



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 — 23 de junio  
de 2019

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



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>IV</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	4
<b>Apéndice B Especificación de Requisitos</b>	<b>5</b>
B.1. Introducción . . . . .	5
B.2. Objetivos generales . . . . .	5
B.3. Catalogo de requisitos . . . . .	5
B.4. Especificación de requisitos . . . . .	5
<b>Apéndice C Especificación de diseño</b>	<b>7</b>
C.1. Introducción . . . . .	7
C.2. Diseño de datos . . . . .	7
C.3. Diseño procedimental . . . . .	7
C.4. Diseño arquitectónico . . . . .	7
<b>Apéndice D Documentación técnica de programación</b>	<b>9</b>
D.1. Introducción . . . . .	9
D.2. Estructura de directorios . . . . .	9
D.3. Manual del programador . . . . .	9

D.4. Compilación, instalación y ejecución del proyecto . . . . .	9
D.5. Pruebas del sistema . . . . .	9
<b>Apéndice E Documentación de usuario</b>	<b>11</b>
E.1. Introducción . . . . .	11
E.2. Requisitos de usuarios . . . . .	11
E.3. Instalación . . . . .	11
E.4. Manual del usuario . . . . .	11
<b>Bibliografía</b>	<b>13</b>

---

## Índice de figuras

---

---

## Índice de tablas

---

## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este apartado se comentará

### 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** [4] 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) [3] 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** [1] para realizar las conexiones al no ser una de dependencia externa. Sin embargo los resultados fueron infructuosos y finalmente se empleó Jsoup [2], 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.

Por otra parte, se refactoriza códigos de anteriores *sprints*.

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

Los objetivos son arreglar los errores de programación encontrados. También se 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.

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

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

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

Empezar la documentación de la memoria. Probar si se puede añadir grafica en funcion de componente y evento del registro

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

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

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

### **A.3. Estudio de viabilidad**

**Viabilidad económica**

**Viabilidad legal**

## *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
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario



---

## Bibliografía

---

- [1] *java.net (Java Platform SE 8 )*. URL: <https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html> (visitado 22-06-2019).
- [2] *jsoup Java HTML Parser, with best of DOM, CSS, and jquery*. URL: <https://jsoup.org/> (visitado 22-06-2019).
- [3] 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).
- [4] *Schneier on Security: The Blowfish Encryption Algorithm*. URL: <https://www.schneier.com/academic/blowfish/> (visitado 18-06-2019).