



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

**Sistema de Información
sobre Matriculación**

Documentación Técnica



Presentado por Mario de la Parte Izquierdo
en Universidad de Burgos — 3 de julio
de 2019

Tutor: Carlos Pardo Aguilar

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	10
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catalogo de requisitos	13
B.4. Especificación de requisitos	16
Apéndice C Especificación de diseño	25
C.1. Introducción	25
C.2. Diseño de datos	25
C.3. Diseño procedimental	30
C.4. Diseño de interfaces	32
Apéndice D Documentación técnica de programación	37
D.1. Introducción	37
D.2. Estructura de directorios	37
D.3. Manual del programador	38

D.4. Compilación, instalación y ejecución del proyecto	40
Apéndice E Documentación de usuario	43
E.1. Introducción	43
E.2. Requisitos de usuarios	43
E.3. Instalación	43
E.4. Manual del usuario	44
Bibliografía	55

Índice de figuras

A.1. Gráfico con el resumen de las horas empleadas en el proyecto . . .	9
A.2. Gráfico con los <i>commits</i> que se ha realizado cada semana	9
A.3. Gráfico donde se aprecia el desarrollo constante	10
B.1. Diagrama UML con todos los Casos de Uso	16
C.1. Modelo Entidad-Relación(ER)	26
C.2. Tabla ASIGNATURAS	27
C.3. Tabla GRUPOS	28
C.4. Tabla PROFESORES	29
C.5. Diagrama de Navegabilidad	33
C.6. Pantalla principal de la interfaz	34
C.7. Ventana emergente de tipo Warning de la interfaz	35
C.8. Ventana emergente informativa de la interfaz	35
C.9. Textos de ayuda en la selección de archivos	36
C.10. Migas de pan de la ventana secundaria	36
D.1. Anaconda Navigator	39
D.2. Ejecución de <i>Jupyter Notebook</i>	41
D.3. Ejecución de <i>Jupyter Notebook</i>	42
E.1. Botón de creación de la Base de Datos	44
E.2. Mensaje de <i>Warning</i> de BBDD ya creada	45
E.3. Pantalla principal de la aplicación	46
E.4. Ventana del Sistema Operativo (seleccionar .xls)	47
E.5. Resumen de botón <i>Preprocesado</i>	47
E.6. Ventana del Sistema Operativo (seleccionar .csv)	48
E.7. Resumen de botón <i>Cargar Archivos</i>	48

E.8. Ventana secundaria para configurar el gráfico 1	49
E.9. Ventana secundaria para configurar el gráfico 2	50
E.10. Gráfico 1: Gráfico Apilado de Asignaturas de año 2018-2019 de 1º de Grado en Ingeniería Informática de Tipología Teoría	51
E.11. Gráfico 1: Gráfico Apilado de Asignaturas del año 2018-2019 de 1º del Grado en Ingeniería Informática de Tipología Prácticas . .	52
E.12. Gráfico 2: Gráfico Apilado de Máximos, Mínimos y Medias por Curso del año 2018-2019 del Grado en Ingeniería Informática . .	52
E.13. Gráfico 3: Gráfico Apilado de Máximos, Mínimos y Medias por Semestre del año 2018-2019 del Grado en Ingeniería Informática	53
E.14. Opción de <i>Ayuda</i>	53
E.15. Contenido de <i>Acerca de...</i>	54

Índice de tablas

A.1. Horas empleadas en el proyecto.	8
A.2. Costes de personal anual.	10
A.3. Costes totales.	11
B.1. Caso de uso 1: Crear Base de Datos	17
B.2. Caso de uso 2: Preprocesar los ficheros originales corruptos . . .	18
B.3. Caso de uso 3: Cargar Archivos	19
B.4. Caso de uso 4: Generar Gráficos	20
B.5. Caso de uso 5: Gráfico apilado de Asignaturas por Curso	21
B.6. Caso de uso 6: Gráfico de máximos, mínimos y medias por curso	22
B.7. Caso de uso 7: Gráfico de máximos, mínimos y medias por semestre	23
B.8. Caso de uso 8: Salir de la aplicación	24

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 *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.** En una tarea del anterior *sprint* se apreció que no se podían leer los archivos Excel con Python, ya que los archivos tenían un error de formato y extensión. Por esta razón se optó por realizar un algoritmo que fuera capaz de leer estos archivos Excel(.xls) en modo texto, y parsear todo el contenido para generar otro archivo nuevo(.csv).

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.
- **Redactar casos de uso 1 y 2**
- **Eliminar apartado *Objetivos personales* de la documentación**
- **Descargar *GitHub Desktop* y *gVim 8.1***

Se estiman 18 horas de trabajo.

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

En la cuarta reunión se propusieron las siguientes tareas:

- **Cambiar la forma de parsear los datos del algoritmo.** Obtener primero la información por filas, después por celdas y por último por DATA o contenido de las celdas. Pudiendo de esta manera obtener valores como *MergeDown* y *MergeAcross* que aportan información necesaria sobre la separación o combinación de celdas.

Se estiman 20 horas de trabajo.

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

En la quinta reunión se propusieron las siguientes tareas:

- Mejorar los nombres de variables del algoritmo
- Mejorar los casos de uso 1 y 2
- Redactar caso de uso 3
- Pensar si es necesario crear una Base de Datos y cómo crearla en tal caso
- Crear gráfico apilado horizontalmente(caso de uso 2)

Se estiman 22 horas de trabajo.

Sprint 5 (08/04/19 - 03/05/19)

Hay que indicar que al coincidir con las vacaciones de Semana Santa, este *sprint* tuvo una duración superior a una semana. En la sexta reunión se propusieron las siguientes tareas:

- **Empezar a crear la interfaz gráfica.** Sacar un cuadro de diálogo cuando se pulse un botón de *Cargar* que permita seleccionar ficheros.
- **Crear modelo Entidad-Relación**
- **Probar a entrecomillar los *strings* de los (.csv) para poder leerlos bien**
- **Añadir apartado *Planificación Temporal* en la Documentación**
- **Resolver problema con la obtención de la ruta.** Al seleccionar un fichero (mediante el cuadro de diálogo de la interfaz gráfica) se obtenía una ruta como la siguiente *C:/Users/mdmar/Desktop/ficheroBueno.csv*, pero la ruta debería tener contrabarras o barras invertidas en vez de barras inclinadas.

Se estiman 45 horas de trabajo.

Sprint 6 (04/05/19 - 07/05/19)

En la séptima reunión (fue la primera videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- **Añadir un *Acerca de* en la interfaz gráfica.** Introducir escudo de la Universidad de Burgos, autor, tutor, fecha y versión.
- **Añadir líneas verticales al primer gráfico(apilado horizontal).** De esta manera se mejora la visualización del mismo.
- **Decidir si es necesario crear una pantalla de autenticación.** Incluso si se trata de una aplicación local, siempre es más seguro pedir la autenticación en caso de que dejemos el ordenador encendido.
- **Crear el logo de la aplicación**
- **Seguir con la creación del modelo Entidad-Relación(Sprint 5)**

Se estiman 18 horas de trabajo.

Sprint 7 (17/05/19 - 22/05/19)

En la octava reunión (segunda videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- **Cambiar el logo y el nombre del proyecto.** Poner un nombre menos genérico al proyecto.
- **Modificar el *Acerca de* en la interfaz gráfica.** Recortar el pdf para que no se vea tanto espacio en blanco, añadir el logo y el nombre de la aplicación.
- **Automatizar las líneas verticales introducidas en el primer gráfico.** Tratar de automatizar las líneas verticales del primer gráfico, no introducirlas a mano una a una.
- **Crear el segundo gráfico.** Crear el segundo gráfico, indicando máximos, mínimos, medias y cuartiles de las asignaturas de cada curso de un año académico.

Se estiman 14 horas de trabajo.

Sprint 8 (23/05/19 - 03/06/19)

En la novena reunión (tercera videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- **Crear Base de Datos(BBDD).** Crear la Base de Datos del proyecto, tal y como se define en el Modelo Entidad-Relación creado.
- **Añadir el *Modelo ER* y las *tablas a crear* en la Documentación.**
- **Crear otro tipo/modelo de gráfico, similar al segundo gráfico.** En vez de asignaturas por cada curso de un año, que se visualicen asignaturas por cada semestre de un año. Es decir, visualizar el doble de columnas(8) o de cajas que en el segundo gráfico(4).
- **Buscar información sobre cómo comparar gráficos en *Python*.**

Se estiman 25 horas de trabajo.

Sprint 9 (04/06/19 - 10/06/19)

En la décima reunión (cuarta videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- **Permitir carga de datos de fichero con múltiples titulaciones.** Permitir que el sistema pueda cargar datos en la BBDD a partir de un fichero con múltiples titulaciones.
- **Crear casos de usos para los botones de Crear BBDD, Cargar Datos y Salir.**
- **Cambiar Botón de Crear BBDD a menú superior.** Es un botón o función que se utilizará con poca frecuencia, ya que una vez creada la BBDD, no hará falta volverla a crear.
- **Pensar el modo de permitir seleccionar las gráficas al usuario.** Lista jerárquica, lista desplegable... etc.
- **Arreglar problema de la Documentación con la barra baja(_) en Latex**

Se estiman 28 horas de trabajo.

Sprint 10 (11/06/19 - 18/06/19)

En la undécima reunión (quinta videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- Mejorar *Introducción*, hacer *Resumen* y *Palabras Clave*.
- Mejorar apartado *C.2. Diseño de datos* de los anexos. También girar imagen de modelo ER, cambiar posición de imágenes de la BBDD...).
- Crear caso de uso para el boton de *Preprocesado*.
- Crear y diferenciar 2 botones en la interfaz gráfica (*Preprocesado* y *Cargar Archivos*).
- Crear 3 botones (con una imagen cada uno) en la interfaz gráfica. Para poder seleccionar el tipo de gráfico.
- Hacer *Técnicas y Herramientas* de la memoria.
- Hacer *C.3. Diseño procedimental* y *C.4. Diseño arquitectónico* de los anexos.

Se estiman 35 horas de trabajo.

Sprint 11 (19/06/19 - 25/06/19)

En la duodécima reunión (sexta videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- Mejorar traducción del *Resumen*.
- Hacer *Conceptos teóricos* de la memoria.
- Obtener los gráficos en cada ventana de los diferentes tipos de gráficos. Obtener los gráficos creados a partir de la información de la BBDD y de lo que haya seleccionado el usuario.
- Añadir botón de *Descargar Gráfico*. El botón permitirá descargar el gráfico como una imagen (.png) en la ruta donde se encuentre la aplicación.
- Crear un diagrama UML con todos los casos de uso.

Se estiman 40 horas de trabajo.

Sprint 12 (26/06/19 - 01/07/19)

En la decimotercera reunión (séptima videoconferencia que se realizó vía Skype) se propusieron las siguientes tareas:

- **Finalizar la interfaz gráfica.**
- **Crear un ejecutable para poder ejecutar la aplicación.**
- **Mejorar tamaño de imágenes y pies de página en la documentación.**
- **Descargar los gráficos visualizando toda la información.** Evitar que al descargar las imágenes de los gráficos, por defecto se corten los nombres de las asignaturas.
- **Finalizar los anexos.**
- **Finalizar la memoria.**

Se estiman 45 horas de trabajo.

Resumen

En la siguiente tabla se puede apreciar el tiempo dedicado a cada tarea:

Tarea	Tiempo (horas)
<i>Programación</i>	140
<i>Documentación</i>	80
<i>Investigación</i>	70
<i>Corrección de errores</i>	52
<i>Reuniones de seguimiento</i>	13
Total	355

Tabla A.1: Horas empleadas en el proyecto.

