



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

UBUMonitor

**Monitorización de alumnos
en la plataforma Moodle.**



Presentado por Yi Peng Ji
en Universidad de Burgos — 19 de junio
de 2019

Tutor: Dr. Raúl Marticorena Sánchez
y Dr. Carlos Pardo Aguilar



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



Dr. Raúl Marticorena Sánchez y Dr. Carlos Pardo Aguilar, profesores del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Yi Peng Ji, con NIE X2149097T, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado “UBUMonitor - Monitorización de alumnos en la plataforma Moodle”.

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, 19 de junio de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Dr. Raúl Marticorena Sánchez

Dr. Carlos Pardo Aguilar

Resumen

El auge de las tecnologías ha alcanzado el ámbito educacional con las aulas de aprendizaje *online* y enseñanza virtual con Moodle como uno de los protagonistas más importantes.

Este hecho ha transformado también el método de seguimiento de la evolución del alumnado durante el curso.

En Moodle permite el seguimiento mediante la visualización de las calificaciones y los registros de acceso de los alumnos. Sin embargo la forma de mostrar las calificaciones y registros de acceso al curso de los alumnos en Moodle es bastante laxa y poco práctica. Este inconveniente ha motivado la creación del presente proyecto como solución.

UBUMonitor es una aplicación de escritorio hecho en Java orientado a docentes que usen los servicios de Moodle. Permite a los profesores, previa identificación, acceder a datos estadísticos diferentes sobre las calificaciones y registros de los alumnos y mostrarlos en distintos tipos de gráficas, una forma mucho más práctica y visual.

Descriptores

Moodle, Educational data mining, Visualización, JavaFX, Servicios web, JSON, ChartJs.

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

Moodle, Educational data mining, Visualization, JavaFX, Web services, JSON, ChartJs.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
2.3. Objetivos personales	4
Conceptos teóricos	5
3.1. Conceptos de Moodle	5
Técnicas y herramientas	13
4.1. Metodología SCRUM	13
4.2. Moodle	13
4.3. REST API	13
4.4. Java	14
4.5. JavaFX	14
4.6. JavaScript	15
4.7. CodePen	15
4.8. JSON	15
4.9. Eclipse	16
4.10. GitHub	16

4.11. LaTeX	16
4.12. Python	17
4.13. Dependencias externas de Java	17
4.14. Librerías de JavaScript	18
Aspectos relevantes del desarrollo del proyecto	21
5.1. Sin acceso a permisos de profesor en el Moodle de la Universidad (UBUVirtual)	21
5.2. Instalación de la versión limpia de Moodle	22
5.3. Empezar de cero la parte de los registros	23
5.4. Decisión sobre seguir el código fuente de UBUGrades o refactorizar	23
5.5. Guardar los datos de los usuarios en ficheros locales	24
5.6. Error en la función gradereport_user_get_grade_items	24
5.7. Gráfica para mostrar los registros	27
5.8. Más tiempo pensando que programando	28
5.9. Herramienta para buscar los Componentes y Eventos únicos	28
5.10. Herramientas de generación de código Java	28
Trabajos relacionados	29
6.1. Heatmap	29
6.2. SmartKlass TM Learning Analytics Moodle	30
6.3. Engagement analytics	31
6.4. Course dedication	31
Conclusiones y Líneas de trabajo futuras	33
7.1. Conclusiones	33
7.2. Líneas de trabajo futuras	34
Bibliografía	35

Índice de figuras

3.1. Ejemplos de módulos del curso que actividades y recursos. . . .	6
3.2. Extracto de la función del servicio web	7
3.3. Enlace que se genera mientras se crea el módulo	7
3.4. Ejemplo de una tabla con varios Ítem de calificación.	9
3.5. Ejemplo de tabla parcial con los usuarios e Ítems de calificación	9
3.6. Ejemplo de categorías del curso enmarcados en rojo.	9
3.7. Ejemplo de varias entradas de un registro.	10
4.8. Herramienta de Scene Builder	15
5.9. Ejemplo de Login Token en el inicio de sesión de la página Mount Orange School	22
5.10. Extracto de la respuesta <i>gradereport_user_get_grade_items</i>	25
5.11. Mensaje de error cuando la retroalimentación del calificador está oculto	26
5.12. Extracto de la función <i>gradereport_user_get_grades_table</i>	26
5.13. Ejemplo de barras apiladas con dos usuarios, cuatro componentes y de lunes a domingo	27
6.14. Ejemplo de Heatmap	30
6.15. Ejemplo del panel SmartKlass	31

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	8
---	---

Introducción

El presente proyecto es una continuación de dos trabajos anteriores, UBUGrades 2.0[21] de Félix Nogal Santamaría y UBULogs[19] de Oscar Fernández Armengol.

En ambos trabajos, la idea principal de la aplicación es el seguimiento de las calificaciones por una parte y registros de eventos por otra. Todo esto representado mediante diversas gráficas, una forma sencilla y visual de mostrar los datos.

El desarrollo de la nueva aplicación se ha integrado ambos proyectos usando UBUGrades como base, sin embargo, UBULogs no funcionaba correctamente debido a los cambios en Moodle. Este hecho ha motivado empezar de cero en esta parte siguiendo una vía alternativa.

Se ha centrado también en la refactorización del código, guardar los datos en ficheros cifrados y nuevos gráficos para los registros de accesos.

En resumidas cuentas, esta nueva versión del proyecto permite visualizar en diferentes gráficas las calificaciones de los alumnos por una parte y sobre los registros de eventos por otra parte. Todo esto con medidas de almacenaje de datos en local, y cifrados al contener información sensible. Esto posibilita acceder a los datos sin tener que realizar peticiones de recuperación de datos al servidor constantemente cada vez que se use la aplicación.

Objetivos del proyecto

En este apartado mostraremos los diferentes objetivos que han surgido para realizar el proyecto.

2.1. Objetivos generales

- Modificar el código para conseguir una mejor mantenibilidad y facilidad de añadir nuevas funcionalidades.
- Guardar los datos de Moodle en ficheros cifrados con la finalidad de acceder más rápidamente a los datos sin tener que actualizar del servidor continuamente.
- Implementar la parte de los registros (*logs*), la descarga de los ficheros CSV desde Moodle hasta su clasificación según su tipo de evento.
- Gráficas nuevas para mostrar el número de registros por cada usuario y evento.

