



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

Data warehouse Sistema de  
ayuda para la asignación en  
sigma



Presentado por Álvaro Urdiales Santidrián  
en Universidad de Burgos — 4 de febrero  
de 2020

Tutor: Carlos Pardo Aguilar







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Carlos Pardo Aguilar, profesor del departamento de ingeniería civil, área de lenguajes y sistemas informáticos.

Expone:

Que el alumno D. Álvaro Urdiales Santidrián, con DNI 71309907V, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Data warehouse Sistema de ayuda para la asignación en sigmade TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 4 de febrero de 2020

Vº. Bº. del Tutor:

D. Carlos Pardo Aguilar





## **Resumen**

Poder obtener información útil de un conjunto de datos es algo que ahorra mucho tiempo.

Los profesores, para conocer información sobre las horas que destinan a las docencias y los diferentes grados, solo poseen una herramienta que solo ofrece una tabla con gran cantidad de datos desorganizados y con problemas de visualización ya que se generan errores al abrir el archivo.

Con este proyecto se ofrece una solución a este problema permitiendo la lectura de este archivo, dando la posibilidad de verlo de forma más organizada y útil para una mejor comprensión de todos estos datos.

## **Descriptores**

Sistema de información, data warehouse, base de datos, gestión de información, aplicación web.

### **Abstract**

Being able to obtain useful information from a data set is a great time-saver.

Nowadays the teachers to know information about the hours they spend on teaching and the different grades, only have a tool that only offers a table with a lot of disorganized data and with visualization problems because it has errors when you try to open the file.

This project offers a solution to this problem by allowing the reading of this file, giving the possibility to see it in a more organized and useful way for a better understanding of all this data.

### **Keywords**

Information system, data warehouse, database, information management, web application.



---

# Índice general

---

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales . . . . .	3
2.2. Objetivos técnicos . . . . .	3
Conceptos teóricos	5
3.1. Sistema de información . . . . .	5
3.2. <i>User Experience UX</i> . . . . .	9
Técnicas y herramientas	13
4.1. Metodologías . . . . .	13
4.2. Patrones de diseño . . . . .	13
4.3. Control de versiones y hosting de repositorio . . . . .	14
4.4. Comunicación . . . . .	14
4.5. Entorno de desarrollo integrado (IDE) . . . . .	15
Aspectos relevantes del desarrollo del proyecto	17
5.1. Inicio del proyecto . . . . .	17
5.2. Metodologías . . . . .	18
5.3. Formación . . . . .	18
5.4. Desarrollo de la Base de datos . . . . .	19

5.5. Desarrollo del programa . . . . .	21
5.6. Desarrollo de la interfaz . . . . .	24
<b>Trabajos relacionados</b>	<b>27</b>
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>29</b>
7.1. Conclusiones . . . . .	29
7.2. Líneas de trabajo futuras . . . . .	30
<b>Bibliografía</b>	<b>31</b>

---

# Índice de figuras

---

1.1. Error documento corrupto. . . . .	1
1.2. Excel obtenido en sigma. . . . .	2
3.3. Esquema de un sistema de información.[16] . . . . .	6
4.4. Modelo vista presentador. . . . .	14
5.5. Diagrama entidad relación. . . . .	20

---

# Índice de tablas

---

---

# Introducción

---

La información consiste en un conjunto de datos procesados que constituirán un mensaje que cambiará el estado del conocimiento del sujeto que recibirá dicho mensaje. Por otro lado, podríamos considerar a la información como conocimiento explícito el cual a diferencia de los datos, posee estructura útil.[13]

Actualmente, los profesores poseen una herramienta llamada sigma, de la cual obtienen todos los datos necesarios para la gestión y el control de los horarios en las diferentes carreras. Esto suponía un problema a la hora de obtener información de esos datos, puesto que no solo se obtenían grandes tablas con numerosos números y datos 1.2, sino que el archivo excel en el que se encontraban estaba corrupto 1.1, lo cual producía mayor dificultad a la hora de intentar extraer información de todos esos datos.

Para solucionar estos problemas y poder automatizar tanto la extracción de información como el almacenamiento de la misma, se propuso la idea de crear un sistema de información, el cual realizaría las tareas de lectura de las hojas de calculo corruptas, como de almacenamiento de forma lógica y ordenada de los datos y obtención de información a partir de los mismos.

Debido a que no existe ningún sistema similar, para la obtención de información, esta herramienta podría ahorrar numerosos costes en cuanto a tiempo a los profesores que requieran obtener información útil de cualquier

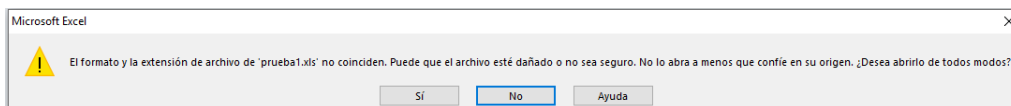


Figura 1.1: Error documento corrupto.

