



Universidad
Zaragoza

Trabajo Fin de Grado

Sistema de Información para la gestión
académica del Centro Universitario de la
Defensa

Information System for academic management
of the Centro Universitario de la Defensa

Autor

Víctor Marín Esteban

Directores

D. Julio Alberto Sangüesa Escorihuela
D^a Piedad Garrido Picazo

Escuela Universitaria Politécnica de Teruel / Universidad de Zaragoza

2020



Sistema de Información "GACUD" by Víctor Marín Esteban is licensed under a Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License.

Sistema de Información para la gestión académica del Centro Universitario de la Defensa

Resumen

Este Trabajo Fin de Grado (TFG) tiene como objetivo desarrollar un Sistema de Información (SI) para la gestión académica del CUD (Centro Universitario de la Defensa). Para ello, se van a facilitar datos almacenados por el CUD en distintas fuentes, con previa firma de acuerdo de confidencialidad. Dichos datos sufrirán un proceso de extracción, transformación y carga en la Base de Datos (BD) mediante una aplicación web desarrollada en este TFG denominada GACUD (Gestión Académica del Centro Universitario de la Defensa). Finalmente, los usuarios de esta aplicación podrán consultar información de interés y detectar patrones para tomar decisiones en los procesos de gestión del centro.

Palabras clave

Sistema de información, aplicación web, base de datos, OLTP.

Siglas y acrónimos

TFG: Trabajo de Fin de Grado.

SI: Sistema de Información.

BD: Base de Datos.

CUD: Centro Universitario de la Defensa.

GACUD: Gestión Académica del Centro Universitario de la Defensa.

OLTP: Procesamiento de Transacciones En Línea (OnLine Transaction Processing).

OLAP: Procesamiento Analítico En Línea (On-Line Analytical Processing).

IGU: Interfaces Gráficas de Usuario.

MDR: Modelo de Datos Relacional.

DBD: Diseño Base de Datos.

SGBD: Sistema Gestor de Bases de Datos.

Information System for academic management of the Centro Universitario de la Defensa

Abstract

The objective of this Final Degree Project is to develop an Information System for the Centro Universitario de la Defensa (CUD) management. For it, they will provide stored data in different sources, under prior signing of confidentiality agreement. These data will undergo a process of extraction, transformation and load in an OnLine Transaction Processing (OLTP) database through a web application called GACUD (Gestión Académica del Centro Universitario de la Defensa). Finally, users of this application can find information of interest and detect patterns to decision-making in the center's management processes.

Keywords

Information system, web application, database, OLTP.

Tabla de Contenidos

1.	Introducción.....	1
2.	Objetivos.....	3
3.	Estado del Arte	4
4.	Fase de Análisis	7
5.	Fase de Diseño.....	12
5.1	Prototipos de baja fidelidad	12
5.1.1	IGU01 – Panel de control	12
5.1.2	IGU02 – Carga de datos	13
5.1.3	IGU03 – FicheroCUD	13
5.1.4	IGU04 – Gestión de usuarios	14
5.1.5	IGU05 – Consulta de datos.....	15
5.2	Base de datos	16
5.2.1	Diseño Conceptual.....	19
5.2.2	Diseño Lógico	20
5.2.3	Diseño Físico	21
5.3	Diseño Conexión con la BD	22
6.	Fase de implementación	25
6.1	Tecnologías utilizadas y puesta en marcha.....	25
6.2	Implementación BD	28
6.3	Desarrollo GACUD	30
6.3.1	Inicio de sesión	30
6.3.2	Carga de datos	32
6.3.3	Gestión de usuarios.....	34
6.3.4	Consulta de datos.....	34
6.3.5	Alojamiento web	37
7.	Estimación de costes.....	38
8.	Análisis de usabilidad y accesibilidad	39
9.	Aspectos legales	42
10.	Conclusiones y trabajo futuro.....	43
11.	Bibliografía.....	44

Anexo A.1: Diseño lógico	46
Anexo A.2: Licencia Software y Documental.....	49

Índice de ilustraciones

Figura 1 Fuentes de la BD	1
Figura 2: Diagrama de Venn	8
Figura 3: Diagrama de casos de uso administrador.....	10
Figura 4: Diagrama de casos de uso profesor.....	10
Figura 5: Diseño arquitectónico	11
Figura 6: IU01 - Panel de control	12
Figura 7: IU02 - Carga de datos	13
Figura 8: IU03 - FicheroCUD.txt	14
Figura 9: IU04 - Gestión de usuarios	14
Figura 10: IU05 - Consulta de datos.....	15
Figura 11: Diseño BD.....	16
Figura 12: Arquitectura OLAP	17
Figura 13: Sistema OLTP	18
Figura 14: Diagrama E-R	19
Figura 15: Diseño físico	21
Figura 16: Librería database	23
Figura 17: Archivo de configuración de la BD	24
Figura 18: Flujo aplicación.....	26
Figura 19: Ruta del proyecto	26
Figura 20: Base URL CodeIgniter.....	27
Figura 21: Librerías CodeIgniter	27
Figura 22: Forward Engineering MySQL Workbench.....	28
Figura 23: Fallo palabra clave MySQL Workbench	28
Figura 24: Importar fichero SQL.....	29
Figura 25: Inicio de sesión	30
Figura 26 Panel de control admin.....	31
Figura 27: Inicio profesor	31
Figura 28: Carga de datos GACUD.....	32
Figura 29: Variable max_execution_time	33
Figura 30: Advertencia subir archivo	33
Figura 31: Gestión usuarios GACUD.....	34

Figura 32: Consulta matrículas GACUD	35
Figura 33: Consulta notas alumnos GACUD	36
Figura 34: Consulta convocatorias alumno GACUD	36
Figura 35: Consulta asignaturas pendientes alumno GACUD	37
Figura 36: Configuración base_url	38
Figura 37: Estimación de costes GANTT	38
Figura 38: Campos autocompletados en formularios	40
Figura 39: Formato solicitado para la contraseña	40
Figura 40: Error password	41

Índice de tablas

Tabla 1: Softwares de gestión académica	6
---	---

1. Introducción

El hecho de tener un gran volumen de información desestructurada, con un difícil acceso a los datos, así como a la manipulación de los mismos, supone un reto a resolver por cualquier entidad. El CUD (Centro Universitario de la Defensa) requiere solventar dicha problemática para la gestión académica de su centro. Hasta la fecha, el CUD ha intentado plantear varias soluciones poco ortodoxas, desde tener una multitud de tablas y macros en el software para Base de Datos (BD) de Microsoft Access hasta contratar a una empresa externa que se encargara de ello, pero todas las soluciones no han conseguido solventar este problema de una forma eficiente.

Para afrontar dicho reto, se mantuvo una reunión con el coronel Pablo Díaz, responsable de la gestión académica del CUD, que tuvo lugar en la academia general militar de Zaragoza, en la que se facilitaron los datos provenientes de varias fuentes, los cuales se pueden observar en la figura 1. Actualmente, con todos estos datos, es el mismo coronel, Pablo Díaz, el que se encarga de manipular todos estos archivos y crear otro archivo Excel en el que va integrando toda la información. Posteriormente, con una sola tabla en Access, se consigue extraer algo de información con esos datos.

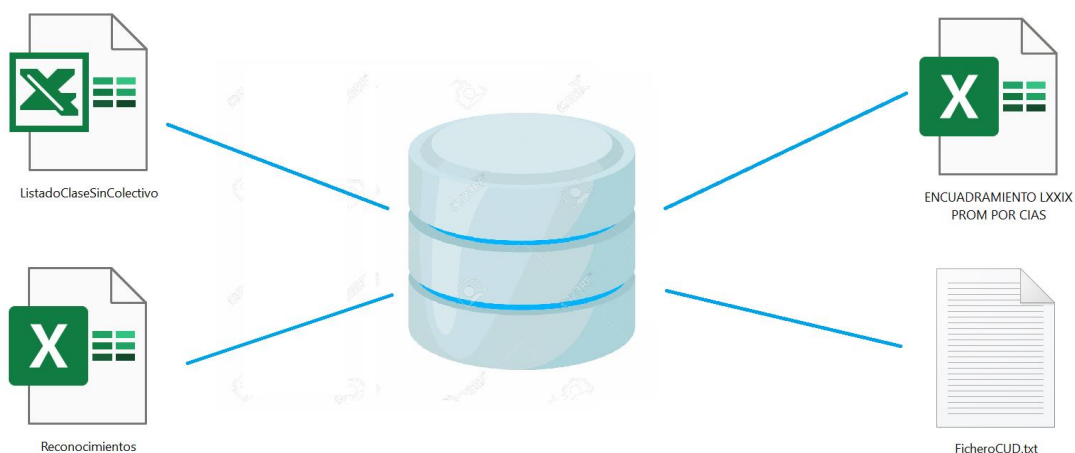


Figura 1 Fuentes de la BD



Este proceso realizado hasta la fecha es muy costoso y tedioso, con una alta probabilidad de fallos y dependiente de una persona. Al introducir muchos datos manualmente existen muchas probabilidades de cometer un fallo o un error en cualquier campo. Este TFG aborda dicho problema. En primer lugar, se realizará un análisis y un diseño que recoja toda la información entregada por el CUD con un previo acuerdo de confidencialidad. A continuación, se pasará a la fase de implementación, en la cual se dará forma física al Sistema de Información (SI). Para finalmente, concluir con el procesamiento analítico de la información a través de consultas para extraer información de interés.

Se va a exponer el soporte tecnológico que se ha utilizado para poder llevar a cabo la aplicación, detallando las tecnologías utilizadas y por qué se ha decidido utilizar cada una de ellas. Diferenciando el proceso de creación del sistema, las características del mismo, las licencias seleccionadas, las conclusiones, el trabajo futuro y la bibliografía de las fuentes consultadas durante el proceso de desarrollo.



2. Objetivos

El propósito de este TFG consiste en diseñar un SI que facilite la gestión académica del CUD. El sistema recoge la gran cantidad de datos de los que se dispone (provenientes de distintas fuentes de información), realizando un tratamiento de los mismos para insertarlos en una BD, hecha a medida, y finalmente extraer información de interés.

En este TFG también se pretende abordar la manipulación de grandes volúmenes de datos, partiendo de un inicial tratado de los mismos para poder insertarlos en la BD del sistema hasta la realización de consultas para poder extraer información sensible y útil.

Finalmente, se consigue desarrollar una aplicación de gestión académica que permita interactuar con la BD y extraer datos relevantes para el CUD.



3. Estado del Arte

Actualmente, en los demás Centros Universitarios de la Defensa (CUD) del país, Cartagena, Madrid y Vigo, la gestión académica está integrada en la universidad a la que el centro está adscrito. En la Universidad de Vigo (UV) la gestión del centro reside en el área de Tecnologías de la Información y de las Comunicaciones (TIC), en la Universidad Politécnica de Cartagena (UPCT) se encarga el vicerrectorado TIC que desarrolla sus propias herramientas y en la Universidad Autónoma de Madrid (UAM) se encargan los servicios informáticos con la plataforma SIGMA Academic.

Se ha realizado un estudio de algunos de los posibles softwares o herramientas de gestión académica que existen actualmente para poder solucionar los problemas a los que se enfrenta este TFG, para ello se han seleccionado varias herramientas tanto libres como propietarias (véase tabla 1).

SIGMA Academic es un software que abarca todo el ciclo de vida de la gestión académica universitaria. Sus funcionalidades parten desde los trámites iniciales universitarios hasta la gestión completa del expediente del alumno. Los procesos básicos de esta herramienta se basan en el acceso y la admisión del alumno, la evaluación de estudios previos, la gestión de la matrícula, la evaluación de los estudiantes y la solicitud del título universitario. Además, SIGMA Academic ofrece otros servicios complementarios y procesos administrativos que abarcan muchas funcionalidades, véase en SIGMA (*SIGMA*, s.f.).

Otra solución para la gestión de centros educativos es Brainbond, este software ha sido desarrollado por una empresa española que facilita la gestión académica. Brainbond gestiona todo el ciclo de la vida del alumno, así como las inscripciones, matrículas, fichas de los alumnos, evaluaciones. Además, abarca más allá de la gestión académica, ocupándose de la gestión económica, comercial, gestión de los profesores, el acceso a las familias etc. Véase en Brainbond (*Brainbond*, s.f.). Esta



empresa también oferta la carga de datos desde Excel, funcionalidad importante a contemplar para resolver la problemática que se plantea en este TFG.

