



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

Data-WareHouse
Documentación Técnica



Presentado por Mario de la Parte Izquierdo
en Universidad de Burgos — 7 de abril
de 2019

Tutor: 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	2
A.3. Estudio de viabilidad	3
Apéndice B Especificación de Requisitos	5
B.1. Introducción	5
B.2. Objetivos generales	5
B.3. Catalogo de requisitos	5
B.4. Especificación de requisitos	7
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

Índice de figuras

Índice de tablas

B.1. Caso de uso 1: Importar Datos	8
B.2. Caso de uso 2: Gráfico apilado de Asignaturas por Curso	9
B.3. Caso de uso 3: Gráfico de máximos, mínimos y medias por curso	10

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Una de las fases más destacadas e imprescindible de un proyecto es la planificación. En esta fase se fijan los requisitos y se estima el tiempo y dinero que va a suponer la realización del proyecto. Para esto, es necesario tener una idea global y a la vez concisa del proyecto que se va a realizar; de manera que ambas partes que forman el proyecto estén de acuerdo.

Así pues, vamos a dividir esta fase en dos apartados:

- **Planificación temporal:** en este primer apartado se realizará una estimación de los tiempos esperados, así como fijar la fecha de inicio y fecha de fin del proyecto.
- **Estudio de viabilidad:** en este apartado se realizará un estudio de la viabilidad, es decir, ser capaces de apreciar si el proyecto ha sido exitoso o por el contrario ha sido un fracaso.

Dentro de este último apartado se diferenciarán dos subapartados:

- **Viabilidad económica:** se estimarán los costes y beneficios del proyecto.
- **Viabilidad legal:** se estudiarán las regulaciones legales que pudieran afectar al proyecto.

A.2. Planificación temporal

Para realizar una planificación correcta del proyecto, se decidió utilizar una metodología ágil de desarrollo, para ello se utilizó la metodología *Scrum*. Como se explica en la memoria:

- Se ha utilizado una estrategia orientada a un desarrollo incremental y basada en *sprints*.
- La duración media de cada *sprint* era aproximadamente de una semana.
- Al inicio de cada *sprint* se definían las tareas o *issues* a realizar, las cuales tenían que ser realizadas en un cierto intervalo de tiempo.
- Cada *sprint* se planificaba cuando se finalizaban las tareas o *issues* del anterior *sprint*.
- Al final de cada *sprint* se revisan todas las tareas realizadas, así como ver si se han logrado los objetivos fijados y solucionado los problemas encontrados.

A continuación, se van a detallar los *sprints* que se han realizado a lo largo del proyecto:

Sprint 0 (25/02/19-10/03/19)

En la primera reunión de planificación de proyecto se desarrollaron y expusieron las ideas del mismo. Además se propusieron las siguientes tareas:

- **Crear repositorio de HitHub.** Además de crear el repositorio principal, se solicitó y posteriormente se adquirió el *Github Student Developer Pack*.
- **Descargar plantilla para la documentación**
- **Redactar los objetivos del proyecto**
- **Probar a leer archivos Excel con Python**

Se estiman 12 horas de trabajo.

Sprint 1 (11/03/19-17/03/19)

En la segunda reunión se propusieron las siguientes tareas:

- **Crear un algoritmo o proceso capaz de leer los archivos Excel**

Se estiman 20 horas de trabajo.

Sprint 2 (18/03/19-24/03/19)

En la tercera reunión se propusieron las siguientes tareas:

- **Mejorar el algoritmo para leer los archivos Excel:** mejorar la expresión regular para que no fuera tan genérica y fuera más específica.
- **Tareas adicionales:** eliminar apartado *Objetivos personales* de la documentación y descargar *GitHub Desktop* y *gVim 8.1*.

Se estiman 20 horas de trabajo.

Sprint 3 (25/03/19-31/03/19)

Sprint 4 (01/04/19-07/04/19)

Sprint 5 (08/04/19-14/04/19)

Sprint 6

Sprint 7

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este anexo se va a realizar y formalizar la especificación de requisitos que define el comportamiento del sistema desarrollado en el proyecto.

B.2. Objetivos generales

Los objetivos generales que se persiguen en este proyecto son los siguientes:

- Desarrollar una herramienta que permita la extracción, tratamiento y análisis de datos relacionados con la matriculación de alumnos en la Universidad de Burgos (UBU).
- Realizar gráficos y estadísticos que resulten visuales y aporten información valiosa al usuario final.

B.3. Catalogo de requisitos

En este apartado se van a enumerar los requisitos específicos derivados de los objetivos proyecto, divididos en funcionales y no funcionales.

Requisitos funcionales

- **RF-1: Importar datos:** La aplicación debe ser capaz de importar los datos desde ficheros Excel(.xls) cuyo formato y extensión de archivo no coinciden.
 - **RF-1.1** Pasar datos de fichero original (.xls) a un fichero (.csv): se realizará un parseo de datos.
 - **RF-1.2** Pasar datos de fichero (.csv) a la Base de Datos.
- **RF-2: Gráfico apilado de Asignaturas por Curso o Semestre:** La aplicación debe ser capaz de mostrar un gráfico apilado cuyo *eje x* muestre las diferentes asignaturas y su *eje y* muestre la cantidad total de alumnos matriculados en las mismas. Al tratarse de un gráfico apilado, se podrán diferenciar los grupos existentes en las asignaturas(grupo online, grupo presencial 1, grupo presencial 2...etc).
- **RF-3: Gráfico de máximos, mínimos y medias por curso:** La aplicación debe ser capaz de mostrar un gráfico cuyo *eje x* muestre los diferentes cursos(1º, 2º, 3º y 4º) y su *eje y* muestre la cantidad total de alumnos matriculados, indicando los máximos, mínimos y medias por cada curso.