En la figura [A.1](#) se puede apreciar un gráfico sobre la distribución del tiempo empleado durante el desarrollo del proyecto.



Figura A.1: Gráfico con el resumen de las horas empleadas en el proyecto



Figura A.2: Gráfico con los *commits* que se ha realizado cada semana

Gráficos y estadísticas

Con la ayuda de los datos que ofrece GitHub, se van a visualizar una serie de gráficos sobre el desarrollo del proyecto.

- **Commits.** En la figura A.2 se aprecia un gráfico de barras, donde se muestra la cantidad de *commits* que se ha realizado cada semana.
- **Contribuciones.** En la gráfica de la figura A.3 se aprecia que el desarrollo del proyecto ha sido constante, con numerosos *issues*, *commits* y aportaciones a lo largo de los cuatro meses.

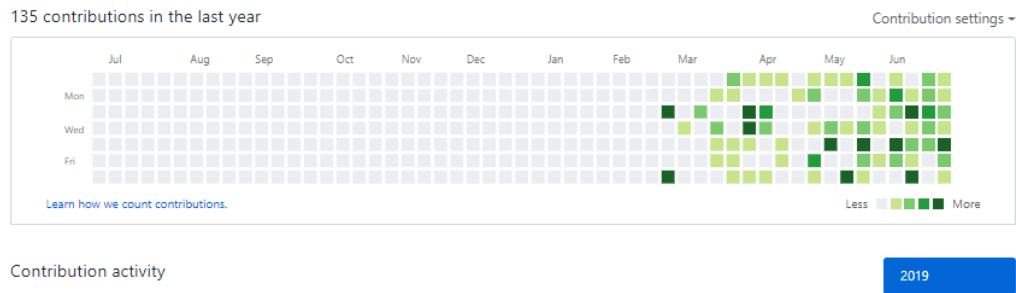


Figura A.3: Gráfico donde se aprecia el desarrollo constante

A.3. Estudio de viabilidad

En este apartado se va a analizar tanto la viabilidad económica como la viabilidad legal del proyecto.

Viabilidad económica

En este subapartado se van a exponer los costes y beneficios que hubiera tenido el desarrollo del proyecto en un entorno empresarial real.

■ Costes de personal:

El proyecto ha sido realizado por una única persona (desarrollador junior a tiempo completo) durante un total de tres meses. De acuerdo a lo anterior, se consideran los siguientes valores:

Concepto	Coste (€)
Salario anual bruto	18000
Coste S.Social empresa (36 %)	6480
Coste Total anual	24480

Tabla A.2: Costes de personal anual.

Por lo tanto, el precio por hora se calcula dividiendo el *Coste Total anual* entre la jornada anual estimada de 1700 horas. Luego se obtiene un precio de 14.40€ la hora. Como anteriormente se han estimado 355 horas de realización del proyecto, se obtiene un coste total de 5112€. Hay que destacar que a este coste habría que añadirle el coste del tutor.

- **Costes de hardware:** Para la realización de este proyecto se ha utilizado un ordenador portátil que tiene una antigüedad aproximada de 4 años y se estima un valor aproximado de 800€. El tiempo utilizado ha sido de 4 meses, luego el coste amortizado durante estos meses sería de 48€.
- **Costes de software:** Para la realización del proyecto todas las herramientas y programas utilizados han sido de software libre y gratuito. Las herramientas que no cumplían estas características, se optó por utilizar su versión de prueba (*Nitro Pro* por ejemplo).
- **Costes totales:** La suma total de todos los costes anteriores es la siguiente:

Concepto	Coste
<i>Personal</i>	5112€
<i>Hardware</i>	48€
Software	0€
Total	5160€

Tabla A.3: Costes totales.

Teniendo en cuenta los datos obtenidos anteriormente, se concluye que para que el proyecto resulte rentable en el período aproximado de uno o dos años, sería necesario utilizar la aplicación para la generación de las diferentes gráficas en todas las titulaciones de la Universidad de Burgos que lo consideraran adecuado.

De esta manera se amortizaría la aplicación, ya que vender licencias de la misma, no tiene mucho sentido en un entorno externo a la universidad. No se establece una periodicidad del uso de la aplicación, ya que podría ser utilizada en diferentes momentos del curso académico al poder almacenar datos de años anteriores, así como datos de matriculación del año actual.

Viabilidad legal

A continuación se expone la viabilidad del proyecto en el ámbito legal, es decir, las licencias del software utilizado en el desarrollo de mismo.

El lenguaje de programación *Python* y todos sus módulos y librerías, poseen la licencia *PSFL* (*Python Software Foundation License*)[2]. Este tipo

de licencia, es exclusivo de *Python* y cumple con los requisitos *OSI*¹, por lo tanto es categorizada dentro de las licencias de software libre.

Todas las herramientas que se han utilizado han sido de software libre, como se ha comentado en la memoria en el apartado de *Técnicas y herramientas*.

Por lo tanto, teniendo en cuenta las licencias de las herramientas utilizadas, la licencia que mejor se ajusta a nuestro proyecto es la *GNU (General Public License)*^[1] en su última versión (versión 3).

¹Open Source Initiative

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, así como el primer y tercer cuartil(el segundo cuartil coincide con la media).
- **RF-4: Gráfico de máximos, mínimos y medias por semestre:** La aplicación debe ser capaz de mostrar un gráfico cuyo *eje x* muestre los diferentes semestres(1º, 2º, 3º, 4º, 5º, 6º, 7º y 8º) y su *eje y* muestre la cantidad total de alumnos matriculados, indicando los máximos, mínimos y medias por cada curso, así como el primer y tercer cuartil(el segundo cuartil coincide con la media).

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-5: Integridad:** La herramienta debe cumplir con la integración de datos y no tener pérdidas de información, así como modificación de la misma.

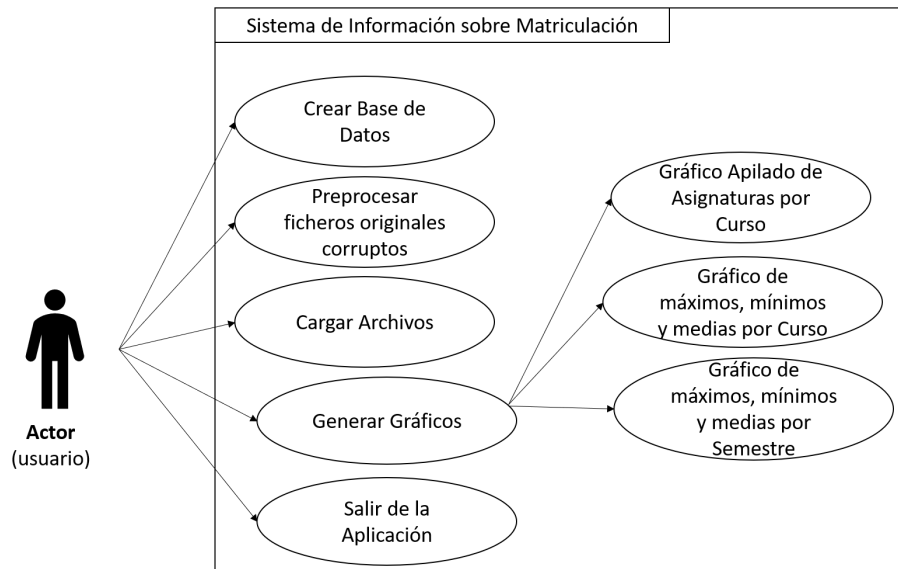


Figura B.1: Diagrama UML con todos los Casos de Uso

B.4. Especificación de requisitos

A través del diagrama de la figura B.1 se van a visualizar todos los casos de uso de los requisitos funcionales previamente definidos. Para esto, se va a utilizar la notación UML. Posteriormente se detallarán todos los casos de uso mediante tablas.

Actores

El actor del sistema es el usuario o persona que maneja la aplicación.

Casos de Uso

Caso de uso 1: Crear Base de Datos	
Descripción	La aplicación debe ser capaz de crear una Base de Datos llamada BBDD y formada por 3 tablas principales denominadas ASIGNATURAS, GRUPOS y PROFESORES. Estas 3 tablas deben contar con los atributos y claves primarias definidas en el Modelo Entidad Relación(ER).
Requisitos	RF-1
Precondiciones	Ejecutar la aplicación con un usuario con permisos de escritura o creación, para poder crear la Base de Datos.
Secuencia normal	Paso Acción
	1 El usuario pulsa el botón de Crear BBDD.
	2 Se crea el fichero llamado BBDD en la ruta sobre la cual se esté ejecutando la aplicación. Este fichero contendrá la estructura ya comentada de 3 tablas principales, para posteriormente poder ir introduciendo los datos.
Postcondiciones	Se crea la Base de Datos(BBDD).
Excepciones	Si ya estuviera creada la BBDD, no se crearía de nuevo y se mostraría una pantalla de tipo warning al usuario indicando que la Base de Datos BBDD ya esta creada.
Importancia	Alta
Urgencia	Alta
Comentarios	Es necesario que la aplicación cuente con la Base de Datos BBDD, para poder ir almacenando la información que posteriormente se utilizará para realizar gráficas.

Tabla B.1: Caso de uso 1: Crear Base de Datos

Caso de uso 2: Preprocesar los ficheros originales corruptos	
Descripción	La aplicación debe contar con un botón de Preprocesado. Una vez pulsado se mostrará una nueva pantalla emergente (pantalla del sistema operativo) visualizando los ficheros y directorios de la ruta donde se está ejecutando la aplicación. Se permitirá seleccionar únicamente ficheros (.xls) que son los que se pueden y deben preprocesar.
Requisitos	RF-1
Precondiciones	La aplicación debe de estar iniciada.
Secuencia normal	Paso Acción
	1 El usuario pulsa el botón de Preprocesado.
	2 Se mostrará una nueva pantalla emergente (pantalla del sistema operativo) visualizando los ficheros (.xls) y directorios de la ruta donde se está ejecutando la aplicación. Se permitirá seleccionar únicamente ficheros (.xls).
	3 Si el usuario selecciona un fichero (.xls) corrupto, se creará en la misma ruta un fichero (.csv) con toda la información del fichero seleccionado parseada y preparada para poder introducirla en la Base de Datos.
Postcondiciones	Si se ha seleccionado un fichero (.xls) corrupto, se creará en la misma ruta otro fichero (.csv) con toda la información parseada y lista para introducir en la Base de Datos
Excepciones	
Importancia	Alta
Urgencia	Alta
Comentarios	Es necesario preprocesar los archivos originales descargados de Sigma, ya que tienen un error de formato y si no se parsean y se convierten en otro fichero (.csv), que además está preparado para introducir los datos en la BBDD, no pueden ser utilizados.

Tabla B.2: Caso de uso 2: Preprocesar los ficheros originales corruptos

Caso de uso 3: Cargar Archivos	
Descripción	La aplicación debe contar con un botón de Cargar Archivos, que permita al usuario seleccionar ficheros no corruptos, es decir ficheros generados nuevos con extensión (.csv) previamente preprocesados. Una vez seleccionado el fichero que el usuario desee, se procederá a la carga en la BBDD de toda la información que contenga dicho fichero.
Requisitos	RF-1
Precondiciones	Existe un fichero (.csv) con datos para importar. Dicho fichero se ha generado a partir de un fichero (.xls) 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 El usuario selecciona el fichero (.csv) que desea cargar en la Base de Datos(BBDD).
	3 Se carga en la Base de Datos(BBDD) toda la información del fichero seleccionado que no estuviera previamente en la BBDD , atendiendo a los requisitos y claves primarias de la BBDD.
Postcondiciones	Se carga la información en la Base de Datos(BBDD).
Excepciones	Si los datos del fichero (.csv) seleccionado ya estuvieran en la BBDD, no se añadirían a la misma.
Importancia	Alta
Urgencia	Alta
Comentarios	.