Atenea es otro software de la empresa Ender que dispone de todos los procesos para dar soporte a la gestión de cualquier centro. Además de incluir las funcionalidades básicas de un software de gestión académica como los citados anteriormente, esta aplicación ofrece todos los servicios, disponibilidad inmediata, backups diarios, escalable, software actualizado, formación personalizada y un gran soporte. Veáse más información en Ender (*Ender, s.f.*). Respecto a las tecnologías, en materia de BD utilizan Oracle, SQL Server, PostGreSQL, Firebird, MySQL entre otras. En cuanto a los lenguajes de programación Python, PHP, Delphi, HTML, JavaScript, Java... y dos plataformas de desarrollo propias como son Olympto y Sapns.

Estos tres softwares o herramientas de gestión académica comentados son todos propietarios. A continuación se van a exponer otras dos soluciones que pueden ser igual de válidas y son Open Source, es decir, de código abierto.

Oduca es un sistema integral ERP (Enterprise Resource Planning) de gestión para centros educativos con una fácil implantación y sin coste de licencias. Además ofrece una solución integral, con gestión económica, académica y de comunicación. Pese a ser de código abierto, Oduca no es gratuito.

Por último, destacar el Sistema de Gestión Académica (SGA) distribuido bajo una licencia GNU (General Public License). Este software que también permite llevar a cabo la gestión de cualquier centro está totalmente disponible en SourceForge (sitio web de colaboración para proyectos software). Esta herramienta está basada en la arquitectura Modelo Vista Controlador (MVC) y utiliza una BD Oracle, veáse en (SourceForge, 2019). Programada principalmente en Java en la parte de backend, también utiliza JSP y JavaScript para el contenido web (frontend) y PL/SQL para el contenido relacionado con la BD.

Este sistema al ser gratuito no da soporte para adaptarse a cualquier circunstancia o cualquier problemática que pueda surgir a la hora de integrarlo en la gestión del



centro. Aunque si que explican cómo se implementa la herramienta, se necesita de una amplia gama de conocimientos para optar por la implantación de una solución como esta.

Estos son algunos ejemplos de los softwares, que existen actualmente, que podrían afrontar el reto que se está planteando. El problema reside en el grado de dificultad a la hora de integrar los datos. Al provenir estos de varias fuentes y en distintos formatos, necesitan un tratamiento óptimo. Todo este procedimiento no lo ofrece una herramienta de software libre como la que se ha propuesto y la idea de una herramienta comercial tampoco ha resultado satisfactoria hasta la fecha para el CUD. Además de ser costosas, dichas herramientas comerciales necesitan un importante trabajo previo para adaptar el software a las necesidades del centro, acoplándose a las distintas características del problema que se han comentado hasta ahora. Por ende, se ha llegado a la conclusión de realizar una aplicación ad-hoc en la cual se afronta dicho reto, permitiendo realizar la integración de todos los datos en una BD, fusionando la parte civil y la parte militar de los alumnos, donde finalmente se va a desarrollar una aplicación web que permita extraer conclusiones e información de interés.






Logotipo	Nombre	Tipo de acceso	Referencia
	SIGMA Academic	Propietario	https://www.sigmaaie.org/es/soluciones/sigma-academic
	Brainbond	Propietario	http://www.brainbond.es/
	Atenea	Libre	https://www.ender.es/
	Oduca	Libre	https://oduca.es/
	SGA	Libre	https://sourceforge.net/projects/sistema-de-gestion-academica/

Tabla 1: Softwares de gestión académica



4. Fase de Análisis

Una vez realizada una investigación previa de cómo se lleva a cabo actualmente la gestión académica y se han visualizado los distintos archivos con los que se va a tratar, toca realizar un análisis exhaustivo de esos ficheros para ver su contenido y el formato en el que están, para poder desarrollar el SI. Como ya se ha indicado anteriormente, se parte de los siguientes archivos:

- *ListadoClaseSinColectivo.xls*
 - Fichero con datos del alumnado, asignaturas, centro, titulación, departamento, docente.
- *Reconocimientos.xlsx*
 - Fichero con la información de las asignaturas reconocidas de los alumnos.
- *EncuadramientoPorCias.xlsx*
 - Fichero con la información militar de los alumnos.
- *FicheroCUD.txt*
 - Fichero con información de los alumnos y datos de acceso de los mismos.

En la figura 2 se puede observar de forma más gráfica, un diagrama de Venn con la información de estos archivos. En las intersecciones se pueden apreciar los campos que se van repitiendo en los distintos ficheros (aunque estén en diferente formato), por lo que es de utilidad para minimizar costes de tiempo a la hora de desarrollar el SI. Este diagrama proporciona una visión global de la información de la que se dispone, de tal manera que ayuda a comprender qué información se está manejando y cómo está relacionada dicha información, aspectos importantes para desarrollar un buen DBD (Diseño de la Base de Datos).

El fichero *Reconocimientos.xlsx* se queda fuera del diagrama, ya que de ese fichero sólo se va a necesitar el campo “reconocida”, para determinar las asignaturas convalidadas o reconocidas por los alumnos.



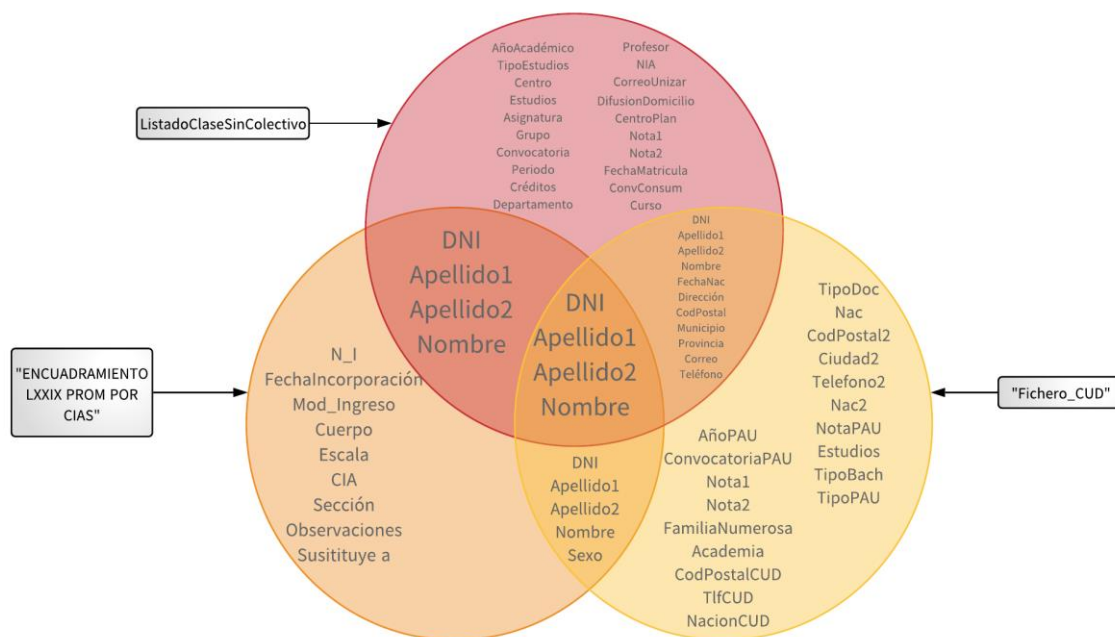


Figura 2: Diagrama de Venn

Posteriormente al estudio de la información de la que se dispone, se procede a un análisis del formato de los datos, para que a la hora de realizar un parseado de los mismos, se minimicen los riesgos y no existan fallos de integridad.

En un primer lugar, se puede observar que los ficheros Excel siguen la misma dinámica o estructura, aunque el fichero *EncuadramientoPorCias.xlsx* esté dividido en varias hojas.

Se puede apreciar que algunos de los campos que están repetidos en los ficheros (figura 2) no tienen el mismo formato. Por ejemplo, el nombre de los alumnos aparece diferenciado en tres columnas (apellido2, apellido1, nombre) en los ficheros *EncuadramientoPorCias.xlsx* y *FicheroCUD.txt*, mientras que en los ficheros *Reconocimientos.xlsx* y en *ListadoClaseSinColectivo.xls* aparece en una misma celda separado por comas.

También destacar que el fichero *ListadoClaseSinColectivo.xls* contiene más de diez mil registros, a diferencia de los demás ficheros, *EncuadramientoPorCias.xlsx* con 544 registros, *FicheroCUD.txt* con 272 y finalmente *FicheroCUD.xlsx* con 170 registros. Por lo que, algunos de los alumnos que aparecen en el fichero más grande



no contendrán información de otros ficheros como, por ejemplo, puede ser la información militar del fichero *EncuadramientoPorCias.xlsx*. El SI deberá contemplar la constante actualización de los ficheros y múltiples cargas de datos a lo largo de su ciclo de vida.

En el *FicheroCUD.txt* los campos están delimitados por el símbolo “|” y existen algunas otras peculiaridades como que el NIF está separado en dos campos diferentes, algunas columnas están vacías, el campo de la dirección del alumno contiene comas (dato importante a contemplar si este fichero se importa cómo CSV) o el fichero no contiene cabecera.

Para el desarrollo completo del SI, se va a desarrollar una aplicación web que permita cargar los ficheros comentados y posteriormente analizar los datos que estos contienen.

En primer lugar, se ha desarrollado una lista de funcionalidades que ha de cumplir el sistema propuesto en este TFG, diferenciando distintos roles, uno para el administrador o los administradores y otro para los profesores. Estas funcionalidades se pueden observar en los diagramas de caso de uso, véase figuras 3 y 4, siguiendo el Lenguaje de Modelado Unificado (UML) versión 2.5.1 (OMG, 2017).

Los administradores pueden desarrollar todas las funciones del sistema, desde la carga de datos hasta la consulta de informes. Además, también podrán dar de alta a distintos usuarios, modificarlos o eliminarlos. Mientras que los profesores sólo podrán acceder a la consulta de datos.

Esta aplicación web deberá ser simple e intuitiva, y dar soporte a estas funcionalidades básicas y esenciales para el CUD.



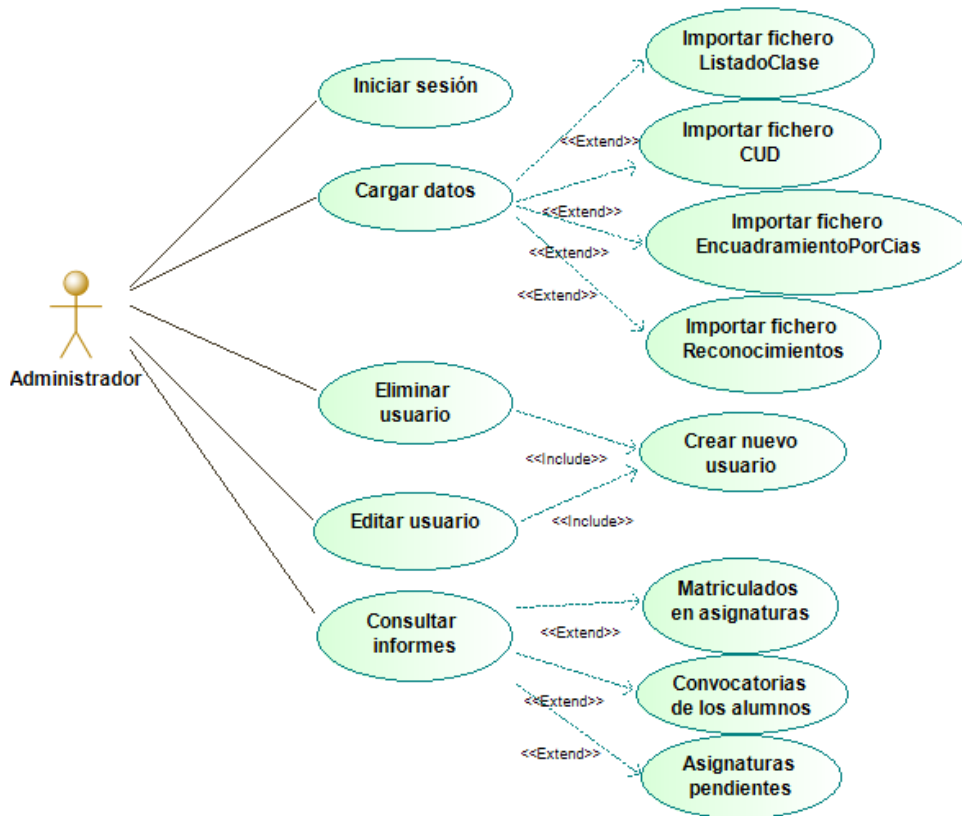


Figura 3: Diagrama de casos de uso administrador

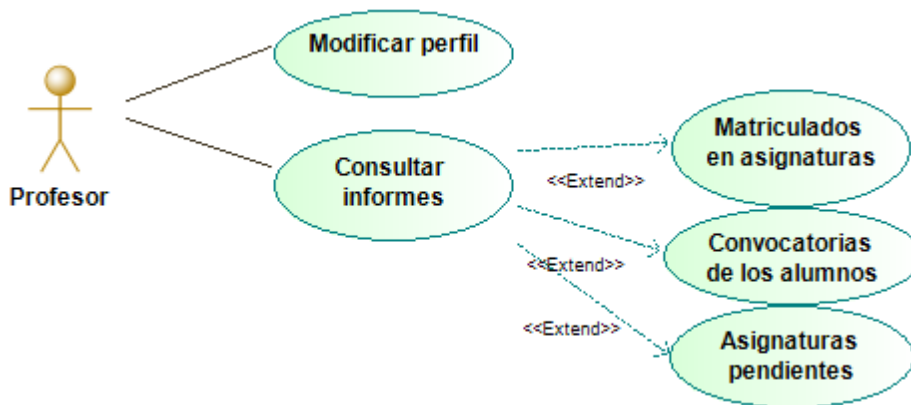


Figura 4: Diagrama de casos de uso profesor

Para comprender el flujo de cómo va a funcionar el SI, se ha desarrollado el diseño arquitectónico del mismo, véase figura 5. Cuando un usuario acceda al sistema realizará una petición al controlador vía URL, éste solicitará los datos al modelo, el cual lanzará una consulta SQL a la BD. Los datos retornarán hasta volver a llegar al

controlador, el cual cargará una vista de la aplicación que será mostrada en el navegador web del usuario.