	A	B	C	D	E	F	G	H	I	J	
1	CARGA DOCENTE POR ESTUDIO										
2	Centro:	13 - Escuela Politécnica Superior									
3	Estudio:	63 GRADO EN INGENIERÍA INFORMÁTICA									
4	Curso académico:	2018-19									
5	Fecha de ejecución:	11/02/2019 11:41	*) Asignatura continua, la docencia se realiza en los trimestres indicados								
6											
7	C.A.	Asignatura	Curso	Tp/Vp 18/19	H.Doc. Asig.	H.Doc.Tp/gr	Tp. Doc.	Grupo	C. prof.	Profesor	
8	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	1	34802	GARCIA REGUI		
9	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	1	1000058	BOL ARREBA, J		
10	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	1	1000231	CALVO RATHER		
11	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	101	1000231	CALVO RATHER		
12	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	102	1000231	CALVO RATHER		
13	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	103	1000231	CALVO RATHER		
14	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	2	34802	GARCIA REGUI		
15	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	2	1000058	BOL ARREBA, J		
16	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	2	1000231	CALVO RATHER		
17	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	201	34802	GARCIA REGUI		
18	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	202	1000058	BOL ARREBA, J		
19	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	203	1000058	BOL ARREBA, J		
20	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	90	1000053	BLANCO MON'		
21	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	1	90	1000058	BOL ARREBA, J		
22	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	901	1000053	BLANCO MON'		
23	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	901	1000058	BOL ARREBA, J		
24	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	902	1000053	BLANCO MON'		
25	6347	FUNDAMENTOS FÍSICOS DE I. I.	SEMESTRE: 1	54	27	2	902	1000058	BOL ARREBA, J		
26	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	1	1	1000656	TORRES CABRE	
27	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	101	1000656	TORRES CABRE	
28	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	102	1000656	TORRES CABRE	
29	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	103	1000656	TORRES CABRE	
30	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	1	2	1000654	SEVILLA GALLC	
31	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	1	2	1000656	TORRES CABRE	
32	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	201	1000656	TORRES CABRE	
33	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	202	1000654	SEVILLA GALLC	
34	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	203	1000656	TORRES CABRE	
35	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	1	90	28423	REGUERA LOPI	
36	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	1	90	100946	LORENTE MAR	
37	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	901	28423	REGUERA LOPI	
38	6345	ÁLGEBRA LINEAL	1	SEMESTRE: 1	54	27	2	902	28423	REGUERA LOPI	
39	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	1	1	49086	LARA PALMA, J	
40	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	1	1	56942	SANCHEZ SAIZ,	
41	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	101	49086	LARA PALMA, J	
42	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	101	56942	SANCHEZ SAIZ,	
43	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	102	49086	LARA PALMA, J	
44	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	102	56942	SANCHEZ SAIZ,	
45	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	103	49086	LARA PALMA, J	
46	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	2	103	56942	SANCHEZ SAIZ,	
47	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	1	2	49086	LARA PALMA, J	
48	6344	FUNDAMENTOS DEONTOLÓG	1	SEMESTRE: 1	54	27	1	2	49086	SANCHEZ SAIZ,	

Figura 1.2: Excel obtenido en sigma.

grado, almacenar información de su grado o simplemente ahorrarse los problemas a la hora de intentar abrir el excel que proporciona sigma.

---

# Objetivos del proyecto

---

El objetivo principal del proyecto consiste en la creación de un data warehouse, con la finalidad de analizar el número de horas que posee cada profesor, para así llevar un control evitando que algunos se encuentren muy descompensados con respecto a otros.

## 2.1. Objetivos generales

- Dadas una serie de hojas de calculo (.xlsx), deberá permitir importarlas analizando y asignando los datos al departamento, y titulación a la que pertenecen los diferentes datos.
- Si se da el caso de que la información importada es una actualización de la información existente, deberá comparar ambas informaciones, mostrando las diferencias y dando la opción de sobrescribir los datos antiguos por los nuevos o desechar lo nuevo.
- Una vez importados los datos, deberá permitir filtrar toda la información que posea para su posterior búsqueda.
- Tras esta búsqueda de información, deberá mostrar la información buscada permitiendo una mejor comprensión de la misma.

## 2.2. Objetivos técnicos

- Desarrollar un analizador gramatical(parse) para los excels entregados capaz de reconocer los datos que se encuentran en la hoja de cálculo.

- Desarrollar una base de datos capaz de recoger todos estos datos, organizándolos de forma correcta, pudiendo acoger gran cantidad de datos.
- Desarrollar una aplicación que posea una parte visual, aplicando lenguajes web como HTML y CSS.
- Uso de Git como herramienta de control de versiones.
- Uso de CherryPy como framework para poder realizar una aplicación web orientada a objetos usando Python como lenguaje.
- Uso de sistema gestor de bases de datos como MySQL para la búsqueda, ordenación, calculo y almacenamiento de la información.
- Uso de librerías en Python para la conexión con la base de datos, el tratamiento de datos y la visualización de los mismos.



---

## Conceptos teóricos

---

En el proyecto, podemos dividir en dos partes la teoría a explicar. Por un lado la explicación de lo que es, en que consiste y las variantes que existen y por otro debido a la implementación de una interfaz en el, que es y en que consiste el *User Experience* (UX).

### 3.1. Sistema de información

Un sistema de información (SI) [3.3](#), es un conjunto de mecanismos ordenados los cuales administran datos e información para poder recuperar o procesar estos datos o información de forma rápida y sencilla.[\[7\]](#)

En lo referente a la informática, un sistema de información ayudará en la administración, recolección, recuperación, procesamiento, almacenamiento y distribución de información importante para la organización que la vaya a utilizar. Su importancia reside en su eficiencia para a partir de una gran cantidad de datos, producir información válida para posteriormente poder ayudar en la toma de decisiones.[\[2\]](#)

### Características

Un sistema de información es caracterizado principalmente por su eficiencia en el procesamiento de datos con respecto al área de acción. Además, son alimentados por procesos y herramientas de diversos temas tales como: estadística, probabilidad, producción, inteligencia de negocio, etc. Para así alcanzar la mejor solución.

