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 — 27 de diciembre de 2017
Tutor: Raúl Marticorena Sanchez

Índice general

Índice de figuras

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.

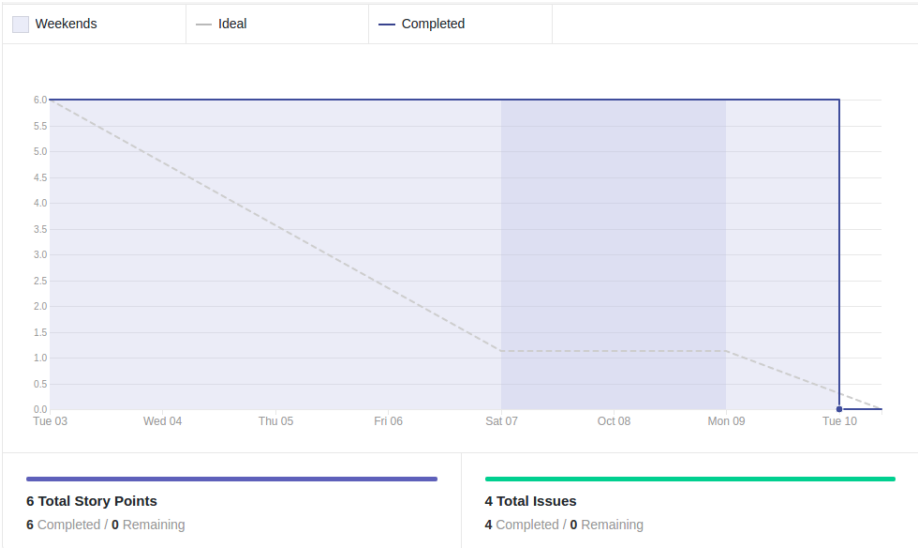

Figura A.1: Sprint 1.

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.

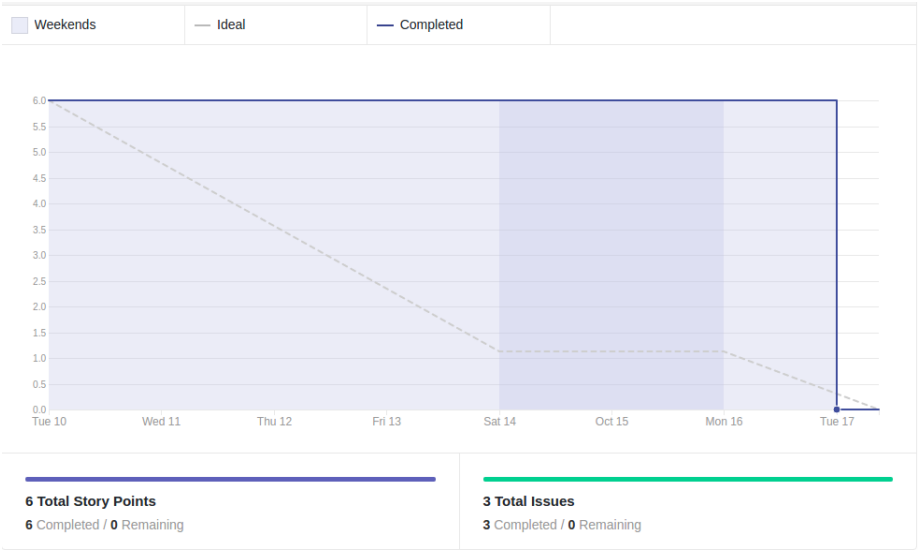

Figura A.2: Sprint 2.

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[4] y SonarQube [2] 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.