En posteriores capítulos de este TFG se expondrán todas las tecnologías que se han utilizado para desarrollar la aplicación web.

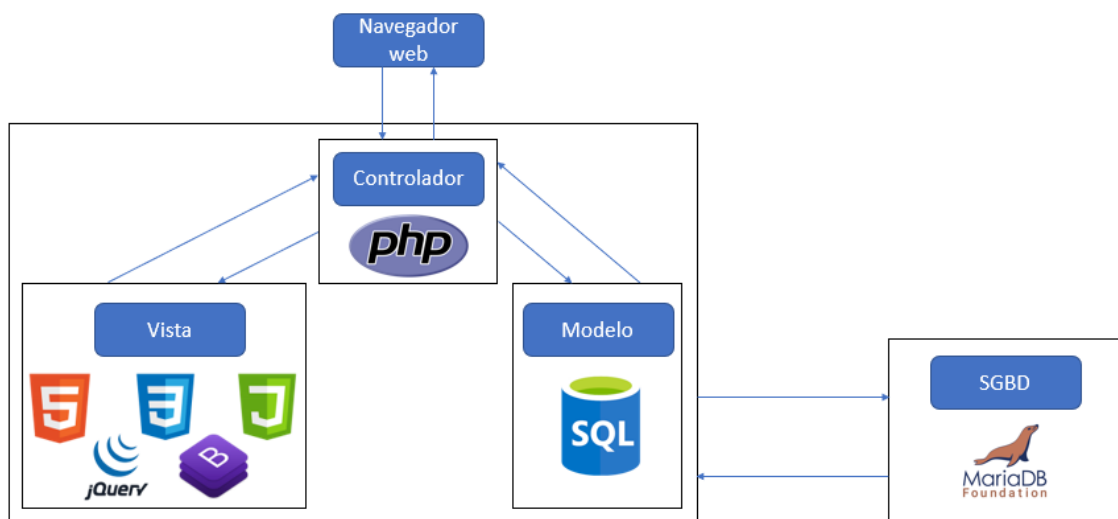


Figura 5: Diseño arquitectónico

5. Fase de Diseño

Una vez finalizada la fase de análisis, se va a dar forma al SI, para ello se van a desarrollar unos prototipos de la aplicación, el diseño completo de la BD, evolucionando en distintas fases y finalmente un capítulo de conexión con la BD.

5.1 Prototipos de baja fidelidad

Se han diseñado una serie de mockups centrados en las Interfaces Gráficas de Usuario (IGU) sobre cómo se va a comportar la aplicación web. Estas IGU sirven de punto de partida para realizar el diseño de las IGU de la aplicación, mostrando cómo los usuarios interactúan en cada una de las pantallas del sistema.

5.1.1 IGU01 – Panel de control

En la figura 6 se puede observar el diseño de la pantalla principal de la aplicación, si el usuario que ha iniciado sesión es administrador. De lo contrario, solo podrá acceder a la funcionalidad de consulta de datos.

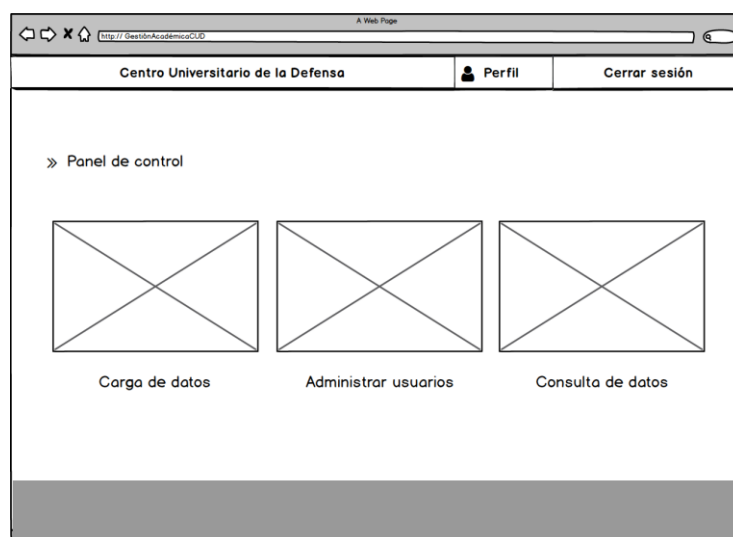


Figura 6: IU01 - Panel de control

5.1.2 IGU02 – Carga de datos

En esta IGU reflejada en la figura 7, se ha diseñado un mockup para la carga de datos, es decir, de los diferentes ficheros de los que se nutre el sistema. Estos se pueden cargar secuencialmente o bien, seleccionando uno, siguiendo los pasos que indique el sistema.

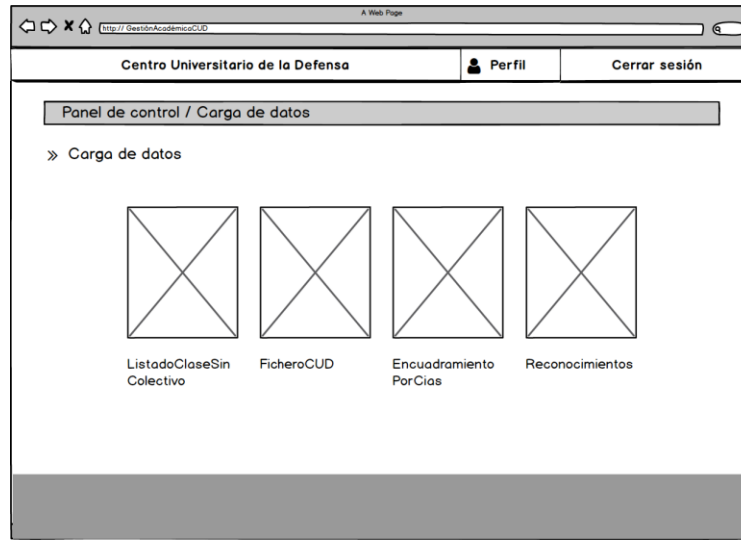


Figura 7: IU02 - Carga de datos

5.1.3 IGU03 – FicheroCUD

En esta otra IGU se puede observar un ejemplo de cómo se cargaría un fichero en el sistema. El usuario seleccionaría un tipo de fichero válido y lo insertaría, también dispone de botones para continuar o volver atrás con la importación de los datos, véase figura 8.



Figura 8: IU03 - FicheroCUD.txt

5.1.4 IGU04 – Gestión de usuarios

En la figura 9 se puede observar la IGU que simula la gestión de los usuarios, diferenciando varias funcionalidades, que son:

- a) Añadir un nuevo usuario.
- b) Editar un usuario existente.
- c) Eliminar un usuario existente.

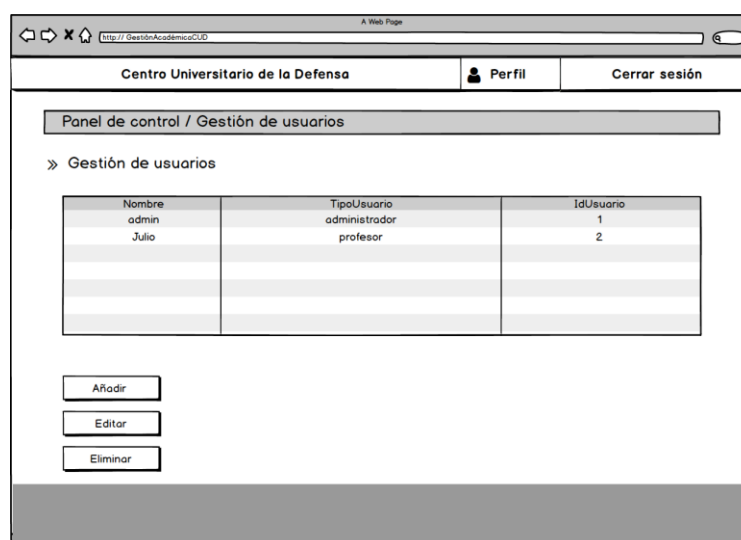


Figura 9: IU04 - Gestión de usuarios



5.1.5 IGU05 – Consulta de datos

Esta interfaz que se puede apreciar en la figura 10 ofrece un ejemplo de consulta de datos, en la cual, un usuario es capaz de consultar los alumnos matriculados en una asignatura para un determinado año académico.

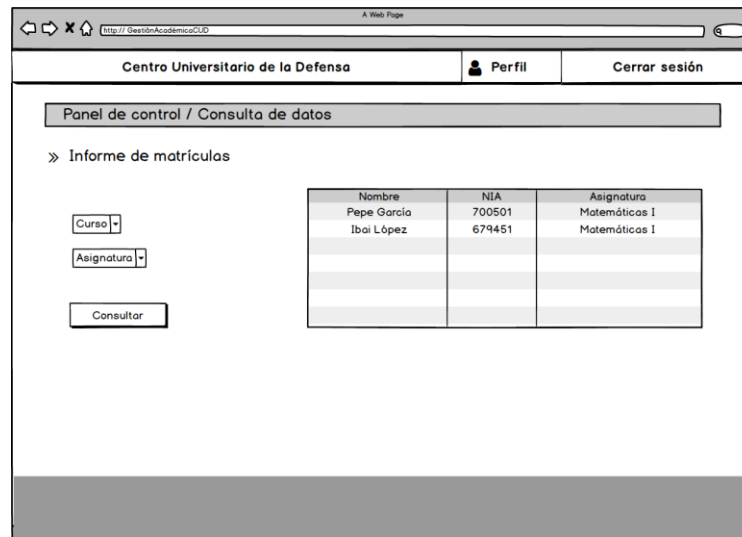


Figura 10: IU05 - Consulta de datos



5.2 Base de datos

Como se puede ver en la figura 11, la fase de DBD se divide en varias etapas secuenciales.

Partiendo de los requisitos o requerimientos que tiene este TFG, se empieza desarrollando el diseño conceptual, después el diseño lógico y finalmente el diseño físico. Se entrará en detalle en los siguientes apartados.

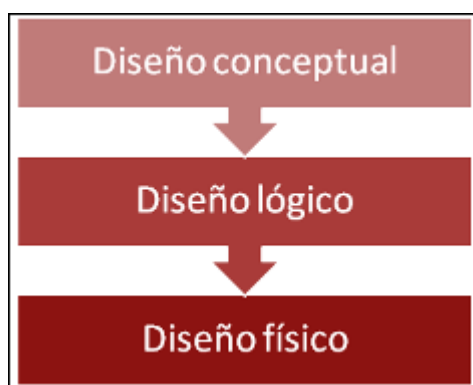


Figura 11: Diseño BD

Tal y como se ha comentado previamente en la introducción, al provenir los datos de distintas fuentes, para abordar el problema, se han valorado dos propuestas de diseño, OLAP (On-Line Analytical Processing) u OLTP (OnLine Transaction Processing). Siendo OLTP un sistema de procesamiento transaccional y OLAP un sistema de procesamiento analítico. OLTP se utiliza como un sistema de gestión y modificación de bases de datos online, por el contrario, OLAP se utiliza como un sistema de recuperación de datos y análisis de datos en línea.

Como ya se ha expuesto en este TFG, se van a manipular varias fuentes de información, que finalmente se integrarán en una única BD. Este proceso de inserción, actualización y eliminación de datos corresponde a un diseño OLTP, aunque la extracción de datos se acerca más a un diseño OLAP, orientado a consultas complejas y a BD que no se modifican con frecuencia.



