



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**UBULog-1.0  
Documentación Técnica**



Presentado por Oscar Fernández Armengol  
en Universidad de Burgos — 12 de diciembre  
de 2017

Tutor: Raúl Marticorena Sanchez



---

# 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 . . . . .                    | 10         |
| <b>Apéndice B Especificación de Requisitos</b>          | <b>11</b>  |
| B.1. Introducción . . . . .                             | 11         |
| B.2. Objetivos generales . . . . .                      | 11         |
| B.3. Catalogo de requisitos . . . . .                   | 11         |
| B.4. Especificación de requisitos . . . . .             | 11         |
| <b>Apéndice C Especificación de diseño</b>              | <b>13</b>  |
| C.1. Introducción . . . . .                             | 13         |
| C.2. Diseño de datos . . . . .                          | 13         |
| C.3. Diseño procedimental . . . . .                     | 13         |
| C.4. Diseño arquitectónico . . . . .                    | 13         |
| <b>Apéndice D Documentación técnica de programación</b> | <b>15</b>  |
| D.1. Introducción . . . . .                             | 15         |
| D.2. Estructura de directorios . . . . .                | 15         |
| D.3. Manual del programador . . . . .                   | 15         |

|  |           |
|--|-----------|
| D.4. Compilación, instalación y ejecución del proyecto . . . . . | 15        |
| D.5. Pruebas del sistema . . . . .                               | 15        |
| <b>Apéndice E Documentación de usuario</b>                       | <b>17</b> |
| E.1. Introducción . . . . .                                      | 17        |
| E.2. Requisitos de usuarios . . . . .                            | 17        |
| E.3. Instalación . . . . .                                       | 17        |
| E.4. Manual del usuario . . . . .                                | 17        |
| <b>Bibliografía</b>  | <b>19</b> |

---

## Índice de figuras

---

|                        |    |
|------------------------|----|
| A.1. Sprint 1. . . . . | 3  |
| A.2. Sprint 2. . . . . | 4  |
| A.3. Sprint 3. . . . . | 5  |
| A.4. Sprint 4. . . . . | 6  |
| A.5. Sprint 5. . . . . | 7  |
| A.6. Sprint 6. . . . . | 8  |
| A.7. Sprint 7. . . . . | 9  |
| A.8. Sprint 8. . . . . | 10 |

---

## Índice de tablas

---

## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este documento hablaremos de la planificación del proyecto, que es una parte muy importante. Aquí analizaremos el tiempo de desarrollo y el presupuesto que se necesitara para llevarlo a cabo.

lo dividiremos en dos partes:

- Planificación temporal del proyecto.
- Estudio de viabilidad del proyecto.

En la planificación temporal hablaremos de cada uno de los *sprints* necesarios para su realización, el tiempo estimado y el tiempo real de realización.

En el estudio de viabilidad hablaremos de los costes y beneficios que nos aporta esta aplicación y su viabilidad legal, que se refiere a las licencias de uso del código ajeno que hemos utilizado en el desarrollo para poder comercializar nuestra aplicación.

### A.2. Planificación temporal

Al inicio del proyecto se propuso utilizar una metodología ágil, en concreto Scrum, ya que se decidió tener reuniones semanales para hablar de los cambios realizados, problemas ocasionados y planificación del siguiente Sprint que se realizara. No se ha conseguido desarrollar la metodología al 100 % ya que el

el equipo de desarrollo constaba de 1 persona (Oscar Fernández Armengol), pero desechando este punto, se ha conseguido un desarrollo ágil en sus demás puntos.