Un sistema de información destaca principalmente por su diseño, usabilidad, flexibilidad, automatización en el mantenimiento de los registros, ayuda

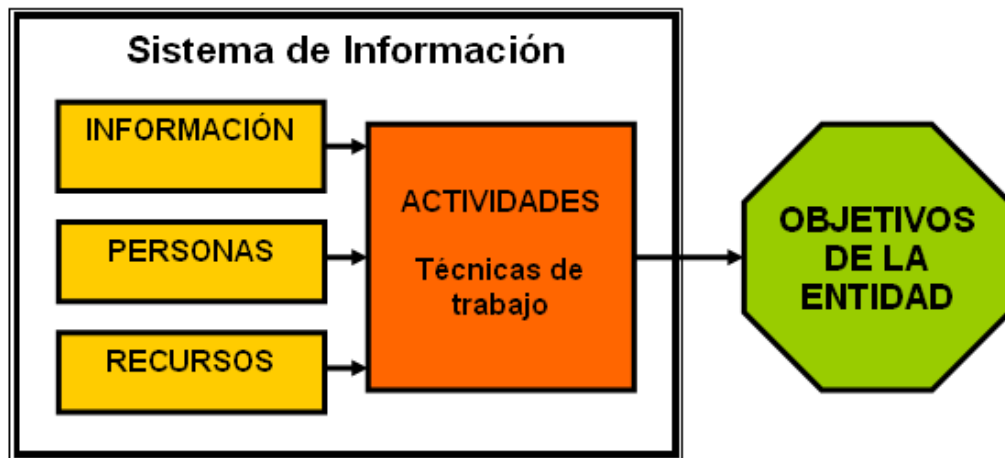


Figura 3.3: Esquema de un sistema de información.[16]

en la toma de decisiones y mantenimiento del anonimato en la información no relevante.

## Recursos

En cualquier sistema de información podemos encontrar diferentes tipos de recursos interconectados y en correlación, de tal forma que mejoren la ayuda frente al problema hacia el que estén dirigidos.

Estos recursos pueden ser de diferentes tipos como:

- **Recursos humanos:** Personal de variada índole y destrezas.
- **Datos:** Cualquier tipo de información que precise organizarse.
- **Actividades:** Pasos a seguir, procedimientos, estaciones de trabajo, etc.
- **Recursos informáticos:** Aquellos determinados por la tecnología.

Hay que tener en cuenta las diferencias entre un sistema de información y un sistema informático, ya que aunque este último suele constituir el grueso de un SI, existen otros métodos para los sistemas de información los cuales no pasen por la informática.

## Componentes

Un sistema de información debe poseer una serie de componentes básicos los cuales estarán interactuando entre si:

- **Hardware:** Consiste en el equipo físico utilizado para el procesamiento y almacenamiento de los datos.
- **Software:** Formado por los procedimientos utilizados para la extracción y transformación de la información y el propio software.
- **Datos:** Son los elementos que representan las actividades de la empresa.
- **Red:** Elemento que permite compartir recursos entre los ordenadores y los dispositivos.
- **Personas:** Son las encargadas del desarrollo, mantenimiento y uso del sistema.

Por otro lado, podemos agrupar los componentes de un sistema de información en tres dimensiones:

- **Dimensión organización:** formada por la estructura de la organización.
- **Dimensión personas:** encargada de fabricar y producir la colaboración necesaria que hace que el sistema funcione.
- **Dimensión tecnología:** referida a la implementación de la estructura hardware y software.

## Actividades

Para producir la información necesaria que las organizaciones necesitan para tomar decisiones, controlar las operaciones, analizar problemas y crear nuevos productos o servicios en un sistema de información existen cuatro actividades:

- **Recopilación:** encargada de la recogida de datos tanto internos como del entorno a la organización.
- **Almacenamiento:** encargada de guardar la información de forma estructurada y ordenada la información recogida.

- **Procesamiento:** encargada de la conversión de toda esa información almacenada en información mas significativa.
- **Distribución:** encargada de transferir toda la información procesada a las personas que lo utilizarán.

Los sistemas de información suelen requerir de una retroalimentación lo cual consiste en la salida que devuelve al personal adecuado de la organización ayudándolo a corregir o evaluar la etapa de entrada. También deben ser evaluados en función de su utilidad o calidad mediado por el uso del sistema por parte del usuario y su satisfacción.

## Ciclo de vida

El desarrollo de un sistema de información es continuo y está formado por una serie de fases:

1. Investigación preliminar, identificación de fortalezas y amenazas.
2. Definición de las necesidades y requerimientos.
3. Diseño.
4. Desarrollo y documentación del software.
5. Pruebas.
6. Implementación y mantenimiento.
7. Identificación de debilidades y oportunidades.

## Tipos

Desde un punto de vista organizacional, existen numerosos tipos de sistemas de información dependiendo del nivel operacional en el que son utilizados. Algunos de los sistemas más comunes son:

- **Sistemas de Procesamiento de Transacciones (TPS):** conocidos como sistemas de gestión operativa, utilizados para la recopilación de información en cuanto al funcionamiento de la organización.
- **Sistemas de Información Ejecutiva (EIS):** utilizados para monitorizar variables gerenciales de un área específica de una organización recogiendo información interna y externa de la misma.

- **Sistemas dde Información Gerencial (MIS):** utilizados para recoger información general de una organización comprendiéndola como un todo.
- **Sistemas de Soporte de Decisiones (DSS):** utilizados para recopilar tanto información intra como extra organizacional ayudando así en la conducción de la empresa.

Existen muchos más tipos de sistemas de información mas especializados dependiendo del campo al que nos refiramos y de las funciones que se precisan de cada uno.

## Elementos

Generalmente se considera que todos los sistemas de información poseen una gran cantidad de elementos, los cuales se pueden clasificar en cinco grandes categorías:

- **Elementos financieros:** referidos al capital y activos de la organización.
- **Elementos tecnológicos:** referidos a la maquinaria especializada y capacidad de procesamiento automatizado de información.
- **Elementos humanos:** referidos al personal de la organización.
- **Elementos materiales:** referidos al soporte tanto físico como a la ubicación de la organización.
- **Elementos administrativos:** referidos a los procesos, la mecánica de conducción, los permisos, informes, transacciones, etc.

