos cursos pero sin resaltar en caso de que no se tengan los permisos necesarios para realizar las consultas para poder contactar al administrador en caso de problemas con los permisos.
- Sería muy interesante poder activar y desactivar las diferentes consultas desde los archivos de configuración si tenemos en cuenta que algunas consultas no aplican para algunos tipos de curso. Sin embargo, a la hora de calcular las estadísticas posteriores habría que adaptarse según las consultas que estén activadas, algo que es difícil teniendo en cuenta que la matriz Rol-Responsabilidad del informe se obtiene multiplicando un vector con los puntos obtenidos de cada consulta por una matriz que contiene la cantidad de puntos que supone el cumplimiento de cada comprobación para cada combinación de rol y perspectiva. El objetivo sería buscar otra forma de realizar el cálculo para facilitar la implementación de lo primero.

---

## Bibliografía

---

- [1] Active storage on Heroku, 5 2018. [Online; Accedido 16-04-2022]. URL: <https://devcenter.heroku.com/articles/active-storage-on-heroku#:~:text=Heroku%20has%20an%20%E2%80%9Cephemeral%E2%80%9D%20hard,after%20the%20application%20is%20restarted.>
- [2] Active Storage Overview, 7 2019. [Online; Accedido 16-04-2022]. URL: [https://guides.rubyonrails.org/active\\_storage\\_overview.html#amazon-s3-service](https://guides.rubyonrails.org/active_storage_overview.html#amazon-s3-service).
- [3] J Dolado and L Fernández. Medición para la gestión de proyectos de software, capítulo 12, 2000.
- [4] Rositsa Doneva and Silvia Gaftandzhieva. Automated e-learning quality evaluation. INTERNATIONAL CONFERENCE ON E-LEARNING, 2015.
- [5] 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.
- [6] FFHS, Adrian Perez Rodriguez, Christoph Karlen, and Sascha Vogel. Moodle Course Checker Plugin, 5 2021. [Online; Accedido 8-Enero-2022]. URL: [https://moodle.org/plugins/block\\_course\\_checker](https://moodle.org/plugins/block_course_checker).
- [7] Erich Gamma, Richard Helm, Ralph Johnson, Ralph E Johnson, John Vlissides, et al. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995.

- [8] Github. Security alert: Attack campaign involving stolen OAuth user tokens issued to two third-party integrators, 4 2022. [Online; Accedido 17-Mayo-2022]. URL: <https://github.blog/2022-04-15-security-alert-stolen-oauth-user-tokens/>.
- [9] Mike Grant, Jez h, Manoj Solanski, and Kevin Moore. Course Checks Block Plugin, 3 2018. [Online; Accedido 8-Enero-2022]. URL: [https://moodle.org/plugins/block\\_bcu\\_course\\_checks](https://moodle.org/plugins/block_bcu_course_checks).
- [10] Heroku. Heroku Security Notification, 4 2022. [Online; Accedido 17-Mayo-2022]. URL: <https://status.heroku.com/incidents/2413>.
- [11] 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.
- [12] Gina Mejía-Madrid, Faraón Llorens-Largo, and Rafael Molina-Carmona. Dashboard for evaluating the quality of open learning courses. *Sustainability*, 12(9):3941, 2020.
- [13] Moodle. Web services - MoodleDocs, 10 2020. [Online; Accedido 4-Noviembre-2021]. URL: [https://docs.moodle.org/dev/Web\\_services](https://docs.moodle.org/dev/Web_services).
- [14] Moodle. Estadísticas del examen de Moodle- Moodle-Docs, 12 2021. [Online; Accedido 11-Febrero-2022]. URL: [https://docs.moodle.org/all/es/Significado\\_de\\_las\\_estad%C3%ADsticas\\_del\\_examen\\_de\\_Moodle#Indice\\_de\\_discriminaci.C3.B3n\\_y\\_Eficiencia\\_de\\_discriminaci.C3.B3n](https://docs.moodle.org/all/es/Significado_de_las_estad%C3%ADsticas_del_examen_de_Moodle#Indice_de_discriminaci.C3.B3n_y_Eficiencia_de_discriminaci.C3.B3n).
- [15] Moodle. Web service API functions, 6 2021. [Online; Accedido 8-Enero-2022]. URL: [https://docs.moodle.org/dev/Web\\_service\\_API\\_functions](https://docs.moodle.org/dev/Web_service_API_functions).
- [16] Moodle. Moodle repository on GitHub, 1 2022. [Online; Accedido 8-Enero-2022]. URL: <https://github.com/moodle/moodle>.
- [17] Moodle. Trademarks - Moodle, 2 2022. [Online; Accedido 26-Febrero-2022]. URL: <https://moodle.com/trademarks>.
- [18] 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.

- [19] Ian Sommerville. *Software Engineering*. Addison-Wesley, 9 edition, 2010.
- [20] 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.
- [21] 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.