



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 — 6 de junio
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	7
Apéndice B Especificación de Requisitos	9
B.1. Introducción	9
B.2. Objetivos generales	9
B.3. Catalogo de requisitos	9
B.4. Especificación de requisitos	11
Apéndice C Especificación de diseño	17
C.1. Introducción	17
C.2. Diseño de datos	17
C.3. Diseño procedimental	23
C.4. Diseño arquitectónico	23
Apéndice D Documentación técnica de programación	25
D.1. Introducción	25
D.2. Estructura de directorios	25
D.3. Manual del programador	25

D.4. Compilación, instalación y ejecución del proyecto	25
D.5. Pruebas del sistema	25
Apéndice E Documentación de usuario	27
E.1. Introducción	27
E.2. Requisitos de usuarios	27
E.3. Instalación	27
E.4. Manual del usuario	27

Índice de figuras

C.1. Modelo Entidad-Relación(ER)	18
C.2. Tabla ASIGNATURAS	20
C.3. Tabla GRUPOS	21
C.4. Tabla PROFESORES	22

Índice de tablas

A.1. Horas empleadas en el proyecto.	7
A.2. Costes de personal.	7
B.1. Caso de uso 1: Importar Datos	12
B.2. Caso de uso 2: Gráfico apilado de Asignaturas por Curso	13
B.3. Caso de uso 3: Gráfico de máximos, mínimos y medias por curso	14
B.4. Caso de uso 4: Crear Base de Datos	15

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 18 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 16 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 10 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 20 horas de trabajo.

Sprint 10

Sprint 11

Resumen

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

Tarea	Tiempo (horas)
<i>Documentación</i>	60
<i>Características</i>	x
<i>Investigación</i>	50
<i>Corrección de errores</i>	x
Total	x

Tabla A.1: Horas empleadas en el proyecto.

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 mensual neto	1000
Retención IRPF (9,65 %)	114,35
Seguridad Social (29,9 %)	359,65
Salario mensual bruto	1300,00
Total 3 meses	3900

Tabla A.2: Costes de personal.

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 Precondiciones	RF-1 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

Caso de uso 4: 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.4: Caso de uso 4: Crear Base de Datos

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 se aprecia el modelo ER o Diagrama Relacional de la Base de Datos de la aplicación:

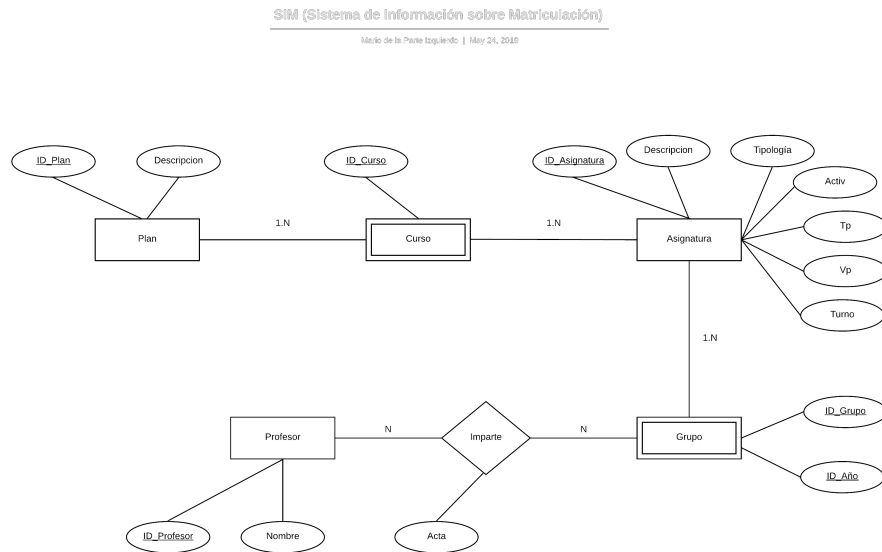


Figura C.1: Modelo Entidad-Relación(ER)

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:

- **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:** Es un campo de caracteres.
 - **Activ:** Es un campo de caracteres.
 - **Tp:** Es un campo de caracteres.
 - **Vp:** Es un campo numérico de 1 dígito.
 - **Turno:** Es un campo de caracteres.

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

- **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 capos (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.



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

Figura C.3: Tabla GRUPOS

- **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:** 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.

 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

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