2.2. Objetivos técnicos

- Aprendizaje sobre las REST API, incluyendo los métodos de petición HTTP[18].
- Uso de Git como método de control de versiones con la plataforma GitHub.
- Continuar el desarrollo con la librería JavaFx para la interfaz gráfica.

- Mejorar la mantenibilidad del software aplicando patrones de diseño.

2.3. Objetivos personales

- Aplicar lo máximo posible de lo aprendido de las asignaturas de la universidad.
- Mejorar en la programación de Java profundizando en la genericidad, programación funcional y patrones de diseño.
- Aportar a la comunidad de Moodle ayudando a la extracción de información de calificaciones y registros de los alumnos.
- Mantener la motivación durante el desarrollo del trabajo, así es más fácil disfrutar realizar el proyecto.

Conceptos teóricos

En este apartado comentaremos sobre conceptos que aparecen en Moodle.

3.1. Conceptos de Moodle

//TODO Introduccion a moodle

Módulo del curso

El módulo del curso[3] (en inglés *course module*) también abreviado como ‘cm’, representa las **actividades** y **recursos** del curso (ejemplos en la figura 3.1). Estos contienen información sobre qué curso y sección se muestra, además de la configuración de visibilidad y otros datos relevantes.

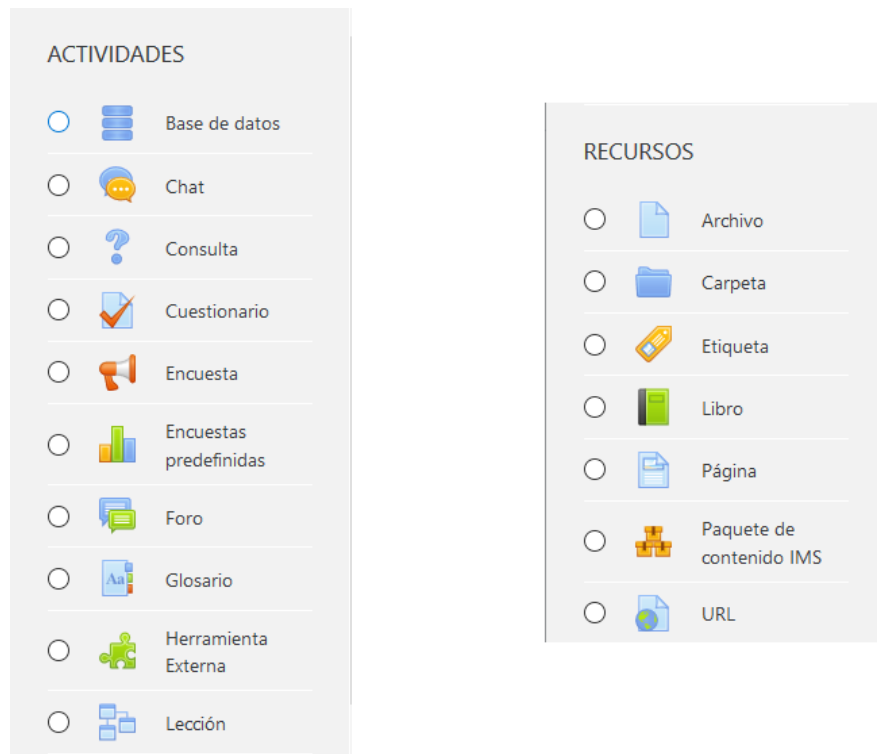


Figura 3.1: Ejemplos de módulos del curso que actividades y recursos.

Los identificadores únicos de los módulos del curso se definen como ‘**cmid**’ y cada tipo de módulo del curso posee un código nemotécnico.

Códigos nemotécnicos de los tipos de módulos del curso

Existen dos formas de buscar el código nemotécnico de un módulo del curso:

- A través de la función de servicio web ***core_course_get_contents*** que devuelve los módulos que hay en la página principal del curso y contiene un campo de tipo de módulo.

```
id : 429
url : "https://school.demo.moodle.net/mod/resource/view.php?id=429"
name : "List of necessary equipment"
instance : 18
description : "
list in pdf format of mountaineering equipment (courtesy of ontopmountaineering)
"
visible : 1
uservisible : true
visibleoncoursepage : 1
modicon : "https://school.demo.moodle.net/theme/image.php/boost/core/1560693690/ff/pdf-24"
modname : "resource"
modplural : "Files"
availability : null
indent : 0
onclick : ""
afterlink : null
customdata : ""a:2:{s:12:"printheading";i:0;s:10:"printintro";i:1;}""
completion : 1
```

Figura 3.2: Extracto de la función del servicio web

- Mediante el enlace al crear un módulo del curso.

```
https://school.demo.moodle.net/course/modedit.php?add=resource&type=&course=628&section=0&return=0&sr=0
```

Figura 3.3: Enlace que se genera mientras se crea el módulo

Tipos de módulos del curso

En la tabla 3.1 muestra los diferentes tipos de módulos que se han encontrado en Moodle.

Código nemotécnico	Nombre en español	Nombre en inglés
assign	Tarea	Assignment
book	Libro	Book
chat	Chat	Chat
choice	Consulta	Choice
data	Base de datos	Database
feedback	Encuesta	Feedback
folder	Carpeta	Folder
forum	Foro	Forum
glossary	Glosario	Glossary
imscp	Paquete de contenido IMS	IMS content package
label	Etiqueta	Label
lesson	Lección	Lesson
lti	Herramienta externa	External tool
page	Página	Page
quiz	Cuestionario	Quiz
resource	Archivo	File
scorm	Paquete SCORM	SCORM package
survey	Encuestas predefinidas	Survey
url	URL	URL
wiki	Wiki	Wiki
workshop	Taller	Workshop

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Ítem de calificación

El Ítem de calificación[5] (en inglés *Grade item*), es cada fila de la tabla de calificaciones. Las carpetas de la tabla, a parte de Ítem de calificación, se considera también como categoría (*category*).