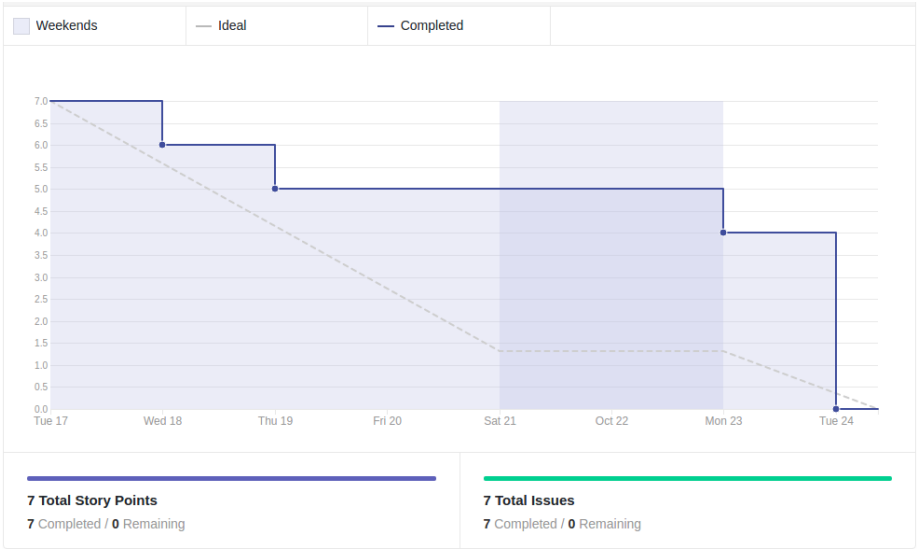

Figura A.3: Sprint 3.

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.

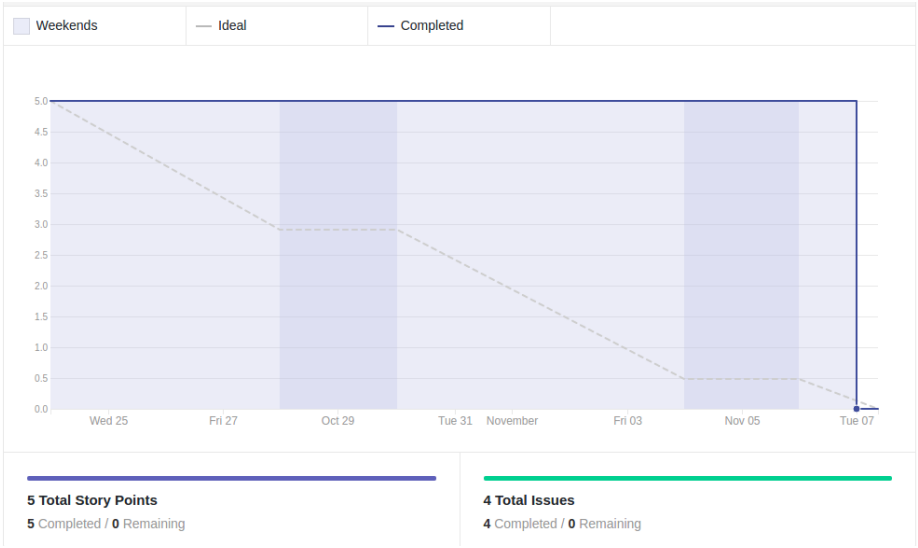

Figura A.4: Sprint 4.

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.

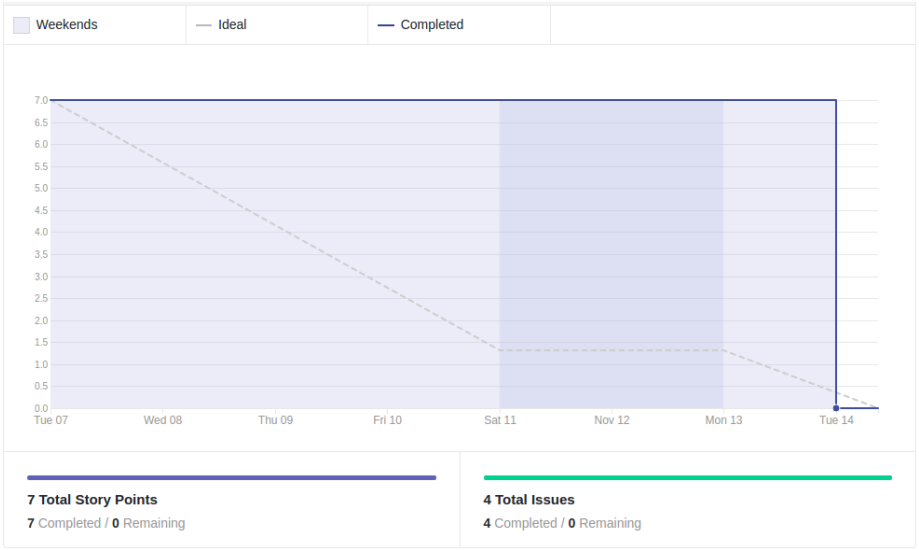

Figura A.5: Sprint 5.

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.