Al trabajar con OLAP, se suelen utilizar Cubos, que contienen grandes cantidades de información que pueden provenir de un diseño OLTP. Para estudiar si fuera factible un posible planteamiento de este diseño, se ha visualizado la arquitectura propuesta en Rubio (*Rubio, 2020*), en el que se proponen varios cubos OLAP que recogen datos de distintas fuentes en diversos formatos, para finalmente extraer información compleja, que permita saber qué factores influyen directamente en la medición de la huella ecológica, en materia del cambio climático. Esta arquitectura se puede observar en la figura 12, en la que se observa como el origen de los datos parte de un diseño OLTP y otras fuentes externas, que tras un proceso ETL (Extracción, Transformación y Carga) se unifica y estructura los datos en una única fuente o repositorio de información.

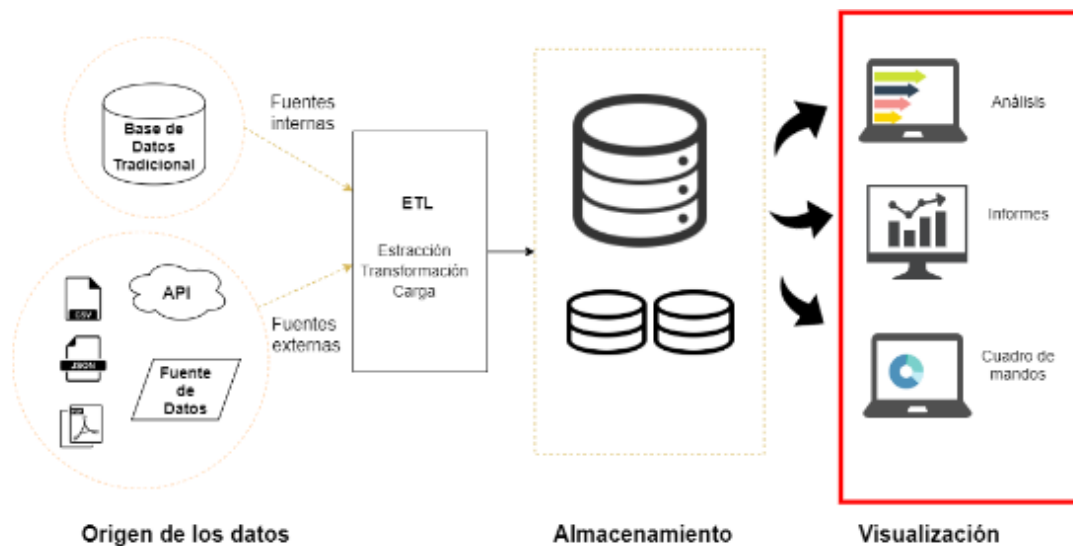


Figura 12: Arquitectura OLAP

Sin embargo, en un sistema operacional OLTP se almacenan datos actuales en detalle y actualizables. En este caso, se desea almacenar los datos académicos, que han de ir actualizándose, siendo las operaciones más importantes las de inserción, actualización y borrado.

En la figura 13 (Date, 1975), se pueden visualizar algunas de las principales características de un sistema transaccional.



Sistema Operacional (OLTP)

- almacena datos actuales
- almacena datos de detalle

- bases de datos medianas (100Mb-1Gb)
- los datos son dinámicos (actualizables)
- los procesos (transacciones) son repetitivos
- el número de transacciones es elevado

- tiempo de respuesta pequeño (segundos)

- dedicado al procesamiento de transacciones
- orientado a los procesos de la organización
- soporta decisiones diarias
- sirve a muchos usuarios (administrativos)

Figura 13: Sistema OLTP

Como se ha podido observar, el problema planteado se asemeja mucho más a un diseño OLTP que a un OLAP. Por lo tanto, para este TFG se va a optar por esa primera opción.



5.2.1 Diseño Conceptual

Esta fase de DBD tiene como objetivo crear un esquema conceptual del contenido de un Sistema de Información (SI). Es independiente de la tecnología, así como del Sistema Gestor de Bases de Datos (SGBD) que se va a utilizar.

Para esta fase se ha optado por utilizar un Modelo de Datos Relacional (MDR). Este modelo, muy utilizado en materia de BD, debido a su simplicidad y facilidad de uso, dispone del diagrama Entidad-Relación (E-R) como herramienta para diseñar gráficamente los elementos, relaciones, características y Restricciones de Integridad (RI) que describen el SI, en un lenguaje común acordado por los profesionales y académicos del gremio. En la figura 14 se puede observar el diseño E-R de la BD.

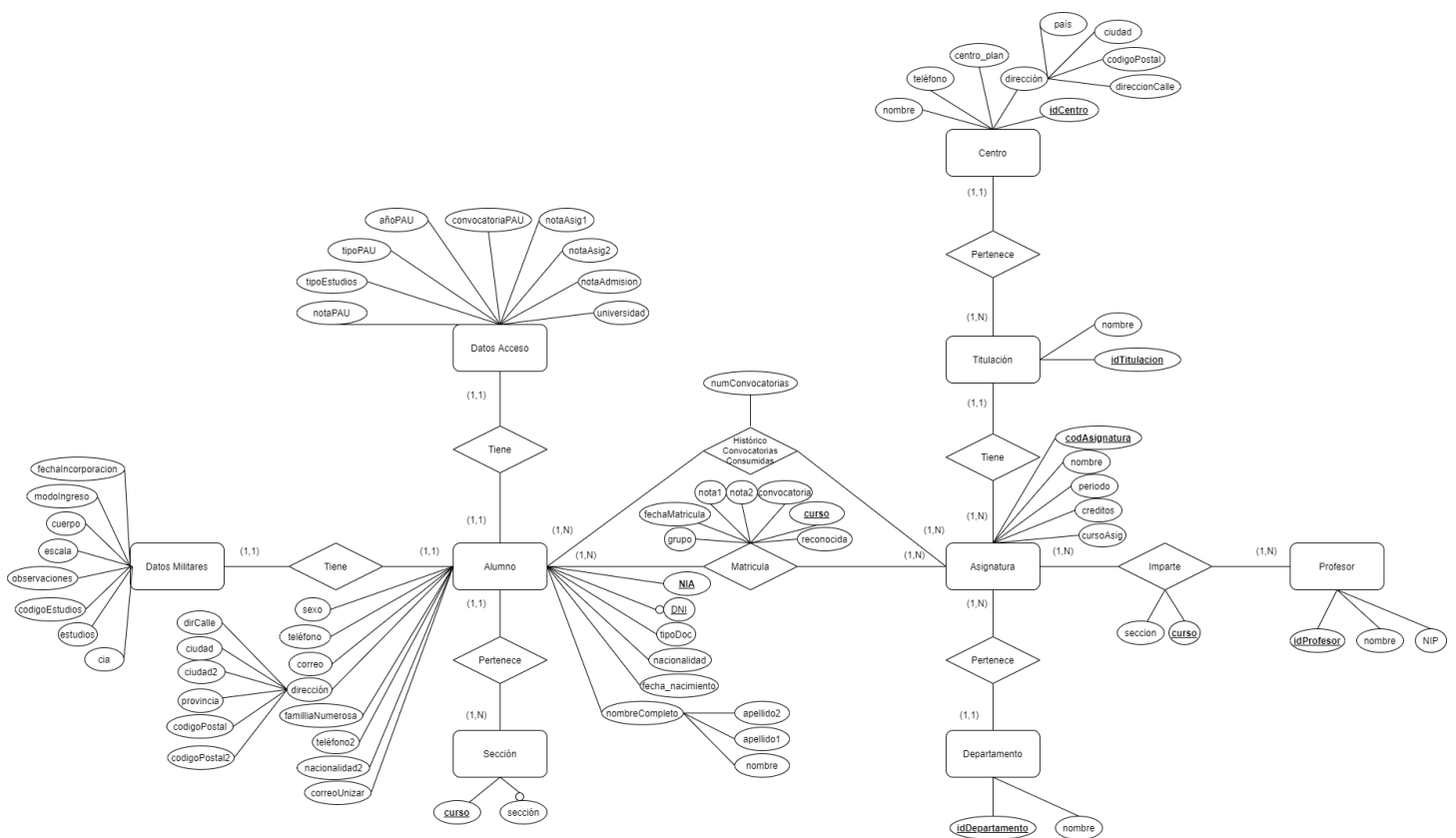


Figura 14: Diagrama E-R



5.2.2 Diseño Lógico

En esta etapa se transforma el esquema conceptual en un esquema lógico, que será dependiente de las pautas de transformación del MDR, bajo el que se va a implementar la BD.

Como la BD va a ser relacional, esta etapa del diseño estará formada por un conjunto de relaciones con sus atributos, claves primarias, claves ajenas y RI que más se ajusten a la realidad.

En esta fase se aplica la normalización para subsanar problemas en las futuras operaciones a realizar con la BD diseñada, que puedan provenir, por ejemplo, de la redundancia de información.

Véase Anexo A.1 para ver el desarrollo del diseño lógico completo.



5.2.3 Diseño Físico

El diseño físico, la última fase del DBD, es un proceso que parte del diseño lógico, enfocado al uso real de la BD. Para desarrollar este diseño (figura 15) se ha utilizado el SGBD y su correspondiente entorno de desarrollo: MySQL Workbench.

Se implementa una configuración física basada en la arquitectura donde se aloje la BD, que permita el almacenamiento de la información, así como su posterior explotación con un rendimiento óptimo. Para ello, se debe de tener en cuenta el hardware donde se va a ubicar la base de datos, la frecuencia con la que se va a acceder a la base de datos y los volúmenes de datos que se van a almacenar.

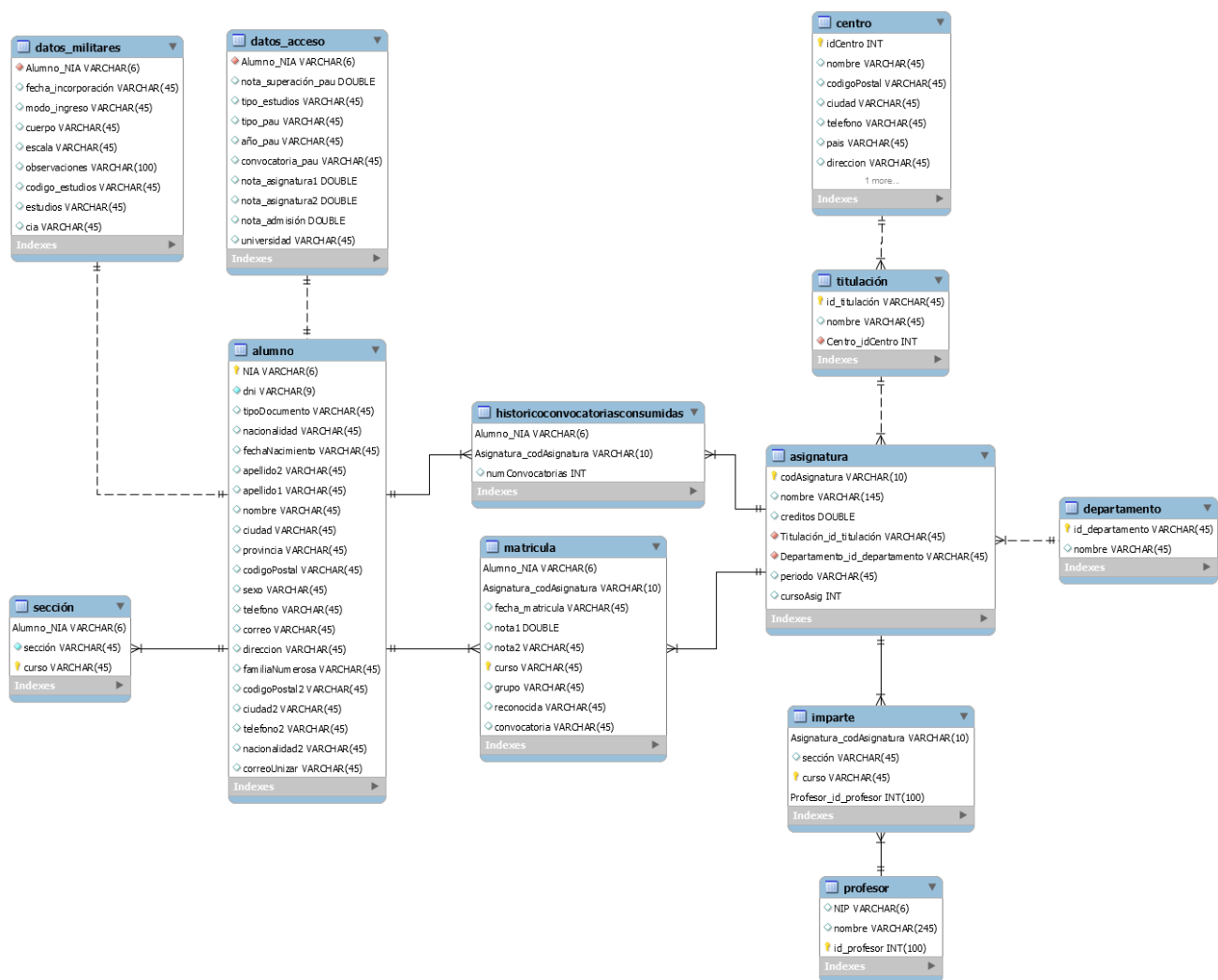


Figura 15: Diseño físico



5.3 Diseño Conexión con la BD

Para este TFG, se ha optado por implementar una BD relacional para su posterior manipulación con el lenguaje SQL, en inglés *Structured Query Language*, pudiendo así elegir entre una amplia gama de diferentes SGBD como Oracle, MySQL, Microsoft SQL-Server, MariaDB, SQLite, PostgreSQL etc.