Los Ítem de calificación se puede mostrar mediante una tabla para un solo alumno (ver figura 3.4) o una tabla de todos los alumnos y todos los Ítems de calificaciones (ver figura 3.5)

Ítem de calificación	Peso calculado	Calificación	Rango	Porcentaje	Retroalimentación	Aporta al total del curso
Moodle and Mountaineering						
Peer Assessment: Rate your mate (submission)	-	-	0-80	-	-	-
Peer Assessment: Rate your mate (assessment)	-	-	0-20	-	-	-
Assignment: Requirements for a night on the Alps	-	-	Fail-Distinction	-	-	-
Total del curso	-	-	0-100	-	-	-
Media ponderada simple de calificaciones.						

Figura 3.4: Ejemplo de una tabla con varios Ítem de calificación.

		Psychology in Cinema			
		Analysis			
First name / Surname	Email address	From Concept to Reality:...	Course discussion	Analysis total	
Frances Banks	francesbanks231@example.com	-	-	-	-
Mark Ellis	markellis267@example.com	55.00	-	55.00	
Brian Franklin	brianfrankli228@example.com	-	-	-	-
Barbara Gardner	barbaragardner249@example.com	-	-	-	-
Amanda Hamilton	amandahamilton205@example.com	95.00	Fair point	93.14	
Joshua Knight	joshuaknight196@example.com	-	-	-	-
George Lopez	georgelopez271@example.com	75.00	Good point	74.51	
Anthony Ramirez	anthonyramirez359@example.com	-	-	-	-
Donna Taylor	donnataylor203@example.com	-	Fair point	0.00	
Brenda Vasquez	brendavasquez355@example.com	65.00	Excellent point	65.69	
Gary Vasquez	garyvasquez366@example.com	75.00	-	75.00	
Overall average		73.00	Good point	60.56	

Figura 3.5: Ejemplo de tabla parcial con los usuarios e Ítems de calificación

Categorías de cursos

Las categorías de cursos[4] (en inglés *course categories*) organizan los cursos de la página de Moodle de forma jerárquica.

Página Principal / Mis cursos / **DOCENCIA REGLADA 2018** / **ESCUELA POLITÉCNICA SUPERIOR** / **GRADO EN INGENIERÍA INFORMÁTICA** / COMPUTACIÓN NEURONAL Y EVOLUTIVA (6382 #1P)

Figura 3.6: Ejemplo de categorías del curso enmarcados en rojo.

Registros

Los registros es un reporte de actividades realizadas por los usuarios, existe registros de sitio web y del curso. En este proyecto solo nos centraremos en el segundo caso, estos se acceden en desde el curso en la rueda de configuración *Aún más...* -> *Informes* -> *Registros* -> *Conseguir estos registros*

Hora	Nombre completo del usuario	Usuario afectado	Contexto del evento	Componente	Nombre evento	Descripción	Origen	Dirección IP
14 de junio de 2019, 23:10	Jeffrey Sanders	-	Encuestas predefinidas: Survey: COLLES	Encuesta predefinida	Módulo de curso visto	The user with id '13' viewed the 'survey' activity with course module id '713'.	web	2a02:2698:3c24:a191:ecaa:7715:bd71:2f25
14 de junio de 2019, 23:09	Jeffrey Sanders	-	Curso: Psychology in Cinema	Sistema	Curso visto	The user with id '13' viewed the course with id '62'.	web	2a02:2698:3c24:a191:ecaa:7715:bd71:2f25
14 de junio de 2019, 23:08	Jeffrey Sanders	-	Curso: Psychology in Cinema	Sistema	Curso visto	The user with id '13' viewed the course with id '62'.	web	83.44.99.163
14 de junio de 2019, 23:07	Barbara Gardner	-	Tarea: Group Project	Tarea	Se ha visualizado el estado de la entrega.	The user with id '56' has viewed the submission status page for the assignment with course module id '715'.	web	2601:647:202:60b6:a182:a447:1156:324d

Figura 3.7: Ejemplo de varias entradas de un registro.

Columnas de los registros

Todas las entradas del registros se muestra en una tabla con varias columnas:

1. **Hora:** La fecha y hora.
2. **Nombre completo del usuario:** Nombre y apellidos del usuario que realiza el evento o acción.
3. **Usuario afectado:** Nombre y apellidos del usuario afectado por el evento o acción.
4. **Contexto del evento:** En qué lugar del curso se ha producido el registro.
5. **Componente:** puede indicar una actividad, un recurso o el sistema.
6. **Nombre del evento:** que evento o acción se ha realizado.
7. **Descripción:** contiene la información de las anteriores columnas pero mostrando su identificador único (*id*) en vez de los nombres usuarios. Cada descripción es única en función del **Componente** y **Nombre de evento**.

8. **Origen:** indica el tipo de fuente donde se ha generado el registro. Pueden ser:

- Servicio web (ws): servicios web de Moodle.
- Web: página web.
- CLI: desde el servidor.

9. **Dirección IP:** la dirección IP del dispositivo que ha activado el registro.

Opciones y descarga

También ofrece la posibilidad de filtrar mediante diferentes filtros y su descarga en varios formatos de texto (CSV, Excel, HTML, JSON, etc).

Problema de visualización

El problema de los registros que ofrece Moodle es la laxa forma de visualizar los datos a pesar de las opciones de filtrado. Sobre todo la información de las **Descripciones** es confusa al mostrar los id's de usuarios, módulos del curso, etc...

Técnicas y herramientas

En este apartado se explicará las técnicas y herramientas usadas para el desarrollo del proyecto.

4.1. Metodología SCRUM

Se ha seguido una de las metodologías ágiles conocido como SCRUM adaptado a para una sola persona con reuniones semanales en vez diarias. Con una estrategia iterativa (incremental) con entregas parciales en los *sprints*.

4.2. Moodle

Moodle es una herramienta de gestión del aprendizaje, distribución libre y escrita en PHP. Ayuda a los docentes en la educación a distancia y en la gestión de los diferentes modos de calificación.

En la página oficial de Moodle ofrece varias versiones de prueba que cualquiera puede acceder con unos usuarios y contraseña predefinidos.

4.3. REST API

REST^[1] son todas las interfaces entre sistemas que usa HTTP para conseguir datos o realizar operaciones sobre esos datos en todos los formatos posibles, como puede ser XML y JSON.