- Se aplicó una estrategia de desarrollo incremental a través de iteraciones (*sprints*) y revisiones.
- La duración media de los *sprints* fue de una semana.
- Al finalizar cada *sprint* se entregaba un producto funcional con la nueva especificación en el caso de que estuviera terminada.
- Se realizaban reuniones de revisión al finalizar cada *sprint*, de resolución de dudas y al mismo tiempo de planificación del nuevo *sprint*.
- En la planificación del *sprint* se generaba una lista de tareas a realizar (nuevas funcionalidades o bugs a solucionar).
- Se estimaba el tiempo de realización de las tareas a realizar en el *canvas*.
- Para monitorizar el progreso del proyecto se utilizan los gráficos generados en github.

### **Sprint 1 (03/10/17 - 10/10/17)**

Este *sprint* fue el comienzo del proyecto, aunque en reuniones previas se hablo con el tutor de las propuestas que tenia para la elección del proyecto, una vez el tutor (Raúl Marticorena Sanchez) acepto tutorizar al alumno (Oscar Fernández Armengol) se puedo empezar el desarrollo.

Los objetivos fueron:

- Preparación del entorno de desarrollo.
- Familiarización con la aplicación heredada **UBUGrades**.
- Investigación del web service de moodle
- Creación de un esqueleto del proyecto para poder empezar a trabajar.

El **Sprint 1** se estimo en 6 días de trabajo y se realizo en esos 6 días.



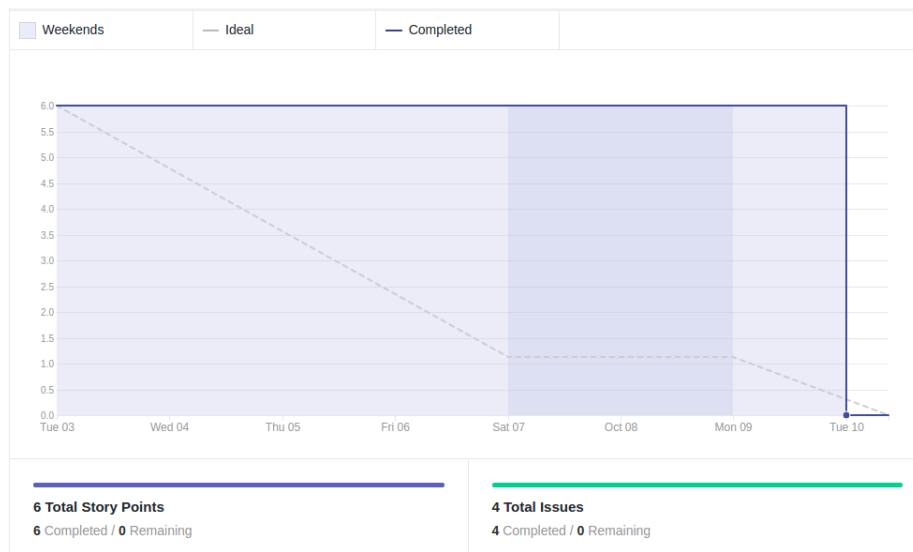


Figura A.1: Sprint 1.

## Sprint 2 (10/10/17 - 17/10/17)

Los objetivos fueron:

- Análisis de modelo de datos.
- Diagrama del modelo de datos.
- Parseo de documentos csv.

El **Sprint 2** se estimo en 6 días de trabajo y se realizo en esos 6 días.