Se ha valorado también el uso de una base de datos no relacional (NoSQL). Esta opción no suele primar operaciones tediosas con comprobaciones de muchas RI, puesto que su objetivo es disponer de muchos datos para su futuro procesamiento analítico. Algunos de los SGBD más utilizados son MongoDB, Redis y Cassandra.

En este caso se ha optado por el primer caso. En concreto, se ha hecho uso de MariaDB. Este SGBD prioriza el desarrollo abierto, publicando código fuente en repositorios públicos. Además, todas las fases de desarrollo también son de dominio público. MariaDB es una bifurcación de MySQL, que con el paso del tiempo se ha convertido en un SGBD autónomo.

En la actualidad, MySQL está bajo el dominio de Oracle, por lo que, si se desea explorar todas las posibilidades, hay que recurrir a los productos de MySQL propietarios. Según Ionos, empresa alemana de telecomunicaciones experta en hosting, dominios y páginas web, el camino de estos dos SGBD tiende a separarse, consolidándose MariaDB como una buena solución de software libre. (IONOS, 2020).

Como lenguajes de programación se han barajado principalmente las opciones de PHP y Java por su popularidad, orientación a objetos, conocimiento sobre ellos y fácil conexión con las bases de datos, pero también otros lenguajes como Python, que es uno de los lenguajes de programación más utilizados actualmente y bastante útil para las aplicaciones de Big Data gracias a las bibliotecas de procesamiento de datos o R, que está orientado al análisis estadístico y relacionado también con la minería de datos.



Finalmente, se ha optado por utilizar el lenguaje de programación PHP junto con el framework de CodeIgniter que permite integrar perfectamente la lógica de negocio de la aplicación con la parte Front-end, basada en HTML y Bootstrap. Este framework cumple las directrices marcadas en la norma ISO, IEC 25010, norma centrada en la evaluación de la calidad del producto, compuesto por ocho características de calidad cómo son usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad, eficiencia de desempeño y adecuación funcional (ISO25000, s.f.).

PHP es un lenguaje de programación muy ligado a la web, empleado para el desarrollo de aplicaciones web de pequeño y mediano tamaño, aunque en menor medida, también se usa en grandes aplicaciones gracias al soporte para clases y objetos. Este lenguaje de programación destaca por una sintaxis sencilla de aprender y manejar. Aunque no sea el lenguaje más utilizado en la carrera, es un lenguaje con el que casi cualquier estudiante del Grado de Ingeniería Informática se siente cómodo, ya que se suele utilizar casi siempre en asignaturas como Bases de Datos, Sistemas de Información, Comercio Electrónico, etc. El propio PHP tiene funciones integradas para la conexión a la BD siendo la generación de código HTML dinámica.

De la conexión con la BD se encarga CodeIgniter, utilizando la carga automática, definida en el archivo *application/config/autoload.php*, agregando la librería *database* en la variable de configuración *libraries*, véase figura 16.

```
$autoload['libraries'] = array('database');
```

Figura 16: Librería database

En el archivo de configuración *database.php* (ubicado en *application/config*) hay que establecer todos los datos de la BD, así como servidor, usuario, contraseña etc. Véase figura 17.



```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'cud',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

Figura 17: Archivo de configuración de la BD



6. Fase de implementación

En esta etapa del ciclo de vida del SI se va a desarrollar la aplicación web GACUD (Gestión Académica del Centro Universitario de la Defensa). Para ello se va a realizar una introducción al entorno de trabajo, diferenciando las diferentes tecnologías a utilizar y la puesta en marcha del entorno de desarrollo, la implementación de la BD y finalmente concluir con unos ejemplos de la aplicación ya desarrollada.

6.1 Tecnologías utilizadas y puesta en marcha

A continuación, se van a detallar las tecnologías utilizadas para el desarrollo del SI, denominado GACUD.

En primer lugar, para poder desplegar la aplicación web GACUD, se necesita de un servidor web, para ello se ha utilizado una distribución de Apache, denominada XAMPP, este paquete de software libre se basa en un SGBD que integra la BD de MariaDB, el lenguaje de programación PHP y el servidor web Apache. *“El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.”* Véase en Wikipedia (Wikipedia, 2020). Este paquete permite integrar todas estas tecnologías fácilmente, además de ser gratuito, está disponible tanto para Windows, como Linux y OS X.

Como entorno de programación o IDE (Integrated Development Environment) se va a utilizar Netbeans. Esta tecnología es una de las más utilizadas para desarrollar software, es gratuito y cuenta con una de las mayores comunidades de desarrolladores, además de estar familiarizado con ella por ser el IDE más utilizado en la carrera, destaca por algunas funcionalidades como autocompletado inteligente, sugerencias, plantillas de código y refactorización. Otros entornos de programación



muy conocidos y que pueden ser también válidos son PHPStorm, Eclipse, Sublime o Atom.

Para facilitar el desarrollo del software se va a utilizar un framework Open Source denominado CodeIgniter. Este framework utiliza la arquitectura MVC (Modelo Vista Controlador) permitiendo una división en el acceso a los datos, la lógica de negocio y la capa de presentación de la aplicación, también proporciona un conjunto de librerías que facilitan la programación. El flujo de la aplicación sigue el esquema que aparece en la figura 18, tomada de la documentación oficial (*CodeIgniter*, s.f.). Todo comienza en el *index.php* de la raíz, posteriormente el enrutador examina la petición HTTP para determinar qué elemento se tiene que procesar. Si la página ya se había cargado antes, se envía directamente, gracias a la caché, además se realiza un tratamiento de seguridad para que posteriormente el controlador cargue cualquier recurso necesario. Finalmente, la vista es procesada y enviada al navegador web.

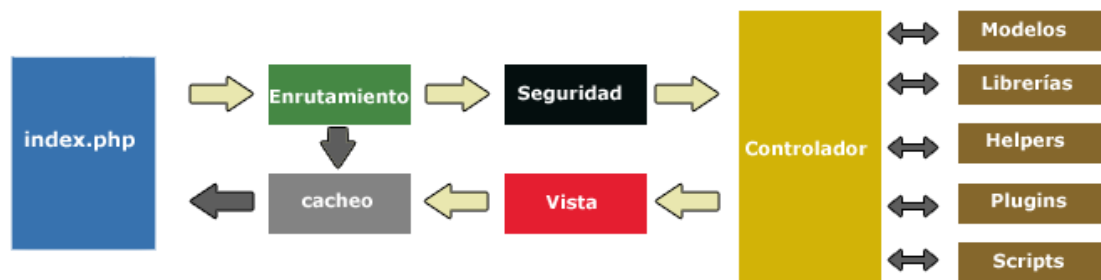


Figura 18: Flujo aplicación

Para empezar a desarrollar la aplicación, se ha de crear un nuevo proyecto PHP en el IDE. Para desplegar la aplicación sobre el servidor Apache, se necesita tener la carpeta del proyecto ubicada en el servidor, en concreto, ha de estar en la carpeta “*htdocs*”, véase figura 19.

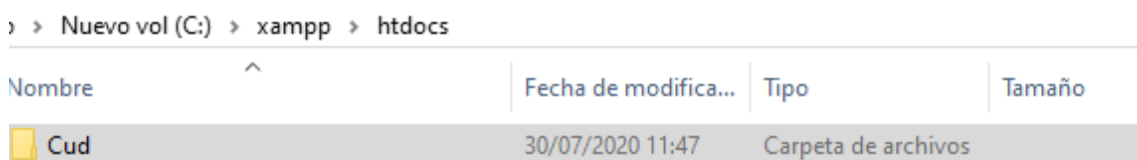


Figura 19: Ruta del proyecto

La integración de CodeIgniter es sencilla, basta con descomprimir el paquete en la carpeta del proyecto. Para la completa instalación, hay que establecer la base URL



en la variable de configuración, en el fichero *application/config/config.php*, véase figura 20. Dicho framework necesita conocer la URL para acceder a la raíz donde se despliega el servidor, de momento, para el desarrollo de la aplicación se va a utilizar el servidor local, *http://localhost/cud*.

```
26 | $config['base_url'] = 'http://localhost/Cud';
```

Figura 20: Base URL CodeIgniter

Como se ha indicado anteriormente en el capítulo de la conexión con la BD, de dicha conexión se encarga CodeIgniter. Además, si se desea utilizar alguna otra librería, ha de especificarse en el fichero *autoload.php*, en este proyecto se va a utilizar la librería *session* como aparece en la figura 21.

```
$autoload['libraries'] = array('session', 'database');
```

Figura 21: Librerías CodeIgniter

También se ha indicado anteriormente el lenguaje de programación que se va a utilizar, PHP, junto con HTML5 para el contenido web. Además, se va a utilizar el framework de Bootstrap y CSS para el diseño y estilo de la aplicación web. Para poder utilizar estas últimas tecnologías en la aplicación, han de estar integradas en el proyecto, para ello hemos de ubicar la carpeta de Bootstrap y las hojas de estilo en la carpeta “*dist*” de nuestro proyecto. En concreto la ruta del Bootstrap es “*C:\xampp\htdocs\Cud\dist\vendor*”.

Por último, se ha utilizado SQL como lenguaje para extraer la información de la BD, este lenguaje ha sido integrado dentro del modelo de la aplicación, se entrará más en detalle en los siguientes apartados.



6.2 Implementación BD

Para comenzar con esta última fase de desarrollo, se parte del diseño físico que se ha expuesto en este TFG (véase figura 15). Una vez que se ha completado este diseño, utilizando la herramienta de MySQL Workbench, se desea transformar este diseño a tablas en la BD. Para ello, MySQL lo pone bastante fácil, detallando los pasos en el manual oficial de MySQL Workbench (MySQL). Siguiendo el manual, aparece un fallo al exportar el diseño físico a la BD, véase figura 22. Este fallo se debe a un error de compatibilidad entre MySQL y MariaDB, este último no acepta la palabra reservada *VISIBLE* que incorpora MySQL Workbench en el script SQL, por lo tanto, para solucionar el fallo simplemente se guarda el script en un fichero y se eliminan todas las referencias a esa keyword que causa el error, véase figura 23.

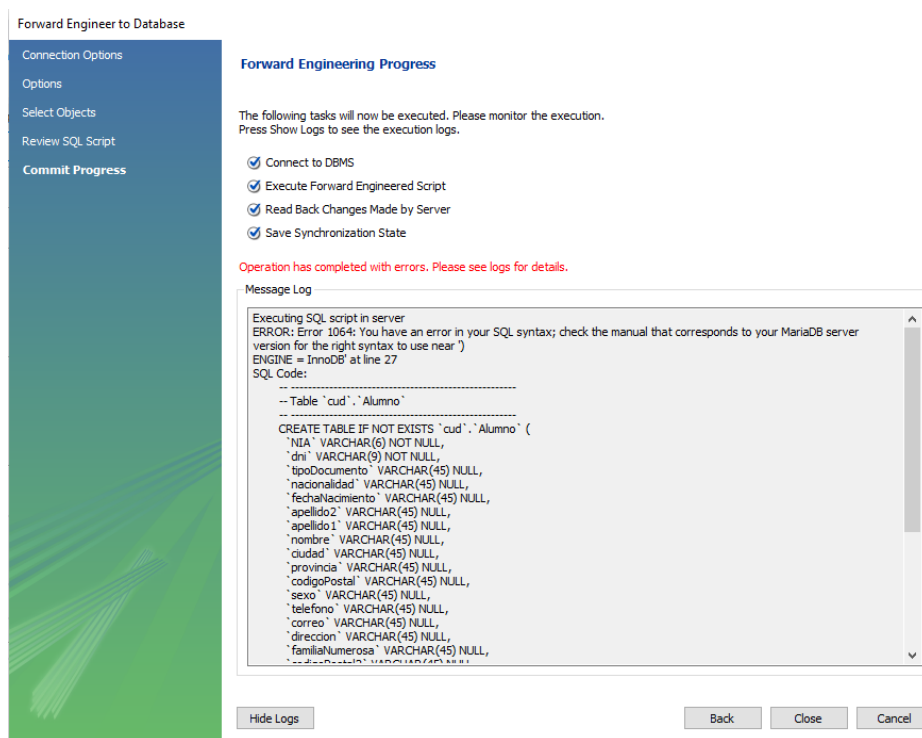


Figura 22: Forward Engineering MySQL Workbench