En Moodle la funcionalidad de REST API se conoce como *Web Services* que a partir de un *token* generado previamente por la función *moo-*

dle_mobile_app se puede acceder a otras funciones del servicio. Puede ser desde ver las calificaciones de alumnos hasta crear actividades o tareas.

4.4. Java

Java es un lenguaje de programación orientado a objetos, diseñado para escribir el código una vez y ejecutar en cualquier otro dispositivo sin tener que recompilar siempre y cuando se tenga instalado *Java Runtime Enviroment (JRE)*. Actualmente es uno de los lenguajes programación con mayor popularidad de uso, tanto en el ámbito de desarrollo como en el de usuarios.

El desarrollo, al igual que el anterior proyecto UBUGrades y UBULogs, se ha seguido con Java 8 al ser una edición con soporte extendido[13].

4.5. JavaFX

Es un paquete gráfico que permite crear aplicación interactivas. Viene incluido en Java 8 dentro de la *Java Development Kit (JDK)* y en *Java Runtime Enviroment (JRE)* por lo que se puede ejecutar en otro dispositivo sin tener que instalar librerías adicionales.

Scene Builder

Es una herramienta que permite diseñar de una forma sencilla ventanas gráficas sin programar código. Los elementos de la ventana se añaden arrastrando y soltando (*Drag and drop*) dentro de la herramienta.

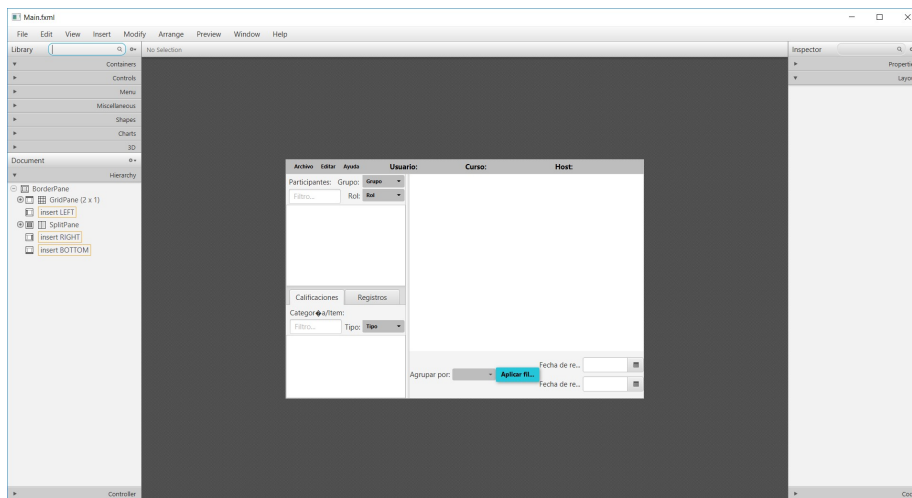


Figura 4.8: Herramienta de Scene Builder

Con Scene Builder generará un archivo FXML, muy parecido al formato XML, donde se guarda las posiciones de los elementos de la ventana gráfica. Desde Java solo tiene que cargar el fichero FXML.

4.6. JavaScript

Lenguaje de programación orientado a páginas web, usado principalmente en el lado del cliente. Al ser compatible con todos los navegadores web del mercado, se ha vuelto un estándar de facto.

En el proyecto se sigue usando Charts.js para la visualización las nuevas gráficas.

4.7. CodePen

CodePen es una herramienta web que permite crear prototipos de páginas HTML, CSS y JavaScript y ver los resultados en tiempo real.

Usado para probar prototipos de gráficas con los registros del curso.

4.8. JSON

JavaScript Object Notation (JSON), formato de texto sencillo para el intercambio de datos. En principio se usaba como subconjunto de notación

en JavaScript pero debido a su masiva popularidad en otros ámbitos, se considera hoy en día como lenguaje independiente.

Los datos de Moodle que se recibe mediante REST API está en formato de texto JSON.

4.9. Eclipse

Eclipse es uno de los entornos de desarrollo (IDE) más famosos, junto con *IntelliJ IDEA* y *NetBeans*, para Java. De código abierto con la capacidad de añadir nuevas características mediante plugins.

Se ha decidido emplear Eclipse frente a las otras alternativas por comodidad y es la herramienta que se maneja durante la carrera.

4.10. GitHub

Es una plataforma de desarrollo colaborativo para alojar proyectos usando el sistema de control de versiones Git. UBUGrades está alojado en GitHub.

Adicionalmente se ha usado la extensión del navegador ZenHub que se integra en GitHub y permite el uso de *canvas* para la organización de los *issues*.

4.11. LaTeX

Sistema de composición de textos dirigido a la creación de documentos con una buena calidad tipográfica. Actualmente muy popular en el ámbito de la investigación con artículos y libros científicos.

El presente proyecto está realizado en \LaTeX usando las versiones portables de MiKTeX y TeXstudio.

MiKTeX

MiKTeX es una distribución de \LaTeX diferentes plataformas de sistemas operativos, es de código libre y viene incluido con múltiples paquetes de tipografías y macros.

TeXstudio

TeXstudio es un entorno de desarrollo integrado (IDE) de edición de texto, código abierto con resaltado de sintaxis orientado a la creación de documentos L^AT_EX.

Zotero

Zotero es un gestor de referencias bibliográficas de código libre creado por el Center for History and New Media de la Universidad George Mason.

4.12. Python

Python es un lenguaje de programación interpretado, orientado a objetos, dinámicamente tipado y de alto nivel. Este lenguaje es muy atractivo para crear prototipos de forma sencilla y rápida.

Se ha empleado Python con el IDE **Jupyter Notebook** programando varias funciones de utilidad para el programador, una de estas funcionalidades genera código Java automáticamente.

4.13. Dependencias externas de Java

En este apartado se explicará las librerías externas de Java usadas durante el desarrollo del proyecto.

Apache Commons CSV

Apache Commons CSV 1.6 permite la lectura y/o escritura de textos en *comma-separated values* comúnmente conocido como formato **CSV**. El proyecto lo emplea para separar cada uno de las columnas y sus elementos de los registros del curso.