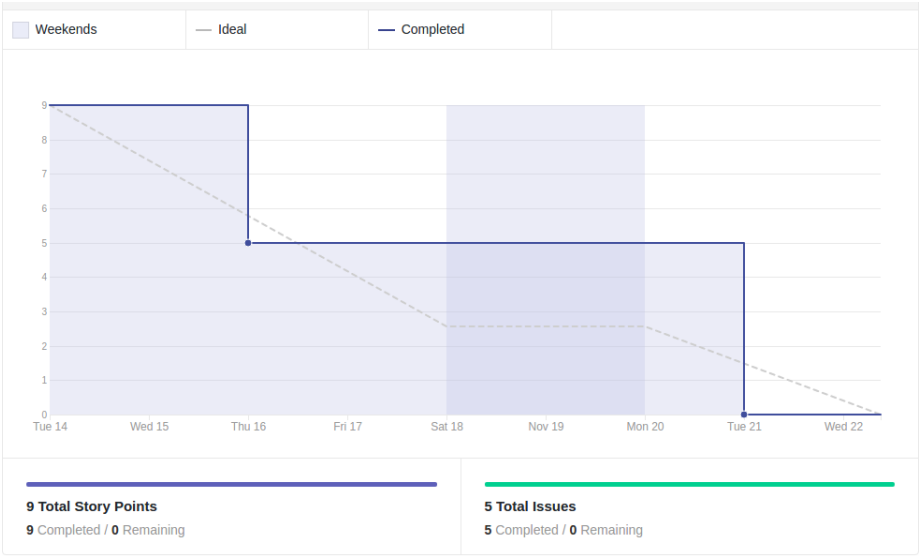

Figura A.6: Sprint 6.

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.

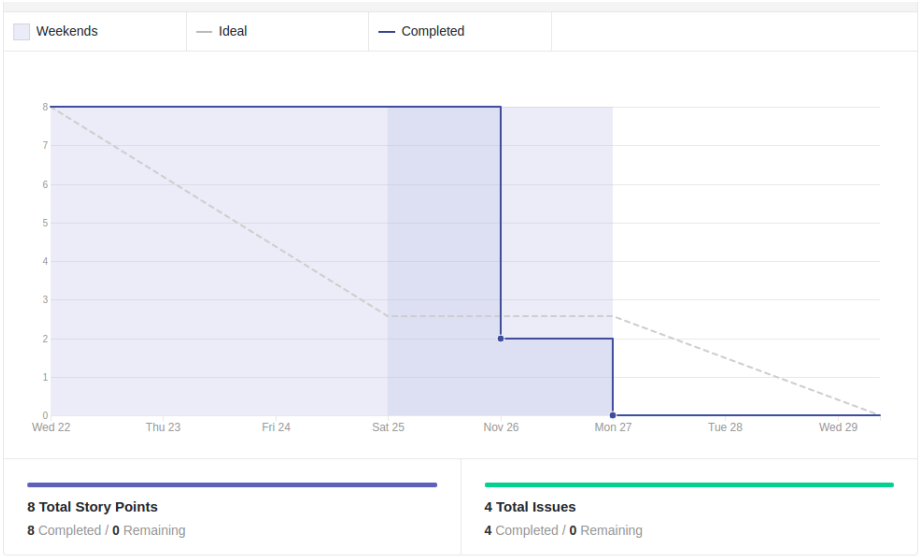

Figura A.7: Sprint 7.

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.

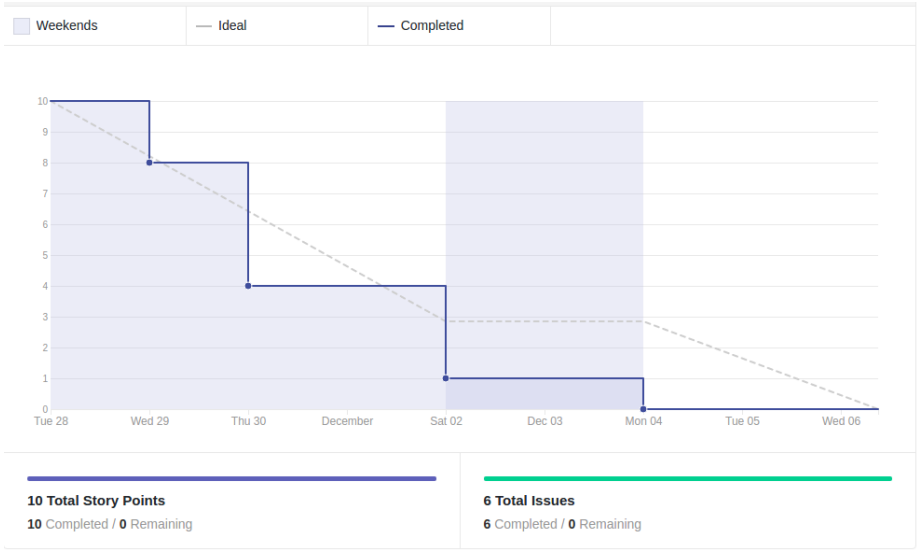

Figura A.8: Sprint 8.