## 3.2. *User Experience UX*

*User experience* (UX) es el término con el que definimos el sentimiento positivo o negativo de una persona al interactuar con un sistema.[1] Este sentimiento no solo depende de los factores relativos al diseño, sino también de aspectos referidos a los sentimientos, emociones, confiabilidad, facilidad de uso, eficiencia, etc.[10]

La Experiencia de Usuario representa una ampliación del concepto usabilidad, incluyendo una perspectiva global que comprenda el sentimiento

del usuario al utilizar el producto. Esta definición se encuentra en continua evolución debido a que inicialmente era aplicada únicamente a la informática, en particular a las webs, pero actualmente se está empezando a aplicar a más campos.<sup>[3]</sup>

## Razones por las que UX es tan importante

Anteriormente únicamente se valoraba la estética del producto, sin embargo ahora se está empezando a valorar otros temas como la usabilidad por encima de la estética.

Hay numerosas razones para ello:

- Debido a la gran cantidad de opciones que se nos presenta cada día, interesa que el usuario sienta preferencia hacia tu producto frente a otros.
- Actualmente los usuarios cuentan con numerosos aparatos con los que pueden acceder a los servicios, es por ello que la UX debe tratar de armonizar la experiencia de los diferentes dispositivos, asegurándose un correcto funcionamiento en cada uno de ellos.
- Cada vez se valora más la accesibilidad, debido a que cuanto más accesible sea un producto hay mayor posibilidad de llegar al mayor público posible.
- Los usuarios cada vez se están volviendo más exigentes, por lo que crear un producto de calidad y único ya no es suficiente teniendo que crear algo que además haga que la experiencia de usuario sea inolvidable.

## Técnicas para lograr un buen UX

*User experience* combina numerosas habilidades, herramientas y técnicas mediante las cuales se pueden construir mejores sistemas. Algunas de las más habituales son:

- **Evaluar el sistema actual**, buscando así los puntos débiles a mejorar.
- **Tests A/B**, probando diferentes variantes de sistema para comprobar su correcto funcionamiento.

- **Encuestas**, preguntando a los usuarios que utilizan nuestro producto.
- **Wireframes y prototipos**, crear esquemas y prototipos basados en los pasos realizados anteriormente.
- **Flujos de usuario**, viendo todos los pasos que hace un usuario la primera vez que utiliza nuestro producto, obteniendo así el comportamiento de las personas.
- **Storytelling**, transmitiendo información y valores de la marca intentando así atraer a nuevos usuarios.
- **Patrones de diseño**, aportando consistencia y elementos de comparación entre diferentes productos, escogiendo los elementos adecuados en función del uso que lo vaya a dar el usuario, permitiéndole así moverse por el producto intuitivamente. Por ello, un buen experto en UX sabrá combinar todos estos patrones anteriores formando un diseño único y adecuado a las necesidades del proyecto.





---

# Técnicas y herramientas

---

## 4.1. Metodologías

### Scrum

Scrum es un proceso iterativo e incremental utilizado para la construcción de productos englobado dentro de las metodologías ágiles[6]. Basado en una estrategia de desarrollo iterativa e incremental, con continuas revisiones y reuniones[15].

## 4.2. Patrones de diseño

### Modelo-vista-presentador MVP

Patrón de diseño derivado del modelo vista controlador utilizado principalmente para la construcción de interfaces de usuario. Es similar al modelo-vista-controlador, con la diferencia de que el controlador desaparece dando paso al presentador, el cual es la capa intermedia la cual posee la lógica de presentación.

Este patrón facilita las pruebas de unidad automatizada y separa los conceptos, permitiendo una simplificación en el desarrollo y mantenimiento de los programas.

Posee 3 componentes distintos 4.4, los cuales interactúan entre ellos:[17].

- **Modelo:** encargado de almacenar y tratar los datos.
- **Vista:** encargada de la visualización del modelo y órdenes del usuario para actuar sobre los datos.



Figura 4.4: Modelo vista presentador.

- **Presentador:** intermediario entre la vista y el modelo que actúa sobre ellos. Recupera los datos del modelo, tratándolos para mostrarlos en la vista.

### 4.3. Control de versiones y hosting de repositorio

La herramienta utilizada para esta labor será **Git** con el repositorio **GitHub**.

#### Git

Software de control de versiones distribuido. Es el más utilizado en la actualidad albergando en el importantes proyectos. Entre sus características mas importantes, podemos encontrar que posee una gestión distribuida, por lo que varios usuarios pueden tener una copia del repositorio en su ordenador y permite una eficiente gestión de proyectos.[11]

#### GitHub

Plataforma de desarrollo colaborativo cuya finalidad consiste en alojar proyectos, basada en el control de versiones de **Git**. Entre sus características principales por las que ha sido elegido, podemos encontrar su gratuidad para proyectos *open source*, gráficos con información sobre cada proyecto, gestion de proyectos con estilo *Kanban*. [12]

### 4.4. Comunicación

La comunicación con el Tutor ha sido principalmente presencial, pero en algunas ocasiones se ha utilizado una herramienta para poder comunicarnos a distancia, **Skype Empresarial**.

## Skype Empresarial

Servicio de mensajería instantánea que permite crear salas de reuniones, compartiéndolas fácilmente mediante un link, compartir la pantalla y archivos, por lo que se convierte en una herramienta muy interesante para realizar reuniones.