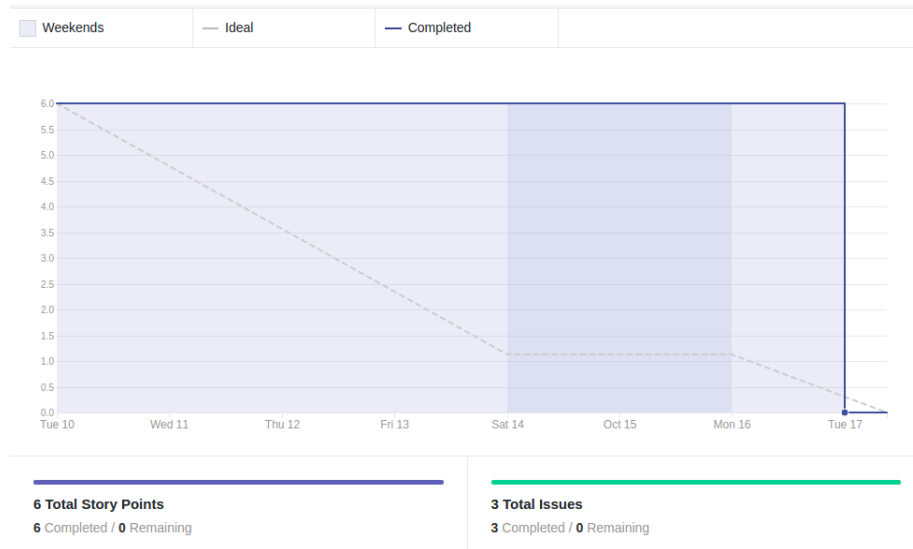


Figura A.2: Sprint 2.

### Sprint 3 (17/10/17 - 24/10/17)

Los objetivos fueron:

- Integración continua con travis CI.
- Integración de sonarqube.
- Recoger token con web service.
- Recoger las asignaturas en las que imparte como profesor el usuario logeado.
- Referencia al usuario con su id en el parseo.
- Interfaz de usuario, primeros pasos.
- Limpieza del carpetas innecesarias en el repositorio.

En este sprint, cabe destacar, la integración de Travis<sup>[4]</sup> y SonarQu-  
be <sup>[2]</sup> para optimizar el tiempo de desarrollo, ya que delegamos en estas  
herramientas las pruebas y el análisis del código.

El **Sprint 3** se estimo en 7 días de trabajo y se realizo en esos 6 días.

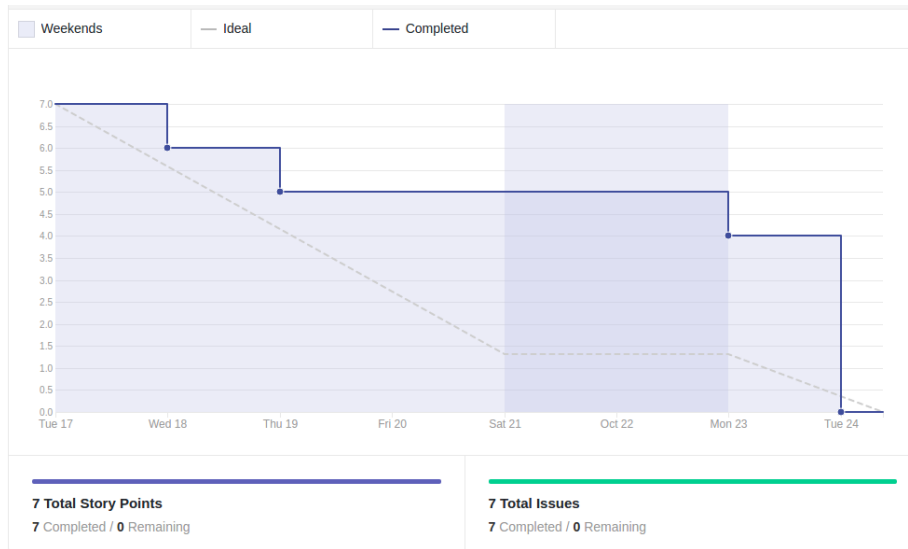


Figura A.3: Sprint 3.

## Sprint 4 (24/10/17 - 7/11/17)

Los objetivos fueron:

- Creación de usuarios ficticios.
- Análisis de datos.
- Eliminación de calificaciones.
- Gestión de errores del documento.

En este sprint, cabe destacar, la creación de usuarios ficticios, ya que hay usuarios que tienen interacciones en las asignaturas y nos interesa saber que están haciendo.

El **Sprint 4** se estimo en 5 días de trabajo y se realizo en esos 6 días.

