



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 — 13 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, 13 de junio de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

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.

La aplicación continua el desarrollo de dos proyectos diferentes, UBUGrades que trata sobre las calificaciones y UBULogs de los registros. Se han integrado ambos desarrollos en uno único, usando la primera como base y reescribiendo totalmente la segunda.

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

Descriptores

Java, JavaFX, REST API, JSON, Moodle, ChartJs, JavaScript.

Abstract

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

Keywords

Java, JavaFX, REST API, JSON, Moodle, ChartJs, JavaScript.

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
3.2. Otros términos	7
Técnicas y herramientas	9
4.1. Metodología SCRUM	9
4.2. Moodle	9
4.3. Java	10
4.4. JavaFX	10
4.5. JavaScript	11
4.6. JSON	11
4.7. Eclipse	11
4.8. GitHub	11
4.9. LaTeX	12

4.10. Dependencias externas de Java	12
4.11. Librerías de JavaScript	13
Aspectos relevantes del desarrollo del proyecto	15
5.1. Sin acceso a permisos de profesor en el Moodle de la Universidad	15
5.2. Instalación de la versión limpia de Moodle	15
Trabajos relacionados	17
Conclusiones y Líneas de trabajo futuras	19
Bibliografía	21

Índice de figuras

3.1. Ejemplos de módulos del curso que actividades y recursos. . . .	6
3.2. Ejemplo de una tabla con varios Ítem de calificación.	6
3.3. Ejemplo de categorías del curso enmarcados en rojo.	7
4.4. Herramienta de Scene Builder	10

Índice de tablas

Introducción

El presente proyecto es una continuación de dos trabajos anteriores, UBU-Grades 2.0[9] de Félix Nogal Santamaría y UBULogs[8] 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, debido a los cambios en Moodle UBULogs no funciona correctamente. Este hecho ha motivado empezar de cero en esta parte siguiendo una vía alternativa.

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[7].
- 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

Moodle

Módulo del curso

El módulo del curso^[3] (en inglés *course module*) también abreviado como ‘cm’, representa las actividades y recursos del curso. Estos contienen información sobre qué curso y sección se muestra, además de la configuración de visibilidad y otros datos relevantes. Los identificadores únicos de los módulos del curso se definen como ‘**cmid**’.

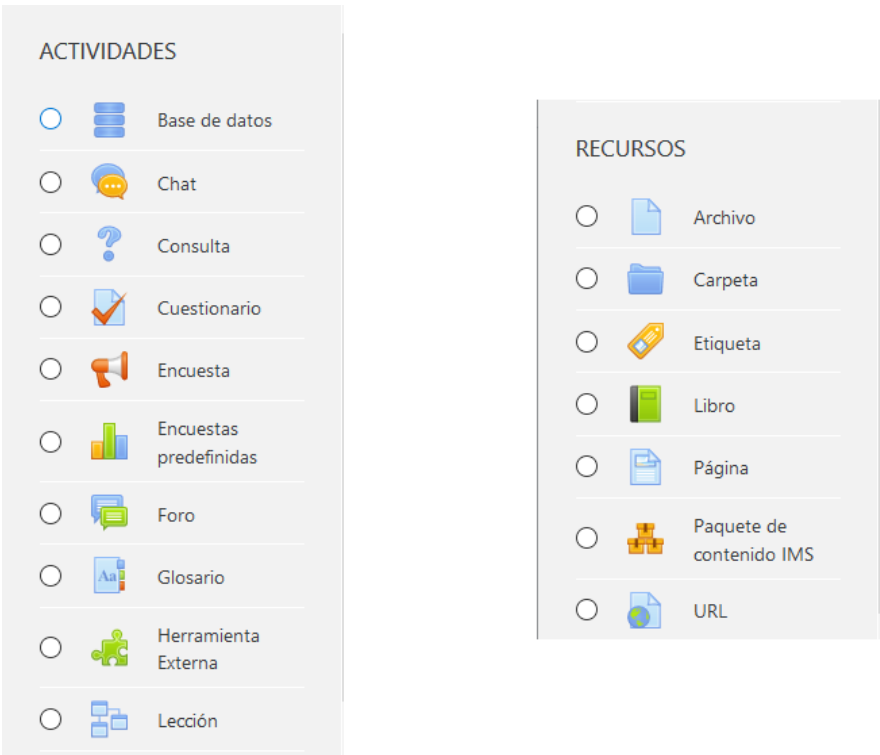


Figura 3.1: Ejemplos de módulos del curso que actividades y recursos.

Ítem de calificación

El Ítem de calificación[4] (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	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 Media ponderada simple de calificaciones.	-	-	0-100	-	-	-

Figura 3.2: Ejemplo de una tabla con varios Ítem de calificación.

Categorías de cursos

Las categorías de cursos[2] (en inglés *course categories*) organizan los cursos de la página de Moodle de forma jerárquica.

A screenshot of a Moodle breadcrumb trail. The text is: 'Página Principal / Mis cursos / DOCENCIA REGLADA 2018 / ESCUELA POLITÉCNICA SUPERIOR / GRADO EN INGENIERÍA INFORMÁTICA / COMPUTACIÓN NEURONAL Y EVOLUTIVA (6382 #1P)'. The last four items are enclosed in red rectangular boxes: 'DOCENCIA REGLADA 2018', 'ESCUELA POLITÉCNICA SUPERIOR', 'GRADO EN INGENIERÍA INFORMÁTICA', and 'COMPUTACIÓN NEURONAL Y EVOLUTIVA (6382 #1P)'.