Tabla B.3: Caso de uso 3: Cargar Archivos

Caso de uso 4: Generar Gráficos	
Descripción	La aplicación debe ser capaz de generar gráficos. El usuario debe seleccionar qué tipo de gráfico quiere generar. Una vez seleccionado el tipo de gráfico, deberá seleccionar diferentes parámetros con la ayuda de listas desplegables para posteriormente poder generarlo. Una vez realizado estos pasos, se generará el tipo de gráfico seleccionado con los parámetros elegidos.
Requisitos	RF-2, RF-3, RF-4
Precondiciones	Los datos o información necesaria se ha cargado correctamente en la Base de Datos. La aplicación debe estar iniciada.
Secuencia normal	Paso Acción
	1 El usuario pulsa el botón del gráfico que quiere generar.
	2 Se abre una nueva ventana con el nombre del gráfico elegido y las opciones necesarias para seleccionar mediante listas desplegables.
	3 Se seleccionan las opciones que se desee de cada lista desplegable(Año, Titulación, Curso...).
	4 Se genera el gráfico con las opciones previamente seleccionadas.
Postcondiciones	Se genera el gráfico deseado con los datos elegidos.
Excepciones	Es necesario seleccionar una opción de cada lista desplegable, en caso contrario, no se podrá generar el gráfico adecuadamente.
Importancia	Alta
Urgencia	Alta
Comentarios	.

Tabla B.4: Caso de uso 4: Generar Gráficos

Caso de uso 5: 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.
	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.
Postcondiciones	La aplicación debe mostrar el gráfico apilado de manera correcta.
Excepciones	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.
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.5: Caso de uso 5: Gráfico apilado de Asignaturas por Curso

Caso de uso 6: 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, así como el primer y tercer cuartil (ya que el segundo cuartil coincide con la media).
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.6: Caso de uso 6: Gráfico de máximos, mínimos y medias por curso

Caso de uso 7: Gráfico de máximos, mínimos y medias por semestre	
Descripción	La aplicación debe ser capaz de mostrar un gráfico cuyo <i>eje x</i> muestre los diferentes semestres(1º, 2º, 3º, 4º, 5º, 6º, 7º y 8º) y su <i>eje y</i> muestre la cantidad total de alumnos matriculados, indicando los máximos, mínimos y medias por cada semestre, así como el primer y tercer cuartil (ya que el segundo cuartil coincide con la media).
Requisitos	RF-4
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 semestre.
Postcondiciones	La aplicación debe mostrar el gráfico de máximos, mínimos y medias por semestre de manera correcta.
Excepciones	
Importancia	Alta
Urgencia	Alta
Comentarios	

Tabla B.7: Caso de uso 7: Gráfico de máximos, mínimos y medias por semestre

Caso de uso 8: Salir de la aplicación	
Descripción	La aplicación debe contar con un botón de Salir. Una vez pulsado se mostrará una nueva pantalla emergente preguntando al usuario si desea salir de la aplicación. En caso afirmativo se cerrará la aplicación y en el caso contrario, la aplicación continuará ejecutándose. Hay que destacar que la aplicación también cuenta con un botón de cerrar en la parte superior derecha(cruz o aspa genérica).
Requisitos	
Precondiciones	La aplicación debe de estar iniciada.
Secuencia normal	Paso Acción
	1 El usuario pulsa el botón de Salir.
	2 Se mostrará una nueva pantalla emergente preguntando al usuario si desea salir de la aplicación.
	3 Si el usuario pulsa la opción Si, se cerrará la aplicación, por el contrario, si pulsa la opción No la aplicación continuará ejecutándose.
Postcondiciones	Se cierra la aplicación.
Excepciones	
Importancia	Baja
Urgencia	Baja
Comentarios	El botón de Salir es un botón adicional.

Tabla B.8: Caso de uso 8: Salir de la aplicación

Apéndice C

Especificación de diseño

C.1. Introducción

En este apartado se va a explicar el análisis y diseño de los datos. También se va a exponer cómo se han resuelto las especificaciones y los casos de uso mencionados en el apartado anterior.

Uno de los objetivos de este apartado es la comprensión de la toma de decisiones y los motivos o causas que han dado lugar a las mismas.

C.2. Diseño de datos

Modelo Entidad-Relación(ER)

A continuación en la figura [C.1](#) se observa el modelo ER o Diagrama Relacional de la Base de Datos de la aplicación.

Se aprecia un total de cinco entidades, dos de las cuales son entidades débiles(*Curso* y *Grupo*). En las relaciones entre entidades se aprecia la cardinalidad entre las mismas, existiendo tres relaciones *1.N* y una relación *N.N* entre *Profesores* y *Grupo*. En cada entidad podemos apreciar sus atributos, así como sus claves primarias(subrayadas).

Base de datos

Se utiliza una base de datos(BBDD) para almacenar toda la información de la aplicación. La BBDD está compuesta por tres tablas principales:

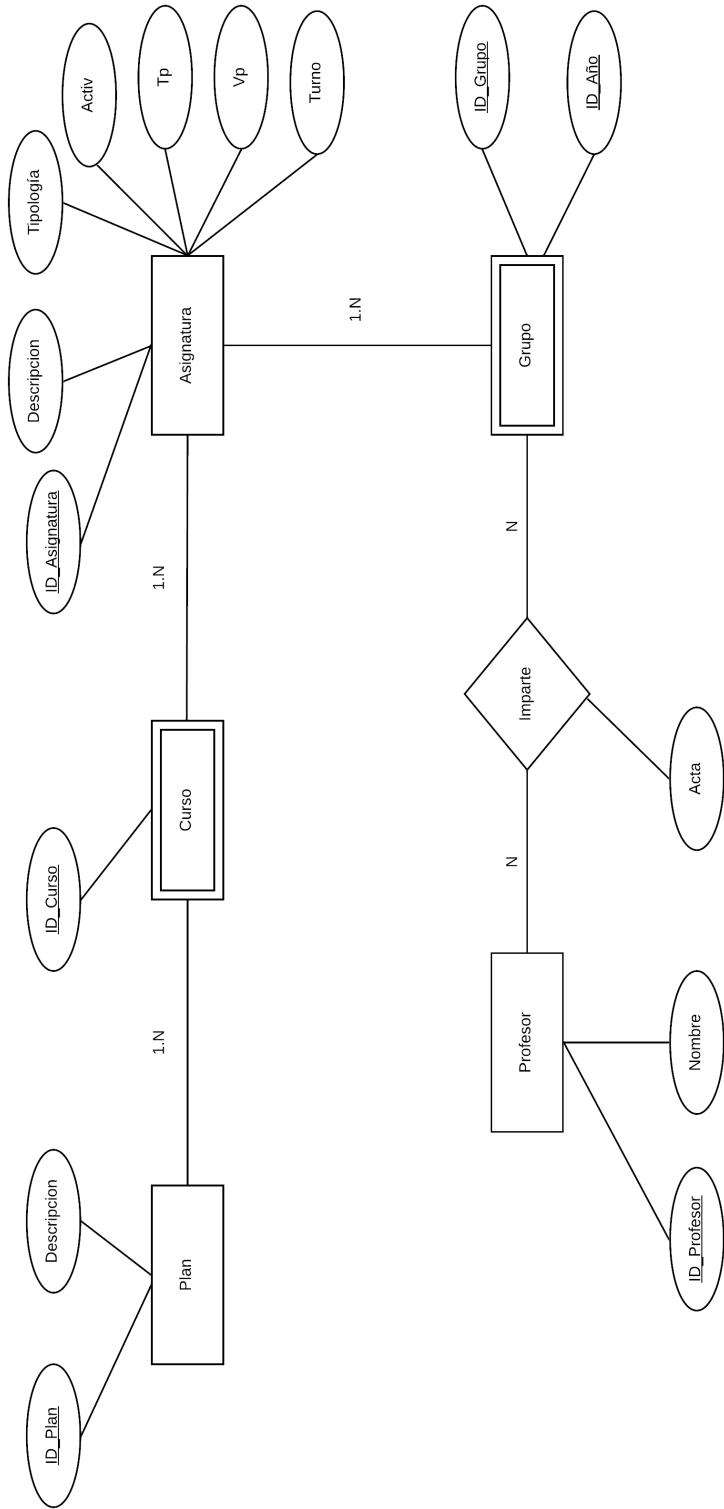


Figura C.1: Modelo Entidad-Relación(ER)

ASIGNATURAS		
	Id_Asignatura	INTEGER
	Descripcion	VARCHAR(50)
	Curso	INTEGER
	Plan	VARCHAR(50)
	Tipologia	VARCHAR(10)
	Activ	VARCHAR(10)
	Tp	VARCHAR(10)
	Vp	INTEGER
	Turno	VARCHAR(10)

Figura C.2: Tabla ASIGNATURAS

- **ASIGNATURAS:** en esta tabla se almacena la información relacionada con las asignaturas que componen un Plan de Estudios o Titulación determinada. Esta tabla está formada por 9 campos, que son los siguientes:
 - **Id_Asignatura:** es la clave primaria de la tabla y almacena el identificador único de la asignatura. Es un campo numérico de 4 dígitos.
 - **Descripcion:** se almacena el nombre de la asignatura. Es un campo de caracteres.
 - **Curso:** se almacena el curso al que pertenece la asignatura. Es un campo numérico de 1 dígito.
 - **Plan:** se almacena el Plan de Estudios o Titulación al que pertenece la asignatura. Es un campo alfanumérico.
 - **Tipologia:** se almacena la tipología del grupo, que puede ser teórica o práctica. Es un campo de caracteres.
 - **Activ:** se trata de un campo de caracteres.
 - **Tp:** es un campo de caracteres.
 - **Vp:** es un campo numérico de 1 dígito. Si es 1 significa que la asignatura es del primer semestre y si es 2 indica que la asignatura se imparte en el segundo semestre.

- **Turno:** indica si el horario es de mañana, de tarde o mezcla de ambos (mixto). Se trata de un campo de caracteres.

GRUPOS	
Id_Asignatura	INTEGER
Id_Grupo	INTEGER
Temporada	VARCHAR(50)
Total_Alumnos	INTEGER

Figura C.3: Tabla GRUPOS

- **GRUPOS:** en esta tabla se almacena la información relacionada con los grupos. Hay que destacar que esta tabla se trata de una entidad débil, como se ha visto anteriormente en el modelo de Entidad Relación (ER). Tiene una clave primaria formada por 3 campos (Id_Asignatura, Id_Grupo y Temporada (Año)) para poder identificar el total de alumnos matriculados en un grupo de una asignatura, en un año académico en concreto. Esta tabla, por lo tanto, está formada por 4 campos:
 - **Id_Asignatura:** es la clave primaria de la tabla *ASIGNATURAS* y almacena el identificador único de la asignatura. Es un campo numérico de 4 dígitos. Como se ha comentado, es un campo que forma parte de la clave primaria de *GRUPOS*, junto a Id_Grupo y Temporada.
 - **Id_Grupo:** es el identificador de grupo y forma parte de la clave primaria junto a Id_Asignatura y Temporada. Es un campo numérico de 2 dígitos.
 - **Temporada:** indica el año que se cursó o se va a cursar un grupo de una asignatura en concreto y por lo tanto, forma parte de la clave primaria junto a Id_Asignatura e Id_Grupo. Es un campo alfanumérico.
 - **Total_Alumnos:** en este campo se almacena la cantidad de alumnos que hay matriculados en un grupo de una asignatura en un año determinado. Se trata, por tanto, de un campo numérico.