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

Los objetivos fueron:

- Sacar filtros del html a nuestra interfaz.
- Implementar funcionalidad filtros de tabla de logs y cargar la nueva gráfica con los log resultantes.
- Investigar la utilización de web scraping para nuestra aplicación.
- Crear botones en la interfaz para coger el csv de forma automática.
- Implementar funcionalidad de botón documento online.

En este sprint, cabe destacar, la investigación del web scraping, se estimo que se haría en 5 días mas 3 dias de implementación, en una mañana, se pudieron probar las librerías Jsoup, Selenium y htmlUnit. Las dos primeras se descartaron, Selenium por problemas de ubicación de exploradores porque no podemos saber en donde el usuario los tendrá instalados y lo descartamos y Jsoup solo parsea la web, no nos deja interaccionar con ella. Con la librería htmlUnt hicimos pruebas y conseguimos traernos los datos correspondientes, en la misma mañana pudimos hacer el trabajo que esperábamos hacer en 8 días.

El Sprint 9 se estimo en 15 días de trabajo y se realizo en esos 3 días.

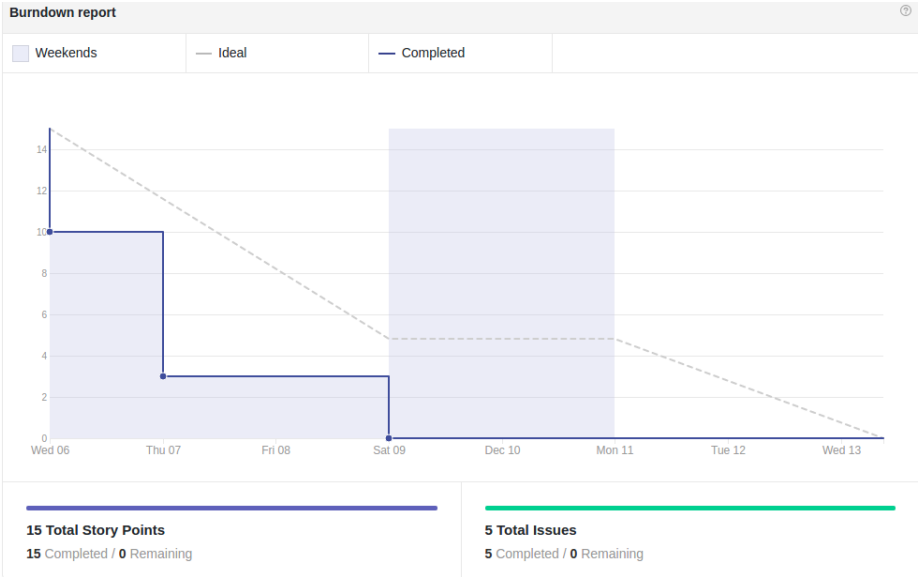

Figura A.9: Sprint 9.

Sprint 10 (13/12/17 - 20/12/17)

Los objetivos fueron:

- WebScripting funcional y refactorizado.
- Mostrar fecha y hora en la tabla de logs.
- Centrar botones inferiores.
- Descartar meses generados por la gráfica que no tengan logs.
- Modal de carga para la lectura de logs.
- Añadir tipos de gráficas.
- Arreglo de la muestra de fechas de la tabla log.
- Aumentar TimeOut de WebScraping.

El Sprint 10 se estimo en 12 días de trabajo y se realizo en esos 5 días.