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

Figura 3.3: Ejemplo de categorías del curso enmarcados en rojo.

3.2. Otros términos

REST API

REST[1] son todas las interfaces entre sistemas que use 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 ***moodle_mobile_app*** se puede acceder a otras funciones del servicio. Puede ser desde ver las calificaciones de alumnos hasta crear actividades o tareas.

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.

Al no tener acceso con privilegios de profesor al servicio Moodle de la universidad, se ha tenido que instalar un servidor de pruebas en localhost. Las comprobaciones en un entorno real lo han realizado varios profesores docentes de la universidad, entre ellos el tutor, en UBUVirtual.

Se ha usado la versión **3.5.4+** de Moodle al no estar disponible para descargar la **3.5.1** instalada en UBUVirtual.

También se ha empleado una versión de prueba oficial del propio equipo de desarrollo *Mount Orange School* en la versión **3.7**, que probablemente se use en la aula virtual de la Universidad de Burgos en el futuro.

4.3. 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 **sophote extendido**.

4.4. 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.

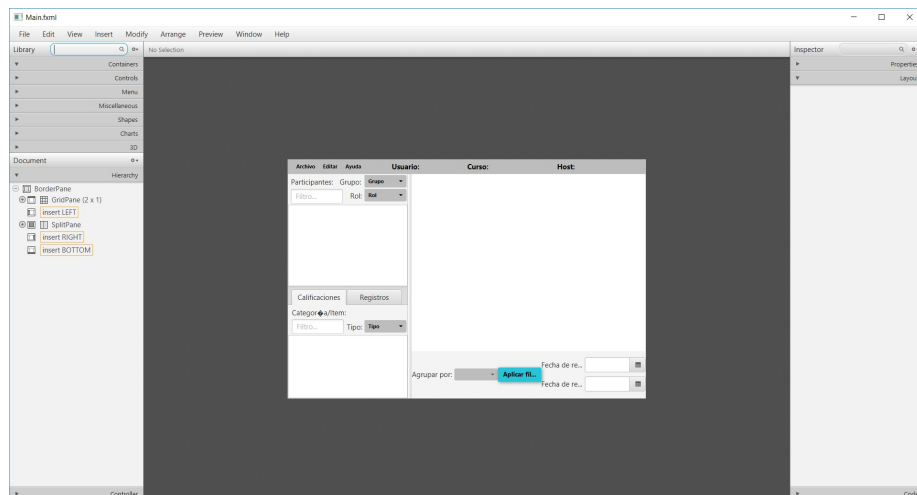


Figura 4.4: Herramienta de Scene Builder

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.

4.5. 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.

4.6. 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.

4.7. 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.

4.8. 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*.

4.9. 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 \LaTeX .

4.10. 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 posibilita 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 en páginas web hasta su tratamiento y extracción de datos de forma 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 se complementa con Java Time

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

4.11. 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**

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

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

Una de las soluciones

5.2. Instalación de la versión limpia de Moodle

Después de conocer que versión Moodle de la Universidad (**3.5.1**) a partir de la función `core_webservice_get_site_info`, se ha intentado instalar en mi equipo la misma versión pero ya no estaba disponible [5]. 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 encontraron varias diferencias reseñables.

Token para el ingreso

Por una parte añaden el Token para ingreso[6] (*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 al iniciar sesión y se envía el formulario de ingreso junto con el usuario y contraseña.

- version de ubuvirtual no disponible, usado como pruebas una version parecida
- debate sobre si trabajar con el codigo o empezar de 0 el back end, 2 semanas de reflexion. Apuesta arriesgada debido a quedarme a medio camino en la entrega o problemas de integración con el front end.
- la importancia de las librerias externas que facilitan el trabajo. se intento trabajar con el paquete java net pero fallo estrepitosamente, jsoup salvacion en unas pocas lineas de codigo
- usado mas tiempo en pensar como estructurar las clases de forma que sea facil de leer y facil de modificar, que programando
- retraso de tiempos al haber un error de moodle en una funcion que saltaba error si el feedback del calificador esta oculto, se ha tenido que usar otra funcion mas engorrosa de trabajar (HTML)
- dedicado varias semanas a elegir como mostrar la grafica de registros, probado con varios prototipos de graficas de barras diferentes, con y sin plugins para chartjs
- ademas de los prototipos de las barras tambien se ha estado pensando y probando como estructurar el codigo para que sea lo mas sencillo posible añadir nuevos tipos de agrupaciones de fechas, crear los datasets de las barras, que sea sencillo añadir para componentes, eventos

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

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

Bibliografía

- [1] API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos.
- [2] Course categories - MoodleDocs.
- [3] Course module - MoodleDocs.
- [4] Grade items - MoodleDocs.
- [5] Moodle packages for Windows.
- [6] Token para ingreso - MoodleDocs.
- [7] W3School . HTTP Methods GET vs POST.
- [8] Oscar 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. original-date: 2017-10-09T13:32:44Z.
- [9] Felix Nogal Santamaría. Continuación y mejora de la aplicación UBU-Grades. Contribute to huco95/UBUGrades development by creating an account on GitHub, October 2017. original-date: 2017-10-10T23:30:55Z.