PROFESORES	
 Id_Profesor	INTEGER
 Id_Asignatura	INTEGER
 Id_Grupo	INTEGER
 Temporada	VARCHAR(50)
 Acta	VARCHAR(1)
 Nombre_Apellidos	VARCHAR(50)

Figura C.4: Tabla PROFESORES

- **PROFESORES:** esta tabla surge de la relación entre las entidades *Grupos* (entidad débil) y *Profesores* como se ha apreciado anteriormente en el modelo Entidad Relación (ER). En esta tabla se almacena la información relacionada con los profesores y los grupos de asignaturas que imparten clase en un año determinado. Tiene una clave primaria formada por 4 campos (Id_Profesor, Id_Asignatura, Id_Grupo y Temporada (Año)). Esta tabla, está formada por estos 6 campos:
 - **Id_Profesor:** forma parte de la clave primaria de la tabla y almacena el identificador único de profesores. Es un campo numérico de 4 dígitos.
 - **Id_Asignatura:** es un campo que forma parte de la clave primaria y almacena el identificador único de la asignatura. Se trata de un campo numérico de 4 dígitos.
 - **Id_Grupo:** es el identificador de grupo y forma parte de la clave primaria. Es un campo numérico de 2 dígitos..
 - **Temporada:** indica el año en el que un determinado profesor impartió un grupo de una asignatura. Por lo tanto, forma parte de la clave primaria y es un campo alfanumérico.
 - **Acta:** se almacena la información relevante sobre si el profesor firma o no el acta. Es un campo de caracteres.
 - **Nombre_Apellidos:** es un campo que almacena el nombre y apellidos de un profesor. Se trata de un campo de caracteres.

C.3. Diseño procedimental

Para entender mejor el comportamiento de la aplicación, en la figura C.5 se puede apreciar el diagrama de navegabilidad de la aplicación. En dicho diagrama se pueden observar los diferentes procedimientos o funcionalidades que existen en el proyecto. Por lo general, se ha tratado de crear una función por cada una de las cajas representadas en el diagrama anterior, pero en algún caso no ha sido posible y se ha dividido en dos o más funciones, por tratarse de procesos más complejos. A continuación se enumeran las funciones más relevantes del proyecto:

- **preprocesar.** Función que se encarga de convertir los (.xls) corruptos que seleccionemos en ficheros (.csv) para su posterior carga en la BBDD. Para esta labor, llama a la función algoritmo pasándole el nombre del archivo seleccionado por el usuario con anterioridad.
- **algoritmo.** Es una de las funciones más complejas de la aplicación, y realiza las siguientes funciones:
 - 1. Abrir un fichero Excel (.xls) como si fuera un (.txt), para obtener un (.xml).
 - 2. Parsear el contenido del (.xml) y obtener toda la información del Excel(todas las celdas).
 - 2.1. Se obtiene toda la información de cada fila del Excel ->ROW.
 - 2.2. Se obtiene toda la información de cada celda del Excel ->CELL.
 - ◊ 2.2.1. Se obtienen los valores MergeDown y MergeAcross de cada celda.
 - 2.3. Se obtiene la información *visual* de cada celda ->DATA. 3. Con toda la información anterior, se crea un fichero (.csv) y se va guardando la información correctamente.
- **crearBBDD.** Función que se encarga de la creación de la Base de Datos. Crea una Base de Datos llamada *BBDD* con 3 tablas: ASIGNATURAS, GRUPOS y PROFESORES. Si ya existiera creada la Base de Datos *BBDD* muestra una ventana de tipo warning mostrando por pantalla el mensaje: *La BBDD ya está creada.*
- **funcionCargar.** Función que se encarga de:

- 1. Mostrar una ventana del sistema operativo que nos permite seleccionar archivos.
 - 2. Una vez seleccionado un archivo, obtiene el nombre y se lo pasa a la función *cargarDatos*, para que devuelva un dataframe con todos los datos.
 - 3. Se rellenan los datos en las 3 tablas de la BBDD, llamando a las funciones *addAsignaturasRows*, *addGruposRows* y *addProfesoresRows*.
 - 4. También se llama a la función *obtenerTemporada*, ya que es la única forma de recuperar u obtener dicho valor.
- **cargarDatos.** Función que se encarga de abrir el fichero que se le pasa por parámetro (ruta), siempre y cuando, este fichero se encuentre en el directorio donde nos encontremos. Devuelve un dataframe con todos los datos del fichero (SIN CONTAR LAS 4 PRIMERAS FILAS DE DATOS).
 - **ventanaGrafica1.** Función que se encarga de generar la ventana secundaria para posteriormente poder personalizar, generar y descargar el tipo de Gráfico 1.
 - 1. Se crea la ventana *hija* raiz2 con las mismas características que la ventana principal.
 - 2. Se realizan consultas a la BBDD para poder rellenar la información a mostrar en los diferentes desplegables.
 - 3. Cada vez que se selecciona una opción en los desplegables, se actualiza el valor de la variable global que almacena dicho valor.
 - 4. Finalmente se cuenta con 2 botones (*Salir* y *Descargar Gráfica*). El segundo llama a la función *descargarG1* para que se proceda a la descarga.
 - **descargarG1.** Función que se encarga de generar y descargar el tipo de Gráfico 1 en función de las opciones que el usuario seleccione en los desplegables.
 - 1. Se realiza una consulta a la BBDD en función a los datos seleccionados por el usuario
 - 2. Se genera el gráfico, agrupando correctamente los datos anteriormente devueltos, así como personalizando el gráfico (tipo, ejes, nombres, tamaños...etc).

- 3. Se descarga el gráfico con el nombre generado con los parámetros seleccionados por el usuario.
- **ventanaGrafica2 y ventanaGrafica3.** Funciones muy parecidas a `ventanaGrafica1`, teniendo la particularidad de que además de llamar a `descargarG2` y `descargarG3` respectivamente, llaman a `rellenarListaG2` y `rellenarListaG3`. Estas funciones se encargan de hacer consultas a la BBDD en función de los parámetros que se le pasen por cabecera y devuelven una lista con el número de matriculados en las asignaturas del curso que se le pase por parámetro.
- **obtenerTemporada.** Función que se encarga de abrir el fichero que se le pasa por parámetro, siempre y cuando, este fichero se encuentre en el directorio donde nos encontremos; y devuelve el valor del año académico de la titulación/plan del fichero.
- **addAsignaturasRows, addGruposRows y addProfesoresRows.** Son funciones que se encargan de añadir los datos necesarios del dataframe y de la temporada que se le pasa por parámetro a la tabla determinada de la BBDD.
- **hacer_consulta.** Función que se encarga de realizar consultas a la Base de Datos. Realiza la consulta que se le pase por parámetro (consulta) y devuelve un cursor con el resultado de la misma.
- **abrirArchivo.** Función que se encarga de abrir y visualizar el archivo que se le pasa por parámetro.

Hay que destacar que para la creación del menú superior de la aplicación, se utiliza una clase denominada *menuSuperior*, con las siguientes funciones:

- **abrirAyudaWeb.**
- **abrirAbout.**
- **abrirAyudaLocal.**

C.4. Diseño de interfaces

Para el diseño de la interfaz de usuario, el objetivo principal era el de dotar de funcionalidad y sencillez a la aplicación. Se ha conseguido crear una