Apache Commons Math

Apache Commons Math 3.6.1 es una librería orientado a las matemáticas y estadísticas. Nos posibilita sacar datos estadísticos como las medias, desviaciones, máximos y mínimos solo enviando los valores.

Utilizado en el proyecto para las medias del curso tanto en las calificaciones como los registros.

Gson

Gson 2.8.5 ofrece la importación y exportación de textos JSON a partir de los métodos *toJson()* y *fromJson()*.

Ignorado de momento para el proyecto. A implementar en el futuro para guardar los datos de los registros de todos los usuarios.

SLF4J y Log4j

Ambas librerías está dedicado a registrar en fichero y/o consola las acciones realizadas durante la ejecución del programa. De gran utilidad para encontrar errores del programa. Hay que tener en cuenta que **SLF4J** es la capa de abstracción y **Log4j** la implementación.

jsoup

jsoup 1.11.3 es una librería que trabaja con textos HTML, desde la descarga de páginas web hasta su tratamiento y extracción de datos de manera sencilla y cómoda.

Ha resultado ser realmente útil, muy polivalente en las funcionalidades. Empleado para la descarga de los registros, las imágenes de los usuarios y analizar el contenido en una de las funciones de Moodle que devuelve como HTML.

ThreeTen Extra

ThreeTen Extra 1.5.0 proporciona varias formas de medir el tiempo no disponibles en el paquete de Java Time. Las clases de ThreeTen Extra son complementarias y extienden de Java Time.

Ejemplos aplicados en el proyecto han sido la combinación de número de semana y año, o trimestre y año.

4.14. Librerías de JavaScript

Al manejar una ventana de navegador web para mostrar las gráficas, se han apoyado por varias librerías de JavaScript.

Chart.js

Chart.js es una librería de visualización de datos convertidos en múltiples tipos de gráficas.

Se ha mantenido esta librería por su facilidad de crear las gráficas y sus múltiples opciones. Sin embargo por un error de la librería en el anterior proyecto, se ha actualizado Chart.js a la última versión disponible **v2.8.0**

Google Charts

Google Charts es una librería *open source* de gráficas, dispone de múltiples tipos de gráficas interactivas. Se usa para mostrar la tabla de calificaciones.

Aspectos relevantes del desarrollo del proyecto

En este apartado se comenta los aspectos importantes que han surgido durante el desarrollo del proyecto. Las decisiones tomadas relativas a la funcionalidad de la aplicación y qué consecuencias ha tenido. También explicaremos sobre los problemas encontrados y las soluciones aplicadas.

5.1. Sin acceso a permisos de profesor en el Moodle de la Universidad (UBUVirtual)

Uno de los principales problemas para realizar las pruebas no se ha podido tener acceso a asignaturas con rol de profesor. Es decir, muchas funciones de Moodle no son accesibles con el rol de estudiante mientras que los profesores si lo tienen.

Se ha instalado una versión en el equipo para realizar las pruebas, se comentara con más detalle en la sección 5.2. También se ha empleado una versión de prueba oficial del propio equipo de desarrollo *Mount Orange School* en la versión 3.7, probablemente se use esta versión en el aula virtual de la Universidad de Burgos en el futuro.

Las comprobaciones en un entorno real lo han realizado varios profesores docentes de la universidad, entre ellos los tutores, en UBUVirtual.

Cabe destacar que los tutores probaron cada mejora implementada en un corto lapso de tiempo, en menos de 24 horas laborales ya tenía el visto bueno

o errores encontrados. Esto ha facilitado mucho los tiempos de desarrollo de nuevas funcionalidades.

5.2. Instalación de la versión limpia de Moodle

Después de conocer la versión de Moodle que usa la Universidad (**3.5.1**) mediante la función `core_webservice_get_site_info`, se ha intentado instalar en mi equipo la misma versión pero ya no estaba disponible en la página de descargas [8]. Finalmente se ha usado la **3.5.4+**.

A pesar de usar una versión con cambios menores respecto al de la Universidad. Si que se encontró una diferencia reseñable.

Token para el ingreso

Añaden el Token para ingreso[17] (en inglés *login token*), una característica relacionada con la seguridad introducida en las versiones de Moodle 3.1.15, 3.3.9, 3.4.6, 3.5.3 y 3.6.0. Ayuda a proteger frente a vulnerabilidades como el robo de sesión de los usuarios. Este token de ingreso aparece como un *input* HTML oculto[6] al iniciar sesión y se envía el formulario de ingreso junto con el usuario y contraseña.

```
<div class="col-md-5">
  <form id="login" class="mt-3" action="https://school.demo.moodle.net/login/index.php"
    method="post">
    <input id="anchor" type="hidden" name="anchor" value="">
    <script></script>
    <input type="hidden" name="logintoken" value="NGCB483EUa0ET90qUwpNymCqGmQTksJ7">
    <div class="form-group"></div>
    <div class="form-group"></div>
    <div class="rememberpass mt-3"></div>
    <button id="loginbtn" class="btn btn-primary btn-block mt-3" type="submit">Acceder</button>
  </form>
</div>
```

Figura 5.9: Ejemplo de Login Token en el inicio de sesión de la página Mount Orange School

Para solucionar este problema y que funcione en ambas versiones y futuras, cada vez que el usuario inicia sesión, se busca en la página el *input logintoken* tal y como aparece en la figura 5.9 con métodos de Web Scraping[20]. Si lo encuentra manda el valor como parámetro adicional de la URL, en caso contrario no se manda este parámetro.

Hay que destacar que el Token de ingreso solo afecta si se quiere iniciar sesión en la página del servicio de Moodle (necesario para poder descargar los registros del curso), en ningún caso es necesario para los servicios Web.

5.3. Empezar de cero la parte de los registros

Se decidió prescindir totalmente del trabajo de UBULogs y empezar de nuevo debido a las modificaciones de Moodle como el token de ingreso (ver sección 5.2).

Se ha seguido por una vía diferente al mencionado proyecto, descargar los CSV directamente en memoria sin guardarlo en un fichero temporal. También se ha aplicado una mejora para bajarse los registros de forma diaria cuando se actualiza los datos.

