



Universidad de Jaén

Escuela Politécnica Superior
de Jaén

TRABAJO FIN DE GRADO

IA PARA EL ASESORAMIENTO EN ESTUDIOS

Alumno

Javier Martínez Jiménez

Tutor

Arturo Montejo-Ráez

Flor Miriam Plaza del Arco

Junio, 2021



Universidad de Jaén

Departamento de Informática

Don Arturo Montejo-Ráez y Doña Flor Miriam Plaza del Arco, tutores del Trabajo Fin de Grado titulado: '**IA PARA EL ASESORAMIENTO EN ESTUDIOS**', que presenta Don Javier Martínez Jiménez, otorgan el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2021

El alumno:

Javier Martínez Jiménez

Los tutores:

Arturo Montejo-Ráez
Flor Miriam Plaza del Arco

Agradecimientos

Quiero expresar mi más sincero agradecimiento a toda persona que me ha apoyado durante este proceso o ha colaborado conmigo en cualquier ámbito del desarrollo de este trabajo de final de grado.

Me gustaría recalcar la ayuda recibida por mi tutor Don Arturo Montejo-Ráez y a mi cotutora Doña Flor Miriam Plaza del Arco, por haberme guiado durante este periodo.

También me gustaría, de manera especial, agradecer a mis padres todo lo que me han apoyado y han facilitado que este proyecto haya podido ser concluido de la mejor manera posible, puesto que siempre han estado ahí para echarme una mano cuando más lo necesitaba.

NORMAS APLICADAS EN ESTE DOCUMENTO

LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA

NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
<p><i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i></p>	

Contenido

1	Especificación del trabajo	11
1.1	Introducción.....	11
1.2	Motivación	11
1.3	Objetivos del trabajo	12
1.4	Requisitos	12
1.4.1	Requisitos funcionales.....	12
1.4.2	Requisitos no funcionales.....	13
1.5	Hipótesis y restricciones	13
1.6	Tecnologías utilizadas.....	14
1.6.1	Python.....	14
1.6.2	PyCharm.....	14
1.6.3	Web scraping.....	15
1.6.4	XAMPP	15
1.6.5	MySQL.....	16
1.6.6	Pandas.....	16
1.6.7	Flask	17
1.6.8	Bootstrap	17
1.6.9	PLN.....	18
1.6.10	BERT	18
1.7	Metodología de desarrollo de software.....	19
1.8	Estimación del tamaño y esfuerzo	19
1.9	Planificación temporal	19
1.10	Diagrama de Gantt.....	21
1.11	Presupuesto	21
2	Diseño	24
2.1	Diseño Arquitectónico	24
2.2	Diseño de Base de Datos	25
2.3	Diseño del sistema web	26
2.3.1	Mockup	26
2.3.2	Storyboard	26
2.3.3	Vistas del sistema web	26
2.3.3.1	Primera vista: FORMULARIO (HOME)	26
2.3.3.2	Segunda vista: Resultado.....	27
2.3.3.3	Tercera vista: About.....	28
2.3.3.4	Header	29
2.4	Estilo.....	30
2.4.1	Logotipo	30
2.4.2	Colores	31
3	Desarrollo	32

3.1	Primera Parte – Web Scraping	32
3.1.1	¿Qué es Web Scraping y Web Crawler?	33
3.1.2	¿Por qué Web Scraping y Web Crawler?	34
3.1.3	Viabilidad de herramientas para web crawling y web scrapping.....	35
3.2	Segunda parte – Sistema Web	36
3.3	Tercera parte – Aplicación del algoritmo BERT	38
3.3.1	Preparación de los datos (Generación de vectores BERT)	39
3.3.2	Algoritmo para comparar una consulta con el contenido de las guías docentes	41
3.3.3	Insertar algoritmo de consulta en el sistema web	42
4	Pruebas.....	44
4.1	Pruebas iniciales	44
4.2	Pruebas de validación del sistema	46
4.2.1	Primer encuestado.....	46
4.2.2	Segundo encuestado.....	47
4.2.3	Tercer encuestado.....	48
4.2.4	Cuarto encuestado	49
4.2.5	Resultados finales	50
5	Resultados.....	51
5.1	Primer resultado	51
5.2	Segundo resultado	53
5.3	Tercer resultado	55
6	Conclusiones y trabajos futuros.....	57
6.1	Conclusiones.....	57
6.2	Trabajos futuros	59
7	Bibliografía	61
8	Apéndices.....	62
8.1	Código fuente.....	62
8.2	Requerimientos del sistema.....	62
8.3	Creación de la base de datos	62
8.4	Instalación de bibliotecas de Python.....	64
8.5	Instalación y configuración del sistema web.....	64
8.6	Manual de Usuario	65
8.6.1	Manual Utilización Script Web Scraping	65
8.6.2	Manual Usuario Sistema Web	66

Índice de ilustraciones

Ilustración 1 - Logo Python	14
Ilustración 2 - Logo PyCharm	14
Ilustración 3 – BeautifulSoup Ilustración 4 - Resquest	15
Ilustración 5 - Logo XAMPP	15
Ilustración 6 - Logo MySQL	16
Ilustración 7 - Logo Pandas	16
Ilustración 8 - Logo Flask	17
Ilustración 9 - Logo Bootstrap	17
Ilustración 10 - Logo BERT	18
Ilustración 11 - Diseño Arquitectónico general de Flask	24
Ilustración 12 - Diagrama Entidad-Relación	25
Ilustración 13 - Primera vista del sistema web	27
Ilustración 14 - Segunda vista del sistema web	28
Ilustración 15 - Tercera vista del sitio web	29
Ilustración 16 - Header del sitio web	30
Ilustración 17 - Logo sistema web	30
Ilustración 18 - Diagrama Funcionamiento BERT en el Sistema	42
Ilustración 19 - Eliminación asignaturas repetidas	44
Ilustración 20 - Rediseño de vista Resultado	45
Ilustración 21 - Primer resultado de titulaciones	51
Ilustración 22 - Primer resultado de asignaturas	52
Ilustración 23 - Segundo resultado de titulaciones	53
Ilustración 24 - Segundo resultado de asignaturas	54
Ilustración 25 - Tercer resultado de titulaciones	55
Ilustración 26 - Tercer resultado de asignaturas	56
Ilustración 27 - Primer paso importación base de datos	63
Ilustración 28 - Segundo paso importación base de datos	63
Ilustración 29 - Ejemplo script extractorContenidosYDB.py	65
Ilustración 30 - Página Principal Web Manual Usuario	66

Índice de tablas

Tabla 1 - Planificación temporal del proyecto	20
Tabla 2 - Sueldo del personal.....	21
Tabla 3 - Recursos esenciales oficina	22
Tabla 4 - Costes totales del proyecto	22
Tabla 5 - Primer encuestado	46
Tabla 6 - Primera encuesta	47
Tabla 7 - Segundo encuestado.....	47
Tabla 8 - Segunda encuesta	47
Tabla 9 - Tercer encuestado	48
Tabla 10 - Tercera encuesta.....	48
Tabla 11 - Cuarto encuestado	49
Tabla 12 - Cuarta encuesta	49
Tabla 13 - Resultados finales encuestas	50
Tabla 14 - Bibliotecas necesarias.....	64

1 ESPECIFICACIÓN DEL TRABAJO

En este capítulo se presenta la especificación del trabajo, con una estructura y contenidos **inspirados** en los criterios y recomendaciones que establece la norma UNE 157801:2007 - “*Criterios Generales para la elaboración de proyectos de Sistemas de Información*”.

1.1 Introducción

El objetivo de este Trabajo de Fin de Grado es desarrollar un sistema web en el que un futuro estudiante de la Universidad de Jaén, rellenando un formulario sea capaz de ser recomendado con la carrera que se ajusta más a sus gustos y con las asignaturas que se asemejan más a lo introducido en el formulario, con sus respectivos enlaces a las guías docentes y Grados de estos.

1.2 Motivación

Este proyecto es realmente interesante, puesto que a la vez que busca desarrollar un sistema web con el que ayudar a futuras generaciones de estudiantes de la Universidad de Jaén, también permite al desarrollador realizar un estudio de cómo funciona el algoritmo BERT de Google y cómo puede ser implementado en una aplicación.

En la actualidad Google domina y es pionera prácticamente en todos los campos relacionados a cualquier tipo de tecnología, por lo que aprender cómo funciona uno de sus posibles futuros buques insignias siempre es un acierto.

Otro de los intereses de este trabajo es que se deberá producir una solución a medida de Web Scraping, otra tecnología que actualmente es realmente interesante aprender y dominar puesto que cada día hay más necesidad por parte de las empresas de digitalizarse y mostrarse en la Web.

1.3 Objetivos del trabajo

El objetivo principal de este trabajo es desarrollar un sistema web que recoja todas las asignaturas y grados de la universidad y sea capaz de forma rápida y eficiente indicar a una persona según sus gustos, hobbies o cualquier inquietud que se le ocurra al usuario, indicar cual es la opción que se asemeja más del amplio abanico que ofrece la Universidad de Jaén.

La aplicación web en la que se ofrecerá este servicio debe ser sencilla y básica, siguiendo la regla *¡No me hagas pensar! de Steve Krug* que sea lo más fácil posible para el usuario independientemente de su nivel de informática.

1.4 Requisitos

1.4.1 Requisitos funcionales

- 1 El sistema debe permitir la entrada de texto libre para que el usuario describa sus preferencias o intereses.
- 2 El sistema debe, a partir de dichos intereses, facilitar una lista de titulaciones y asignaturas afines.
- 3 El sistema debe facilitar los enlaces a las páginas de las titulaciones y a las guías docentes de las asignaturas mostradas.

1.4.2 Requisitos no funcionales

1 Disponibilidad

- a. El sistema debe mantenerse activo a todas horas, todos los días del año

2 Accesibilidad

- a. La web estará disponible desde cualquier navegador.

3 Interfaz de usuario y diseño

- a. El sistema debe ofrecer al usuario un diseño intuitivo en el que sea capaz de comprender con facilidad el total funcionamiento del sistema.
- b. No se usarán más de tres colores en la interfaz, estos son verde, negro y blanco. Aunque se podrán usar imágenes que contengan más colores.

1.5 Hipótesis y restricciones

El TFT se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto será de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Por consiguiente, la principal restricción aplicable es la limitación de la duración del trabajo.

El sistema debe dar una respuesta en tiempo real y no debe requerir de registro para poder ser usado.

1.6 Tecnologías utilizadas

Para el desarrollo de este trabajo se han utilizado diversas tecnologías.

1.6.1 Python

Python es un lenguaje de programación interpretado, multiparadigma y multiplataforma usado, principalmente, en Big Data, Data Science, frameworks de pruebas y desarrollo web. Esto lo convierte en un lenguaje de propósito general de gran nivel debido a su extensa biblioteca, cuya colección ofrece una amplia gama de instalaciones [1].



Ilustración 1 - Logo Python

1.6.2 PyCharm

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python [2].



Ilustración 2 - Logo PyCharm

1.6.3 Web scraping

Web Scraping es una técnica utilizada para extraer información de páginas web de forma automatizada. Es este trabajo, esta técnica será utilizada para obtener el apartado de Contenidos de todas las guías docentes de cada una de las asignaturas de Universidad de Jaén. Para ello se evaluarán bibliotecas de Python como BeautifulSoup y Resquest [3].