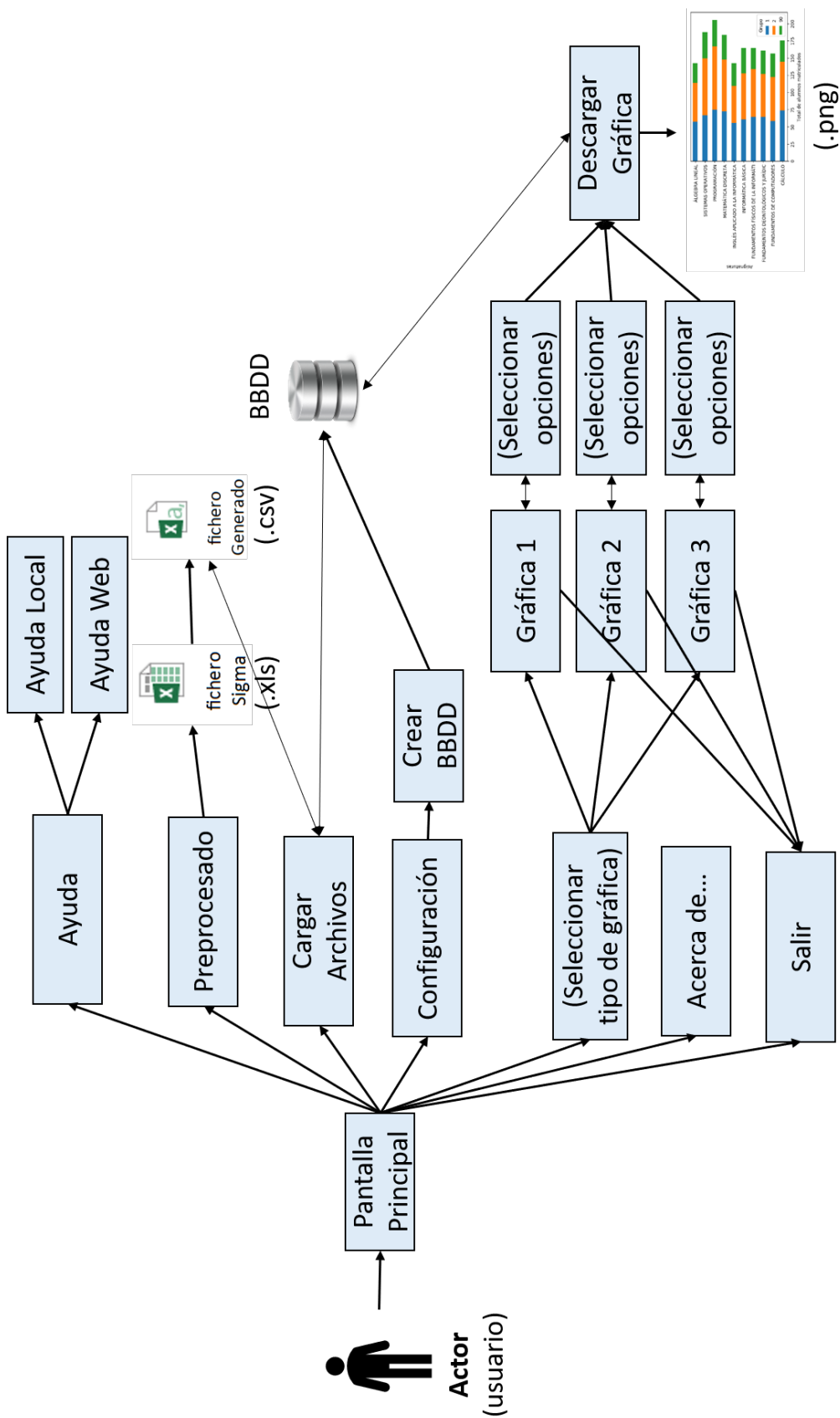


Figura C.5: Diagrama de Navegabilidad

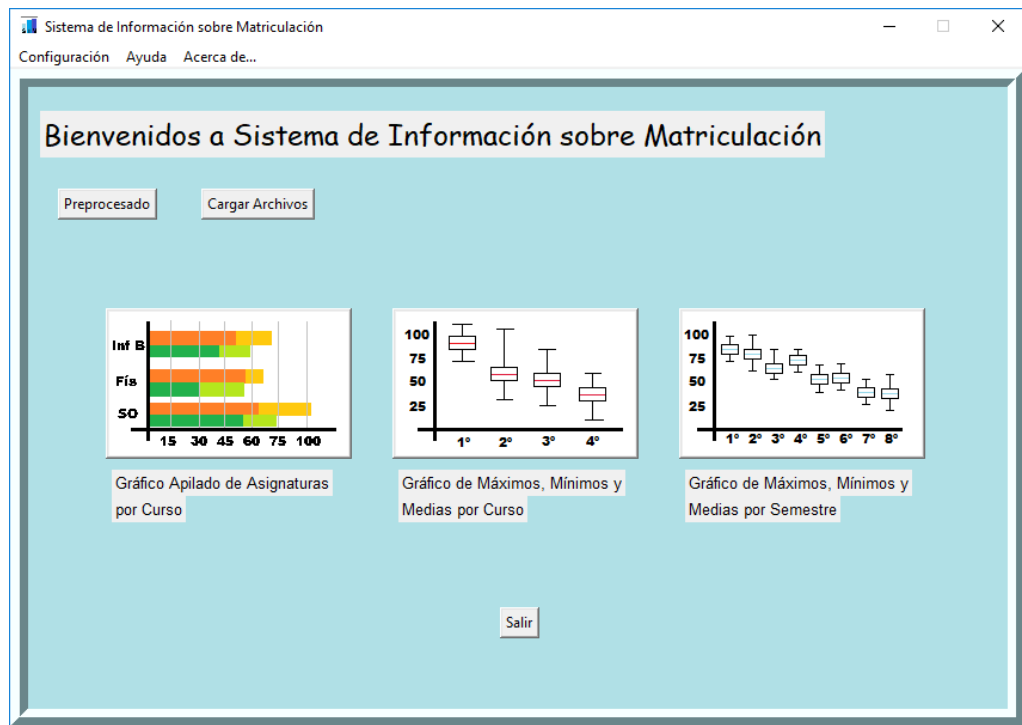


Figura C.6: Pantalla principal de la interfaz

interfaz funcional, limpia y clara para facilitar la interacción con el usuario. Esto se ha logrado en parte gracias a la librería *tkinter* de *Python*, ya que se caracteriza por su funcionalidad y utilidad, en contra del aspecto visual o el orientado hacia el diseño gráfico.

A continuación, en la figura C.6, se muestra la vista de la ventana principal de la aplicación. En la figura se puede apreciar que se trata de una interfaz muy sencilla, sin imágenes ni fondos intrusivos.

Con la finalidad de mejorar la experiencia del usuario se han incorporado una serie de características o mejoras en la interfaz.

Cuando se quiere informar de algún acontecimiento relevante al usuario, se le muestran ventanas emergentes, con la finalidad de captar la atención del mismo. Estas ventanas emergentes, pueden ser de tipo *Warning* como la figura E.2 o simplemente informativas como se aprecia en la figura C.8

Para evitar errores o confusiones, cuando se muestran ventanas propias del sistema operativo, como es el caso de la figura E.6, se indica en la parte

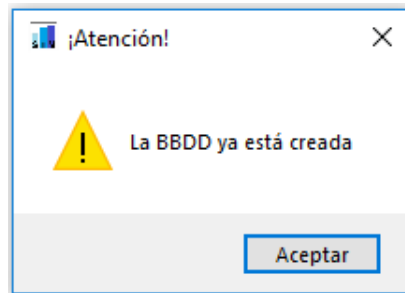


Figura C.7: Ventana emergente de tipo Warning de la interfaz

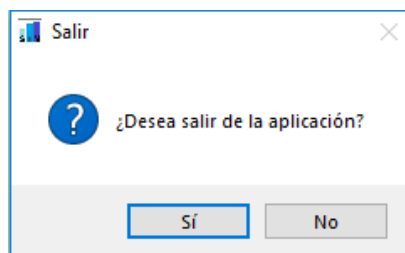


Figura C.8: Ventana emergente informativa de la interfaz

superior de la ventana un mensaje explicando qué tipos de archivos hay que seleccionar.

Adicionalmente, como se puede apreciar en la figura E.6, en la parte inferior sólo nos deja seleccionar el tipo de archivos necesarios (en este caso .csv), evitando así la subida de ficheros no válidos. Además, en el tipo de archivos nos muestra el mensaje .csv (*Ya parseados*), para comprender con mayor facilidad, el tipo de archivos que hay que seleccionar.

En las pantallas secundarias (pantallas de los diferentes tipos de gráficos), se muestra en la parte superior migas de pan o *breadcrumb-trail* como se aprecia en la figura C.10.

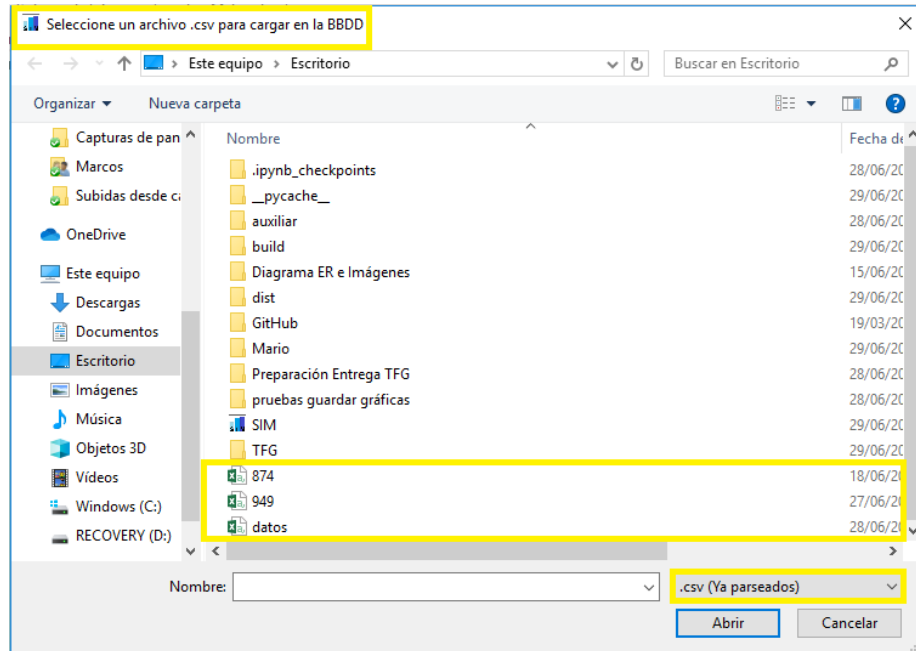


Figura C.9: Textos de ayuda en la selección de archivos

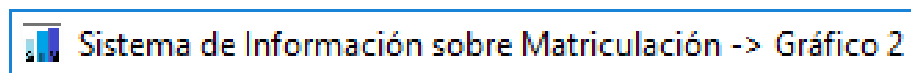


Figura C.10: Migas de pan de la ventana secundaria

Apéndice *D*

Documentación técnica de programación

D.1. Introducción

En este apartado se va a describir la documentación técnica referente a la programación. Se incluirá la preparación del entorno de desarrollo utilizado y la estructura de la aplicación.

D.2. Estructura de directorios

El proyecto tiene la siguiente estructura de directorios:

- **/Application/**: Contiene los siguientes elementos:
 - Ejecutable **SIM.exe**: Haciendo doble clic sobre el ejecutable se ejecuta la aplicación.
 - Carpeta **/Application/img**: Contiene las imágenes que se utilizan en la interfaz gráfica.
 - **about.pdf**: *Acerca de...* de la interfaz gráfica.
 - **ayuda.pdf**: *Ayuda Local* de la interfaz gráfica.
 - **BBDD**: Base de Datos ya creada de la aplicación con un conjunto de datos introducidos. Este fichero se puede crear con la aplicación, pero se suministra para probar la misma.

- ***ficheroSigma.xls***: Fichero descargado de la aplicación Sigma para que pueda ser usado como fichero de entrada de nuestra aplicación.
- ***ficheroSigma2.xls***: Otro ejemplo similar al anterior.
- **/Code/**: En esta carpeta se encuentran los ficheros *SIM.py* y *SIM.ipynb* donde se hayan todas las funcionalidades programadas del proyecto.
- **/Documentation/**: Contiene todos los ficheros relacionados con la documentación del proyecto.
 - **Memoria.pdf**: Documento que contiene la memoria del proyecto.
 - **Anexos.pdf**: Documento que contiene los anexos del proyecto.
 - **/Documentation/tex**: Contiene todos los apartados en los que se desglosa tanto la memoria como los anexos.
 - **/Documentation/img**: Contiene todas las imágenes utilizadas en la memoria y en los anexos.
- **/**: Contiene un fichero *GitHub_Repository_Link.txt* que contiene el enlace al repositorio del proyecto de *GitHub*, un fichero *SIM.mp4* que es un vídeo de demostración de uso de la aplicación y un fichero *README.txt* que contiene una serie de indicaciones, así como la licencia.

D.3. Manual del programador

Esta sección, tiene como objetivo principal, servir de punto de partida o referencia para futuras personas que decidan mejorar la aplicación. Se va a explicar cómo se han instalado las herramientas principales para el desarrollo del proyecto, así como la compilación y ejecución del mismo. También se explicará como obtener el código y ficheros necesarios.

- **Anaconda Navigator**. *Anaconda* ha sido la opción utilizada para poder utilizar *Jupyter Notebook*¹ en la versión 5.7.4 como entorno de desarrollo principal. *Anaconda Navigator* es una plataforma completamente gratuita, y podemos descargarla a través del siguiente enlace ². Antes de descargarlo, deberemos seleccionar la opción de *Python 3.7*

¹www.jupyter.org

²www.anaconda.com/distribution

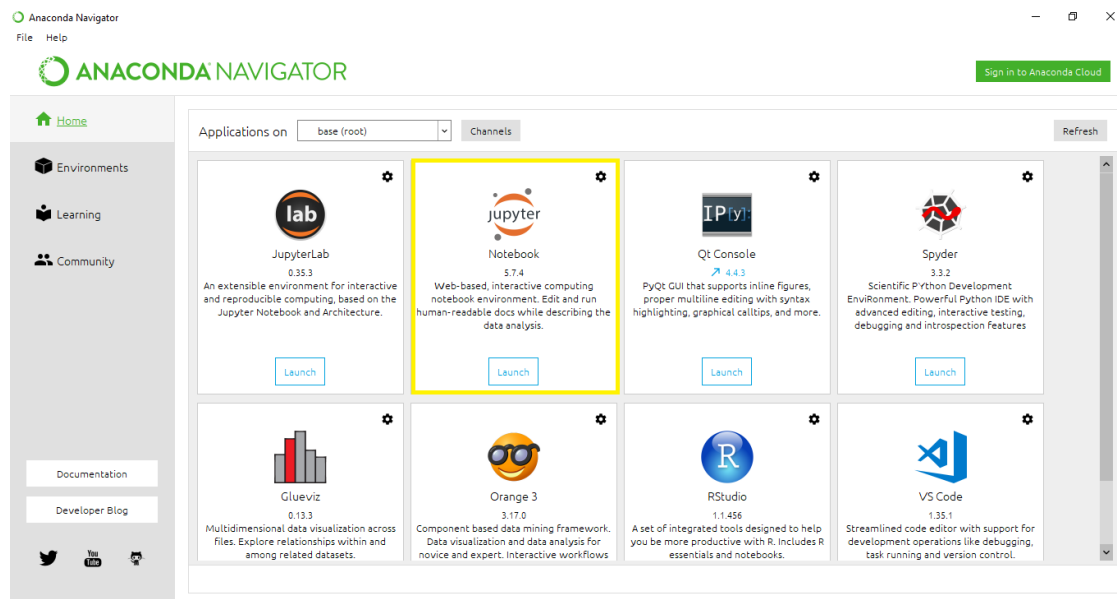


Figura D.1: Anaconda Navigator

version, ya que es la versión que se ha utilizado para el desarrollo del proyecto.