## 4.5. Entorno de desarrollo integrado (IDE)

Dentro de los posibles entornos de desarrollo, tenía que encontrar uno que me permitiera realizar el trabajo tanto en **Python** como en HTML y CSS.

### SQL

Para ello planteo una serie de posibles entornos:

- **Entornos planteados:** **MySQL**, **Oracle DB**, **Mongo DB**.
- **Entorno elegido:** **MySQL**.

**Mongo DB** es un sistema de base de datos NoSQL[14], al contrario que **MySQL** y **Oracle DB**, los cuales son sistemas de bases de datos SQL. A la hora de decidirme entre el tipo de BD y la tecnología a utilizar me decidí por una tipo SQL, puesto que la idea de la aplicación consiste en manejar un gran volumen de datos y un sistema de bases de datos de tipo SQL posee tecnología más madura puesto que aporta mayor seguridad.

Dentro de los dos tipos de sistemas de bases de datos SQL, descarté **Oracle DB**, debido a que una licencia para explotar el producto no es gratuita, al contrario que **MySQL**.

### Python, HTML y CSS

Para ello planteo una serie de posibles entornos:

- **Entornos planteados:** **PyCharm**, **Spyder**.
- **Entorno elegido:** **Spyder**.

Para poder crear una aplicación con una parte gráfica, decidí que la mejor opción era crear una aplicación web. **Python** como tal no soporta la integración del lenguaje web, por lo que elegí **CherryPy** como framework para poder desarrollarlo.

Elegí **Spyder** como entorno de desarrollo debido a la facilidad de instalación del framework frente a **PyCharm**, ya que esta daba numerosos errores en el proceso de instalación.

## **LaTeX**

Debido al conocimiento previo de un editor gratuito para la escritura de los documentos requeridos para el TFG en **LaTeX**, mi primera y única opción para esta tarea fue **TeXstudio**.

**TeXstudio** es un entorno de escritura gratuito con numerosas cualidades tales como la posibilidad de resaltar sintaxis, un visor integrado, verificación de referencias, varios cursores, posibilidad de autocompletar, etc.

---

# Aspectos relevantes del desarrollo del proyecto

---

Este apartado recogerá los aspectos mas interesantes del desarrollo del proyecto, incluyendo desde la exposición del ciclo de vida y las fases de diseño e implementación incluyendo también los posibles problemas que han surgido durante la realización del mismo.

## 5.1. Inicio del proyecto

Me interesé por el proyecto debido a las posibilidades de aprendizaje que este ofrecía, a mi interés por las bases de datos y poder profundizar mas en ellas. La base del proyecto consiste en una gran base de datos la cual almacenará gran cantidad de datos sobre los diferentes grados, profesores y asignaturas.

Debido a que he cursado asignaturas en las que nos enseñan a conectar una base de datos con un lenguaje (**Java**), pudiendo tratar la base de datos realizando inserciones y consultas, me pareció interesante proponer **Python** como lenguaje, permitiéndome así aprender más sobre ello.

Por otro lado aunque el proyecto no requería de una interfaz visual, me pareció interesante la idea de formarme en ello y poder aplicar los conocimientos sobre ux (*User Experience*) dotando así al proyecto de una interfaz.

## 5.2. Metodologías

Para la realización del proyecto, se intentó seguir una de las metodologías dadas durante la carrera, la cual es utilizada en numerosas empresas hoy en día y que ayuda a reducir la complejidad en el desarrollo, satisfaciendo las necesidades del cliente promoviendo la colaboración[15].

Es por ello que durante el proyecto se trato de seguir estas tecnologías ágiles, en concreto Scrum, aunque debido a ser un proyecto educativo y al haber unicamente una persona trabajando en el, no se pudo seguir al pie de la letra.

Para ello, se siguieron una serie de pautas basadas en esta metodología:

- Seguimiento del proceso mediante *Sprints*, cuya duración seria de una semana.
- Desarrollo incremental en el que se realizaban *Sprints* y revisiones.
- En las reuniones de finalización e inicio de los *Sprints*, se revisaba el producto entregado y se planificaba el nuevo *Sprint*, otorgando una serie de tareas a realizar.

Para el diseño del programa se utilizaban metodologías basadas en el ensayo y error, buscando así la eficiencia y eficacia de los algoritmos.

## 5.3. Formación

Debido a que para la realización del proyecto no tenia conocimientos necesarios en alguna de las materias del mismo, he tenido que buscar información sobre cada uno de los temas.

Debido a que quería utilizar **Python** como lenguaje para ampliar mis conocimientos sobre bases de datos, y que quería incluir una interfaz gráfica, utilicé **CherryPy** como framework, ya que me ofrecía una forma sencilla y flexible de poder crear una aplicación web en **Python**.

Para formarme en **CherryPy** utilicé principalmente la documentación de la propia web del framework[8].

## 5.4. Desarrollo de la Base de datos

La base de datos es la parte principal del proyecto, para su desarrollo, se pasó por una serie de etapas.

### Diseño

En la etapa de diseño es donde se repasan los requisitos que debe tener el proyecto y los tipos de datos que vamos a poseer. Para ello, es necesario conocer muy bien cuales son los requerimientos bien documentados, o incluso generar un flujo de trabajo de la aplicación para conocer bien todos los procesos que vamos a requerir.

Esta etapa la podemos dividir en dos partes: diseño conceptual y diseño lógico[5].

### Diseño conceptual

Esta es la etapa de diseño más básica, puesto que únicamente generaremos las diferentes tablas con sus respectivas relaciones entre ellas. Para ello, seguiremos un orden:

1. Primero generaremos las diferentes entidades(tablas), que tendrá nuestra base de datos.
2. Una vez creadas, iremos añadiendo los atributos que poseerán cada una de ellas, indicando también cual de todos ellos será el que identificara a la entidad.
3. Por último, crearemos las relaciones entre las entidades.