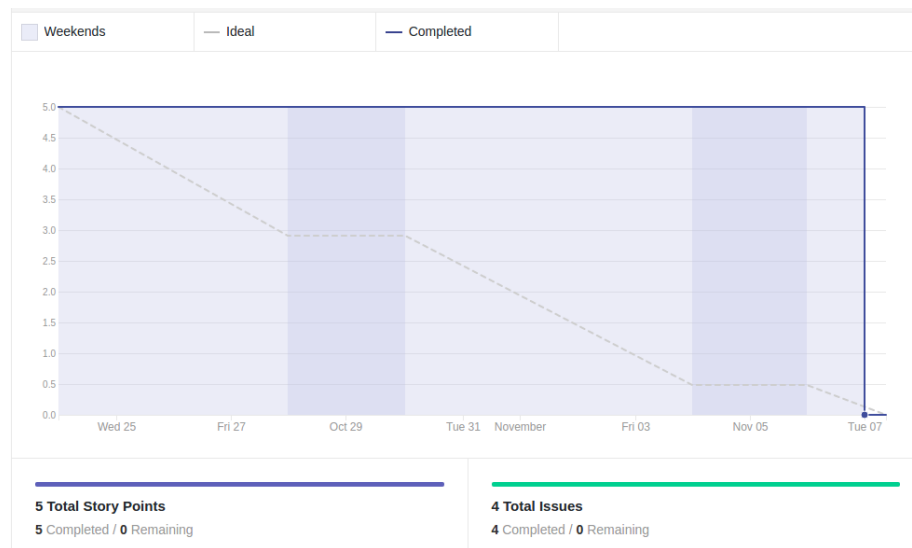


Figura A.4: Sprint 4.

## Sprint 5 (7/11/17 - 14/11/17)

Los objetivos fueron:

- Cambio de árbol de actividades.
- Construir los filtros de la interfaz.
- Creación de eventos
- Interfaz nuevo diseño.

En este sprint, cabe destacar, el nuevo diseño para la ventana principal, la antigua no era re-dimensionable y como había que implementar nuevos filtros y botones mas adelante, se opto por rehacerla.

El **Sprint 5** se estimo en 7 días de trabajo y se realizo en esos 6 días.

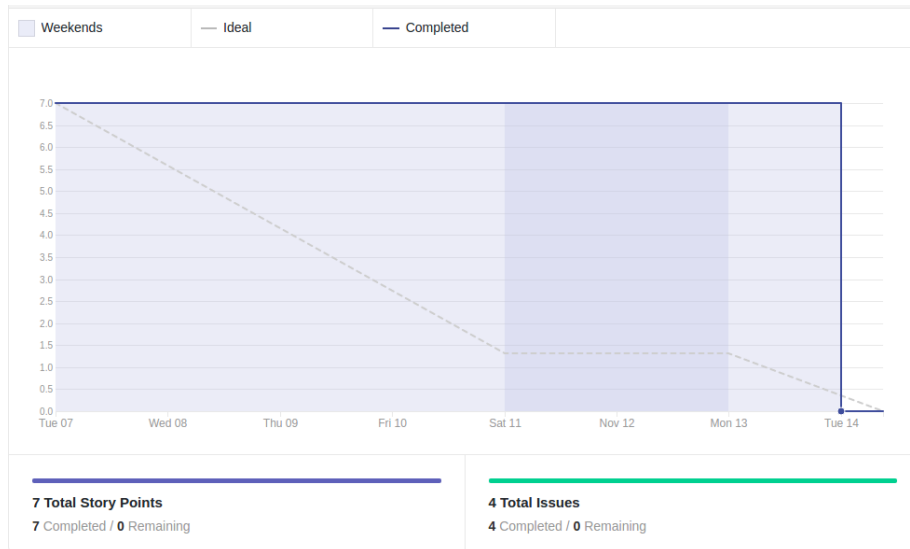


Figura A.5: Sprint 5.