La instalación de la plataforma no tiene ninguna dificultad, pero si hubiera algún inconveniente, en Internet existen numerosos vídeos explicativos de cómo realizar la descarga paso a paso.

Una vez instalada, abrimos la aplicación y veremos una pantalla similar a la figura D.1. Seleccionamos *Jupyter Notebook* y ya tendríamos listo la plataforma de desarrollo.

- **Bibliotecas o librerías utilizadas.** No se necesita la instalación de ninguna de las librerías utilizadas, ya que todas vienen incluidas en la plataforma anteriormente descargada (*Anaconda Navigator*).
- **DB Browser.** Como se ha comentado en la memoria, DB Browser³ es una herramienta gratuita y de código abierto cuyo principal objetivo es la administración de Bases de Datos que utilizan SQLite como motor de las mismas.

³www.sqlitebrowser.org

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Para la descarga de esta herramienta, únicamente tendremos que elegir la opción que mejor se adapte a nuestro equipo y realizar la descarga de la misma a través del siguiente enlace ⁴.

En este punto, ya podremos comenzar a trabajar sobre la aplicación. Sólo nos hace falta disponer del código y de los recursos del proyecto necesarios, que se explicará en la siguiente sección.

D.4. Compilación, instalación y ejecución del proyecto

Descarga e instalación del proyecto

Para obtener el código y los recursos necesarios del proyecto, es necesario descargar o clonar el repositorio de *GitHub* desde el siguiente enlace⁵.

Una vez obtenido el código y con la ayuda de la sección anterior (*Estructura de directorios*), abriendo el fichero de *Jupyter Notebook* llamado *SIM.ipynb*, podremos modificar y continuar el desarrollo del proyecto.

Hay que destacar que no se necesita ninguna instalación adicional, ya que todas las librerías utilizadas ya se encuentran en *Anaconda Navigator*.

Ejecución del proyecto

Para la ejecución del proyecto, únicamente es necesario ejecutar o hacer doble clic en *SIM.exe*. De esta manera se procederá al arranque de la aplicación.

Hay que destacar que la aplicación también se puede ejecutar desde *emphJupyter Notebook*. Sólo se necesita abrir el fichero *SIM.ipynb* y pulsar el botón de ejecución de *Jupyter Notebook* (D.2).

Visualización de la BBDD

Para visualizar la base de datos, únicamente tendremos que arrastrar o abrir el fichero BBDD con *DB Browser*. Como se aprecia en la figura D.3, en el apartado *Hoja de datos* se podrá consultar el contenido de nuestras tablas. Asimismo, en el apartado *Estructura* se podrá visualizar el diseño de la BBDD y en el apartado *Ejecutar SQL* se podrán realizar consultas.

⁴www.sqlitebrowser.org/dl

⁵www.github.com/mdi0007/Sistema-Informacion-sobre-Matriculacion

D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

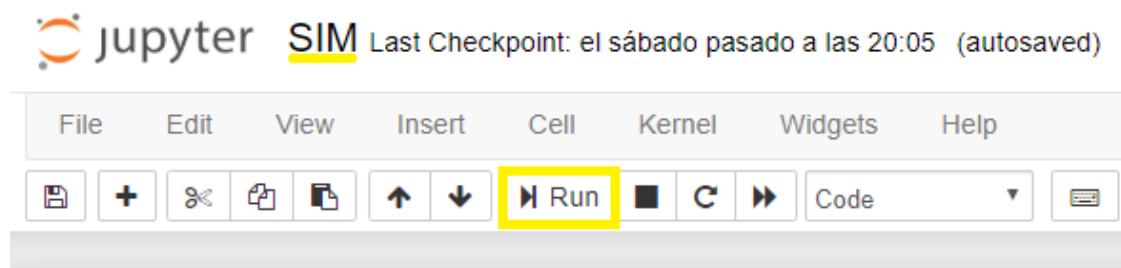


Figura D.2: Ejecución de *Jupyter Notebook*

APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

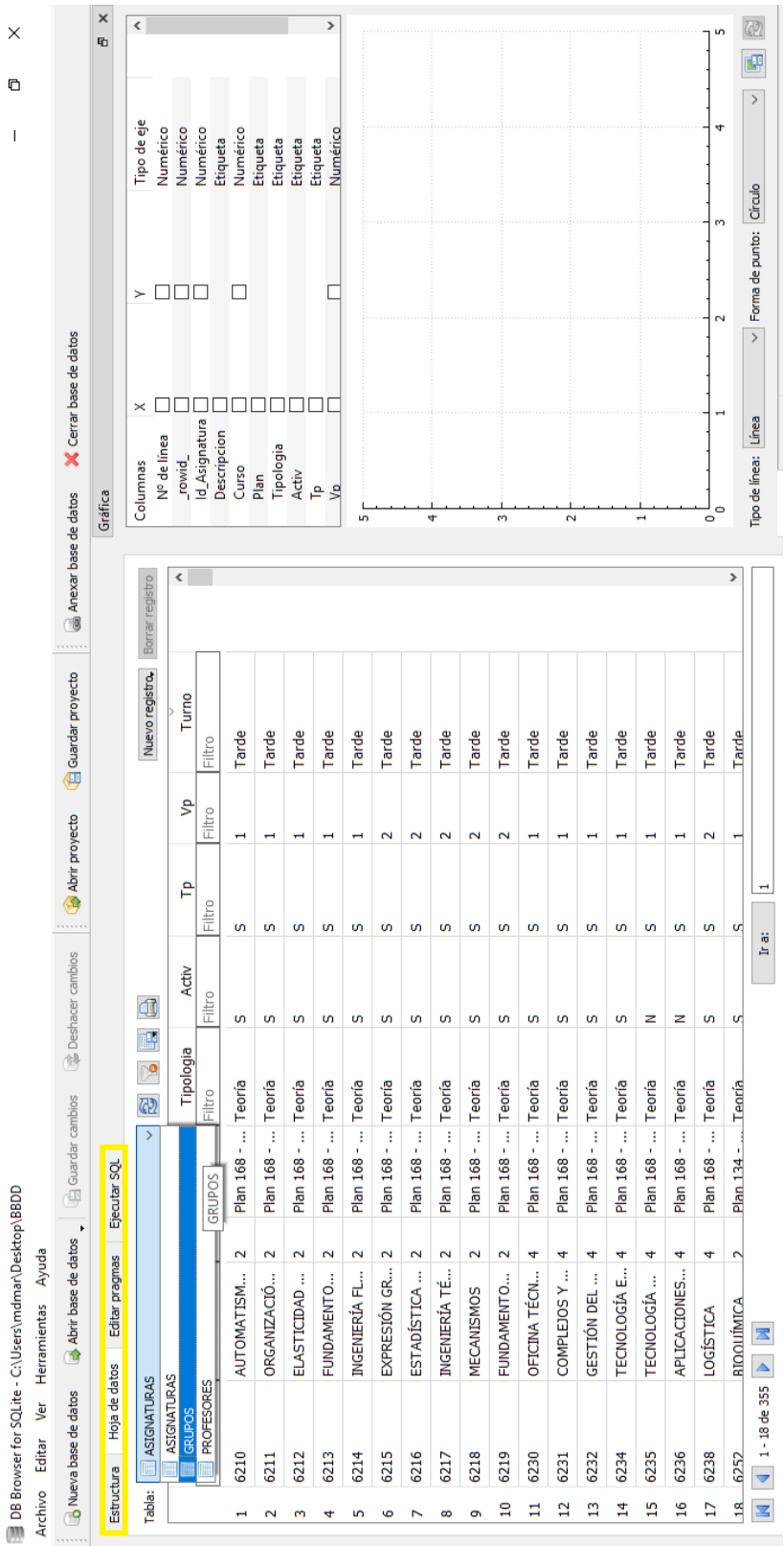


Figura D.3: Ejecución de *Jupyter Notebook*

Apéndice *E*

Documentación de usuario

E.1. Introducción

En el siguiente apartado se van a explicar los requerimientos de la aplicación, así como explicar cómo ejecutarla en un equipo con *Windows* y las indicaciones para su correcto uso por parte del usuario.

E.2. Requisitos de usuarios

Para ejecutar la aplicación no hace falta disponer de requisitos previos, ya que se ha desarrollado un ejecutable denominado ***SIM.exe*** dentro de la carpeta *Application* del proyecto, que es capaz de arrancar la aplicación.

Por lo tanto, únicamente se necesita tener un dispositivo con sistema operativo *Windows*, como se ha comentado en la sección anterior.

E.3. Instalación

No es necesaria realizar una instalación de la aplicación, ya que como se ha comentado anteriormente, se ha desarrollado un ejecutable denominado ***SIM.exe*** dentro de la carpeta *Application* del proyecto, que es capaz de arrancar la aplicación sin realizar ninguna instalación previa por parte del usuario.

Lo que sí que necesitaremos será contar con dicho ejecutable, así como la estructura de carpetas principal del proyecto. Para obtenerlo, se puede

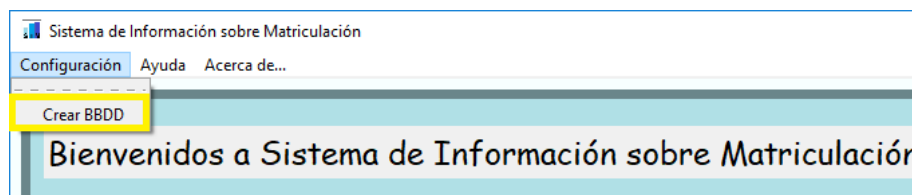


Figura E.1: Botón de creación de la Base de Datos

descargar o clonar el repositorio de *GitHub* desde el siguiente enlace¹.

E.4. Manual del usuario

En esta sección se van a describir todas las funcionalidades de la aplicación con numerosas figuras para entender el funcionamiento de la misma.

Creación de la Base de Datos (BBDD)

En primer lugar es necesario crear la Base de Datos denominada BBDD donde se almacena la estructura de tablas del proyecto. Para crearla únicamente es necesario pulsar sobre el Botón Crear BBDD situado en el menú superior de la aplicación, dentro de *Configuración* (figura E.1). Hay que destacar que este botón se ubica en el menú superior de la aplicación, ya que es un botón que se utilizará con poca frecuencia, ya que únicamente es necesario crear la BBDD la primera vez que se ejecuta la aplicación (si no tuviéramos ya creado un archivo BBDD en la ruta donde se ejecuta la aplicación).

Al pulsar sobre este botón se creará la Base de Datos y por consiguiente el fichero BBDD en la ruta donde estemos ejecutando la aplicación. Si es la primera vez que pulsamos este botón, se creará con éxito; pero si ya existiera la BBDD, la aplicación nos mostraría un mensaje de *Warning* como el de la figura E.2. De esta forma, no se volverá a crear la Base de Datos y mantendremos nuestra BBDD anterior.

Preprocesado de los ficheros Excel (.xls) descargados de *Sigma*

Una vez creada la BBDD, necesitamos introducir o meter los datos. Para esta labor, se utilizan ficheros de datos de matriculación de alumnos

¹www.github.com/mdi0007/Sistema-Informacion-sobre-Matriculacion

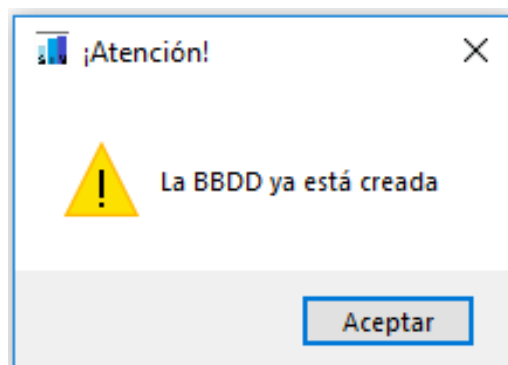


Figura E.2: Mensaje de *Warning* de BBDD ya creada

descargados de la aplicación *Sigma*.