En mi caso, debido a que ya se me daba un archivo *xlsx*, ya tenía todos los atributos de la base de datos, por lo que únicamente tenía que organizarlos en tablas bien relacionadas entre ellas de forma que quedara organizado y pudiera guardar un gran volumen de datos 5.5.

Uno de los problemas de esta etapa fue creer que no iba a tener un gran volumen de datos, por lo que guardar la tabla que me daban tal cual me pareció una buena opción. De ahí, una vez supe que iba a almacenar una gran cantidad de datos, y que por lo tanto al guardarlo así, una gran cantidad de ellos iban a estar duplicados y generarían problemas de cohesión y rapidez (al carecer de ella), planteé un nuevo diseño de base de datos dividido en numerosas tablas evitando así estos problemas.

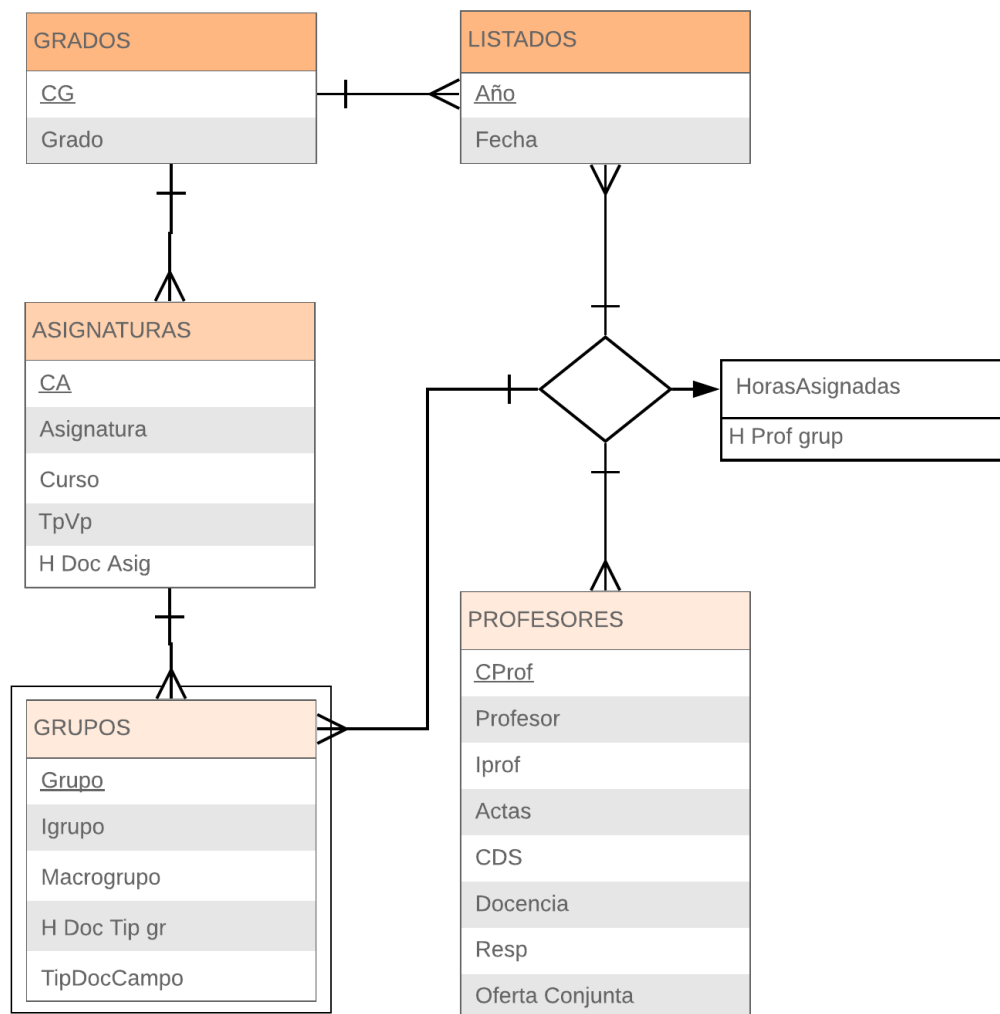


Figura 5.5: Diagrama entidad relación.



### Diseño lógico

Esta etapa es la encargada de normalizar las tablas creadas anteriormente, evitando aspectos negativos como elementos duplicados, evitar redundancias y garantizar la integridad de la base de datos.

Por ello es en esta parte donde comprobaremos las posibles dependencias, los tipos de datos que vamos a manejar, y que atributos son necesarios en cada tabla. Esta parte es bastante importante, puesto que si aquí no se hace correctamente, en la fase de implementación nos va a generar problemas.

### Implementación

Esta fase coincidiría con el diseño físico. Es aquí donde escribiremos en el entorno elegido(**MySQL**), todo lo diseñado anteriormente.

Debido a que el diseño no estaba completamente depurado, en mi caso surgieron una serie de problemas, los cuales fui depurando y arreglando.

## 5.5. Desarrollo del programa

El lenguaje elegido para ello fue **Python**. A la hora de realizarlo, podemos dividir el proceso en varias partes:

### Lectura de datos

Para poder introducir nuevos datos el proyecto deberá ser capaz de leer un archivo **xlsx**, el cual contendrá una tabla con todos los posibles datos sobre los profesores, las asignaturas y las horas.

Debido a la gran cantidad de librerías en **Python**, existen múltiples opciones para realizar esta labor. Ya que una vez leídos los datos, estos hay que procesarlos y prepararlos para subirlos a la base de datos, guardarlos en forma de **DataFrame** puede ser una muy buena opción.