Requisitos no funcionales

- **RNF-1: Usabilidad:** La herramienta debe ser intuitiva y fácil de utilizar, así como tener una estructura clara y contar con una interfaz amigable.
- **RNF-2: Escalabilidad:** La herramienta debe permitir la incorporación de nuevas funcionalidades o nuevos módulos.
- **RNF-3: Rendimiento:** La herramienta debe funcionar de forma fluida sin que la interfaz gráfica se quede bloqueada.
- **RNF-4: Fiabilidad:** La herramienta debe ser segura y debe funcionar correctamente bajo determinadas condiciones.
- **RNF-4: Integridad:** La herramienta debe cumplir con la integración de datos y no tener pérdidas de información, así como modificación de la misma.

B.4. Especificación de requisitos

En este apartado se va a visualizar a través de diagramas los casos de uso de los requisitos funcionales previamente denidos. Para esto, se va a utilizar la notación UML.

Actores

El actor del sistema es la persona que maneja la aplicación.

Casos de uso

Caso de uso 1: Importar Datos	
Descripción	La aplicación debe ser capaz de importar los datos desde ficheros Excel(.xls) cuyo formato y extensión de archivo no coinciden. Por lo tanto, estos ficheros no se podrán abrir con librerías de Python, teniendo que realizar un parseo previo.
Requisitos	RF-1
Precondiciones	Existe un fichero Excel(.xls) con datos para importar. Dicho fichero se ha descargado de <i>Sigma</i> y se encuentra en una carpeta local.
Secuencia normal	Paso Acción
	1 El usuario pulsa el botón de cargar archivos.
	2 Se muestra un desplegable para poder navegar por el directorio y seleccionar los ficheros disponibles. Los ficheros pueden ser (.xls) o (.csv).
	2' Si se selecciona un fichero (.csv) se salta al paso 4.
	3 Se crea un fichero (.csv) a partir del fichero(.xls) para poder tratar la información.
	4 Se lee el archivo (.csv) y se guardan los datos en la Base de Datos.
Postcondiciones	Se crea un fichero (.csv) nuevo y los datos se cargan en la Base de Datos.
Excepciones	Si los datos se encontraran en un fichero (.csv), no sería necesario el paso 3.
Importancia	Alta
Urgencia	Alta
Comentarios	Los ficheros Excel no cumplen el estándar(al abrirlos muestra un error de formato y extensión) y es necesario hacer un parseo previo.

Tabla B.1: Caso de uso 1: Importar Datos

Caso de uso 2: Gráfico apilado de Asignaturas por Curso	
Descripción	La aplicación debe ser capaz de mostrar un gráfico apilado cuyo <i>eje x</i> muestre las diferentes asignaturas y su <i>eje y</i> muestre la cantidad total de alumnos matriculados en las mismas. Al tratarse de un gráfico apilado, se podrán diferenciar los grupos existentes en las asignaturas(grupo online, grupo presencial 1, grupo presencial 2...etc).
Requisitos	RF-2
Precondiciones	Los datos o información necesaria se ha cargado correctamente en la Base de Datos.
Secuencia normal	Paso Acción
	1 Se selecciona si se desea grado o máster.
	2 Se selecciona el curso(1º, 2º, 3º, 4º) que se desea obtener el gráfico apilado si se trata de un grado.
	2' Si se trata de un máster de un único curso académico, se salta el paso anterior (paso 2).
	3 Se cargan los datos necesarios de la Base de Datos.
	4 Se realiza el gráfico apilado de asignaturas por curso.
	5 Una vez visualizado el gráfico, se muestra una opción de guardar el gráfico y otra de salir.
	5' Si se pulsa el botón de guardar el gráfico, el gráfico se guarda en el directorio donde se encuentran los ficheros Excel.
	5'' Si se pulsa el botón de salir volvería al paso 3.
Postcondiciones	La aplicación debe mostrar el gráfico apilado de manera correcta.
Excepciones	
Importancia	Alta
Urgencia	Alta
Comentarios	Con esta gráfica se puede apreciar si existe algún grupo desequilibrado, así como las asignaturas con más matriculados entre otra información destacable.

Tabla B.2: Caso de uso 2: Gráfico apilado de Asignaturas por Curso

Caso de uso 3: Gráfico de máximos, mínimos y medias por curso	
Descripción	La aplicación debe ser capaz de mostrar un gráfico cuyo <i>eje x</i> muestre los diferentes cursos(1º, 2º, 3º y 4º) y su <i>eje y</i> muestre la cantidad total de alumnos matriculados, indicando los máximos, mínimos y medias por cada curso.
Requisitos	RF-3
Precondiciones	Los datos o información necesaria se ha cargado correctamente en la Base de Datos.
Secuencia normal	Paso Acción
	1 Se seleccionan los campos necesarios de la Base de Datos.
	2 Se realiza el gráfico de máximos, mínimos y medias por curso.
Postcondiciones	La aplicación debe mostrar el gráfico de máximos, mínimos y medias por curso de manera correcta.
Excepciones	
Importancia	Alta
Urgencia	Alta
Comentarios	

Tabla B.3: Caso de uso 3: Gráfico de máximos, mínimos y medias por curso

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