Otra mejora reseñable, al estar las horas de los registros mostradas en función de la zona horaria usada por el usuario, se ha decidido tomar como referencia para la aplicación la zona horaria del servidor. No importa si el usuario esta usando otra que no sea la misma que el servidor.

5.4. Decisión sobre seguir el código fuente de UBUGrades o refactorizar

Cuando pasaron dos semanas desde el inicio del proyecto, acabé en la tesitura sobre seguir con el código fuente o refactorizar la parte lógica de la aplicación (back-end) aplicando patrones de diseño y tenga mejor mantenibilidad.

Dos de los temores de tomar la decisión de refactorizar fue quedarme a medio camino en la entrega del proyecto o problemas de integración con el vista e interacción con el usuario (front-end).

Finalmente, después de otras dos semanas de reflexión, se decide refactorizar el código de la parte del back-end completamente. Al empezar de nuevo el código, se dedica un esfuerzo extra en el proyecto para cumplir los tiempos.

5.5. Guardar los datos de los usuarios en ficheros locales

Uno de los problemas que tenía el anterior proyecto te obligaba descargar los datos del servidor de Moodle cada vez que accedías a la aplicación aunque no haya cambios importantes del curso.

Una mejora que se ha implementado da la opción al usuario de guardar los datos en un fichero local y cargarlo de forma instantánea. Cabe destacar que el fichero local está cifrado, al contener datos sensibles, usando el algoritmo Blowfish[15] que está destinado para uso doméstico y además es seguro.

5.6. Error en la función `gradereport__user__get__grade__items`

Al igual que el anterior proyecto de UBUGrades, se volvió a implementar función del servicio web **`gradereport__user__get__grade__items`** ya facilitaba en gran medida recoger las calificaciones de los alumnos. Aunque no devuelve bien la jerarquía.

5.6. ERROR EN LA FUNCIÓN

GRADEREPORT_USER_GET_GRADE_ITEMS

25



▼ 3:	
id:	427
itemname:	"Quiz: Test your knowledge on Alpinism"
itemtype:	"mod"
itemmodule:	"quiz"
iteminstance:	39
itemnumber:	0
categoryid:	74
outcomeid:	null
scaleid:	null
locked:	false
cmid:	439
weightraw:	0.97087
weightformatted:	"97.09 %"
graderaw:	25
gradedatesubmitted:	1560773685
gradedategraded:	1560773685
gradehiddenbydate:	false
gradeneedsupdate:	false
gradeishidden:	false
gradeislocked:	false
gradeisoverridden:	false
gradeformatted:	"25.00"
grademin:	0
grademax:	100
rangeformatted:	"0–100"
percentageformatted:	"25.00 %"
feedback:	""
feedbackformat:	0

Figura 5.10: Extracto de la respuesta *gradereport_user_get_grade_items*

Después de implementar esta funcionalidad, cuando lo probó uno de tutores daba error en la respuesta del servicio web. Finalmente se descubrió que era un error de Moodle cuando se oculta la columna de **Retroalimentación o Feedback** del calificador[23]. Este error aun no se ha solucionado en la versión 3.7 de Moodle.

Debido a este inconveniente, se decidió trabajar con otra función del servicio web, *gradereport_user_get_grades_table*, mucho más incómoda para extraer datos debido a que las respuestas contienen HTML.

```

{
  "exception": "invalid_response_exception",
  "debuginfo": "usergrades => Detectado valor de respuesta no válido: gradeitems
=> Detectado valor de respuesta no válido: Error in response - Missing following
required key in a single structure: feedbackformat",
  "message": "Detectado valor de respuesta no válido",
  "errorcode": "invalidresponse"
}

```

Figura 5.11: Mensaje de error cuando la retroalimentación del calificador está oculto

```

▼ 0:
  ▼ itemname:
    class: "level1 levelodd oddd1 b1b b1t column-itemname"
    colspan: 7
    ▼ content:
      celltype: "th"
      id: "cat_74_48"
  ▼ leader:
    class: "level1 levelodd oddd1 b1t b2b b1l column-leader"
    rowspan: 5
  ▼ 1:
    ▼ itemname:
      class: "level2 leveleven item b1b column-itemname"
      colspan: 1
      ▼ content:
        celltype: "th"
        id: "row_269_48"
    ▼ weight:
      ▼ class: "level2 leveleven item b1b itemcenter column-weight"
        content: "-"
        headers: "cat_74_48 row_269_48 weight"
    ▼ grade:
      class: "level2 leveleven item b1b itemcenter column-grade"
      content: "-"
      headers: "cat_74_48 row_269_48 grade"
    ▼ range:
      class: "level2 leveleven item b1b itemcenter column-range"
      content: "0&ndash;80"
      headers: "cat_74_48 row_269_48 range"
    ▼ percentage:
      ▼ class: "level2 leveleven item b1b itemcenter column-percentage"
        content: "-"
        headers: "cat_74_48 row_269_48 percentage"
    ▼ feedback:
      ▼ class: "level2 leveleven item b1b feedbacktext column-feedback"
        content: "&nbsp;"
        headers: "cat_74_48 row_269_48 feedback"
    ▼ contributiontocoursetotal:
      ▼ class: "level2 leveleven item b1b itemcenter column-contributiontocoursetotal"
        content: "-"
        headers: "cat_74_48 row_269_48 contributiontocoursetotal"

```

Figura 5.12: Extracto de la función *gradereport_user_get_grades_table*

Este imprevisto ha supuesto un retraso de una semana en el proyecto.

5.7. Gráfica para mostrar los registros

Ha sido una tarea titánica buscar una forma de mostrar gráficamente los registros con un determinado número de usuarios, otros tantos de componentes/eventos y en diferentes forma de tiempo (días, meses, años, etc...).

Se probado diferentes maneras de crear gráficas, experimentando varios plugins de Chart.js[14] como *chartjs-plugin-datalabels*[2] o *Chart.BarFunnel.js*[22]. Se han creado varios prototipos con estas librerías pero finalmente se desecharon porque no daban muy buenos resultados y para no tener una excesiva dependencia a librerías externas.

Todos los prototipos se realizaron con CodePen como este:

<https://codepen.io/yipengji/pen/yrPzLr>

Finalmente se decidió usar barras apiladas por componente/evento, agrupadas en el eje X con el formato temporal y una barra por cada usuario.