Ya se han explicado los problemas que existen con estos tipos de ficheros, por esta razón, se deben parsear y preprocesar antes de añadir a la BBDD. Para esta tarea, contamos con el botón de *Preprocesado*. Este botón se ubica en la pantalla principal de la aplicación, ya que es un botón que se utilizará con cierta normalidad y periodicidad.

Cuando pulsemos el botón de *Preprocesado* (E.3), se mostrará una ventana del Sistema Operativo, donde se nos permitirá seleccionar archivos (.xls) únicamente (E.4).

Seleccionamos el archivo que deseemos (deben ser archivos descargados de *Sigma*) y pulsamos abrir. Una vez realizado esto, se generará en la misma ruta un fichero con extensión (.csv) corregido, modificado y preparado para la carga de datos en la BBDD. El resumen del proceso de preprocesado de ficheros *Sigma* se puede apreciar en la figura E.5.

Carga de datos en la Base da Datos (BBDD)

En este punto, ya podemos realizar la carga de datos. Para eso contamos con un botón de *Cargar Archivos* situado también en la pantalla principal de la aplicación (E.3).

De la misma forma que el anterior paso, cuando pulsamos este botón, se mostrará una ventana del Sistema Operativo, donde se nos permitirá seleccionar archivos (.csv) únicamente (E.6).

Seleccionamos el archivo que se ha generado en el paso anterior y pulsamos abrir. Una vez realizado esto, se procederá a realizar la carga de todos los

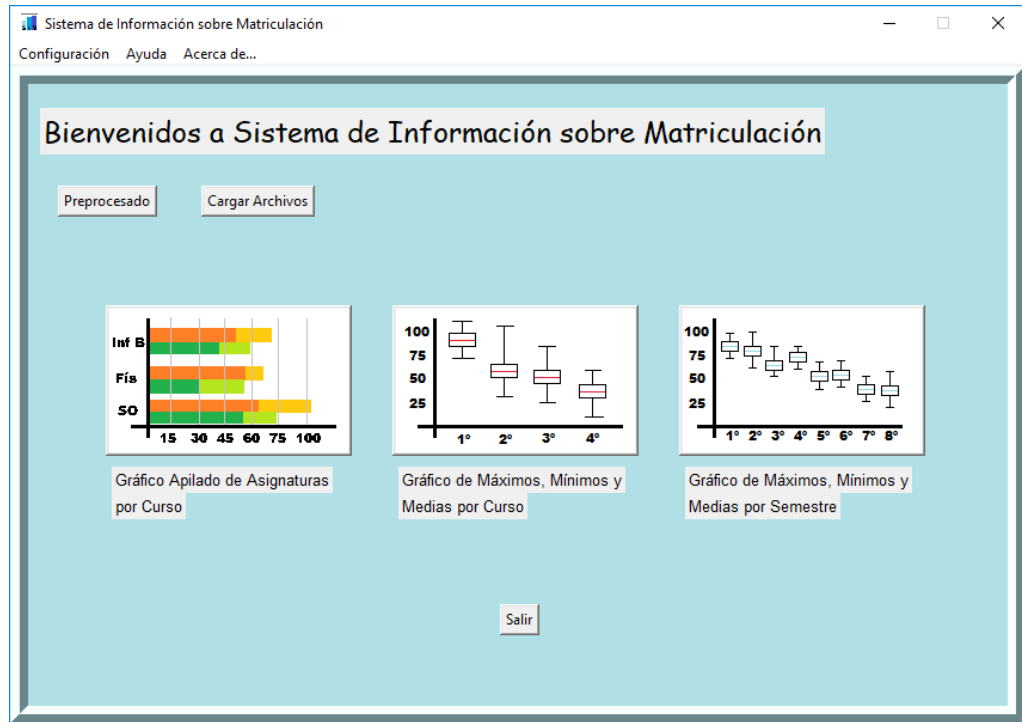


Figura E.3: Pantalla principal de la aplicación

datos del fichero a la BBDD siguiendo las normas de claves primarias definidas en el modelo ER de la aplicación. El resumen del proceso de carga de datos se puede apreciar en la figura [E.7](#).

Selección y personalización de los gráficos

En este punto, ya contamos con información o datos suficientes para realizar los diferentes tipos de gráficos. Por lo tanto pulsamos en una de las imágenes (botones) de la pantalla principal ([E.3](#)) de los 3 tipos de gráficos (el primero por ejemplo) y se nos abrirá una nueva ventana ([E.8](#)).

En esta ventana, se muestran una serie de desplegables con la información necesaria para que el usuario seleccione y escoja los parámetros para generar el gráfico específico que desee. Hay que destacar que en cada desplegable se muestra toda la información diferente que se encuentre en la BBDD.

Para obtener correctamente el gráfico, es necesario seleccionar una opción de cada desplegable. En este ejemplo, es decir, en el tipo de gráfico 1, es necesario seleccionar cuatro parámetros, que son los siguientes ([E.8](#)):

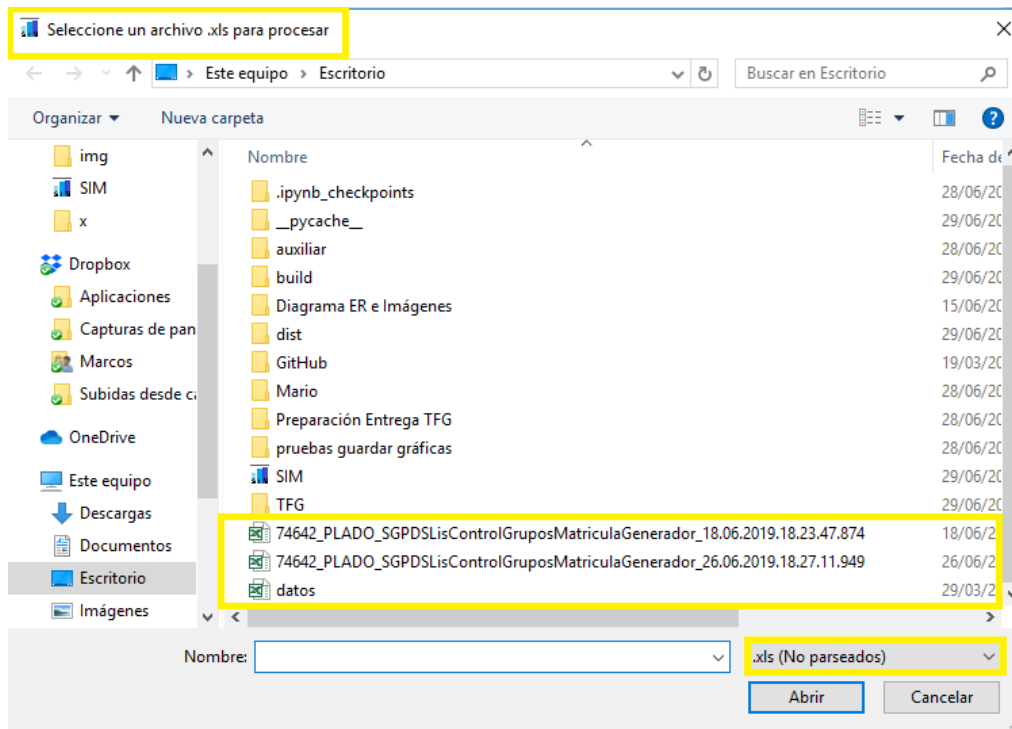
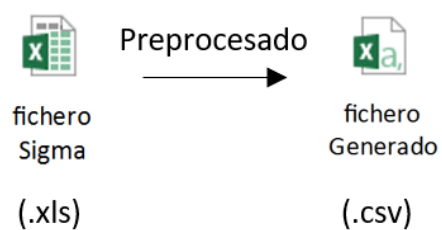


Figura E.4: Ventana del Sistema Operativo (seleccionar .xls)

Figura E.5: Resumen de botón *Preprocesado*

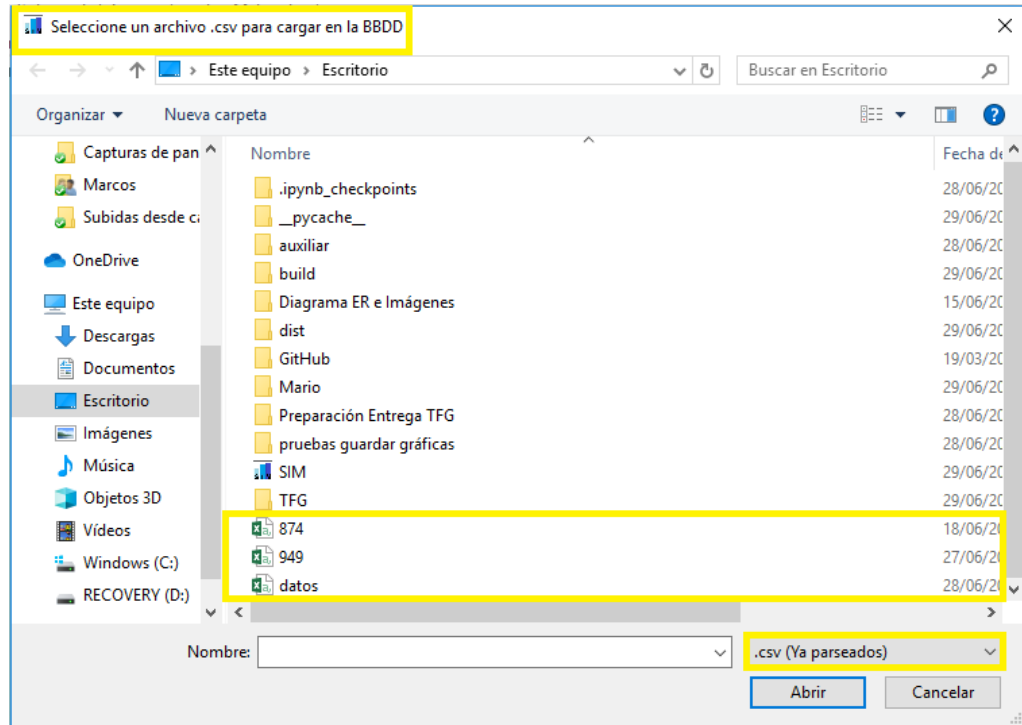
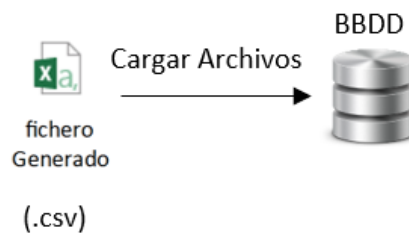


Figura E.6: Ventana del Sistema Operativo (seleccionar .csv)

