



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 — 24 de junio
de 2019

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

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	5
Apéndice B Especificación de Requisitos	9
B.1. Introducción	9
B.2. Objetivos generales	9
B.3. Catalogo de requisitos	9
B.4. Especificación de requisitos	9
Apéndice C Especificación de diseño	11
C.1. Introducción	11
C.2. Diseño de datos	11
C.3. Diseño procedimental	11
C.4. Diseño arquitectónico	11
Apéndice D Documentación técnica de programación	13
D.1. Introducción	13
D.2. Estructura de directorios	13
D.3. Manual del programador	13

D.4. Compilación, instalación y ejecución del proyecto	13
D.5. Pruebas del sistema	13
Apéndice E Documentación de usuario	15
E.1. Introducción	15
E.2. Requisitos de usuarios	15
E.3. Instalación	15
E.4. Manual del usuario	15
Bibliografía	19

Índice de figuras

E.1. Primera pantalla de inicio de sesión.	16
E.2. Pantalla de selección de curso	17

Índice de tablas

A.1. Sueldos anuales, mensuales y de 5 meses.	5
A.2. Dependencias de Java.	6
A.3. Dependencias de JavaScript.	6

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apartado se comentará la planificación temporal de proyecto junto con su viabilidad económica y legal.

A.2. Planificación temporal

El desarrollo de la aplicación esta planteado usando Scrum como metodología de gestión del proyecto. Aunque cabe destacar que está adaptado para trabajar con una persona a diferencia de los 4-8 usuales y también con reuniones semanales en vez de diarias. A continuación se muestran las características que se han seguido de la filosofía ágil:

- Desarrollo iterativo o incremental mediante interacciones (*sprints*).
- La duración de los *sprints* han sido de uno a cuatro semanas y cada una de ellas contiene las tareas o *issues* realizadas.
- Reuniones semanales de más de una hora comentando el avance del desarrollo y presentando partes del producto final.
- También se planifica las siguientes tareas a realizar durante las reuniones.

Sprint 0 (14/01/2019 - 08/02/2019)

Durante la primera semana se buscó reuniones con tres docentes de la Universidad (José Manuel Galán Ordax, Bruno Baruque Zanón y Raúl Marticorena Sánchez) sobre posibles proyectos que tuvieran en mente.

Después de hablar con los tres en esa semana, hubo varios de reflexión sobre la difícil elección del trabajo. Finalmente se elige el proyecto de Raúl Marticorena.

En la siguiente semana se prepara la primera reunión comentando en profundidad los detalles del trabajo y los objetivos finales que se espera.

En este *sprint* también se prepara las instalaciones de las herramientas de necesarias para el desarrollo. Se ha usado el manual del programador aportado por el autor de UBUGrades 2.0 como medida de apoyo.

Sprint 1 (08/02/2019 - 15/02/2019)

Buscar la manera de guardar los datos extraídos del servicio web en ficheros locales cifrados usando la contraseña de acceso a Moodle como clave. Se decide usar el algoritmo de encriptación **Blowfish** [8] que permite claves de 32 hasta 448 bits. Cabe destacar que se han descartado los algoritmos de cifrado que usan tamaños específicos de claves. Un ejemplo de ellos es el algoritmo *Advanced Encryption Standard* (AES) [6] que necesita claves de cifrado de 128, 192 o 256 bits.

Se acuerda empezar de nuevo el trabajo de UBULogs al no funcionar para la nueva versión de Moodle de la Universidad usando una alternativa diferente.

También se arregla un error del anterior proyecto el cual trataba al nombre de usuario como si fuera un correo.

Sprint 2 (15/02/2019 - 13/03/2019)

En este *sprint* se busca la forma de iniciar sesión con el usuario y contraseña de Moodle y así poder descargar los registros del curso. Inicialmente se intentó usar solo el paquete **java.net** [3] para realizar las conexiones al no ser una de dependencia externa. Sin embargo los resultados fueron infructuosos y finalmente se empleó Jsoup [4], una librería muy famosa de web scraping.

Además se decide modificar completamente las clases encargadas de recoger información de los servicios web de Moodle.