```
43 UNIQUE INDEX `dni_UNIQUE` (`dni` ASC) VISIBLE
43 UNIQUE INDEX `dni_UNIQUE` (`dni` ASC) )
```

Figura 23: Fallo palabra clave MySQL Workbench



A continuación, para tener la BD operativa, se ha utilizado la herramienta phpMyAdmin integrada en el paquete XAMPP. Simplemente, en la herramienta hay una opción de importar, en la cual se selecciona el anterior fichero SQL y se importa, una vez cargado, la BD ya estaría disponible, véase figura 24.

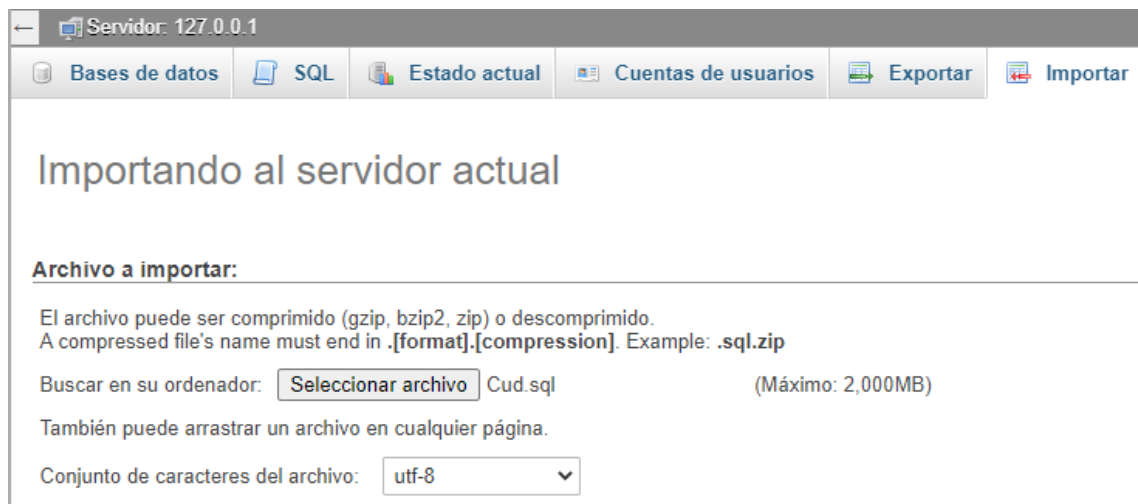


Figura 24: Importar fichero SQL



6.3 Desarrollo GACUD

En esta fase se va a detallar como se ha desarrollado el software GACUD. Para ello se van a distinguir los diferentes casos de uso o funcionalidades del sistema, explicando las funciones los distintos roles del sistema.

6.3.1 Inicio de sesión

Para el diseño del inicio de sesión o login del sistema, se ha elegido una plantilla web, disponible de forma gratuita en Colorlib (*Colorlib*, s.f.).

Esta funcionalidad es una de las más sencillas del sistema, pero no por ello menos importante. En la figura 25 se puede ver la vista inicial de la aplicación.



Figura 25: Inicio de sesión

En función del tipo de usuario que ingrese en el sistema, la aplicación redirigirá a los usuarios a dos pantallas diferentes: 1) Panel de control administrador, 2) Pantalla de inicio profesor, véase figura 26 y 27, respectivamente. En el caso del usuario profesor, aparecerán las asignaturas que imparte en este curso académico y podrá



consultar datos de las asignaturas que imparte, mientras que el administrador puede utilizar todas las funciones del sistema.

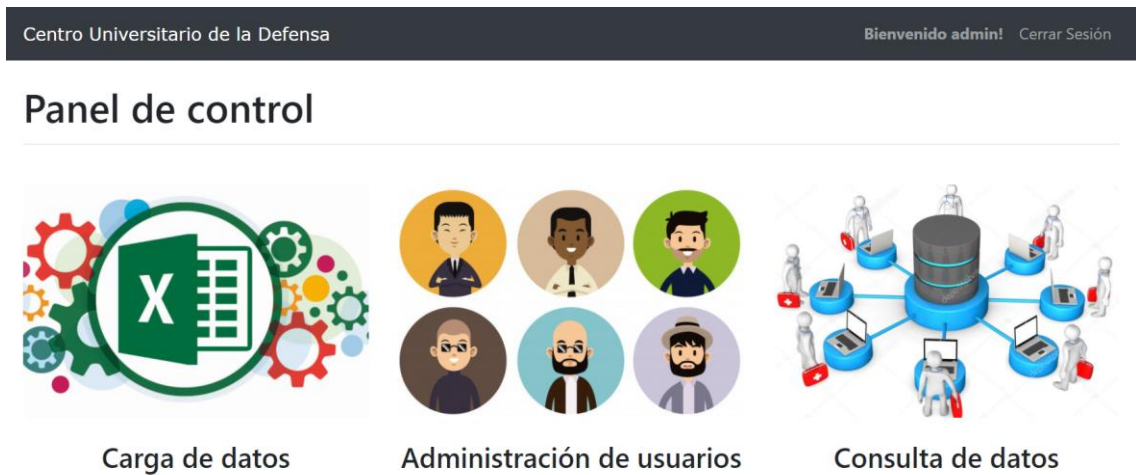


Figura 26 Panel de control admin



Figura 27: Inicio profesor

6.3.2 Carga de datos

Esta funcionalidad es la más relevante del sistema. En esta pantalla que aparece en la figura 28, el administrador puede ir cargando los diferentes archivos fuentes de los que se nutre la aplicación. Los archivos se pueden volver a cargar por si existe alguna actualización de los datos.

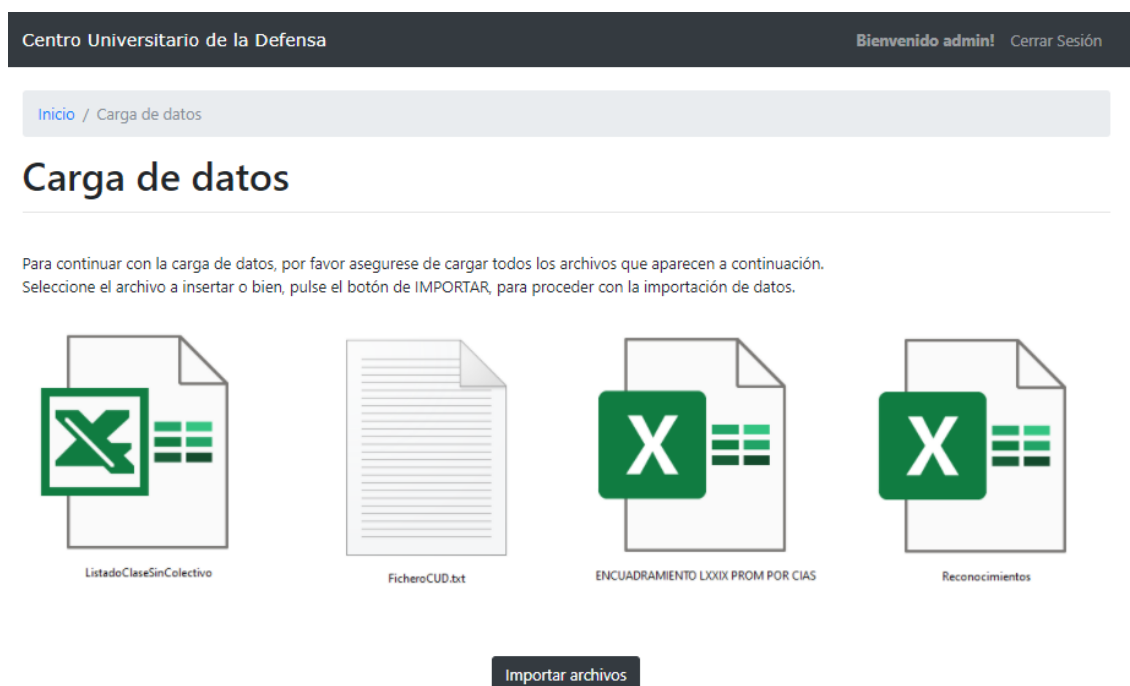


Figura 28: Carga de datos GACUD

La carga de datos no es una tarea sencilla ni tampoco rápida, el primer archivo a importar es el más costoso, ya que cuenta con más de 10.000 registros, es por ello, por lo que se advierte al usuario a la hora de cargar el archivo, que la importación va a tardar varios minutos, véase figura 30. Además, se ha tenido que modificar la variable *max_execution_time* del fichero de configuración de PHP (*php.ini*) para dar más tiempo a la importación de los datos, en este caso se ha establecido en 900 segundos, es decir 15 minutos. Véase figura 29.

El proceso desde que el usuario sube el archivo hasta que finalmente los datos son almacenados en la BD, es un proceso complejo y tedioso, se realizan muchas operaciones para llegar al objetivo. Además, cada fichero ha de ser tratado de



diferente manera, ya que tienen distintas características y algunos datos están en diferentes formatos.

Para cada fichero se va troceando la información que va nutriendo a las diferentes tablas de la BD. Durante este proceso surgen complicaciones o errores que dificultan la tarea, como por ejemplo el juego de caracteres que impide que los datos se inserten correctamente en la base de datos, datos que se repiten en varios ficheros no tienen el mismo formato, etc.