Ilustración 3 – BeautifulSoup



Ilustración 4 – Resquest

1.6.4 XAMPP

XAMPP es un paquete de software libre, que nos interesará porque ofrece un sistema de gestión de bases de datos MySQL y un servidor web completo Apache [4].



Ilustración 5 - Logo XAMPP

1.6.5 MySQL

MySQL es un sistema de gestión de bases de datos relacional que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle [5].



Ilustración 6 - Logo MySQL

1.6.6 Pandas

Pandas es un paquete de Python que proporciona estructuras de datos similares a los dataframes de R. Pandas depende de Numpy, la librería que añade un potente tipo matricial a Python [6].



Ilustración 7 - Logo Pandas

1.6.7 Flask

Flask es un micro Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC. Es muy interesante para nuestro trabajo puesto que estamos siguiendo las reglas de Steve Krug y nuestro sistema web queremos que sea lo más básico posible en cuanto a interfaz [7].



Ilustración 8 - Logo Flask

1.6.8 Bootstrap

Bootstrap es un framework CSS de interfaz de usuario, de código abierto, que hace que el desarrollo web sea más rápido y sencillo. En este proyecto ha sido utilizado para simplificar el diseño del sistema web [8].

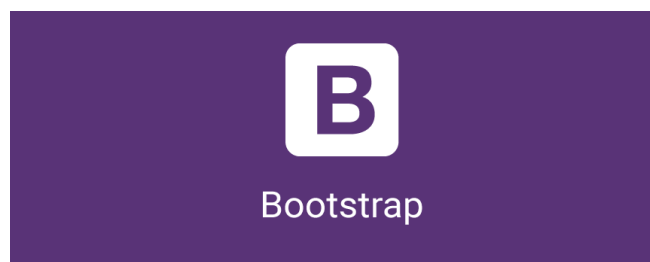


Ilustración 9 - Logo Bootstrap

1.6.9 PLN

PLN significa Procesamiento del Lenguaje Natural. Es un campo dentro de la inteligencia artificial y la lingüística aplicada que estudia las interacciones mediante uso del lenguaje natural entre los seres humanos y las máquinas. Se centra en el procesamiento de las comunicaciones humanas, dividiéndolas en partes, e identificando los elementos más relevantes del mensaje. Con la Compresión y Generación de Lenguaje Natural, busca que las máquinas consigan entender, interpretar y manipular el lenguaje humano [15].

Este es importante en nuestro proyecto puesto que el algoritmo BERT se basa en este campo de la inteligencia artificial.

1.6.10 BERT

BERT es un algoritmo de codificación de textos desarrollado por Google. Se trata de un algoritmo bidireccional, es decir, analiza las frases de búsqueda en ambas direcciones, de tal forma que puede relacionar todas las palabras de una consulta entre sí, en vez de considerarlas de forma individual. También se trata de un algoritmo para el modelado de lenguaje, es decir, modela las distribuciones de palabras a alto nivel, capturando significado que le ayuda a entender nexos, pronombres y preposiciones [9].

Utilizaremos este algoritmo en nuestro sistema web puesto que mediante este podremos generar vectores de consultas y asignaturas y calcular determinar las equivalencias entre el texto introducido por el usuario y el contenido de cada una de las guías docentes de la Universidad.



Ilustración 10 - Logo BERT

1.7 Metodología de desarrollo de software

La metodología seleccionada para el desarrollo del proyecto ha sido Scrum [10], que se trata de un proceso en el que se aplica de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en este caso con ambos de mis tutores.

Se ha elegido Scrum porque este se basa en ir realizando entregas parciales y regulares del producto final que se quiere obtener, en este caso cada semana o dos semanas se realizaba una reunión con los tutores para actualizar el estado del proyecto o también se realizaban reuniones cuando se llegaba a un punto crítico o se alcanzaba un objetivo del proyecto.

1.8 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFG, no existen restricciones de tipo económico, sino de tipo temporal, unas 300 horas. Por consiguiente, los cálculos de tamaño del proyecto están supeditados al tiempo disponible. En cuanto al esfuerzo, se dispone de tan un solo efectivo (Javier Martínez Jiménez).

1.9 Planificación temporal

Puntos a tener en cuenta en la estimación temporal del proyecto:

- ☐ El inicio de este trabajo empieza con una reunión el ocho de febrero de 2021, en la cual al alumno se le introduce el objetivo final a conseguir y se le indica los objetivos intermedios a este objetivo final.
- ☐ De media se le ha dedicado tres horas diarias de lunes a viernes por parte del alumno para conseguir los objetivos.
- ☐ Hay que tener en consideración que del día veintiuno de mayo al cuatro de junio es un periodo de exámenes para el Grado de Ingeniería informática por lo que durante este tramo de tiempo ha sido casi imposible hacer nada sobre este trabajo.

- Además de en otros periodos vacacionales en los que el tiempo dedicado a la consecución de los objetivos se vio reducido.

Tarea	Estimación temporal en semanas
Análisis del problema	2
Estudio de la estructura HTML de la página Web con las guías docentes de la UJA	1
Creación script para extracción de contenidos mediante Web scraping	1
Creación de Mock Up y Storyboard del sistema web final	1
Creación del diseño de base de datos y almacenamiento de datos obtenidos en el proceso de Web scraping	1
Análisis del framework a utilizar para la aplicación final	1
Creación del sistema web y vistas de este	1
Fase de análisis del algoritmo BERT	2
Preparación de los datos que el script utilizará para calcular semejanzas	1
Creación del algoritmo para calcular semejanzas entre un texto entrante y todas las asignaturas existentes en la UJA	2
Implementación del algoritmo en el sistema web	1
Fase de pruebas del sistema web y corrección de errores	1

Tabla 1 - Planificación temporal del proyecto

1.10 Diagrama de Gantt

El diagrama de Gantt del proyecto es el que podemos ver en la Figura 11.

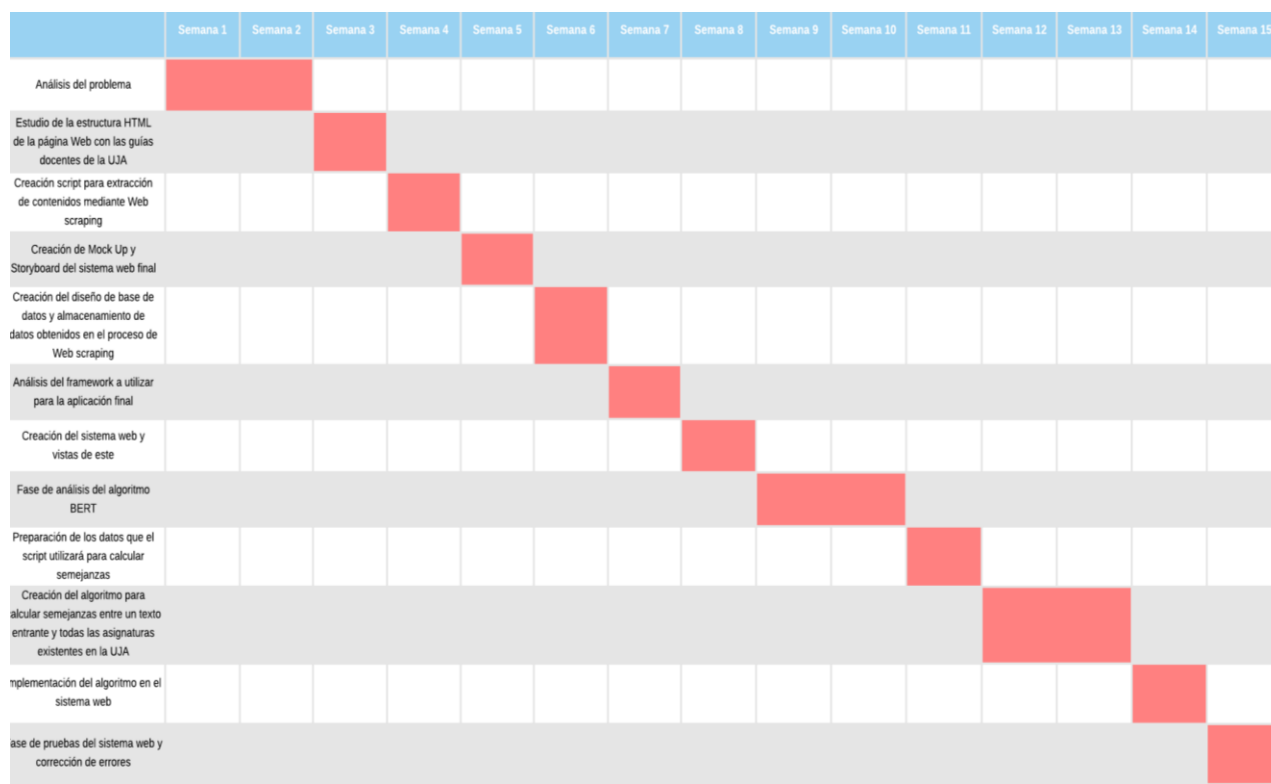


Ilustración 11 - Diagrama de Gantt

1.11 Presupuesto

Respecto al personal empleado en desarrollar el sistema web se tratará de una única persona, que será un desarrollador web con poca experiencia. El sueldo de un profesional de estas características oscila entre los 1.470 y 1.584 euros mensuales en España. Nos pondremos en el peor de los casos y calcularemos el importe total del sueldo del personal como 1.584 euros mensuales.

Concepto	Meses	Importe mensual	Total Importe
Sueldo empleado	5	1.584.00 €	7.920.00 €

Tabla 2 - Sueldo del personal

Todo el software utilizado durante el proyecto es gratuito puesto que se han usado programas con licencia gratuita o se han usado claves académicas, por lo que la suma total de este será de cero euros.

En cuanto a recursos de hardware se ha utilizado un portátil MSI GF62 8RD con un precio de 1.149 euros, puesto que este ordenador ha sido utilizado durante cinco meses, es decir, alrededor de 180 días y su vida útil es de cuatro años, este tiene un coste de amortización de 141,65 euros.

Este proyecto ha sido posible desarrollarlo mediante teletrabajo por lo que no se han producido gastos de alquiler de una oficina.

También tenemos que tener en cuenta que necesitaremos conexión a internet, además, se producirá un consumo de luz y agua, la suma total de todo es 325 euros por los cinco meses que ha costado realizar el proyecto.

Concepto	Meses	Importe mensual	Total Importe
Agua y electricidad	5	45,00 €	225,00 €
Internet	5	20,00 €	100,00 €
			325,00 € totales

Tabla 3 - Recursos esenciales oficina

Concepto	Coste	Total Importe
Sueldo empleado	5 meses * 1.584,00 €	7.920.00 €
Agua y electricidad	5 meses * 45,00 €	225.00 €
Internet	5 meses * 20,00 €	100.00 €
Software	Gratuito	0.00 €
MSI GF62 ORD	Una unidad	141,65 €
		8.386,65 € totales

Tabla 4 - Costes totales del proyecto

Calculados todos los gastos posibles, la suma del coste total para producir el proyecto será de 10372 euros. Hay que tener en cuenta que esto es una aproximación, es decir, son costes estimados del proyecto.