## Sprint 6 (14/11/17 - 22/11/17)

Los objetivos fueron:

- Generar gráficas.
- Mostrar gráficas con char.js [3]
- Creación de botón para generar las gráficas.
- Instalar sublime text para la edición de html y javascript.
- Contar los logs para las gráficas.

En este sprint, cabe destacar, la utilización de la librería chart para la generación de gráficos. Como el gráfico es cambiante dependiendo de los filtros pulsados, optamos en generar los html y javascript en tiempo de ejecución.

El **Sprint 6** se estimo en 8 días de trabajo y se realizo en esos 7 días.



Figura A.6: Sprint 6.

## Sprint 7 (22/11/17 - 29/11/17)

Los objetivos fueron:

- Integración de gráficas.
- Ordenar selección de eventos.
- imagen de usuario logeado.
- UTF-8 para el gráfico.

El **Sprint 7** se estimo en 8 días de trabajo y se realizo en esos 5 días.

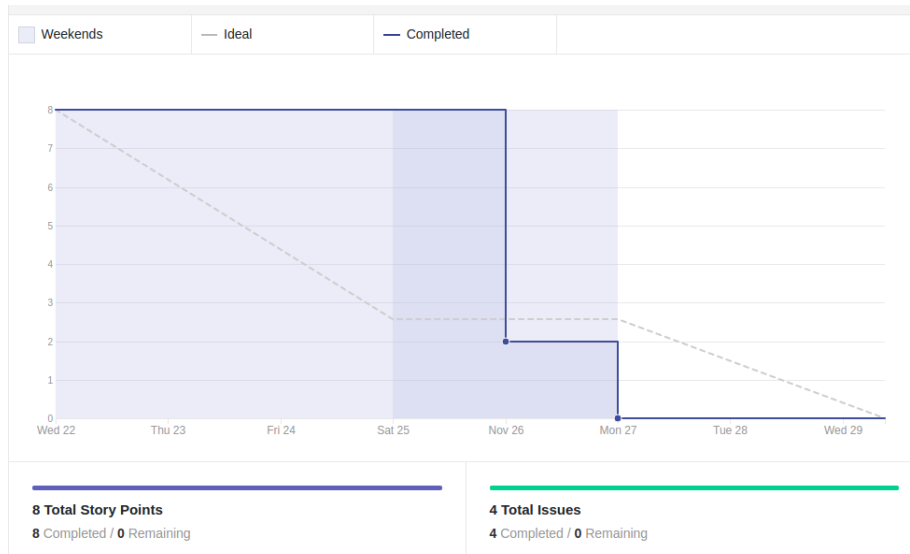


Figura A.7: Sprint 7.

## Sprint 8 (28/11/17 - 06/12/17)

Los objetivos fueron:

- Cambio de parseo de log.
- Creación de tablas para los log en JavaScript.
- Muestreo de tabla en aplicación.
- Implementación de filtros en tabla
- Revisión de logs.
- Cambio de logo.

En este sprint, cabe destacar, el cambio en el parseo, dado que había incongruencias entre los log de pruebas y los reales, para evitar eso, se implementa con la librería `common-csv` [1], con ella cogemos los datos correspondientes a la columna concreta y nos evitamos que en las pruebas el usuario sea `.alumno apellido1 apellido2z` en producción sea `.apellido1 apellido2, alumno`.

Por otra parte, también es interesante la implementación de las tablas, donde el usuario de la aplicación puede ver los log filtrados y desgranar aun mas los datos con filtros adicionales.

El **Sprint 8** se estimo en 10 días de trabajo y se realizo en esos 6 días.



Figura A.8: Sprint 8.

## Sprint 9 (06/12/17 - 13/12/17)

El **Sprint 9**

### 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] Commons CSV – Home.
- [2] GII-17.1B-UBULog-1.0 - trona85.
- [3] Libreria para generar los graficos - Chart.js · Cocumentación.
- [4] Travis CI - Integración continua.