```
; Maximum execution time of each script, in seconds  
; http://php.net/max-execution-time  
; Note: This directive is hardcoded to 0 for the CLI SAPI  
max_execution_time=900
```

Figura 29: Variable max_execution_time

[Inicio](#) / [Carga datos](#) / [Listado Clase](#)

Listado Clase

Paso 1: Subir archivo ListadoClase.csv

⚠ Advertencia: La carga de este archivo puede tardar varios minutos.

Importar archivo

Figura 30: Advertencia subir archivo



6.3.3 Gestión de usuarios

La gestión de usuarios es otra tarea exclusiva del administrador. Es el único que puede dar de alta a un usuario en el sistema para que utilice GACUD. Por defecto, la aplicación viene dada con un usuario administrador. La vista principal de la gestión de usuarios se puede ver en la figura 31, en la cual, el administrador puede dar de alta a un usuario o bien, editarlo o eliminarlo del sistema. Si desea dar de alta un nuevo usuario en el sistema, tendrá que indicar si quiere añadir un nuevo usuario administrador o un nuevo usuario profesor. En este último caso, en el formulario se le mostrará una lista de todos los profesores que imparten clases actualmente, para poder darlos de alta en el sistema.

Gestión de usuarios



Id usuario	Tipo usuario	Nombre usuario	Profesor	Acción
1	Administrador	admin	No	Editar Eliminar
4	Profesor	Javier	ACEÑA MEDINA, JAVIER	Editar Eliminar
8	Profesor	Enrique	Cajal Hernando, Carlos Enrique	Editar Eliminar

Figura 31: Gestión usuarios GACUD

Esta funcionalidad abarca las tareas de inserción, actualización y borrado de la BD.

6.3.4 Consulta de datos

Esta última fase de consulta de datos se puede diferenciar en consultas del administrador y consultas del profesor. En cuanto a las consultas del profesor, sólo va a poder realizar consultas de las asignaturas que él imparte, mientras que el administrador puede realizar consultas globales. Se han implementado algunas de las consultas que ha solicitado el CUD, como los alumnos matriculados en las



asignaturas, el número de convocatorias de los alumnos en las asignaturas y las asignaturas pendientes de los alumnos.

En la figura 32 se puede ver un ejemplo de los alumnos matriculados en una asignatura para un curso determinado. Mientras que, si la consulta la realiza un usuario profesor, tan solo se mostrarán las asignaturas que él imparte. Por lo tanto, sólo puede realizar consultas de los alumnos matriculados en sus asignaturas. Además, el profesor podrá consultar las notas de sus alumnos, véase figura 33. En dicha consulta también aparecerá como información adicional los demás profesores que imparten esa asignatura, si es que los hay.

Curso académico
2019/2020-0

Asignatura
30100-Matemáticas I

Consultar datos

Resultados obtenidos:

Número de filas: 10

Buscar en la tabla:

Curso académico	Código	Asignatura	Grupo	NIA	Nombre
2019/2020-0	30100	Matemáticas I	150	613428	ENRIQUE ALEXANDER LÓPEZ GUALLICHICO
2019/2020-0	30100	Matemáticas I	132	698517	GUILLERMO MOLINÉ BAGDÉS
2019/2020-0	30100	Matemáticas I	123	755425	ALFONSO LASALA
2019/2020-0	30100	Matemáticas I	122	755824	JAVIER VILLACAMPA
2019/2020-0	30100	Matemáticas I	121	760129	NATALIA ORTEGA
2019/2020-0	30100	Matemáticas I	114	760344	SANTIAGO ARNAU MORENO
2019/2020-0	30100	Matemáticas I	121	768822	EMILIO GALIPIENSO AVILES
2019/2020-0	30100	Matemáticas I	131	770141	CRISTIAN GONZALEZ RIVERA
2019/2020-0	30100	Matemáticas I	150	771370	ANTONIO ESTEBAN NAVAS
2019/2020-0	30100	Matemáticas I	111	774232	NICOLAS MENDEZ

Mostrando del 1 al 10 de un total de 366 elementos

Anterior 1 2 3 4 5 ... 37 Siguiente

Figura 32: Consulta matrículas GACUD



Consulta de datos para el docente: **IGLESIA PEDRAZA, OSCAR DE LA**

Curso académico: 2019/2020-0

Asignatura: 30170-Ingeniería del medio :

Consultar datos

Resultados obtenidos:

Profesores que imparten esta asignatura:

- FONTS AMADOR, MARIA ISABEL
- IGLESIA PEDRAZA, OSCAR DE LA
- ORTIN GRACIA, JORGE
- SÁNCHEZ TAPIA, CARLOS
- URBIZTONDO CASTRO, MIGUEL ÁNGEL

Número de filas: 10

Buscar en la tabla:

NIA	Nombre	Calificación	Nota	Reconocida
685032	ANTONIO JOSÉ MARTÍNEZ TORREGROSA	5	Aprobado	Reconocida (TTS ET)
771345	DANIEL SEBASTIAN ORTIZ SANCHEZ	5	Aprobado	Reconocida (TTS ET)
542777	HECTOR MAINAR RUBIO	Pendiente calificar		
559853	JHONATAN DUESO GUERRERO	Pendiente calificar		

Figura 33: Consulta notas alumnos GACUD

Otros informes que son de interés para el CUD son el número de convocatorias en la que se encuentra cada alumno, figura 34, o las asignaturas que quedan pendientes para cada alumno, figura 35. Estos datos son fáciles de obtener gracias al gran trabajo realizado anteriormente, ya que el diseño de la BD facilita la complejidad de la consulta para poder realizar estos informes.

Convocatorias de los alumnos

Informe que muestra todos los alumnos y el número de convocatorias en cada asignatura.

(*) Usted puede filtrar en la tabla utilizando el campo de búsqueda o bien, puede ordenar la tabla por columnas.

⚠ Esta consulta puede tardar unos segundos.

Resultados obtenidos:

Número de filas: 10

Buscar en la tabla:

NIA	Alumno	Código	Asignatura	Curso	Periodo	Convocatorias
702264	CARLOS JAVIER LÓPEZ MARTINEZ	30123	Resistencia de materiales	3	Segundo Semestre	4
712233	ISABEL ESTEBAN LIMA	30173	Economía de la empresa (Perfil Defensa)	3	Segundo Semestre	4
713295	JAVIER TARIFA VARON	30123	Resistencia de materiales	3	Segundo Semestre	4

Figura 34: Consulta convocatorias alumno GACUD



Asignaturas pendientes por aprobar

Informe que muestra las asignaturas pendientes por aprobar de cada alumno.

(*) Usted puede filtrar en la tabla utilizando el campo de búsqueda o bien, puede ordenar la tabla por columnas.

⚠ Esta consulta puede tardar unos segundos.

Resultados obtenidos:

Número de filas: Buscar en la tabla:

NIA	Alumno	Código	Asignatura	Curso	Periodo	Grupo
812486	AARON SANTANA	30100	Matemáticas I	1	Primer Semestre	113
812486	AARON SANTANA	30101	Física I	1	Primer Semestre	113
812486	AARON SANTANA	30102	Química	1	Segundo Semestre	113

Figura 35: Consulta asignaturas pendientes alumno GACUD

6.3.5 Alojamiento web

Para que GACUD se accesible desde cualquier lugar se ha creado un nuevo alojamiento web (*gacud.unizar.es*) en el servidor web de la Universidad de Zaragoza. Además, se ha creado una BD MySQL.

Una vez se ha dado el acceso para poder entrar al servidor, se ha subido el proyecto GACUD al servidor web. El acceso a este servidor está cortado para direcciones IP externas a la red universitaria, por lo que si se desea entrar desde fuera hay que utilizar una conexión VPN (Virtual Private Network) a través de la aplicación FortiClient (*FortiClient*, s.f.).

El paso de migrar la aplicación desde el equipo en local a este servidor, se ha realizado utilizando la aplicación FileZilla (*FileZilla*, s.f.), una aplicación FTP libre y de código abierto. Esta fase también conlleva dificultades y problemas a resolver, no es tan sencillo como copiar el directorio del proyecto al servidor. En el fichero *config.php* hay que cambiar la variable *base_url* (véase figura 36) con la nueva dirección del servidor.