Por ello, para la lectura de datos inicialmente intenté utilizar la librería de **pandas**, la cual ofrece una función `pandas.read_excel()`, la cual nos permitirá transformar a un **DataFrame** los datos del excel.

A la hora de intentarlo, me di cuenta de que el excel del que se deben leer los datos, estaba corrupto, lo que dificultaba la lectura del mismo. La única posible solución para poderlo leer consistiría por lo tanto en crear un *parser*, el cual leyera dato a dato el documento. Para ello genere una carpeta

temporal, donde se guardará un archivo .csv, en el cual se almacenaran estos datos para posteriormente leerlos con **pandas**, transformándolo así en un **DataFrame** y pudiéndolo formatear de acorde a la base de datos para su posterior inserción.

## Inserción de datos

Una vez tengamos los datos en un **DataFrame** formateado según el tipo de dato que requiere la base de datos, dividiré el dataframe en tantos dataframes como tablas haya en la base de datos con todos los atributos que esta posea, borrando los datos duplicados. Esto es debido a que **pandas** posee una función mediante la cual, permite la subida de datos de un dataframe a una base de datos únicamente con darle la conexión a esta, el nombre de la tabla y si ya existe la tabla como introducir los datos. Esta función sería: `dataframe.to_sql()`.

Para la lectura de datos, es necesario que el archivo a leer se encuentre en la misma ruta que el programa, por ello también se creo un aviso en caso de no encontrarse en la ruta.

En esta etapa surgieron fallos menores por incoherencias en los nombres entre la base de datos y el dataframe, los cuales se solucionaron depurando ambas partes. También surgió un fallo debido a que a la hora de formatear los datos, sobrescribía un valor, borrando los datos de las horas asignadas de cada profesor, pasando de ser un número concreto de horas a un valor nulo. Este error fue solucionando depurando el código a la hora de tratar este valor hasta encontrar el momento en el que era borrado.

Un fallo importante que descubrí al final del proyecto es que a la hora de la subida de datos, el programa no realizaba ninguna comprobación, por lo que a la hora de añadir nuevos datos, este intentaba subirlos a la base de datos tal cual. Esto generaba errores debido a que aunque fueran datos de otro año, intentaba volver a subir datos ya existentes en la base de datos generando errores de duplicidad de *primary keys* y por lo tanto no subiendo ningún dato a la base de datos.

Este error fue solucionado añadiendo comprobaciones para únicamente subir datos nuevos a la base de datos evitando así la duplicación de los ya existentes.

## Actualización de datos

La actualización de datos es un proceso similar a introducir los datos por primera vez, con la diferencia de que hay que buscar las diferencias entre los datos nuevos y los ya existentes en la base de datos, mostrando las diferencias entre ambos. En mi caso, también he añadido una pequeña comprobación al principio para ver si realmente ya existen esos datos o son datos nuevos.

Para ello, primero deberemos leer los datos del excel nuevos, guardándolos en un **DataFrame** tal y como se indicaba en la lectura de datos. Una vez tengamos los datos nuevos leídos, leeremos los datos de la base de datos mediante una consulta SQL, en la que buscaremos el dataframe con todos los datos en el que coincidan los mismos datos del año y del código de grado que en los datos nuevos.

Si no se encuentran coincidencias, es porque los datos introducidos para actualizar, no se encuentran en la base de datos, por lo que en lugar de actualizarlos, habría que insertarlos, y el programa lanzará un aviso al usuario. Por el contrario, si se encuentran datos, la función continuaría puesto que si hay posibles datos a actualizar.

Una vez realizada la comprobación, dividiremos los datos por tablas, para una mayor comprensión a la hora de mostrar las diferencias, y compararemos ambas tablas, mostrando en la opción de mostrar las tablas, tanto la tabla antigua, con los campos a actualizar, como la tabla nueva con los campos que han cambiado.

Llegados a este punto hay dos opciones:

- **Volver:** mediante la cual cancelaríamos la actualización de los datos
- **Actualizar los datos:** en esta opción, se actualizarían los datos, borrando primero los datos obsoletos e introduciendo después los datos nuevos evitando así inconsistencias en la base de datos.

Durante esta etapa me han surgido numerosos problemas, casi todos por fallos de consistencia en la base de datos, al intentar borrar y subir los datos a la vez o subir los nuevos datos sin haber borrado los anteriores, puesto que se duplicaban campos de *Primary keys*, al no haberse borrado la información anterior.

Otro problema surgió por tener tipos de datos distintos, ya que al subir el dataframe a la base de datos, algún tipo de dato cambiaba, por lo que a la hora de compararlos siempre daban false aunque fueran iguales.

## Visualización de datos

Para la visualización de datos hemos creado diferentes filtros mediante los cuales se puede mostrar la información mas relevante para la persona que vaya a manejar el programa, todos ellos filtrados por el código de grado del que queramos obtener la información.

Inicialmente seleccionaremos el tipo de filtro a elegir entre:

- **Asignaturas grado:** Nos mostrará las diferentes asignaturas que hay en el grado ordenadas por curso y semestre.
- **Profesores en asignaturas:** Nos mostrará los diferentes profesores que hay en cada asignatura del grado ordenados por asignatura, curso y semestre.
- **Profesores por grupo:** Nos mostrará los diferentes profesores que hay en cada grupo de cada asignatura del grado ordenados por grupo, asignatura, curso y semestre.
- **Horas profesor:** Nos mostrará las diferentes horas que tienen de docencia asignadas cada uno de los profesores del grado dividido por semestres e indicando la suma de horas totales por año.