2 DISEÑO

Realizar un buen diseño es fundamental para obtener un resultado que para el usuario final resulte de confianza, profesional y además le permita navegar por el sistema web de forma rápida, segura y con comodidad sin tener altos niveles de conocimiento informáticos.

Un diseño de calidad también nos permitirá como desarrolladores web poder desarrollar el sistema web de forma más rápida y con seguridad de que todo se esta realizando de la forma correcta.

2.1 Diseño Arquitectónico

El diseño elegido para el desarrollo de este sistema web es el modelo MVC (Modelo-Vista-Controlador), como podemos ver en la Figura 12, puesto que es el modelo perfecto para el framework elegido en este caso que es Flask.

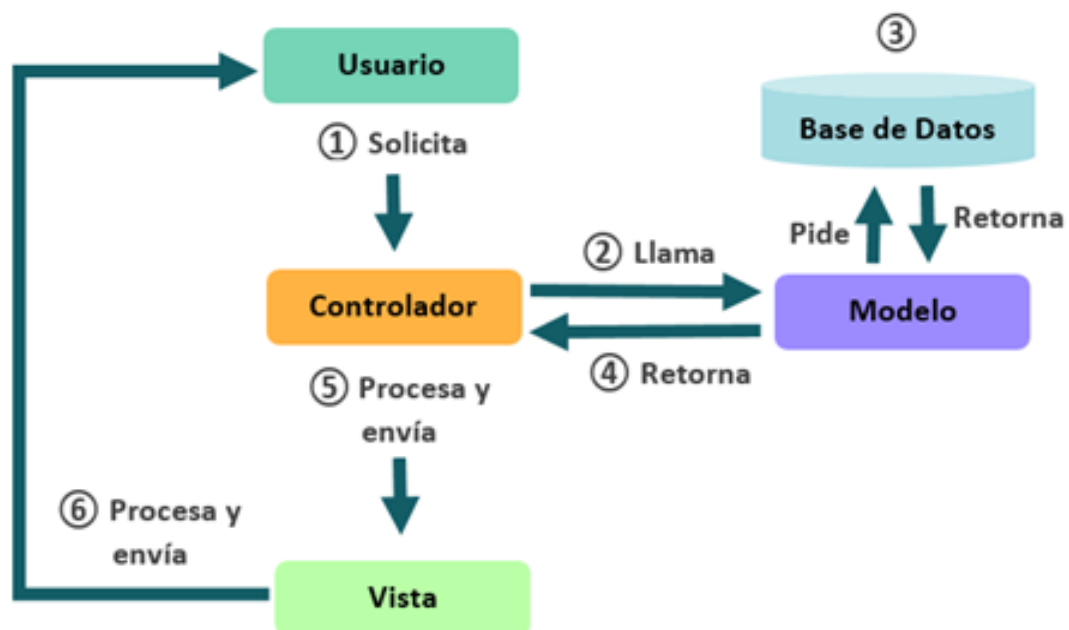


Ilustración 12 - Diseño Arquitectónico general de Flask

2.2 Diseño de Base de Datos

El modelo elegido para la base de datos de este sistema web es el de entidad-relación ER, mediante el cual se representará la abstracción y percepción del sistema mediante un conjunto de objetos, el modelo ER de la base de datos es el que podemos ver en la Figura 13.

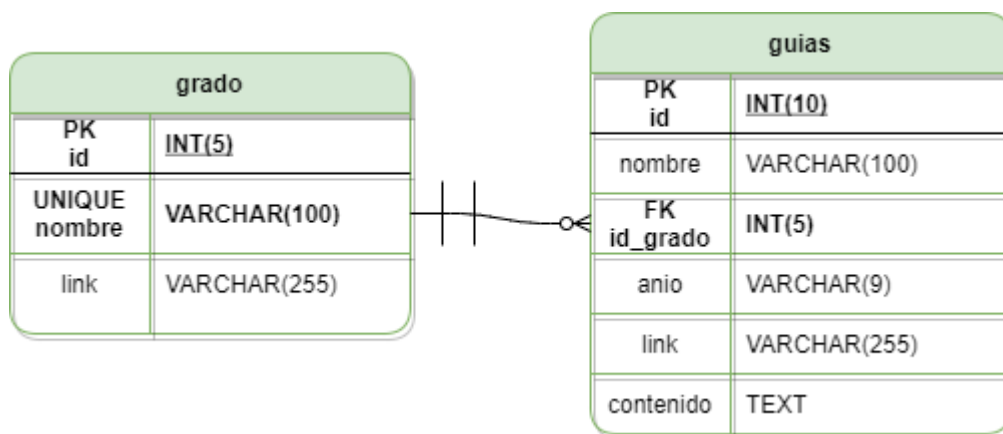


Ilustración 13 - Diagrama Entidad-Relación

Como se puede observar es un diagrama simple en el que solo encontramos dos entidades (**Grado y Guías**), con una relación de **Varios a uno**, cabe recalcar de estas entidades que la clave primaria de la entidad Grado es un entero que será el ID del grado y además este será una clave foránea en la entidad Guías mediante la cual se realizará la relación entre estas identidades y la clave primaria de Guías será un entero que corresponde con el ID de la guía docente de la asignatura.

2.3 Diseño del sistema web

El diseño del sistema web se ha realizado mediante un *mockup* y un *storyboard*.

2.3.1 Mockup

Un *mockup* es un modelo o un prototipo utilizado para exhibir o probar un diseño. Gracias a este, el diseñador puede analizar y mostrar cómo avanza su trabajo a un cliente. De esta forma, si es necesario realizar cambios, dichas modificaciones se podrán llevar a cabo antes de la presentación final del producto [12].

2.3.2 Storyboard

Un *storyboard* es un conjunto de ilustraciones presentadas de forma secuencial con el objetivo de servir de guía para entender el funcionamiento de un sistema [13].

2.3.3 Vistas del sistema web

Las siguientes vistas del *mockup* y *storyboard* del sistema han sido generadas mediante la herramienta **Draw.io** de Google.

2.3.3.1 Primera vista: FORMULARIO (HOME)

Esta es la vista principal del sistema web, la que aparecería al usuario final cuando entra en la página web. En esta, el usuario final rellenaría el cuadro de texto hablando sobre sí mismo, es decir, contando sus intereses, hobbies, gustos o lo que más le ha gustado estudiar en etapas formativas previas. Una vez rellenado el cuadro con lo que le interesa, pulsaría en el botón **Enviar** del centro y sería redirigido a la siguiente vista. Esta vista la podemos observar en la Figura 14.

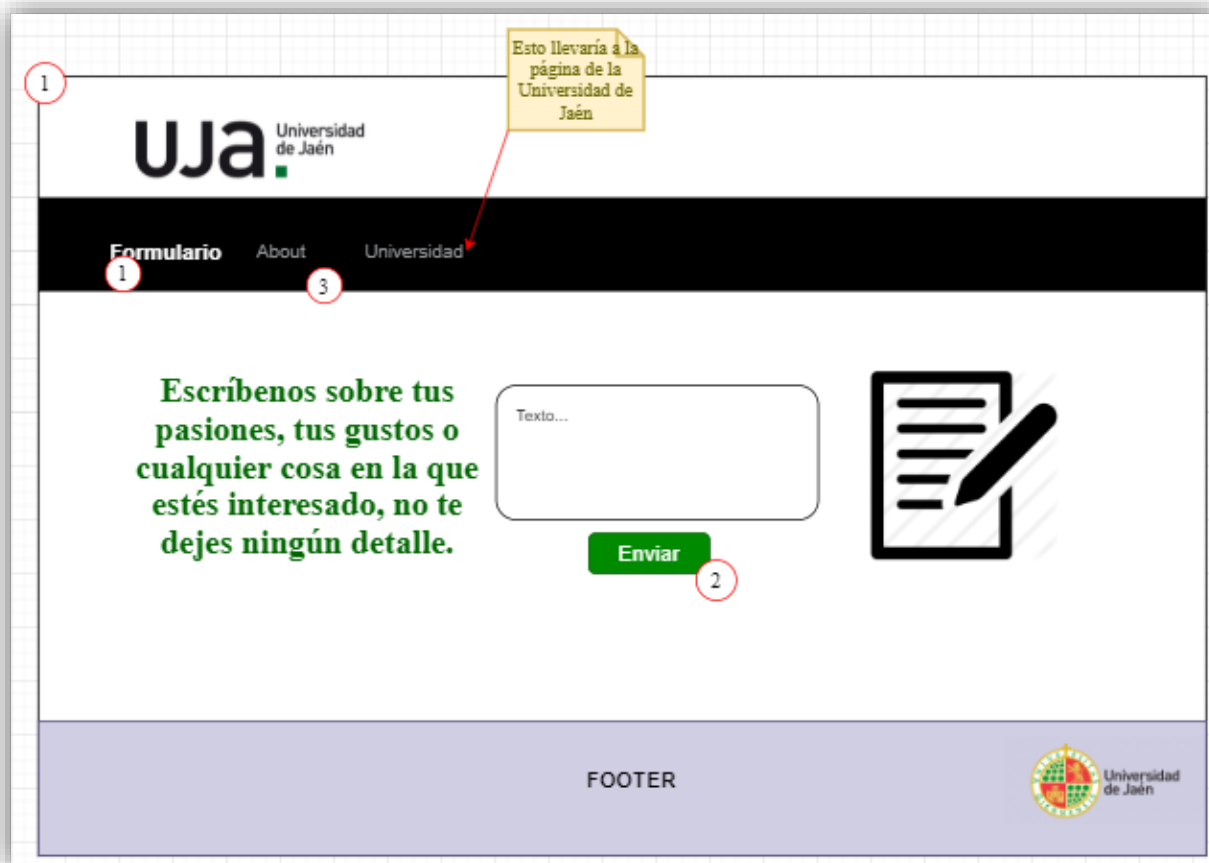


Ilustración 14 - Primera vista del sistema web

2.3.3.2 Segunda vista: Resultado

Esta es la vista que se genera cuando se pulsa el botón Enviar en la vista anterior. Podemos observarla en la Figura 15. En esta encontramos la carrera que más se asemeja a lo introducido por el usuario. Además, aparece una tabla con el top diez de asignaturas que se asemejan a la información introducida por el usuario, si se pincha sobre el nombre de la asignatura el usuario será guiado a la guía de la asignatura.

En la tabla también aparece el grado que corresponde con la asignatura, el año en la que se cursó la asignatura y el tanto por ciento de semejanza de la asignatura con respecto a la consulta realizada por el usuario.