Sprint 3 (13/03/2019 - 21/03/2019)

Los objetivos fueron arreglar los errores encontrados en el anterior *sprint* y cambiar el paquete model modificando las clases y añadiendo todos los tipos disponibles de actividades y recursos del curso.

Sprint 4 (21/03/2019 - 26/03/2019)

Centrado exclusivamente en recoger los datos de los servicios web y transformarlo en objetos de Java. La información que nos interesa son los usuarios matriculados, los contenidos (actividades y recursos), las calificaciones, los roles y grupos del curso.

Todo esto se extrae de los JSON que devuelve el servicio web de Moodle, se han creado métodos que recogen los campos más útiles del JSON.

Sprint 5 (26/03/2019 - 03/04/2019)

Buscado la forma de encontrar cómo actualizar los datos, tanto de las calificaciones como de los registros del curso.

En cuanto a los registros del curso se descubre una forma de descargar los datos por días a través de un parámetro de la URL. Los tiempos de actualización de datos baja considerablemente con este método al descargar registros parciales en vez de completos.

También se refactoriza códigos de anteriores *sprints* y **Carlos Pardo Aguilar** se incorpora como segundo tutor.

Se acuerda en qué tipo de gráfica debe mostrar los registros del curso, barras apiladas.

Sprint 6 (03/04/2019 - 26/04/2019)

Los objetivos son arreglar los errores de programación encontrados. También aparece un bug de los servicios web con la función **gradereport_user_get_grade_items** que devuelve un mensaje de error si la **Retroalimentación** está oculta en el calificador.

Después de varias propuestas se llega el consenso de usar la otra función del servicio web **gradereport_user_get_grades_table** mucho más incómoda de trabajar. Este inconveniente ha supuesto un retraso de una semana en el proyecto.

Se añadió un fichero de configuración para guardar los nombres de usuario y el servidor de Moodle en la pantalla de inicio de sesión.

Sprint 7 (26/04/2019 - 11/05/2019)

Durante este proceso se implementó la agrupación de los registros por usuarios, componente y distintos tipos de tiempo (días, meses, años, etc) y finalmente contándolos. Se calcula también la media de todos los participantes según el número de registros.

Con todos estos datos se crean métodos que transforma en formato específico de Chart.js.

Sprint 8 (11/05/2019 - 18/05/2019)

Durante este *sprint* se ha buscado una manera de añadir fácilmente nuevos tipos de registros según el componente y evento. Finalmente se ha creado varias herramientas en Python que busca los diferentes tipos que hay a partir de los registros en CSV y los guarda en un fichero JSON. Con este fichero se ha creado otra herramienta para generar las enumeraciones Java de los componentes y eventos junto con las traducciones. Todo esto se comenta con más detalles en el manual del programador (D.3).

Sprint 9 (18/05/2019 - 23/05/2019)

Los objetivos fueron empezar la documentación de la memoria y probar si se puede añadir la opción de agrupar por componente y evento en los registros. Para esto se creó una nueva rama con resultados satisfactorios.

También se añadieron las fotografías de los participantes del curso.

Sprint 10 (23/05/2019 - 08/06/2019)

En este *sprint* se añadieron varias mejoras de interacción con el usuario:

- Valores predefinido en el campo *host* de la pantalla de inicio.
- Mostrar la categoría del curso al en la pantalla de selección del curso.
- Añadir una opción de ajustar el valor máximo del eje Y en el gráfico de barras.

Sprint 11 (08/06/2019 - 22/06/2019)

Los objetivos fueron completar la memoria, documentación de los métodos de Java, arreglar varios errores y refactorización del código arreglando la gran mayoría de *code smells* detectados por Sonarqube y Codacy.

Sprint 12 (22/06/2019 - 30/06/2019)

Completar los apartados restantes del anexo y su revisión final.