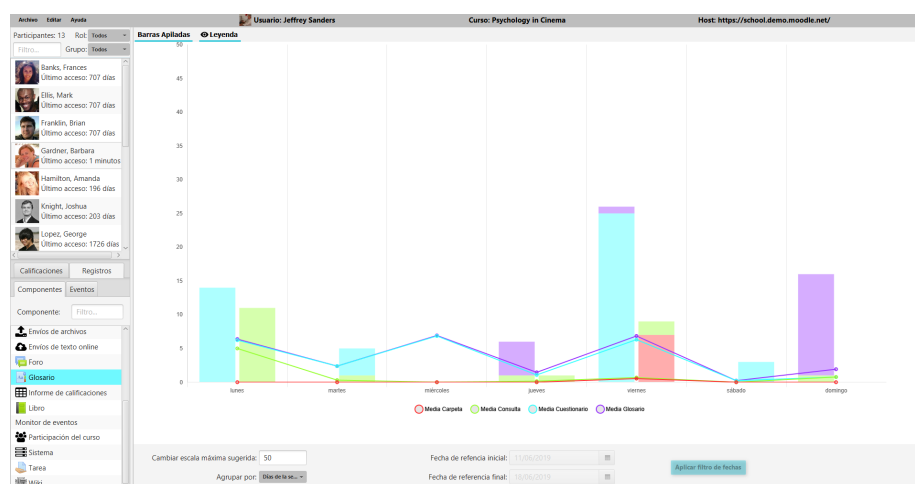


Figura 5.13: Ejemplo de barras apiladas con dos usuarios, cuatro componentes y de lunes a domingo

5.8. Más tiempo pensando que programando

En general, se ha dedicado más tiempo en pensar cómo estructurar el código que programando en sí.

Se ha hecho un gran esfuerzo para que el código sea mantenible en la mayoría de los paquetes excepto el de *controllers* por falta de tiempo.

Por ejemplo, es muy fácil añadir una nueva forma de agrupar los tiempos, se puede agregar nuevas combinaciones de Componentes y Eventos de los registros del curso.

5.9. Herramienta para buscar los Componentes y Eventos únicos

Se ha creado una herramienta en *Jupyter Notebook* usando Python. Busca todas las combinaciones de **Componente**, **Nombre de evento** y **Descripción** diferentes. Todo esto se guardan en fichero JSON para facilitar la legibilidad y por si es necesario emplearlo en proyectos futuros.

5.10. Herramientas de generación de código Java

También creado otra herramienta que usa el fichero JSON comentado anteriormente en 5.9 para genera código en Java de las enumeraciones de Componentes y Eventos, además de las traducciones en inglés o español.

Trabajos relacionados

En cuanto a los trabajos relacionados tenemos dos precedentes comentados anteriormente, UBUGrades y UBULogs hechos como Trabajos de Fin de Grado por alumnos de la universidad.

Existe varios plugins de Moodle que realizan trabajos parecidos al proyecto.

6.1. Heatmap

Heatmap^[9] superpone un mapa de calor (gradiente de color) en un curso para resaltar los módulos dependiendo del nivel de actividad y así ayudar a los docentes a mejorar sus cursos.

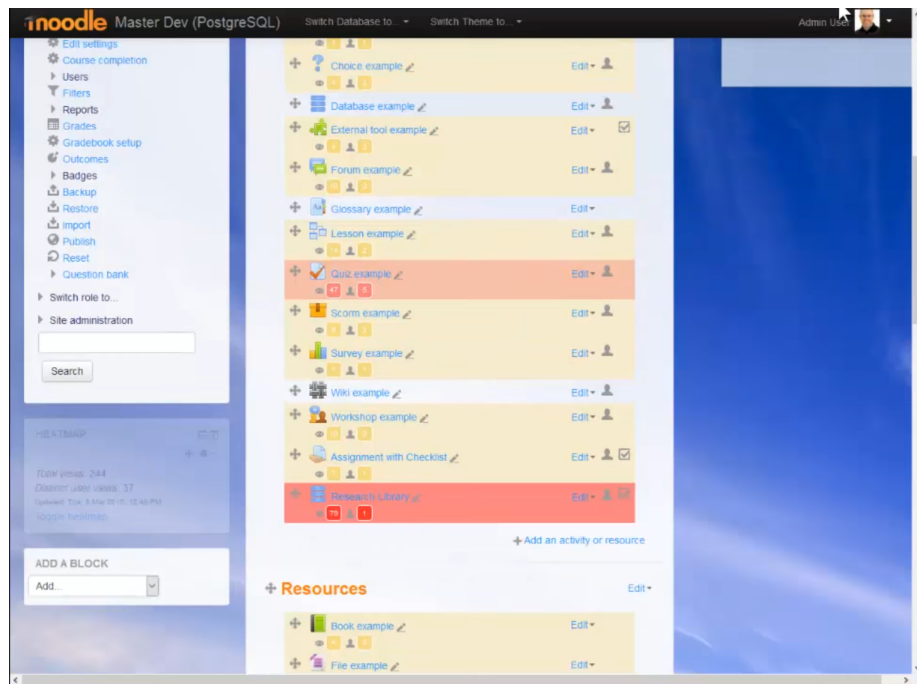


Figura 6.14: Ejemplo de Heatmap

6.2. SmartKlassTM Learning Analytics Moodle

SmartKlassTM Learning Analytics Moodle[10] es un panel de análisis dirigido a docentes y estudiantes que muestra la evolución del rendimiento sobre el curso.

Aparentemente el proyecto lleva sin actualizarse desde hace 3 años[7].