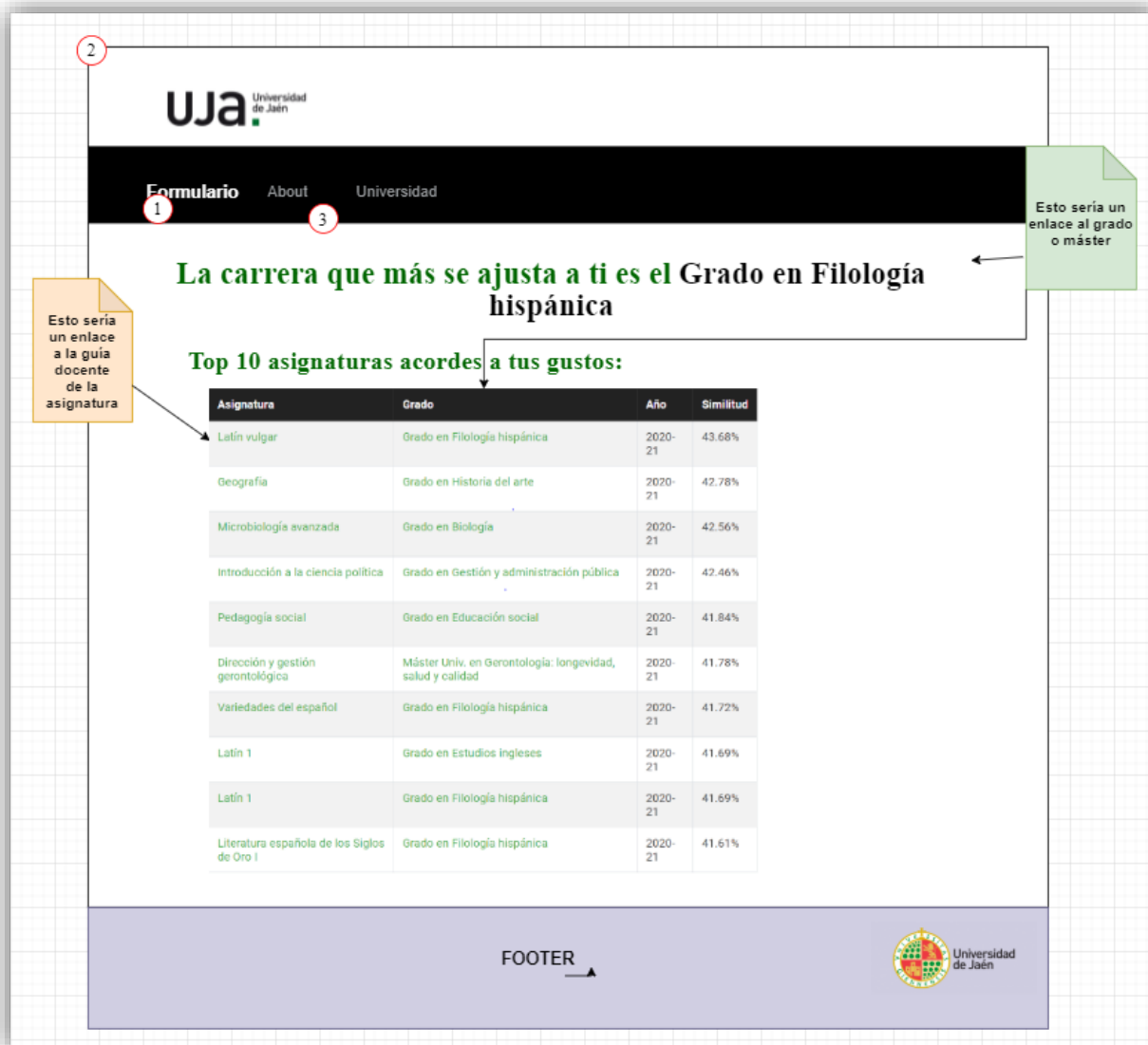


Ilustración 15 - Segunda vista del sistema web

2.3.3.3 Tercera vista: About

La tercera vista está orientada a dar un poco de información sobre lo que hace el sistema web, en el caso de que el usuario final no supiese como usar la página web.

En esta vista tenemos un botón que es **Descubre tu futuro** y cuando lo pulsamos somos guiados a la página principal del sistema web para poder iniciar la consulta sobre las asignaturas más acordes al usuario. Esta vista la podemos observar en la Figura 16.



Ilustración 16 - Tercera vista del sitio web

2.3.3.4 Header

El header de este sistema web estará compuesto por dos secciones.

La primera de estas ocupa la parte superior de las vistas, contiene un fondo blanco y además el logo de la página web, si el usuario hace click sobre este será redirigido a la página principal del sistema, que sería la vista **Formulario (Home)**.

La segunda sección del header se trata de una barra de navegación, con un fondo negro para hacer contraste con respecto a la sección superior, en esta barra de navegación encontramos tres posibles destinos, el primero que es **Formulario** nos lleva a la página inicial del sistema web donde el usuario puede rellenar un formulario para obtener la información sobre las asignaturas y grados que más se ajustan a su persona. El segundo destino es **About**, este nos lleva a la vista donde el usuario puede obtener más información sobre la funcionalidad del sistema y la última posibilidad es **Universidad**, esta opción se trata de un enlace a la página principal de la universidad de Jaén, es decir, a <https://www.ujaen.es/>.



Ilustración 17 - Header del sitio web

2.4 Estilo

En este sistema web se ha seguido un estilo continuista con respecto a la página web oficial de la universidad de Jaén (<https://www.ujaen.es/>), puesto que la funcionalidad de este sistema web está estrechamente unido con un servicio que podría llegar a implementar la universidad para ofrecérselo a futuros integrantes de la comunidad universitaria jienense.

2.4.1 Logotipo

En este apartado del estilo se ha elegido directamente el mismo logo que el que encontramos en la página web oficial de la universidad.

Este está compuesto por las iniciales de la universidad (**UJA**), además del nombre completo de la universidad (**Universidad de Jaén**).



Ilustración 18 - Logo sistema web

2.4.2 Colores

Los colores que predominan en el sistema web son el verde, el negro y el blanco, puesto que estos son los colores con los que más se puede relacionar a la universidad de Jaén, sobre todo con el color verde, que será el que se utilice para la gran mayoría de texto del sistema.

3 DESARROLLO

En el proceso de desarrollo de este proyecto se puede diferenciar claramente tres partes: web scraping, desarrollo del sistema web y desarrollo del algoritmo BERT e implementación en aplicación web.

3.1 Primera Parte – Web Scraping

Esta parte comprende el Sprint 1 (**Análisis del problema**), el Sprint 2 (**Estudio de la estructura HTML de la página Web con las guías docentes de la UJA**) y el Sprint 3 (**Creación script para extracción de contenidos mediante Web scraping**) de la planificación del proyecto.

El primer problema a solucionar que surgió durante el desarrollo de este proyecto fue el de recopilar todas las guías docentes de todas las titulaciones que se imparten en la Universidad de Jaén y extraer de cada una de ellas el temario, es decir, la sección **5. CONTENIDOS** de las guías docentes.

Este problema se podría solucionar descargando manualmente todas las guías docentes y almacenándolas, pero esto sería una solución tediosa, que nos ocuparía mucho tiempo y en el que sería muy fácil que nos saltásemos guías docentes y no nos la descargásemos, por lo tanto, tendríamos errores en nuestro sistema web. Debido a esto había que buscar una solución con la que automatizásemos esta tarea, que nos asegurará no tener fallos en el sistema y poder obtener las guías docentes de forma rápida.

Tras realizar un análisis de como confrontar este problema, llegamos a la conclusión de que la mejor forma de solucionarlo era mediante la técnica de Web Scraping y Web Crawler.

3.1.1 ¿Qué es Web Scraping y Web Crawler?

El Web Scraping es un proceso por el cual se utilizan bots para extraer datos y contenidos de una web de forma automática. Mediante estos bots se simula la navegación de un humano en la World Wide Web. Con esta técnica podemos transformar datos sin estructura en la web en datos estructurados que podemos almacenar y analizar. [3]

El proceso de Web Scraping sigue los siguientes pasos:

- **Solicitud-respuesta:**
 - Lo primero que hace es solicitar al sitio web de destino el contenido de una URL específica.
 - A cambio, el scraping obtiene la información solicitada en HTML.
- **Analizar y extraer:**
 - Mediante un lenguaje de programación, generalmente Python, se procesa el código como texto y se produce una estructura en la memoria sobre la que el lenguaje entiende y puede trabajar.
- **Descarga de los datos:**
 - La parte final es la descargar los datos y guardarlos, se pueden guardar en formatos CSV, JSON o directamente en una base de datos con los que podremos realizar cualquier tipo de operación.

El Web Crawler se basa en un bot que navega a través de un sitio web y localiza y recupera la información a través de las distintas capas del lugar, es decir, navega a través de los datos e información y obtienen lo que sea relevante para el proyecto, generalmente hipervínculos. [11]

El proceso de Web Crawler sigue los siguientes pasos:

- ☐ Se elige una URL inicial o URL iniciales.
- ☐ Se indica como parte de la frontera.
- ☐ Selecciona la URL de la frontera.
- ☐ Se obtiene la página web correspondiente a esta URL.
- ☐ Analiza esta página web para encontrar nuevos enlaces URL.
- ☐ Todas las URL recién encontradas se añaden a la frontera.
- ☐ Se vuelve al tercer paso y se repite hasta que la frontera este vacía.

3.1.2 ¿Por qué Web Scraping y Web Crawler?

Estas tecnologías son elegidas para este proyecto porque nos ofrecen una solución perfecta al problema de obtener todos los contenidos de las guías docentes de la Universidad de Jaén, puesto que todas las guías docentes se pueden obtener a partir de la siguiente URL inicial

<https://uvirtual.ujaen.es/pub/es/informacionacademica/catalogoguiasdocentes/>, con esta podemos ir obteniendo los enlaces de URL que nos interesa, hasta que lleguemos a ciertas URL frontera donde aplicaremos el Web Scraping para obtener de forma automatizada el contenido que nos interesa de las guías.

El contenido extraído de las guías es el siguiente:

- ☐ ID de la guía.
- ☐ Nombre de la asignatura de la guía.
- ☐ Nombre del grado.
- ☐ Año.
- ☐ Link de la guía docente.

Este contenido será almacenado directamente en una base de datos, para su posterior procesamiento y uso en nuestro sistema web.

3.1.3 Viabilidad de herramientas para web crawling y web scrapping

Actualmente, existen numerosas herramientas software gratuitas y de pago tanto para realizar Web Scraping como Web Crawler, pero realmente estas no nos ofrecen un servicio de calidad, suelen solo funcionar para determinados casos específicos y para parsear información en páginas web de determinadas grandes empresas como Facebook o Amazon.

Algunas de estas herramientas software para hacer Web Scraping son Octoparse, Web Sundew, ParseHub o Easy Web Extract. Todas estas herramientas fueron probadas, pero ninguna satisfacía lo que se buscaba.

Por esto, resulto una mucho mejor solución desarrollar un script o un programa con el que realizar estas acciones, es decir, realizar una solución a medida para un determinado sitio web.

Una muy buena opción para llevar a cabo esto es utilizar el lenguaje de programación Python, puesto que cuenta con una amplia gama de bibliotecas con las que realizar estas técnicas como puede ser BeautifulSoup, Requests, Scrapy o Selenium, además estas bibliotecas pueden ser utilizadas de forma simultánea para obtener la solución deseada.

En el caso de este proyecto, tras probar todas las bibliotecas antes mencionadas se determinó que las mejores para solucionar el problema de obtener el contenido de todas las guías docentes eran las bibliotecas de BeautifulSoup, con la que podemos convertir el contenido HTML o XML del sitio web en un árbol complejo de objetos Python, que en este caso nos servirá para acceder a las URL hijas hasta llegar a la URL frontera deseada, y Requests, con la que haremos peticiones web para poder extraer la información de la página.