A.3. Estudio de viabilidad**Viabilidad económica**

Desarrollado por una sola persona durante 5 meses. Calculando usando salarios medios brutos de 19.485€ para un programador junior [7]. Empleando la calculadora de sueldos netos de Cinco Días [5] nos da los resultados en costes anuales. Calculamos también el sueldo mensual y el coste en 5 meses:

Concepto	Coste anual	Coste mensual	Coste 5 meses
Sueldo bruto	19.485,00 €	1.623,75 €	8.118,75 €
Retenciones por IRPF	2.221,30 €	185,11 €	925,54 €
Cuotas a la Seg. Social	1.237,30 €	103,11 €	515,54 €
Sueldo neto	16.026,40 €	1.335,53 €	6.677,67 €

Tabla A.1: Sueldos anuales, mensuales y de 5 meses.

En principio el producto no está pensado venderlo, la intención es que sea de código libre y gratuito. Los potenciales clientes serían las Universidades que usen Moodle, por lo que si podríamos cobrar por crear versiones específicas de la aplicación para cada una de las Universidades. Otra forma de ingresos sería a partir de donaciones de la comunidad.

Viabilidad legal

Se mantiene la licencia de MIT al ser una de las más permisivas para productos de código libre. Permite a la gente hacer casi todo lo que quieran con el proyecto, incluyendo crear y distribuir versiones de código cerrado.

En el desarrollo del proyecto se han empleado varias librerías externas, algunas de ellas nuevas y otras que estaban en el anterior proyecto se han quitado en este desarrollo.

En definitiva, se han usado varias librerías de Java, JavaScript y un paquete de iconos.

Dependencias de Java

En la siguiente tabla se muestran las dependencias de Java junto con su versión y licencia.

Dependencia	Versión	Licencia
Apache Commons CSV	1.6	Apache-2.0
Apache Commons Math	3.6.1	Apache-2.0
Apache Log4j	1.2.17	Apache-2.0
Gson	2.8.5	Apache-2.0
JSON In Java	20180813	MIT
JSoup Java HTML Parser	1.11.3	MIT
SLF4J API Module	1.7.26	MIT
SLF4J LOG4J 12 Binding	1.7.26	MIT
ThreeTen Extra	1.5.0	BSD 3-Clause

Tabla A.2: Dependencias de Java.

Dependencias de JavaScript

En esta otra tabla se muestra las dependencias de JavaScript. Cabe destacar que Google Charts no es de código libre, no permite descargar y usarlo en un fichero local sin conexión [2].

Dependencia	Versión	Licencia
Chart.js	2.8.0	MIT
color-hash	1.0.3	MIT
Google Charts	Desconocida	Privada
html2canvas	0.4.1	MIT

Tabla A.3: Dependencias de JavaScript.

Paquete de iconos

Se ha usado el paquete de iconos Font Awesome [1] que también se emplea en Moodle.

Los iconos de Font Awesome están sacados de **fa2png** versión **4.7.0** y licencia **SIL OFL 1.1**.

Apéndice B

Especificación de Requisitos

- B.1. Introducción
- B.2. Objetivos generales
- B.3. Catalogo de requisitos
- B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

E.1. Introducción

En este manual se explica los requerimientos de la aplicación, su ejecución en el ordenador e instrucciones de cómo usarlo correctamente.

E.2. Requisitos de usuarios

- Tener instalado **Java 8**.
- Obtener el jar ejecutable de la última versión disponible en:
<https://github.com/yjx0003/UBUMonitor/releases/latest>
- Debe disponer conexión a Internet.
- Una cuenta de Moodle con acceso a asignaturas con rol de Profesor.