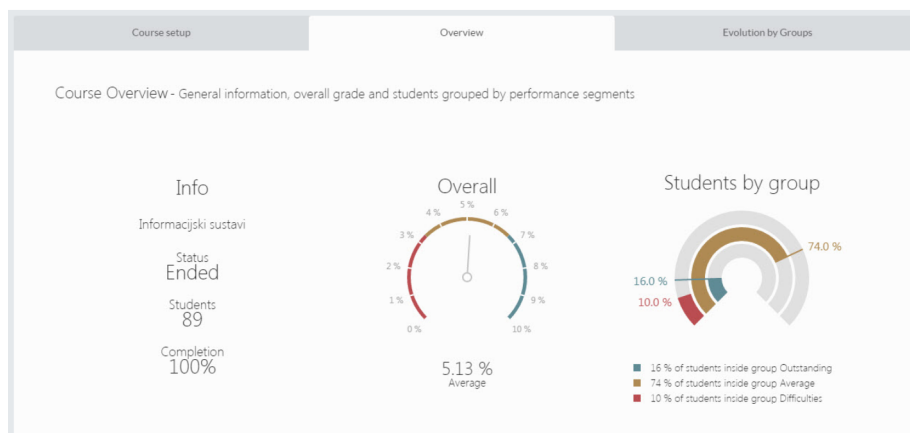


Figura 6.15: Ejemplo del panel SmartKlass

6.3. Engagement analytics

Engagement analytics[11] proporciona información sobre el progreso de los estudiantes en relación con una serie de indicadores.

Como su nombre indica, analiza el nivel de compromiso o participación del estudiante en el curso.

6.4. Course dedication

Course dedication[12] Calcula el tiempo estimado de dedicación de los participantes en el curso.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Después de finalizar el proyecto, una de las conclusiones que podemos sacar es que se ha cumplido bastante bien los objetivos planeados. Uno de los cambios más importantes fue empezar y modificar prácticamente todo el proyecto excepto la interfaz gráfica, ha salido mucho mejor de lo esperado.

Se han implementado multitud de nuevas funcionalidades, algunas de ellas por iniciativa propia, como guardar en un fichero de configuración los nombres de usuario y el *host*, imágenes de los participantes del curso facilitando la identificación y también mostrar la última vez que se accedió al servidor de Moodle.

He aplicado lo aprendido de muchas de las asignaturas de la carrera, procesadores de lenguaje en las expresiones regulares, patrones de diseño y Java. De esto último he aplicado prácticamente todas sus características tales como herencia, genericidad, programación funcional o Java Stream. La curva de aprendizaje con JavaFx no fue muy complicada ya que había tenido experiencias previas con varios trabajos de la universidad con Tkinter[16], una librería de creación de interfaces gráficas pero en Python y con ciertos parecidos.

En resumidas cuentas, en general he disfrutado mucho realizando el proyecto, con motivación se puede realizar grandes trabajos y bien mimados.

7.2. Líneas de trabajo futuras

A pesar de cumplir los objetivos planteados, sí que se han encontrado otras funcionalidades durante el desarrollo del proyecto. Estas nuevas características, generalmente no se han podido implementar por falta de tiempo.

- Empleo de técnicas de minería de datos que generan clúster de usuarios del curso. Estos clústeres se muestran en una gráfica de dispersión (en inglés *scatter plot*)
-

Bibliografía

- [1] API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos, . URL <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.
- [2] chartjs-plugin-datalabels, . URL <https://chartjs-plugin-datalabels.netlify.com/>.
- [3] Course module - MoodleDocs, . URL https://docs.moodle.org/dev/Course_module.
- [4] Course categories - MoodleDocs, . URL https://docs.moodle.org/37/en/Course_categories.
- [5] Grade items - MoodleDocs, . URL https://docs.moodle.org/37/en/Grade_items.
- [6] HTML input type="hidden", . URL https://www.w3schools.com/tags/att_input_type_hidden.asp.
- [7] klassdata/moodle-local_smart_klass: Moodle Plugin for SmartKlass Learning Analytics Tool, . URL https://github.com/klassdata/moodle-local_smart_klass.
- [8] Moodle packages for Windows, . URL <https://download.moodle.org/windows/>.
- [9] Moodle plugins directory: Heatmap, . URL https://moodle.org/plugins/block_heatmap.
- [10] Moodle plugins directory: SmartKlassTM Learning Analytics Moodle, . URL https://moodle.org/plugins/local_smart_klass.

- [11] Moodle plugins directory: Engagement analytics, . URL https://moodle.org/plugins/report_engagement.
- [12] Moodle plugins directory: Course dedication: Stats, . URL https://moodle.org/plugins/stats.php?plugin=block_dedication.
- [13] Oracle Java SE Support Roadmap, . URL <https://www.oracle.com/technetwork/java/java-se-support-roadmap.html>.
- [14] Plugins · Chart.js documentation, . URL <https://www.chartjs.org/docs/latest/developers/plugins.html>.
- [15] Schneier on Security: The Blowfish Encryption Algorithm, . URL <https://www.schneier.com/academic/blowfish/>.
- [16] TkInter - Python Wiki, . URL <https://wiki.python.org/moin/TkInter>.
- [17] Token para ingreso - MoodleDocs, . URL https://docs.moodle.org/all/es/Token_para_ingreso.
- [18] W3School . HTTP Methods GET vs POST. URL https://www.w3schools.com/tags/ref_httpmethods.asp.
- [19] Oscar Fernández Fernández Armengol. Aplicación de escritorio para visualizar y comparar la actividad de los distintos usuarios de Moodle en los cursos. La herramienta debe permitir la integración con UBUVirtual. Tecnologías: Java, Ja., December 2017. URL <https://github.com/trona85/GII-17.1B-UBULog-1.0>. original-date: 2017-10-09T13:32:44Z.
- [20] Marq Martí. Qué es el Web scraping? Introducción y herramientas, April 2016. URL <https://sitelabs.es/web-scraping-introduccion-y-herramientas/>.
- [21] Felix Nogal Santamaría. Continuación y mejora de la aplicación UBU-Grades. Contribute to huco95/UBUGrades development by creating an account on GitHub, October 2017. URL <https://github.com/huco95/UBUGrades>. original-date: 2017-10-10T23:30:55Z.
- [22] y takey. This plugin for Chart.js that makes your bar chart to 100 % stacked bar chart.: y-takey/chartjs-plugin-stacked100, June 2019. URL <https://github.com/y-takey/chartjs-plugin-stacked100>. original-date: 2017-03-23T06:43:40Z.

- [23] Kin Zadok. [MDL-64298] `gradereport_user_get_grade_items` fails when "Course Grade Settings" when "Show Feedback" is set to "False" for user reports - Moodle Tracker. URL <https://tracker.moodle.org/browse/MDL-64298>.