El script final para la extracción de los contenidos es **extractorContenidosYDB.py**.

3.2 Segunda parte – Sistema Web

Esta parte comprende el Sprint 4 (**Creación de mockup y storyboard del sistema web final**), el Sprint 5 (**Creación del diseño de base de datos y almacenamiento de datos obtenidos en el proceso de Web scraping**), el Sprint 6 (**Análisis del framework a utilizar para la aplicación final**) y el Sprint 7 (**Creación del sistema web y vistas de este**) de la planificación del proyecto.

El segundo problema a resolver de este proyecto es el de montar un sistema web en que mostrar nuestra solución.

Empezaremos a resolver este problema mediante la creación de un Mock Up y Storyboard de lo que sería el sitio web final para que este sea validado por nuestro superior y nos dé luz verde para continuar con el desarrollo del proyecto. Esta tarea la podemos llevar a cabo mediante Draw.io de Google, puesto que es una herramienta gratuita que no envidia nada de otras herramientas premium de pago puesto que incluye miles de opciones de personalización.

Una vez se recibe el visto bueno se pasa al siguiente Sprint, en este caso el 5, este consiste en volver a realizar el diseño de la base de datos que se utilizara en el sistema web, por lo que volveremos a usar Draw.io de Google puesto que también es una gran opción a la hora de diseñar cualquier tipo de diagrama.

Cuando el diseño de la base de datos está hecho, debemos empezar a almacenar la información que obtendremos mediante el script de extracción de contenidos de las guías docentes (**extractorContenidosYDB.py**). Para poder empezar a ejecutar este script previamente debe ser creada la base de datos. Esta será creada mediante MySQL, que en este caso la hemos obtenido mediante el entorno XAMPP que ofrece una distribución gratuita de esta. Una vez las tablas han sido creadas siguiendo el diseño Entidad-Relación de la base de datos y la tabla **“grado”** ha sido rellenada manualmente con la información necesaria de cada grado, podemos empezar a ejecutar el script de Web Scraping, con el que obtendremos todos los contenidos de las guías docentes y serán almacenados en la tabla **“guias”**.

Teniendo todos los datos almacenados hay que empezar a pensar en un framework mediante el cual podamos crear nuestro sistema web. En la elección del framework tenemos que tener en cuenta que estamos trabajando con el lenguaje Python, por lo que demos seleccionar un sistema que trabaje en este lenguaje, las dos opciones que resaltaron sobre todas las posibles opciones son los framework **Django** y **Flask**.

□ **Django**

- Django se trata del framework por excelencia de Python, es gratuito y de código abierto. Cuenta con una comunidad enorme y activa. Su principal característica es que permite desarrollar aplicaciones de cualquier tipo de complejidad en tiempos bastantes razonables. Además, ofrece una gran seguridad, es muy escalable y versátil. Utiliza un ORM (Object Relational Mapper) para asignar sus objetos a tablas de base de datos, con las que hace consultas. Trabaja con MySQL, PostgreSQL, SQLite y Oracle, aunque otras bases de datos también pueden ser utilizadas con controladores de terceros.

□ **Flask**

- Flask se trata de un microframework de código abierto, cuya principal intención es ser simple y pequeño, consiste en un grupo de módulos. Su curva de aprendizaje es bastante rápida y permite desarrollar sitios web de forma muy veloz. Sigue el patrón MVC (Modelo-Vista-Controlador). Incluye un servidor web de desarrollo y un depurador integrado para poder probar las aplicaciones que estamos desarrollando y así poder ir comprobando los resultados.

Tras el estudio de ambos frameworks se decide que para este proyecto se va a seleccionar **Flask**, puesto que el sistema web que se va a desarrollar en este proyecto no es de una gran complejidad, por lo que no se le podría llegar a sacar todo el partido a un framework como Django, mientras que Flask encaja perfectamente en la solución buscada, puesto que buscamos un sistema web sencillo y pequeño en el que implementar únicamente unas pocas funcionalidades.

El último ciclo de esta parte es crear la base del sistema web y sus vistas, para empezar esta iteración debemos tener un entorno Python en el que poder desarrollar el sitio web. En este caso se ha elegido PyCharm que nos permite crear entornos virtuales para Python de forma muy sencilla. Una vez creado el entorno debemos instalar Flask en este (`pip install flask`). Realizado este paso ya se puede empezar con el desarrollo del sistema web, con sus correspondientes vistas, tal y como se especificó en el Mock Up y Storyboard creados en el Sprint 4.

Hay que tener en cuenta que en esta parte no se obtendrá la aplicación final puesto que todavía hay que desarrollar los scripts con los que obtener las funcionalidades finales del sistema.

3.3 Tercera parte – Aplicación del algoritmo BERT

Esta parte comprende el Sprint 8 (**Fase de análisis del algoritmo BERT**), el Sprint 9 (**Preparación de los datos que el script utilizará para calcular semejanzas**), el Sprint 10 (**Creación del algoritmo para calcular semejanzas entre un texto entrante y todas las asignaturas existentes en la UJA**), el Sprint 11 (**Implementación del algoritmo en el sistema web**) y el Sprint 12 (**Fase de pruebas del sistema web y corrección de errores**) de la planificación del proyecto.

El tercer problema que se debe tratar en este proyecto es el generar un **vector BERT** para cada uno de los temarios de las guías docentes que hemos extraído en las fases anteriores del desarrollo.

Para poder generar estos vectores lo primero que haremos será realizar un análisis de qué es y cómo funciona **BERT**.

- **BERT** (Bidirectional Encoder Representations from Transformers) es una técnica basada en redes neuronales para el preentrenamiento del procesamiento del lenguaje natural (PLN) desarrollada por Google y publicada en 2018 por Jacob Devlin.
- El **objetivo de Bert** es interpretar nuestro lenguaje de búsqueda de un modo más natural.
- El funcionamiento del **algoritmo BERT** consiste en utilizar la bidireccionalidad, mediante esta analiza una misma frase en dos direcciones: tanto a la izquierda como a la derecha de la palabra clave. Siguiendo este patrón el algoritmo es capaz de entender con profundidad el contexto y la temática de cada oración.

3.3.1 Preparación de los datos (Generación de vectores BERT)

Una vez hemos analizado cómo funciona el algoritmo BERT de Google podemos dar por completado la iteración ocho del proyecto, ahora nos toca pasar al siguiente ciclo.

El ciclo nueve consiste en crear los vectores BERT de los contenidos que tenemos almacenados en nuestra base de datos. Este problema lo podemos resolver mediante un script con que el que automatizaremos esta tarea de generación de vectores. Para la generación de los vectores necesitaremos importar la biblioteca de Transformers de Huggingface y las bibliotecas de PyTorch y Pandas.

En este script deberemos importar **BertTokenizer** y **BertModel** de la biblioteca **transformers**, **BertTokenizer** es un generador de tokens, para nuestro caso nos interesa ya que este cuenta con una funcionalidad para descargar un tokenizador preentrenado mediante una simple línea de código. **BertModel** nos interesa básicamente por lo mismo que **BertTokenizer**, con el podemos descargar un modelo concreto preentrenado que concuerde con los parámetros que necesitamos. Para este proyecto utilizaremos '**dccuchile/bert-base-spanish-wwm-uncased**' que el mejor que podemos encontrar para el castellano.

Una vez cargado el tokenizador y el modelo podemos conectarnos a la base de datos para obtener todos los contenidos de las guías docentes.

Llegado a ese punto debemos pensar en una estructura para almacenar los vectores BERT que generemos en este script junto con los contenidos de las guías docentes y el ID de estas para poder identificarlas. Procesar el contenido de alrededor de tres mil guías docentes y generar sus vectores BERT en un proceso demasiado largo como para estar repitiéndolo cada vez que un usuario haga una consulta en nuestro sistema web.

Debido a estas circunstancias, la mejor solución a este problema es utilizar la librería **Pandas**. Esta está especializada en el manejo y análisis de estructuras de datos, se basa en los arrays de la librería NumPy. La estructura principal de la biblioteca es los **DataFrame** que son estructuras de dos dimensiones (tablas). Nos Con esta podemos leer y obtener el contenido de una base de datos SQL fácilmente. Además, contiene una funcionalidad como **.to_pickle("./nombre_archivo.pkl")** que nos permitirá almacenar un determinado DataFrame de forma permanente para más tarde poder ser leído mediante la siguiente funcionalidad de la biblioteca, **.read_pickle("./nombre_archivo.pkl")**

La generación de todos los vectores BERT de las guías docentes de todas las asignaturas de la Universidad de Jaén y su almacenamiento de forma permanente en un archivo formato PKL, se puede realizar mediante el script **creadorTablaPandas.py**, el archivo en formato PKL que generará tendrá el nombre

de **pdGuia.pkl**. En conclusión, los vectores BERT de todas las guías docentes están ya precalculados.

3.3.2 Algoritmo para comparar una consulta con el contenido de las guías docentes

La siguiente fase de desarrollo del sistema web es el Sprint 10, en esta parte, debemos crear un algoritmo con el que se generé un nuevo vector BERT sobre la consulta de un usuario, este nuevo vector BERT no será incluido en el DataFrame en el que almacenamos todos los vectores BERT.

Este problema tiene fácil solución en el punto de desarrollo que nos encontramos puesto que podemos reciclar la función con la que generamos los vectores BERT del punto anterior y esto estaría totalmente resuelto.

La novedad de este ciclo es que debemos crear una función que compare el vector BERT de la consulta del usuario con todos los vectores BERT de las guías docentes que ya hemos creado previamente. Esta función también debe devolvernos los diez de los vectores BERT que más se asemejan a la consulta y las distancias entre los vectores; Estas distancias podemos interpretarlas como el grado de similitud que hay entre el texto de la consulta y el texto del contenido de una guía docente.

Este problema lo podemos resolver usando **Torch**, que es una biblioteca de código abierto para aprendizaje automático y computación científica. Proporciona una amplia gama de algoritmos de aprendizaje profundo, como el que nos interesa en este caso “**Torch.fucntional.cosine_similarity()**”, que lo que hace es devolver la similitud entre dos entradas de vectores calculadas a lo largo de toda su dimensión.