E.3. Instalación

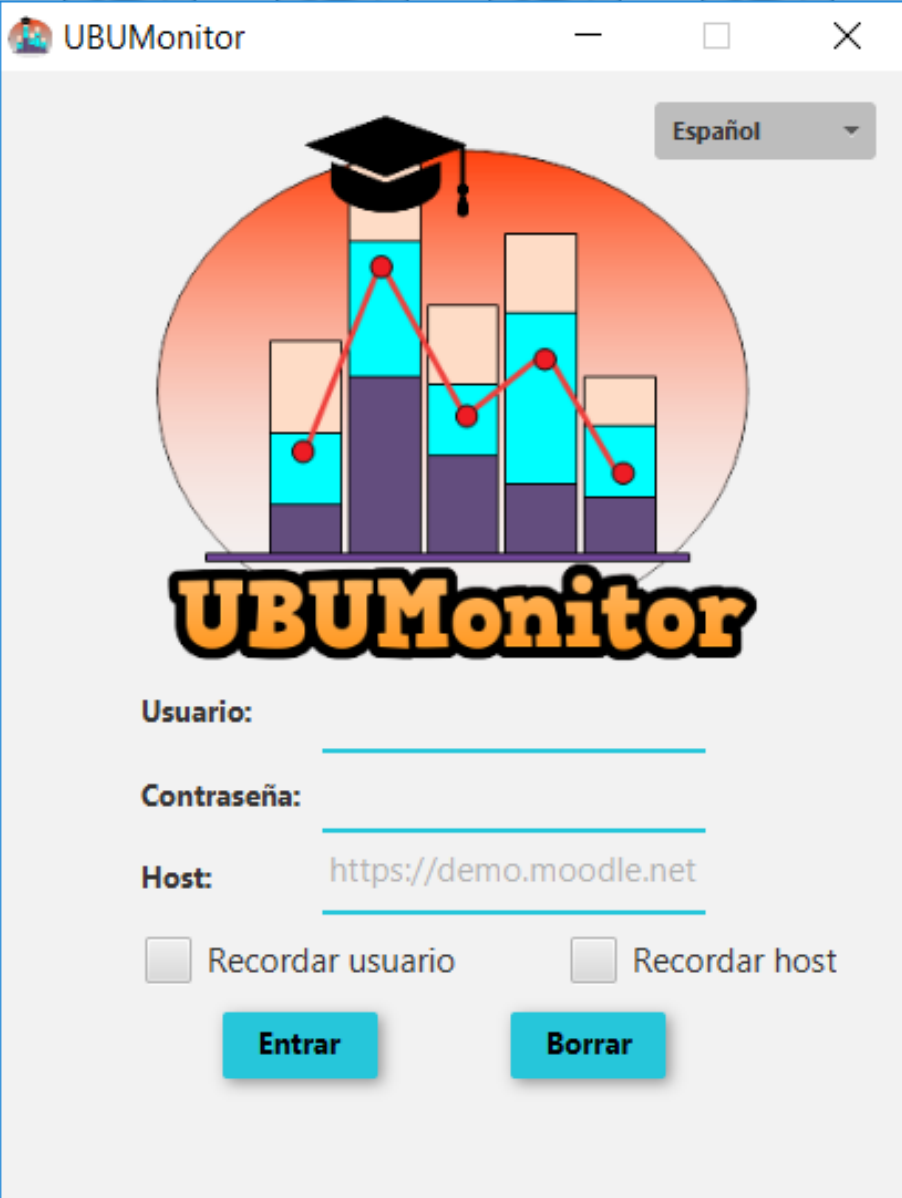
La aplicación no necesita instalar nada más a parte de poseer Java 8.

E.4. Manual del usuario

Después de ejecutar el jar previamente descargado, inicialmente nos mostrará una pantalla de inicio.

Inicio de sesión

En esta primera pantalla es la de inicio de sesión. Ahí introducimos las credenciales de usuario, contraseña y en el campos host la URL de la página web de la plataforma Moodle.



The screenshot shows the UBUMonitor login window. At the top left is the UBUMonitor logo and title. At the top right is a language dropdown menu set to 'Español'. The main graphic features a graduation cap above a bar chart with a red line graph overlaid. Below the graphic is the 'UBUMonitor' logo in large, stylized orange letters. The login form includes three input fields: 'Usuario:', 'Contraseña:', and 'Host:'. The 'Host:' field contains the text 'https://demo.moodle.net'. Below the fields are two checkboxes: 'Recordar usuario' and 'Recordar host', both of which are unchecked. At the bottom are two buttons: 'Entrar' (blue) and 'Borrar' (blue).

Figura E.1: Primera pantalla de inicio de sesión.

Da la opción de recordar el usuario y host para los siguientes usos de la

aplicación. Esa información se guarda en un fichero **config.properties** si marcamos una o ambas opciones.

En la esquina superior derecha nos permite cambiar el idioma de la aplicación y el botón **Borrar** para eliminar el texto de los tres campos.