Figura E.7: Resumen de botón *Cargar Archivos*

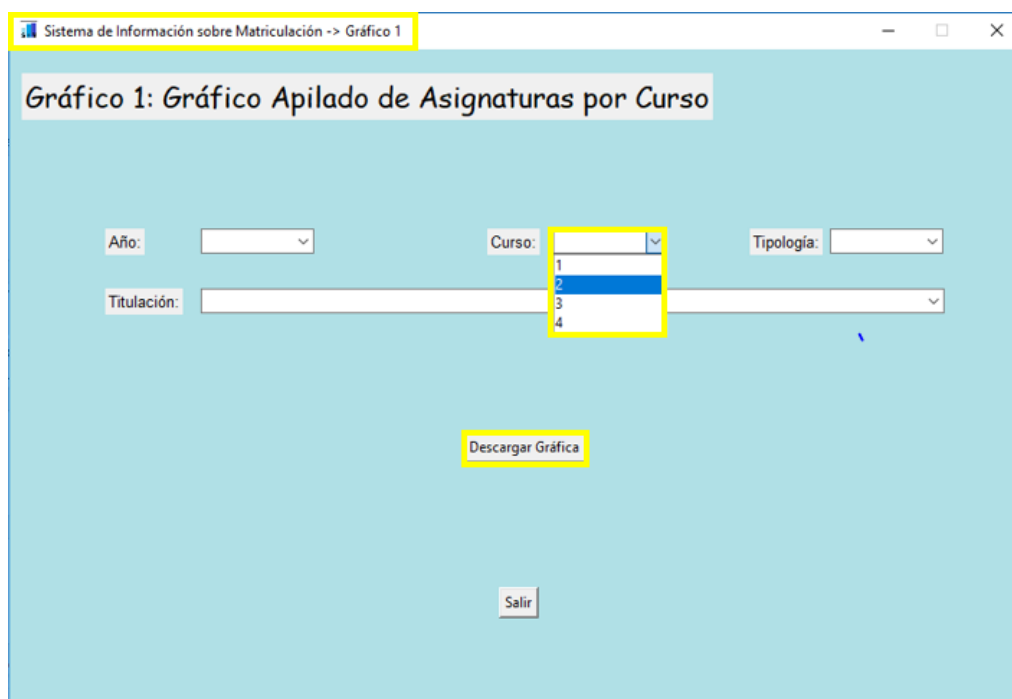


Figura E.8: Ventana secundaria para configurar el gráfico 1

- **Año.** Año académico que se desea visualizar en la gráfica.
- **Curso.** Curso que se desea visualizar. Los posibles valores son cuatro cursos (1º, 2º, 3º y 4º) en el caso de disponer de información sobre los cuatro cursos en la BBDD.
- **Tipología.** Tipología académica que se quiere visualizar. Los posibles valores son dos tipologías (Teoría y Prácticas).
- **Titulación.** Nombre de la titulación o plan que se desea obtener el gráfico.

Una vez hayamos escogido estos cuatro parámetros, podemos proceder a la descarga del gráfico.

Como se aprecia en la figura E.9, para los tipos de gráfico 2 y 3, es necesario seleccionar únicamente dos parámetros, que ya han sido explicados anteriormente, que son:

- **Año.**



Figura E.9: Ventana secundaria para configurar el gráfico 2

- **Titulación.**

Descarga de los gráficos

Para descargar el gráfico, únicamente debemos pulsar el botón de *Descargar Gráfica* situado en las ventanas secundarias de configuración de los diferentes tipos de gráficos. Este botón se puede apreciar en las figuras [E.8](#) y [E.9](#).

Una vez pulsado dicho botón se descargará la gráfica con extensión (.png) en la ruta donde estemos ejecutando la aplicación. La nomenclatura de estos archivos es la siguiente:

- **Gráfico 1.** *Grafica1* seguido de una *T* (si se ha seleccionado la tipología de teoría) o una *P* (si se ha escogido la tipología de prácticas), seguido de *__curso_x* (siendo *x* el tipo de curso seleccionado y por último el plan o titulación).

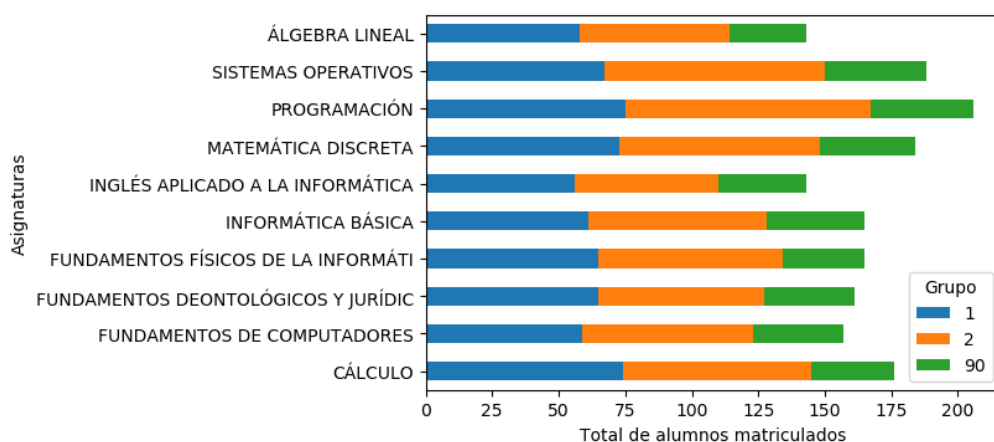


Figura E.10: Gráfico 1: Gráfico Apilado de Asignaturas de año 2018-2019 de 1º de Grado en Ingeniería Informática de Tipología Teoría

- **Gráfico 2.** *Grafica2_cursos_* seguido del plan o titulación seleccionado.
- **Gráfico 3.** *Grafica3_semestres_* seguido del plan o titulación seleccionado.

Gráficos obtenidos

Los diferentes gráficos principales que se pueden obtener gracias a la aplicación son los siguientes: [E.10](#), [E.11](#), [E.12](#) y [E.13](#).

Otras funcionalidades

Hay que señalar que la aplicación cuenta con una opción de *Ayuda* en el menú superior. Al pulsar este botón se desplegarán dos opciones (*Ayuda Local* y *Ayuda Web*) como se aprecia en la figura [E.14](#). Si pulsamos la primera opción, nos abrirá el (.pdf) de ayuda que se incorpora en la aplicación; y si pulsamos la segunda opción nos abrirá el mismo (.pdf) de ayuda pero dicho archivo se encuentra subido en el repositorio de GitHub. De esta manera, mediante esta segunda opción, se podría actualizar la información relevante a la *Ayuda* de manera inmediata para el usuario.

Para finalizar, hay que comentar que la aplicación cuenta con un botón de *Acerca de...*, situado a la derecha del botón de ayuda, como se aprecia en

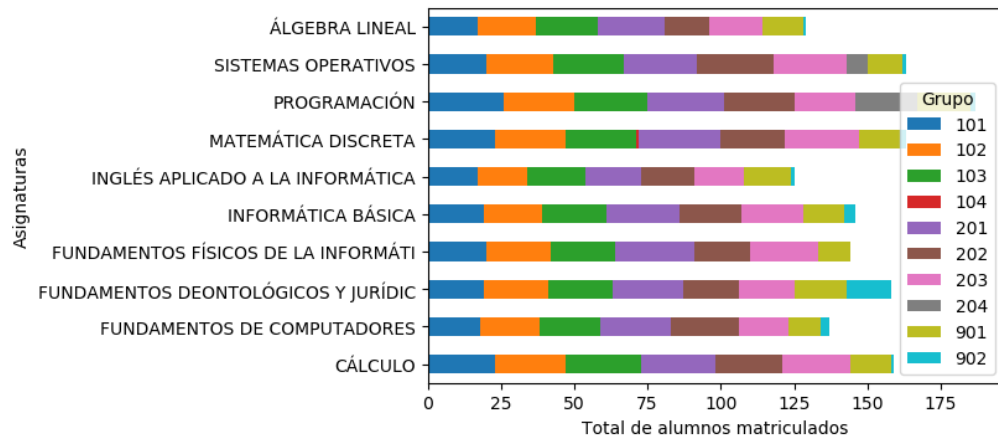


Figura E.11: Gráfico 1: Gráfico Apilado de Asignaturas del año 2018-2019 de 1º del Grado en Ingeniería Informática de Tipología Prácticas

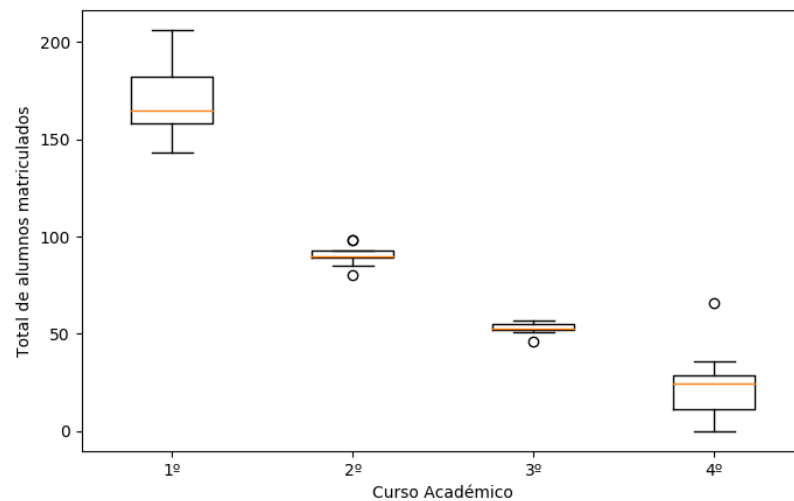


Figura E.12: Gráfico 2: Gráfico Apilado de Máximos, Mínimos y Medias por Curso del año 2018-2019 del Grado en Ingeniería Informática

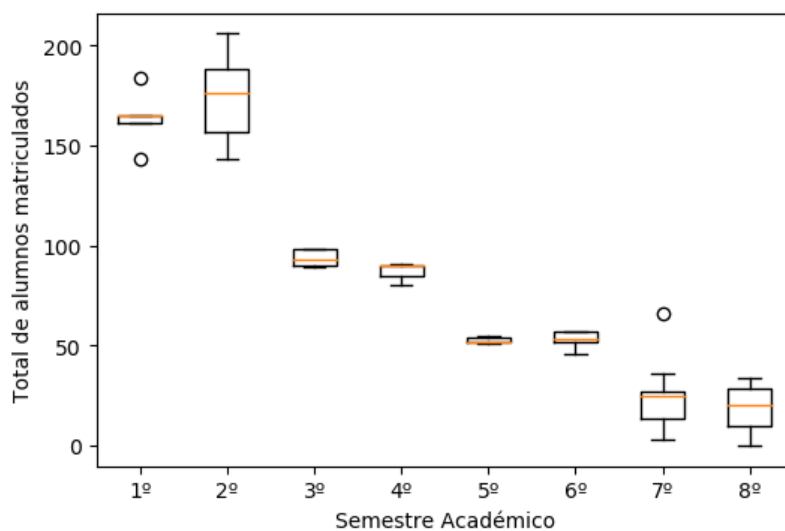


Figura E.13: Gráfico 3: Gráfico Apilado de Máximos, Mínimos y Medias por Semestre del año 2018-2019 del Grado en Ingeniería Informática

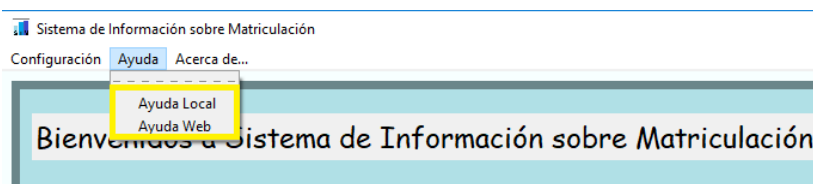


Figura E.14: Opción de *Ayuda*

la figura E.14.

Si pulsamos sobre este botón nos mostrará información relevante del proyecto (nombre, logotipo, autor, tutor, versión, licencia...etc) como se aprecia en la figura E.15.



Sistema de Información sobre Matriculación



Autor: Mario de la Parte Izquierdo

Tutor: Carlos Pardo Aguilar

Versión 1.0

Copyright (c) 2019 Mario de la Parte Izquierdo

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

Figura E.15: Contenido de *Acerca de...*

Bibliografía

- [1] Wikipedia. Gnu general public license — wikipedia, la enciclopedia libre, 2019.
- [2] Wikipedia. Python software foundation license — wikipedia, la enciclopedia libre, 2019.