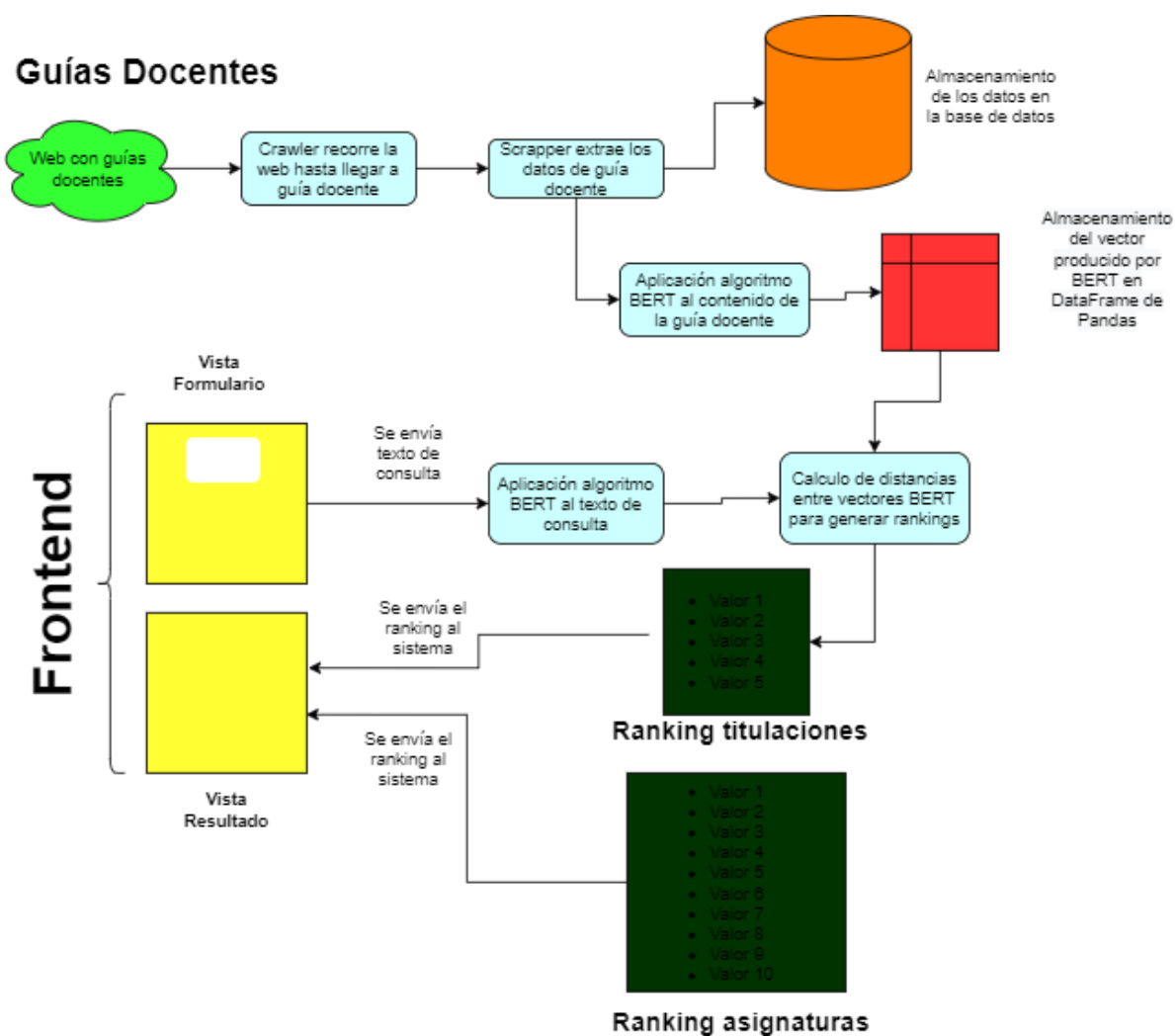


Ilustración 19 - Diagrama Funcionamiento BERT en el Sistema

El funcionamiento en general del sistema lo podemos observar en la Figura 19.

3.3.3 Insertar algoritmo de consulta en el sistema web

Una vez el algoritmo de consulta ha sido testeado para comprobar que funciona perfectamente, debemos integrarlo en el sistema web para que este ofrezca la funcionalidad que perseguíamos desde el inicio del proyecto.

Las funciones creadas anteriormente tanto para generar el vector BERT como para calcular las similitudes entre los distintos vectores las podemos copiar directamente en el fichero principal de la aplicación Flask (**App.py**), para ser utilizadas en cualquier módulo.

Haremos uso de estas funciones desde el método **resultado()**, que está asociado a la segunda vista del sistema web y es donde mostraremos la comparación del texto introducido por el usuario y el contenido de las guías docentes. También será necesario una función de apoyo extra con la que acceder a la base de datos y obtener los datos de las guías docentes semejantes al texto de consulta, además de una función que calcule cuáles son los grados más afines al usuario.

4 PRUEBAS

Una vez realizado el desarrollo del proyecto debemos pasar a realizar pruebas para determinar si el funcionamiento del sistema web es correcto y satisface los requisitos iniciales, además, de detectar posibles errores que ocurran en el sistema.

4.1 Pruebas iniciales

En las primeras pruebas de las funcionalidades del sistema web se ha detectado un error que perjudica bastante al funcionamiento de la aplicación. Este error consiste en que todas las titulaciones cuentan con dos asignaturas que son genéricas, es decir, que las podemos encontrar en todas y cada una de los grados o máster de la Universidad de Jaén, el problema en que deriva esto es que a la hora de hacer las comparaciones el grado de similitud entre las distintas titulaciones y asignaturas cuando se realiza la consulta se vea perjudicado. Las asignaturas previamente mencionadas son “**Trabajo fin de Grado**” y “**Prácticas externas**” o “**Prácticas de empresa**” o “**Prácticas en empresa**”. La solución a este problema es eliminarlas de la base de datos, de la misma forma que en la Figura 20.

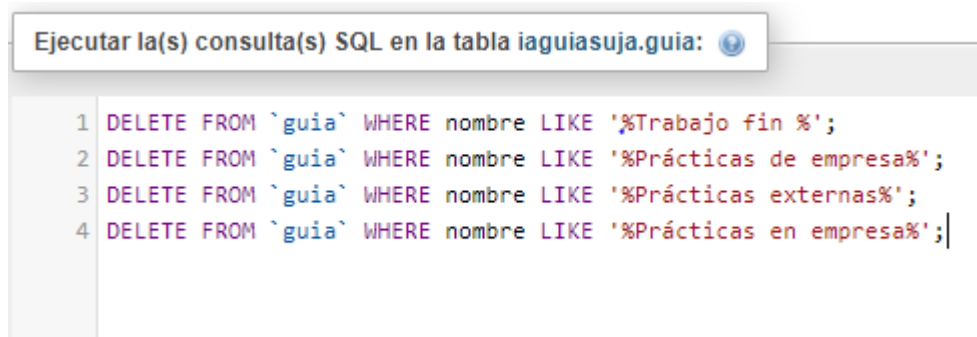



Ilustración 20 - Eliminación asignaturas repetidas

Además, se debe volver a generar el archivo PKL donde están almacenados los vectores BERT de las asignaturas.

Otro de los errores detectados es que realmente el diseño de la **vista Resultado** (Segunda vista) no era muy comprensible, en el sentido de que era fácil que el usuario no prestase atención a la información de interés, puesto que por ejemplo la indicación de la titulación que más se ajustaba al usuario parecía más un


título que realmente lo importante de la página web. Por esto, se ha reconsiderado el diseño de la vista y se ha cambiado por otro, como podemos ver en la Figura 21, dónde, ahora se muestra el top cinco de titulaciones que se ajustan más al usuario, quedando la vista como en el siguiente ejemplo:



[Formulario](#)
[About](#)
[Universidad](#)

Titulaciones

- Máster Univ. en Ingeniería geodésica y geofísica aplicada
- Máster Univ. en Ingeniería mecatrónica
- Grado en Ingeniería química industrial EPS Linares
- Grado en Psicología
- Grado en Gestión y administración pública



Asignaturas más acordes a tus intereses	Grado	Año	Similitud
Biocombustibles	Grado en Ingeniería química industrial EPS Linares	2020-21	42.21%
Monitorización y control geodésico de deformaciones	Máster Univ. en Ingeniería geodésica y geofísica aplicada	2020-21	41.79%
Gestión de proyectos y desarrollo de aplicaciones software	Máster Univ. en Ingeniería mecatrónica	2020-21	41.77%
GGOS: sistema de observación geodésico global	Máster Univ. en Ingeniería geodésica y geofísica aplicada	2020-21	41.66%
Fundamentos de intervención psicológica	Grado en Psicología	2020-21	41.54%
Estadística II	Grado en Gestión y administración pública	2020-21	41.15%
Domótica e inmótica	Grado en Ingeniería electrónica industrial	2020-21	41.01%
INSAR: interferometría radar de satélite	Máster Univ. en Ingeniería geodésica y geofísica aplicada	2020-21	40.95%
Nutrientes tecnológicos	Máster Univ. en Ingeniería de los materiales y construcción sostenible EPS Linares	2020-21	40.94%
Estadística II	Doble Grado en Administración y dirección de empresas y Finanzas y contabil	2020-21	40.82%

© 2020-2021, Javier Martínez Jiménez, Universidad de Jaén

Ilustración 21 - Rediseño de vista Resultado

4.2 Pruebas de validación del sistema

En este apartado lo que haremos es realizar una encuesta a diferentes tipos de usuarios con distintos niveles de conocimiento de informática y de distintas edades para comprobar la usabilidad del sistema web.

La respuesta de los encuestados consistirá en determinar como de bueno es el sistema web en un determinado campo, valorándolo del 1 al 5, siendo el 1 la peor puntuación y el 5 la mejor puntuación posible.

Alguna de las encuestas realizadas son las siguientes:

4.2.1 Primer encuestado

PRIMERA ENCUESTA		
NOMBRE	NIVEL	EDAD
Ignacio Moral Rodríguez	Alto	21

Tabla 5 - Primer encuestado

PREGUNTA	VALORACIÓN
¿Las imágenes que aparecen en la página web tienen relación con el tema?	3
¿Sabe dónde se encuentra en todo momento dentro del sistema web?	4
¿Es sencillo navegar a través del sistema web?	5
¿La respuesta obtenida le ha ayudado?	4
¿Los colores utilizados les resultan agradables?	4
¿Qué le parece el diseño?	3
¿Los enlaces a las titulaciones corresponden con las titulaciones mostradas?	5

¿Los enlaces a las guías docentes corresponden con las titulaciones mostradas?	5
---	----------

Tabla 6 - Primera encuesta

4.2.2 Segundo encuestado

PRIMERA ENCUESTA		
NOMBRE	NIVEL	EDAD
Annalisa Giasi	Medio	35

Tabla 7 - Segundo encuestado

PREGUNTA	VALORACIÓN
¿Las imágenes que aparecen en la página web tienen relación con el tema?	3
¿Sabe dónde se encuentra en todo momento dentro del sistema web?	5
¿Es sencillo navegar a través del sistema web?	5
¿La respuesta obtenida le ha ayudado?	3
¿Los colores utilizados les resultan agradables?	5
¿Qué le parece el diseño?	4
¿Los enlaces a las titulaciones corresponden con las titulaciones mostradas?	5
¿Los enlaces a las guías docentes corresponden con las titulaciones mostradas?	5

Tabla 8 - Segunda encuesta

4.2.3 Tercer encuestado

PRIMERA ENCUESTA		
NOMBRE	NIVEL	EDAD
Juan Martínez Rubio	Nulo	58

Tabla 9 - Tercer encuestado

PREGUNTA	VALORACIÓN
¿Las imágenes que aparecen en la página web tienen relación con el tema?	4
¿Sabe dónde se encuentra en todo momento dentro del sistema web?	4
¿Es sencillo navegar a través del sistema web?	3
¿La respuesta obtenida le ha ayudado?	5
¿Los colores utilizados les resultan agradables?	4
¿Qué le parece el diseño?	4
¿Los enlaces a las titulaciones corresponden con las titulaciones mostradas?	5
¿Los enlaces a las guías docentes corresponden con las titulaciones mostradas?	5