```
$config['base_url'] = 'http://gacud.unizar.es/';
```

Figura 36: Configuración base_url

Otro error a destacar es que el servidor es Case Sensitive, es decir, distingue entre mayúsculas y minúsculas, por lo que la vista *Admin/consultas.php* no es la misma que la ubicada en *Admin/Consultas.php*. Este error se desconocía en la realización del proyecto, por lo que existían bastantes erratas que en el equipo local funcionaban bien, pero en el servidor no.

7. Estimación de costes

En la figura 37 se puede observar una estimación real del coste temporal del proyecto en un diagrama de Gantt. Para ello se ha utilizado un programa de código abierto denominado GanttProject (*GanttProject*, s.f.). Este programa permite visualizar el desglose de tareas o actividades programadas, así como las relaciones jerárquicas y la interdependencia entre tareas.

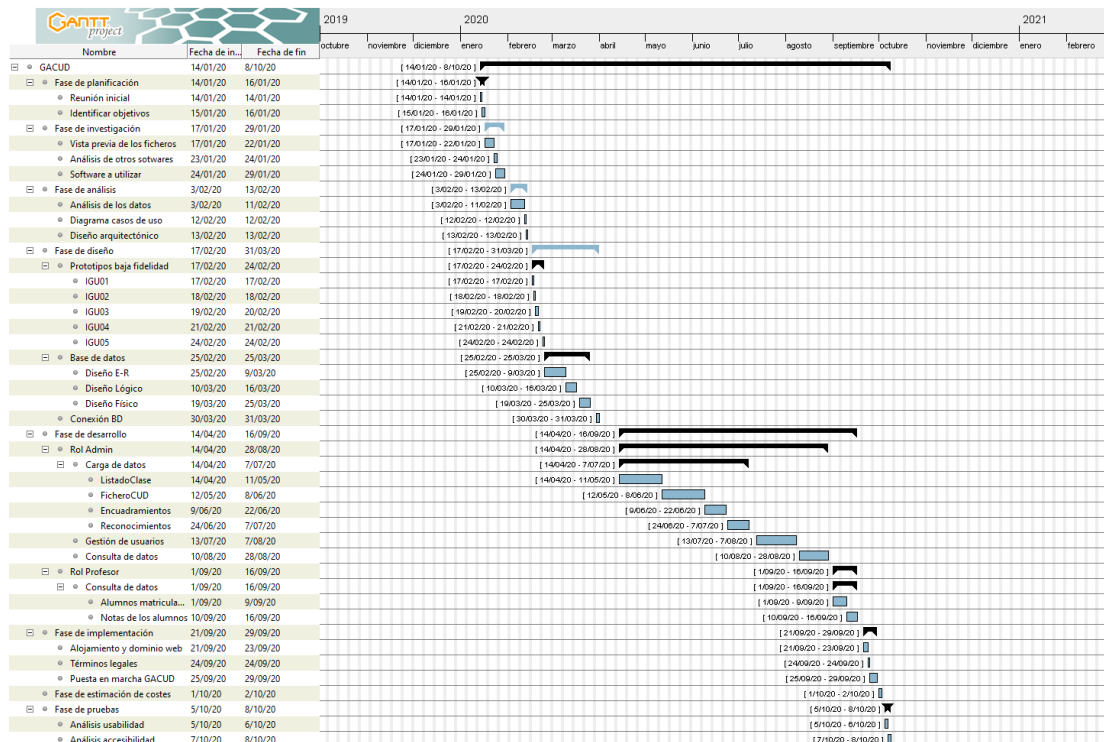


Figura 37: Estimación de costes GANTT

8. Análisis de usabilidad y accesibilidad

GACUD se centra en la norma ISO 25010, intentando cumplir todos los requisitos. En cuanto a la usabilidad del sistema, intenta seguir las siguientes características (*ISO25000*, s.f.):

- Capacidad para reconocer su adecuación: Permite al usuario entender si GACUD es adecuado a sus necesidades.
- Capacidad de aprendizaje: Permite al usuario aprender la aplicación.
- Capacidad para ser usado: Permite al usuario operar y controlar GACUD con facilidad.
- Protección contra errores de usuario: Protege a los usuarios de cometer errores.
- Estética de la interfaz de usuario: Capacidad de la IGU de satisfacer la interacción con el usuario.
- Accesibilidad: Capacidad de que GACUD sea utilizado por usuarios con determinadas características y discapacidades.

A continuación, se van a exponer ejemplos de cómo se cumplen estos criterios.

GACUD se ajusta a las necesidades de los usuarios ya que el software se adapta a los requisitos del CUD, eso sí, con un trabajo futuro para conseguir de GACUD una buena herramienta de gestión académica. El software no realiza ninguna funcionalidad que no se adecue a las necesidades de los usuarios. En cuanto a la capacidad de aprendizaje y capacidad para ser usado, sigue la misma directriz, es decir las IGU son sencillas y fáciles de usar, no hay ninguna pantalla que resulte tediosa o con mayor dificultad para interactuar con ella por parte del usuario.

En cuanto a la protección contra errores de usuario, GACUD solventa dicha problemática evitando introducir datos inválidos en los formularios, por ejemplo, si en una consulta el usuario ha de introducir su NIA personal, el software solo permite introducir números. Además, la mayoría de los campos están autocompletados o en listas previamente cargadas de la BD, por ejemplo, en la figura 38 se requiere del



curso académico y de la asignatura. Por lo tanto, estos datos son cargados automáticamente para que el usuario escoja uno de la lista y no tenga que introducir información, de tal manera que existe una gran prevención de errores.

Curso académico
2019/2020-0

Asignatura
30100-Matemáticas I

Consultar datos

Figura 38: Campos autocompletados en formularios

También se han tenido en cuenta patrones de interacción, como por ejemplo el campo contraseña en formularios. Para ello, se utiliza el tipo *password*. En este caso, el software utiliza el atributo *pattern* para crear una contraseña segura (Véase figura 39). Además, GACUD cumple el principio de prevención de errores, con el doble campo *password*, cumpliendo así con el principio de usabilidad de Jakob Nielsen (*Profile*, s.f.). Si las contraseñas no coinciden, la aplicación mostrará un error (véase figura 40).

Nombre de usuario
Victor

Contraseña
.....

Utiliza un formato que coincida con el solicitado
Una contraseña válida es un conjunto de caracteres, donde cada uno consiste de una letra mayúscula o minúscula, o un dígito. La contraseña debe empezar con una letra, contener al menor un dígito y un mínimo de 8 caracteres.

.....

Añadir

Figura 39: Formato solicitado para la contraseña



Error. Las contraseñas no coinciden. Por favor, asegurese que las contraseñas coinciden

Figura 40: Error password

En cuanto a la accesibilidad web del sistema, todas las imágenes contienen el atributo *alt*, es decir, texto alternativo para brindar información y ayudar a contextualizar el contenido a personas con discapacidad visual. Para que GACUD sea accesible también se han asociado etiquetas a todos los formularios como información complementaria de lo que hay que insertar en ese campo, tampoco existe ningún link sin texto que los referencie o presencia de texto subrayado sin que sea un link.

Como trabajo futuro, se recomienda realizar una evaluación de la accesibilidad web utilizando herramientas on line como WAVE (WAVE, s.f.) o TAW (TAW, s.f.), siguiendo las guías WCAG (Web Content Accessibility Guidelines) 2.1. También se recomienda realizar un test de UX (User Experience) con un mínimo de usuarios para realizar las pruebas de las IGU del sistema. Algunos de los más utilizados son USE (USE, s.f.) y SUS (SUS, s.f.). Además, otra posible mejora es el perfeccionamiento de la aplicación para que sea totalmente responsive.



9. Aspectos legales

Para la manipulación de datos de carácter sensible en este TFG, se tuvo que firmar un documento de compromiso de confidencialidad de acuerdo con la LOPD 3/2018 (Ley Orgánica de Protección de Datos) (BOE, 2016) y con el RGPD 679/2016 (Reglamento Europeo Protección de Datos) (BOE, 2018).

Para definir los aspectos legales de GACUD habría que tener en cuenta:

- Aviso legal y Condiciones de uso.
 - Referencias legales para regular las condiciones en las que los usuarios pueden acceder, navegar y utilizar la aplicación web. La LSSI (Ley de Servicios de la Sociedad de la Información) obliga a identificar al titular de la web. Además, habría que contemplar los requisitos para darse de alta y baja como usuario.
- Política de privacidad.
 - Documento obligatorio para toda página web que recoja datos personales. Esta política informa a los usuarios de cómo se van a tratar sus datos. El artículo 12 del RGPD exige que se comunique la información sobre su procesamiento de manera precisa, transparente, en un lenguaje claro y sencillo, inteligible, fácilmente accesible y gratuita.
 - En la política de privacidad se deberá de indicar: 1) Datos de contacto del responsable. 2) Licitud del tratamiento de los datos. 3) Propósitos del tratamiento de los datos. 4) Destinatarios de los datos. 5) Plazo de conservación de los datos. 6) Referencia a los derechos del interesado. 7) Deber legal de recolectar datos. 8) Explicación sobre el uso de decisiones individuales automatizadas. (IONOS D. G., IONOS)



10. Conclusiones y trabajo futuro

En el presente TFG se ha logrado desarrollar un SI capaz de llevar a cabo la gestión académica del CUD, procesando los distintos datos almacenados en distintos ficheros para finalmente almacenarlos en la BD y extraer información de interés.

El proceso más complejo ha sido el análisis y parseo de los distintos ficheros para su posterior inserción en la BD. Este proceso tiene un mayor grado de dificultad por los diferentes formatos de los ficheros que por su volumen. Además, este proceso puede ser repetitivo para que los ficheros puedan ir actualizándose. Por lo que hay que tener en cuenta a parte de la inserción de los datos, su posterior actualización, diseñando y codificando un código que lo sostenga, para ello el uso del framework CodeIgniter ha resultado bastante potente para el desarrollo de la aplicación, consolidándose como una buena solución para el desarrollo de proyectos web, de la mano de BootStrap y CSS para el diseño de la página web se han conseguido los resultados esperados.

GACUD puede tener mucho trabajo por delante, ya que se pueden realizar muchos más informes o bien, llevar a cabo un análisis de los datos con técnicas de datamining o minería de datos. Se puede dotar a la aplicación de muchas más funcionalidades, como añadir un nuevo tipo de usuario alumno, de tal manera que pudiera consultar su expediente y sus datos personales. También quedaría pendiente implementar los aspectos legales de la aplicación.

En cuanto a las pruebas del sistema, habría que realizar un test de usabilidad y algún análisis de accesibilidad con alguna herramienta como se ha citado en anteriores apartados. También sería conveniente realizar pruebas de la carga que puede soportar el servidor y pruebas de detección de errores.

Otras funcionalidades o extras que podría incorporar GACUD serían gestión documental, gestión económica, exportación de los datos, matriculaciones etc.



11. Bibliografía

BOE. (27 de Abril de 2016). Obtenido de BOE:

<https://www.boe.es/doue/2016/119/L00001-00088.pdf>

Brainbond. (s.f.). *Brainbond*. Obtenido de <http://www.brainbond.es/>

CodeIgniter. (s.f.). *CodeIgniter*. Obtenido de <https://codeigniter.com/docs>

Colorlib. (s.f.). *Colorlib*. Obtenido de <https://colorlib.com/wp/cat/login-forms/>

Date, C. (1975). *Introducción a los sistemas de bases de datos*.

Ender. (s.f.). *Ender*. Obtenido de <https://www.ender.es/>

Estado, B. O. (06 de Diciembre de 2018). *BOE*. Obtenido de

<https://www.boe.es/eli/es/lo/2018/12/05/3/dof/spa/pdf>

FileZilla. (s.f.). Obtenido de <https://filezilla-project.org/>

FortiClient. (s.f.). Obtenido de <https://www.forticlient.com/>

GanttProject. (s.f.). Obtenido de <https://www.ganttproject.biz/>

IONOS. (s.f.). Obtenido de Digital Guide IONOS:

[https://www.ionos.es/digitalguide/paginas-web/derecho-digital/politica-de-privacidad-gana-la-confianza-de-tus-usuarios/#:~:text=Una%20pol%C3%ADtica%20de%20privacidad%20\(privacy,contexto%20de%20la%20relaci%C3%B3n%20comercial](https://www.ionos.es/digitalguide/paginas-web/derecho-digital/politica-de-privacidad-gana-la-confianza-de-tus-usuarios/#:~:text=Una%20pol%C3%ADtica%20de%20privacidad%20(privacy,contexto%20de%20la%20relaci%C3%B3n%20comercial)

IONOS, D. G. (20 de Marzo de 2020). *Digital Guide IONOS*. Obtenido de

<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/mariadb-vs-mysql/>

IONOS, D. G. (s.f.). *IONOS*. Obtenido de

[https://www.ionos.es/digitalguide/paginas-web/derecho-digital/politica-de-privacidad-gana-la-confianza-de-tus-usuarios/#:~:text=Una%20pol%C3%ADtica%20de%20privacidad%20\(privacy,contexto%20de%20la%20relaci%C3%B3n%20comercial](https://www.ionos.es/digitalguide/paginas-web/derecho-digital/politica-de-privacidad-gana-la-confianza-de-tus-usuarios/#:~:text=Una%20pol%C3%ADtica%20de%20privacidad%20(privacy,contexto%20de%20la%20relaci%C3%B3n%20comercial)

ISO25000. (s.f.). *ISO/IEC 25010*. Obtenido de

<https://iso25000.com/index.php/normas-iso-25000/iso-25010>

MySQL. (s.f.). *MySQL*. Obtenido de <https://dev.mysql.com/doc/workbench/en/wb-forward-engineering-live-server.html>

OMG. (Diciembre de 2017). Obtenido de <https://www.omg.org/spec/UML/>



Profile. (s.f.). Obtenido de <https://profile.es/blog/los-10-principios-de-usabilidad-web-de-jakob-nielsen/>

Rubio, M. L. (2020). *Arquitectura abierta para diseño de Data Marts*.

SIGMA. (s.f.). *SIGMA*. Obtenido de <https://www.sigmaaie.org/es/soluciones/sigma-academic?id=1&t=1>

SourceForge. (10 de 06 de 2019). *SourceForge*. Obtenido de <https://sourceforge.net/projects/sistema-de-gestion-academica/>

SUS. (s.f.). Obtenido de <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

TAW. (s.f.). Obtenido de <https://www.tawdis.net/#>

USE. (s.f.). Obtenido de <https://garyperلمان.com/quest/quest.cgi?form=USE>

WAVE. (s.f.). Obtenido de <https://wave.webaim.org/>

Wikipedia. (20 de Agosto de 2020). *Wikipedia*. Obtenido de <https://es.wikipedia.org/wiki/XAMPP>



Anexo A.1: Diseño lógico

Alumno (NIA: dom_nia, dni: dom_dni, tipoDocumento: dom_tipoDocumento, nacionalidad: dom_nacionalidad, fechaNacimiento: dom_fechaNacimiento, nombreCompleto.apellido2: dom_apellido2, nombreCompleto.apellido1: dom_apellido1, nombreCompleto.nombre: dom_nombre, dirección.ciudad: dom_ciudad, dirección.provincia: dom_provincia, dirección.dirCalle: dom_dirCalle, dirección.codigoPostal: dom_codigoPostal, dirección.codigoPostal2: dom_codigoPostal2, dirección.ciudad2: dom_ciudad2, sexo: dom_sexo, teléfono: dom_telefono, correo: dom_correo, familiaNumerosa: dom_familiaNumerosa telefono2: dom_telefono2, nacionalidad2: dom_nacionalidad2, correoUnizar: dom_correoUnizar)

CP: {NIA}

UQ: {dni}

VNN: {dni}

Centro (idCentro: dom_idCentro, nombre: dom_nombre, teléfono: dom_telefono, centro_plan: dom_centro_plan, dirección.direccion_calle: dom_direccion_calle, dirección.codigoPostal: dom_codigoPostal, dirección.ciudad: dom_ciudad, dirección.pais: dom_pais)

CP: {idCentro}

Titulación (idTitulacion: dom_idTitulacion, nombre: dom_nombre, idCentro: dom_idCentro)

CP: {idTitulacion}

Caj: {idCentro} → Centro

Departamento (idDepartamento: dom_idDepartamento, nombre: dom_nombre)

CP: {idDepartamento}



Asignatura (codAsignatura: dom_codAsignatura, nombre: dom_nombre, periodo: dom_periodo, créditos: dom_creditos, curso_asig: dom_curso_asig, idTitulacion: dom_idTitulacion, idDepartamento: dom_idDepartamento)

CP: {codAsignatura}

Caj: {idTitulacion} —————> Titulacion

Caj: {idDepartamento} —————> Departamento

Profesor (idProfesor: dom_idProfesor, nombre: dom_nombre, NIP: dom_nip)

CP AutoIncrement: {idProfesor}

Historico_Convocatorias_Consumidas (NIA: dom_nia, codAsignatura: dom_codAsignatura, numConvocatorias: dom_numConvocatorias)

CP: {NIA, codAsignatura}

Caj: {NIA} —————> Alumno

Caj: {codAsignatura} —————> Asignatura

Datos_Militares (fechaIncorporacion: dom_fechaIncorporacion, modoIngreso: dom_modoingreso, cuerpo: dom_cuerpo, escala: dom_escalas, observaciones: dom_observaciones, codigoEstudios: dom_codigoEstudios, estudios: dom_estudios, cia: dom_cia, NIA: dom_nia)

Caj: {NIA} —————> Alumno

Datos_Acceso (notaPAU: dom_notapau, tipoEstudios: dom_tipoEstudios, tipoPAU: dom_tipopau, añoPAU: dom_añoPAU, convocatoriaPAU: dom_convocatoriaPAU, notaAsig1: dom_notasig1, notaAsig2: dom_notasig2, notaAdmision: dom_notadmisión, universidad: dom_universidad, NIA: dom_nia)

Caj: {NIA} —————> Alumno

Seccion (seccion: dom_seccion, curso: dom_curso, NIA: dom_nia)

CP: {NIA, curso}

Caj: {NIA} —————> Alumno

VNN: {seccion}



Matricula (grupo: dom_grupo, fechaMatricula: dom_fechaMatricula, nota1: dom_nota1, nota2: dom_nota2, convocatoria: dom_convocatoria, curso: dom_curso, reconocida: dom_reconocida, NIA: dom_nia, codAsignatura: dom_codAsignatura)

CP: {NIA, codAsignatura, curso}

Caj: {NIA} → Alumno

Caj: {codAsignatura} → Asignatura

Imparte (curso: dom_curso, seccion: dom_seccion, idProfesor: dom_idProfesor, codAsignatura: dom_codAsignatura)

CP: {curso, idProfesor, codAsignatura}

Caj: {idProfesor} → Profesor

Caj: {codAsignatura} → Asignatura



Anexo A.2: Licencia Software y Documental

La licencia BSD de 3 cláusulas es la licencia resultante de la modificación de la licencia BSD de 4 cláusulas. Es una licencia de software libre permisiva:

BSD

Copyright (c) <2020>, <Víctor Marín Esteban>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS



INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Las Licencias de derechos de autor Creative Commons y sus herramientas, forman un equilibrio dentro de la premisa tradicional de "todos los derechos reservados" que la Ley de Propiedad Intelectual establece.



Para este TFG, se ha decidido elegir una licencia Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0), que tiene las siguientes características de utilización:

- Compartir: copiar y redistribuir el material en cualquier medio o formato.
- Adaptar: reescribir, transformar y construir a partir del material.

Y bajo los siguientes términos:

- Atribución — Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- NoComercial — Usted no puede hacer uso del material con propósitos comerciales.
- CompartirIgual — Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.
- No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