Después de introducir las credenciales pulsamos el botón de **Entrar**. Si los datos son válidos cargará la siguiente pantalla y si no mostrará un mensaje de error.

Elección del curso

En esta pantalla muestra un listado de cursos que está matriculado el usuario y permite seleccionar uno de ellos y acceder a la siguiente pantalla. Cuando se elige uno de los curso cambia la información de fecha de la última actualización en local. Esta fecha se muestra en función del fichero local que este guardado en el directorio **cache** en caso contrario indica el texto **Nunca**.

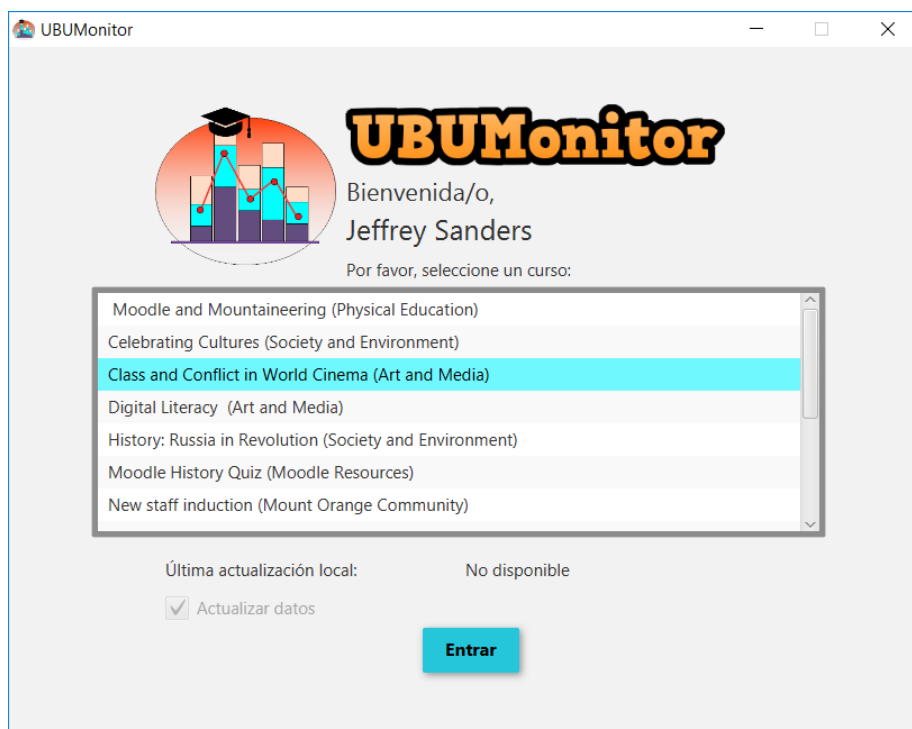


Figura E.2: Pantalla de selección de curso

Bibliografía

- [1] *Font Awesome*. URL: <https://fontawesome.com/> (visitado 23-06-2019).
- [2] *Frequently Asked Questions / Charts*. en. URL: <https://developers.google.com/chart/interactive/faq?csw=1#offline> (visitado 23-06-2019).
- [3] *java.net (Java Platform SE 8)*. URL: <https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html> (visitado 22-06-2019).
- [4] *jsoup Java HTML Parser, with best of DOM, CSS, and jquery*. URL: <https://jsoup.org/> (visitado 22-06-2019).
- [5] Ediciones El País. *Calculadora sueldo neto*. es. URL: <https://cincodias.elpais.com/herramientas/calculadora-sueldo-neto/> (visitado 23-06-2019).
- [6] Margaret Rouse. *What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com*. en. Marzo de 2017. URL: <https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard> (visitado 22-06-2019).
- [7] *Salarios para empleos de Programador/a junior en España / Indeed.es*. Junio de 2019. URL: <https://www.indeed.es/salaries/Programador/a-junior-Salaries?period=yearly> (visitado 23-06-2019).
- [8] *Schneier on Security: The Blowfish Encryption Algorithm*. URL: <https://www.schneier.com/academic/blowfish/> (visitado 18-06-2019).