Tabla 10 - Tercera encuesta

4.2.4 Cuarto encuestado

PRIMERA ENCUESTA		
NOMBRE	NIVEL	EDAD
Lourdes Jiménez Moral	Bajo	16

Tabla 11 - Cuarto encuestado

PREGUNTA	VALORACIÓN
¿Las imágenes que aparecen en la página web tienen relación con el tema?	4
¿Sabe dónde se encuentra en todo momento dentro del sistema web?	5
¿Es sencillo navegar a través del sistema web?	4
¿La respuesta obtenida le ha ayudado?	5
¿Los colores utilizados les resultan agradables?	4
¿Qué le parece el diseño?	4
¿Los enlaces a las titulaciones corresponden con las titulaciones mostradas?	5
¿Los enlaces a las guías docentes corresponden con las titulaciones mostradas?	5

Tabla 12 - Cuarta encuesta

4.2.5 Resultados finales

Finalmente, tras realizar la encuesta a más de 10 usuarios, los valores medios obtenidos son los siguientes:

PREGUNTA	VALORACIÓN
¿Las imágenes que aparecen en la página web tienen relación con el tema?	3.4
¿Sabe dónde se encuentra en todo momento dentro del sistema web?	4.7
¿Es sencillo navegar a través del sistema web?	4.2
¿La respuesta obtenida le ha ayudado?	4
¿Los colores utilizados les resultan agradables?	4.2
¿Qué le parece el diseño?	3.8
¿Los enlaces a las titulaciones corresponden con las titulaciones mostradas?	5
¿Los enlaces a las guías docentes corresponden con las titulaciones mostradas?	5

Tabla 13 - Resultados finales encuestas

Tras la realización de estas encuestas podemos observar que todos los campos encuestados sobre el sistema web están por encima del 3, es decir, por encima de la media, por lo que podemos determinar que la usabilidad de la aplicación web es correcta.

5 RESULTADOS

En este apartado se van a mostrar algunos ejemplos de resultados obtenidos mediante la introducción de un determinado texto:

5.1 Primer resultado

El primer texto que probaremos será **“Desde pequeño me ha encantado la historia del mundo, me gustaría conocer más sobre esta.”**

Los resultados obtenidos para el texto introducido son los de la Figura 22.

Titulaciones
<u>Máster Univ. en Análisis histórico del mundo actual</u>
Grado en Geografía e historia
Grado en Historia del arte
Máster Univ. en Industria conectada EPS Linares
Grado en Estudios ingleses

Ilustración 22 - Primer resultado de titulaciones

Podemos observar cómo en el top cinco de titulaciones recomendadas por el sistema web aparecen titulaciones relacionadas con el texto de consulta.

Asignaturas más acordes a tus intereses	Grado	Año	Similitud
Los debates sobre la construcción del Estado contemporáneo: nacionalismo, culturas, identidades	Máster Univ. en Análisis histórico del mundo actual	2020-21	62.22%
Historia moderna	Grado en Arqueología	2020-21	60.37%
Historia contemporánea de los países mediterráneos	Grado en Geografía e historia	2020-21	60.16%
Antropología social del Mediterráneo	Grado en Geografía e historia	2020-21	59.82%
Culturas políticas y opinión pública	Máster Univ. en Análisis histórico del mundo actual	2020-21	59.56%
Entornos virtuales y simulación	Máster en Ingeniería informática	2020-21	59.42%
Entornos virtuales y simulación	Doble Máster en Ingeniería informática y Seguridad informática	2020-21	59.42%
Antropología del arte	Grado en Historia del arte	2020-21	59.32%
Introducción a la fábrica digital	Máster Univ. en Industria conectada EPS Linares	2020-21	59.04%
Competencia comunicativa en el ámbito académico y profesional	Grado en Educación social	2020-21	58.84%

Ilustración 23 - Primer resultado de asignaturas

En el caso de la Figura 23 podemos observar que las asignaturas recomendadas en el top diez también están relacionadas con las titulaciones recomendadas.

5.2 Segundo resultado

El segundo texto que probaremos será **“No me gustan nada las matemáticas, prefiero las letras.”**

Los resultados obtenidos para el texto introducido son los de la Figura 24.

Titulaciones
Grado en Filología hispánica
Grado en Estudios ingleses
Máster Univ. en Análisis histórico del mundo actual
Grado en Historia del arte
Grado en Geografía e historia

Ilustración 24 - Segundo resultado de titulaciones

Otra vez podemos observar cómo en el top cinco aparecen titulaciones relacionadas con nuestra consulta, además no aparece nada relacionado con las matemáticas, lo cual es un claro ejemplo de la capacidad de los modelos BERT para capturar la semántica completa de un texto.

Asignaturas más acordes a tus intereses	Grado	Año	Similitud
Literatura española de los Siglos de Oro I	Grado en Filología hispánica	2020-21	57.54%
Literatura española e intermedialidad (prensa, cine, internet)	Grado en Filología hispánica	2020-21	56.36%
Iconografía profana	Grado en Historia del arte	2020-21	55.66%
Culturas políticas y opinión pública	Máster Univ. en Análisis histórico del mundo actual	2020-21	55.64%
Orígenes de la narrativa de ficción en la literatura española	Grado en Filología hispánica	2020-21	55.54%
Los debates sobre la construcción del Estado contemporáneo: nacionalismo, culturas, identidades	Máster Univ. en Análisis histórico del mundo actual	2020-21	55.15%
Literatura comparada inglés-español	Grado en Filología hispánica	2020-21	54.79%
Literatura comparada inglés-español	Grado en Estudios ingleses	2020-21	54.79%
Literatura española comparada con la europea	Grado en Estudios ingleses	2020-21	54.22%
Literatura española comparada con la europea	Grado en Filología hispánica	2020-21	54.22%

Ilustración 25 - Segundo resultado de asignaturas

En el caso de la Figura 25 podemos observar que las asignaturas recomendadas en el top diez también están relacionadas con las titulaciones recomendadas.

5.3 Tercer resultado

El tercer texto que probaremos será **“Me encanta ayudar a la gente.”**

Los resultados obtenidos para el texto introducido son los de la Figura 26.

Titulaciones
Grado en Educación social
Grado en Trabajo social
Máster Univ. en Psicología general sanitaria
Máster Univ. en Gerontología: longevidad, salud y calidad
Máster Univ. en Economía y desarrollo territorial

Ilustración 26 - Tercer resultado de titulaciones

Otra vez podemos observar cómo en el top cinco aparecen titulaciones relacionadas con nuestra consulta, ya que nos recomienda titulaciones en la que hay que implicarse con la gente para ayudarles a tener una mejor vida.

Asignaturas más acordes a tus intereses	Grado	Año	Similitud
Educación del ocio y creatividad social por medio de las artes	Grado en Educación social	2020-21	50.58%
Valoración e intervención terapéutica	Máster Univ. en Gerontología: longevidad, salud y calidad	2020-21	50.13%
Innovación social y emprendimiento	Máster Univ. en Economía y desarrollo territorial	2020-21	49.68%
Evaluación e intervención psicológica en niños y adolescentes	Máster Univ. en Psicología general sanitaria	2020-21	49.6%
Prácticas en instituciones de bienestar social I	Grado en Trabajo social	2020-21	49.5%
Trabajo social, animación sociocultural y participación	Grado en Trabajo social	2020-21	49.45%
Evaluación e intervención psicológica en adultos I	Máster Univ. en Psicología general sanitaria	2020-21	49.37%
Sociología de la desviación y la exclusión social	Grado en Educación social	2020-21	49.26%
Heurística y metodología del diseño	Máster Univ. en Investigación y educación estética: artes, música y diseño	2020-21	49.14%
Desarrollo ágil	Grado en Ingeniería informática	2020-21	49.04%

Ilustración 27 - Tercer resultado de asignaturas

En la figura 27 también obtenemos un top diez de asignaturas, donde estas están relacionadas con el texto de consulta.

6 CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

Este proyecto nació buscando montar un motor de consulta de titulaciones de la Universidad de Jaén que devolviera las asignaturas que respondiesen a los intereses de un estudiante a partir de una consulta.

La primera meta era **obtener todas las guías docentes** de todos los grados de la Universidad de Jaén y extraer el contenido de estas, para generar un vector BERT de cada uno de los temarios. Para este objetivo se tiene una conclusión satisfactoria, puesto que se ha podido conseguir todas las asignaturas impartidas en la Universidad de Jaén, extraer su contenido y almacenarlo en una base de datos.

El segundo objetivo era desarrollar un **sistema web** donde un usuario pudiese realizar una consulta en la que indicase sus intereses y se mostrará como resultado un top de grados y asignaturas acordes a lo introducido en la consulta, además que desde los tops se puede redirigir a la guía docente completa de la asignatura o a la página oficial del grado.

El resultado final es que se ha conseguido realizar todas las metas impuestas al principio del proyecto, aunque hay que mencionar que durante la realización de estas han surgido distintos problemas.

El problema más complicado de resolver que ha aparecido durante el desarrollo del proyecto es que la página donde se encuentran todas las guías docentes (<https://uvirtual.ujaen.es/pub/es/informacionacademica/catalogoguiasdocentes/>) tiene un mecanismo de defensa anti descargas de forma automatizada de un bot, cuando se realizaba la descarga del PDF de una de las guías docentes, el archivo que se obtenía estaba en blanco sin contenido en el interior. Debido a este problema se tuvo que buscar otra solución, esta resultó en que una vez se llegaba a la página HTML de las guías docentes, en vez de realizar un proceso de descarga del archivo PDF de la guía docente, lo que se haría es obtener directamente el contenido de las guías

docentes recorriendo el cuerpo de la página HTML. Finalmente, esto resultó en un total acierto porque nos permitía obtener más datos sobre la asignatura de la guía docente; como cuál es su grado, su ID, el año en la que se imparte, etc. También de esta forma podíamos guardar directamente todos los datos extraídos en nuestra base de datos.

Para acabar con el apartado de conclusiones, me gustaría aportar una reflexión personal. Realizar el trabajo de final de grado ha sido un punto de inflexión en la carrera para mí, es la primera vez que he sentido un verdadero reto en este ciclo formativo de mi educación, puesto que no ha sido como cuando tenía que realizar una práctica de cualquiera de las asignaturas de la carrera, en las que había unos apuntes o un guion específico sobre cómo realizar la práctica en cuestión. A lo largo de este proceso he sentido cientos de emociones, desde que no era capaz de realizarlo, hasta un minuto más tarde sentirme el mejor ingeniero informático del mundo.

Creo que este trabajo me ha ayudado a desarrollar una serie importante de capacidades, como la de ser capaz de solucionar problemas por mí mismo y buscarme la vida de forma independiente, esta es la parte que más me ha gustado y parecido interesante junto con aprender ciertas nociones de web scraping y ser capaz de utilizarlas con resultado exitoso.

Con respecto a mi futuro laboral creo que este trabajo me ha dado un impulso gigante en cuanto a preparación, por lo que solo tengo agradecimientos en este sentido hacía mis tutores.