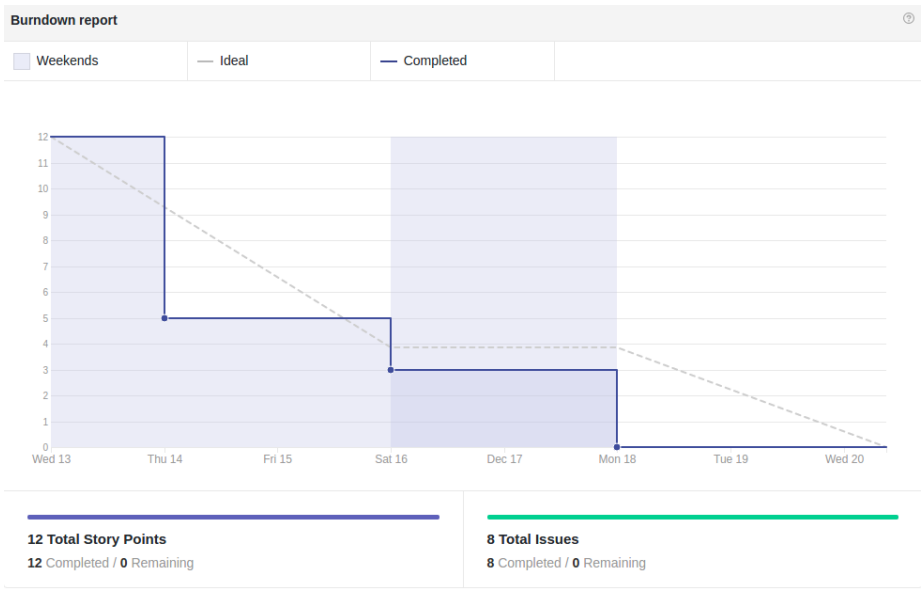

Figura A.10: Sprint 10.

Sprint 11 (20/12/17 - 12/01/18)

Los objetivos fueron:

- Documentación de código.
- Documentación de memoria.
- Refactorización.
- Arreglo de bugs para la entrega.

Este es el ultimo spring del proyecto, y el mas largo, se completara la memoria, anexos, documentación de código y se arreglaran bugs que se han observado durante el desarrollo y el testeo manual.

El Sprint 10

FALTA IMASLKEFÑSJFDÑASJF

A.3. Estudio de viabilidad

Podemos desglosar los costes del proyecto de la siguiente manera.

Concepto	Coste
Salario mensual bruto	1.300
IRPF (10 %)	130
S.S(29.9 %)	388.88
Salario programador	1.688.88
Total 3 meses y medio(20H/S)	2955.54

Tabla A.1: Coste personal

Concepto	Coste	Coste amortizado
Portatil	1.000	4.86
Total	1000	4.86

Tabla A.2: Coste hardware

Viabilidad económica

En el desglose podemos diferenciarlo en las siguientes categorías.

Costes de personal

El proyecto se lleva a cabo por un programador a media jornada ya que esta de practicas en otra empresa 25H/S, empezando en Octubre del 2017 asta Enero del 2018.

Coste hardware

El proyecto se realiza con el ordenador personal y se considera la amortización a 5 años y se utiliza durante 3 meses y medio.

Beneficios

El proyecto se ha pensado para ayudar al profesorado a ver la interacción de los alumnos en la asignatura. Se puede considerar un coste de 5 euros por licencia y que en la universidad hay, redondeando, 700 profesores estamos hablando de 3500 euros, la aplicación en un futuro cambiara y llegara a la 2.0 con lo cual habría que volver a pagar la nueva licencia.

Herramientas	Licencia
commons-codec1.9	Apache License Version 2.0
commons-csv-1.5	Apache License Version 2.0
commons-logging-1.2	Apache License Version 2.0
gson-2.8.0	Apache License Version 2.0
htmlunit-2.28-OSGi	Apache License Version 2.0
httpasyncclient-4.1.2	Apache License Version 2.0
httpasyncclient-cache-4.1.2	Apache License Version 2.0
httpclient-4.5.2	Apache License Version 2.0
httpclient-cache-4.5.2	Apache License Version 2.0
httpcore-4.4.5	Apache License Version 2.0
httpcore-nio-4.4.5	Apache License Version 2.0
java-json	Json license
log4j-1.2.17	Apache License Version 2.0
slf4j-api-1.7.1	Apache License Version 2.0
httpclient-cache-4.5.2	Apache License Version 2.0
slf4j-log4j12-1.7.1	Apache License Version 2.0
UBULog 1.0	Eclipse Public License - v 1.0

Tabla A.3: Viabilidad legal

La aplicación se puede extender a diferentes centros aparte de la UBU. Si lo observamos en ese sentido se podría cobrear 500 euros por licencia, para su utilización en esa universidad. Si tenemos 83 universidades es España, podríamos obtener unos beneficios de 83 * 500 = 41.500 euros.

Viabilidad legal

En esta sección hablaremos de la viabilidad del software utilizado, de si las licencias nos permiten la explotación del mismo o su propia utilización.

Para la aplicación desarrollada y para lo que esta pensado, por las licencias que se están utilizando no hay ningún problema para su utilización, mientras compartamos todo el código utilizado.

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.