Una vez seleccionado el filtro que deseemos, el programa mostrará los diferentes códigos de grado con la carrera correspondiente, dándonos un cuadro de texto donde introduciremos el código de grado seleccionado. Una vez introducido el código, nos mostrará la tabla con los datos deseados.

El único 'error' que me ha surgido en esta etapa consistía en un descuadre en la interfaz entre las tablas y el diseño de esta. Esto se corrigió añadiendo mayor espacio en la parte de visualización para permitir que la tabla se mostrara sin quedar descuadrada.

## 5.6. Desarrollo de la interfaz

Debido al deseo de aprender mas sobre el desarrollo de interfaces, quise añadir una parte que añadiera al proyecto una interfaz gráfica. Para ello, debido a que el programa está en **Python**, la solución elegida fue la creación de una aplicación web basada en HTML y CSS, mediante el framework **CherryPy**.

Para ello cree una interfaz sencilla y facil de utilizar basada en menús con botones y entradas de texto o archivos de acuerdo a los principios de UX(*User Experience*).

En esta parte he tenido algunos problemas a ala hora de conectar la parte del *Back end* con el *Front end*, debido a que los errores de **Python**, no eran recogidos por esta interfaz, dándome errores no concretos, los cuales me llevaban a una depuración del código para encontrar el error.



---

## Trabajos relacionados

---

No se han encontrado ningún trabajo relacionado con este proyecto. Lo más parecido que se ha encontrado han sido los informes obtenidos mediante sigma.





---

# Conclusiones y Líneas de trabajo futuras

---

Este apartado recogerá todas las conclusiones obtenidas tras el desarrollo del proyecto, así como las posibles líneas de trabajo futuras para la mejora del mismo.

## 7.1. Conclusiones

Tras obtener el resultado del proyecto y haber estado trabajando en el todo este tiempo, obtengo las siguientes conclusiones:

- Con este proyecto, se facilitará el conocimiento de las horas que tienen asignadas de docencia todos los profesores, permitiendo consultar y almacenar numerosos datos de sigma de diferentes grados y años pudiendo tener toda la información en un mismo lugar mostrándola de manera mas sencilla de comprender al usuario.
- Tras el desarrollo del proyecto, los profesores contarán con una aplicación web que les permitirá controlar la horas que tiene asignada cada profesor de una forma sencilla y rápida en cada uno de los diferentes grados, por lo que el objetivo del proyecto ha sido cumplido de manera satisfactoria.
- La utilización de una aplicación web como solución al problema que se me planteaba, no solo me ha supuesto el uso de los conocimientos de UX (*User Experience*) adquiridos en los congresos permitiéndome crear una aplicación sencilla y de fácil uso, sino aprender a programar en HTML y CSS y conocer un nuevo framework para python.

- El conocimiento, estudio y posterior utilización de metodologías ágiles, me ha permitido flexibilidad en las diferentes etapas del desarrollo en cuanto a cambios se refiere, posibilitándome así que pase a un segundo plano la estimación de las diferentes tareas, ya que al no tener conocimientos previos sobre muchas de las materias, resultaba complicado. Pudiendo así entregar el proyecto en el plazo establecido.
- El proyecto me ha permitido no solo aumentar y asentar conocimientos adquiridos durante mi paso por la carrera, sino que me ha permitido adquirir nuevos como librerías antes desconocidas para mi en python, un nuevo framework y programación web entre otras.

## **7.2. Líneas de trabajo futuras**

El proyecto en si, ya estaría acabado aunque se podrían añadir mejoras para facilitar su uso y hacerlo más global.

- Conectar el proyecto a la red, permitiendo una conexión a través de sigma, logrando así la descarga directa de los datos.
- Mejorar la visualización de los datos almacenados mostrando no solo tablas, sino también gráficos.
- Crear una aplicación de escritorio o subir el proyecto a la web, de tal forma que sea mas fácilmente accesible por los diferentes usuarios que necesiten acceder a ello.
- Posibilidad de poder enviar los datos visualizados por correo o permitir exportarlos generando un informe sobre ellos.

---

## Bibliografía

---

- [1] Nerea Boada. ¿por qué es tan importante el user experience o experiencia del usuario?, 2017. [Internet; accedido 10-enero-2020].
- [2] Caterina Chen. Sistema de información, 2019. [Internet; accedido 10-enero-2020].
- [3] innovando. User experience – experiencia de usuario (ux), 2015. [Internet; accedido 10-enero-2020].
- [4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [5] Alexys Lozada. Las etapas del diseño de una correcta base de datos relacional, 2018. [Internet; accedido 02-enero-2020].
- [6] MiriadaX. Gestión de proyectos con metodologías Ágiles y enfoques lean, 2017. [Internet; Mooc realizado en el 2017].
- [7] María Estela Raffino. Sistema de información, 2019. [Internet; accedido 10-enero-2020].
- [8] The CherryPy team. CherryPy — a minimalist python web framework, 2019. [Internet; accedido 02-enero-2020].
- [9] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].
- [10] Wikipedia. Experiencia de usuario, 2019. [Internet; accedido 10-enero-2020].
- [11] Wikipedia. Git, 2019. [Internet; accedido 02-enero-2020].

- [12] Wikipedia. Github, 2019. [Internet; accedido 02-enero-2020].
- [13] Wikipedia. Información, 2019. [Internet; accedido 10-enero-2020].
- [14] Wikipedia. MongoDB, 2019. [Internet; accedido 02-enero-2020].
- [15] Wikipedia. Scrum (desarrollo de software), 2019. [Internet; accedido 02-enero-2020].
- [16] Wikipedia. Sistema de información, 2019. [Internet; accedido 10-enero-2020].
- [17] Wikipedia. Modelo–vista–presentador, 2020. [Internet; accedido 02-enero-2020].