6.2 Trabajos futuros

Una vez se han cumplimentado todos los objetivos impuestos en el proyecto y se ha alcanzado la limitación del tiempo, podemos empezar a plantear posibles funcionales que podrían ser interesantes para implementarlas en el futuro al sistema web.

- Una de las principales funcionalidades que se le podrían implementar a este proyecto es el de que el usuario pueda elegir si solo quiere que se le recomienden titulaciones de grados, másteres o ambos.
- Otra opción muy interesante es que se pueda diferenciar a la hora de realizar la consulta que aparezcan titulaciones del campus que se encuentra en la ciudad de Jaén o del campus de la ciudad de Linares.
- Una posible característica más que se puede introducir al sistema web es que se pueda realizar la consulta por años. Lo mejor de esta posible funcionalidad es que podría ser que los usuarios tuvieran curiosidad por consultar que titulaciones se impartían en el pasado. Si se llegará a implementar esta funcionalidad se debería programar para que sea totalmente opcional, por defecto se deberían mostrar las titulaciones del año presente en el que se está realizando la consulta.
- La automatización de la regeneración de los datos cada año, con las nuevas guías docentes, podría ser otra funcionalidad interesante.

7 BIBLIOGRAFÍA

- [1] *Python*. (21 de Junio de 2021). Obtenido de <https://es.wikipedia.org/wiki/Python>
- [2] *PyCharm*. (2021). (Jet Brains) Obtenido de <https://www.jetbrains.com/es-es/pycharm/>
- [3] *Web scraping más BeautifulSoup*. (2020). (J2Logo) Obtenido de <https://j2logo.com/python/web-scraping-con-python-guia-inicio-beautifulsoup/>
- [4] *XAMPP*. (2021). (Apache Friends) Obtenido de <https://www.apachefriends.org/es/index.html>
- [5] *MySQL*. (OpenWebinars) - Robledano, A. (Ed.). (24 de Septiembre de 2019). Obtenido de <https://openwebinars.net/blog/que-es-mysql/>
- [6] *Pandas*. (14 de Mayo de 2021). Obtenido de <https://aprendeconalf.es/docencia/python/manual/pandas/>
- [7] *Flask*. (OpenWebinars) - Domingo Muñoz, J. (17 de Noviembre de 2017). Obtenido de <https://openwebinars.net/blog/que-es-flask/>
- [8] *Bootstrap*. (28 de Agosto de 2021). (Hostingers) Obtenido de <https://www.hostinger.es/tutoriales/que-es-bootstrap>
- [9] *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* - Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (24 de Mayo de 2019). Obtenido de <https://arxiv.org/pdf/1810.04805.pdf>
- [10] *Scrum*. (s.f.). Obtenido de <https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/>
- [11] *Web crawler*. (21 de Octubre de 2020). Obtenido de <https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-un-web-crawler/>
- [12] *Mockup*. (2019). Obtenido de <https://definicion.de/mockup/>
- [13] *Storyboard*. (29 de Abril de 2019). Obtenido de <https://www.esdesignbarcelona.com/es/expertos-diseno/que-es-y-como-crear-un-storyboard>
- [14] *¡No me hagas pensar!* - Krug, S. (2000). Reino Unido.
- [15] *PLN*. (12 de Septiembre de 2019). Obtenido de <https://decidesoluciones.es/procesamiento-del-lenguaje-natural-pln-o-nlp-que-es-y-para-que-se-utiliza/>

8 APÉNDICES

8.1 Código fuente

El código fuente del proyecto puede descargarse desde el siguiente repositorio GitHub:

<https://github.com/jmj00019/IAasesoramientoEstudios>

8.2 Requerimientos del sistema

Los requerimientos básicos que se necesitan para poder probar el proyecto son:

- ☐ Ordenador o entorno virtual con Python 3.6.6 o superior instalado.
- ☐ Tener instalado XAMPP, para poder acceder a la base de datos.
- ☐ Crear una base de datos SQL a partir del archivo **esquema.sql** de la carpeta **db** del repositorio del proyecto.

8.3 Creación de la base de datos

Para la creación de la base de datos debemos asegurarnos que tenemos instalado XAMPP y los módulos de Apache y MySQL están en funcionamiento.

Una vez cumplido este requerimiento debemos acceder en nuestro navegador web de preferencia a la siguiente URL, <http://localhost/phpmyadmin/>, ahora tenemos dos opciones:

- 1 Importar la base de datos completa creada durante este proyecto, con el archivo **iaguiasuja.sql** del repositorio de la carpeta **db** y partiríamos desde el punto final de este proyecto.

- 2 Importar el esquema de la base de datos con el archivo **esquema.sql** e importar la tabla “**grado**” con el archivo **grado.sql**. Ambos archivos se encuentran en la carpeta **db** del repositorio.

Para importar los archivos se deben seguir los siguientes pasos:

1. En la barra de navegación de <http://localhost/phpmyadmin> pinchamos sobre **Importar**, Figura 28.



Ilustración 28 - Primer paso importación base de datos

2. Ahora pinchamos sobre Seleccionar archivo y seleccionamos el archivo o archivos necesarios, Figura 29.

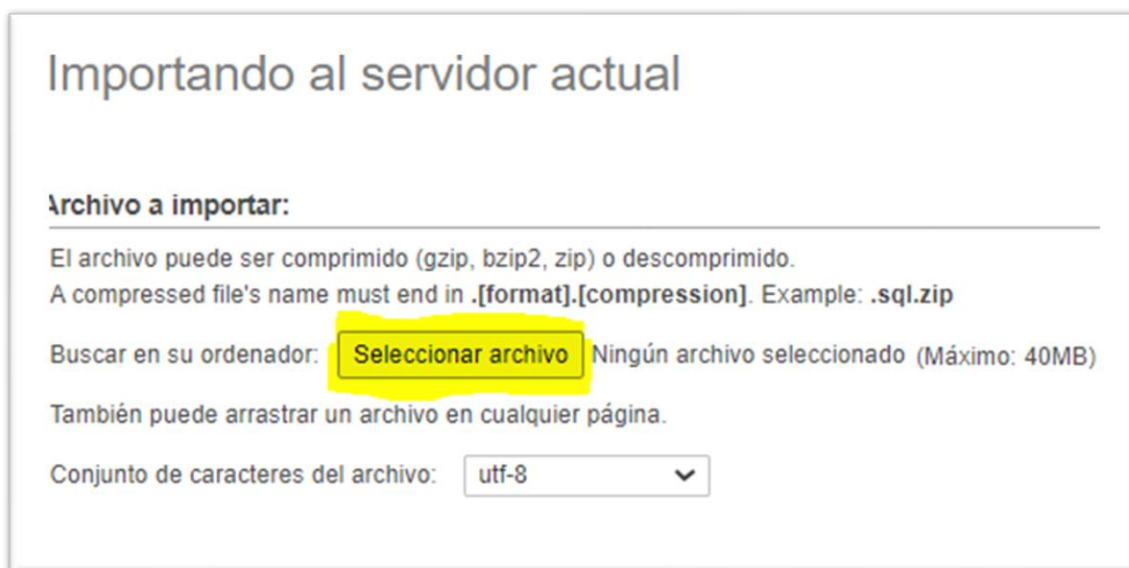


Ilustración 29 - Segundo paso importación base de datos

8.4 Instalación de bibliotecas de Python

Para poder ejecutar los distintos scripts del proyecto como para poder correr el servidor del sistema web debemos tener ciertas bibliotecas instaladas en nuestro ordenador o entorno virtual.

Las bibliotecas que necesitamos que estén instaladas en nuestro entorno son las siguientes:

Biblioteca	Comando de instalación
BeautifulSoup	<code>pip install beautifulsoup4</code>
Requests	<code>pip install requests</code>
LXML	<code>pip install lxml</code>
MySQL driver in Python	<code>pip install mysql-connector-python</code>
Os-SYS	<code>pip install os-sys</code>
Flask	<code>pip install flask</code>
Flask-MySQL	<code>pip install flask-mysqldb</code>
Pandas	<code>pip install pandas</code>
Torch	<code>pip install Torch</code>
Transformers (Bert)	<code>pip install transformers</code>

Tabla 14 - Bibliotecas necesarias

8.5 Instalación y configuración del sistema web

- 1 Debemos comprobar que XAMPP está instalado.
- 2 Encender los módulos Apache y MySQL de XAMPP.
- 3 Importar la base de datos mediante el archivo **iaguiasua.sql** de la carpeta **db** del repositorio.
- 4 Instalar todas las bibliotecas necesarias de Python en el ordenador o entorno virtual.
- 5 Ejecutar el script App.py, para iniciar el servidor.

- 6 Abrir navegador web de preferencia y buscar <http://127.0.0.1:3000/>.
- 7 Probar que todo funciona perfectamente en el sistema web.

8.6 Manual de Usuario

8.6.1 Manual Utilización Script Web Scraping

En el caso de qué se quieran obtener las guías docentes de un curso diferente a los del curso 20/21 debemos utilizar el script `extractorContenidosYDB.py`

Antes de empezar a utilizar el script debemos asegurarnos de que tenemos instalado XAMPP y los módulos Apache y MySQL estén funcionando.

También debemos tener creada la base de datos **iaguiasuja** de cualquiera de las maneras indicadas en el **punto 8.3** e instaladas las bibliotecas mencionadas en el **punto 8.4**.

Al script `extractorContenidosYDB.py` se le deben pasar tres atributos en la línea de comando en este orden:

- 1 El primer atributo qué le debemos pasar es la URL de la titulación de la que queramos obtener las guías docentes.
- 2 El segundo atributo es nombre de la titulación.
- 3 El tercer atributo es el curso, con el siguiente formato, **“2021-21”**

Podemos observar un ejemplo en la Figura 30.

```
python extractorContenidosYDB.py  
"https://uvirtual.ujaen.es/pub/es/informacionacademica/catalogoguiasdocentes/p/2020-  
21/205/780A" "Doble Máster en Análisis, conservación y restauración de habitats y MPFES"  
"2020-21"
```

Ilustración 30 - Ejemplo script extractorContenidosYDB.py

8.6.2 Manual Usuario Sistema Web

- 1 Una vez nos encontramos en la página principal del sistema web, Figura 31. Podemos empezar a hacer consultas sobre nuestros intereses.



Ilustración 31 - Página Principal Web Manual Usuario

- 2 Una vez hemos escrito nuestro texto de consulta, hacemos click en **ENVIAR** y seremos redirigidos a la vista final del sistema web.
- 3 Tras haber pulsado **ENVIAR**, nos encontraremos en la página Resultado del sistema web en la que se nos mostrará un top cinco de titulaciones que son más acordes a nuestro texto de consulta. También aparecerán las diez asignaturas más acordes sobre nuestro texto de consulta, además, sobre estas tendremos información extra como a qué titulación pertenecen, en qué año se imparten o el porcentaje de similitud que tiene con nuestro texto. Si hacemos click sobre cualquiera de las titulaciones mostradas se nos abrirá en una ventana la página oficial de la titulación y si hacemos click sobre cualquiera de las asignaturas mostradas se nos abrirá en una ventana la guía docente oficial de la asignatura.
- 4 Si deseamos realizar otra consulta basta que pulsemos sobre el logo de la aplicación web o sobre Formulario de la barra de navegación.