



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática  
título del TFG



Presentado por Nombre del alumno  
en Universidad de Burgos — 2 de marzo de 2017  
Tutor: nombre tutor





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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

En Burgos, 2 de marzo de 2017

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor



## **Resumen**

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## **Descriptores**

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android . . .

### **Abstract**

A **brief** presentation of the topic addressed in the project.

### **Keywords**

keywords separated by commas.

---

# Índice general

---

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	2
Conceptos teóricos	3
3.1. Secciones . . . . .	3
3.2. Referencias . . . . .	3
3.3. Imágenes . . . . .	3
3.4. Listas de ítems . . . . .	4
3.5. Tablas . . . . .	5
Técnicas y herramientas	6
4.1. Herramientas Candidatas (Aplicación Web) . . . . .	6
4.2. Herramientas candidatas (Bases de Datos de texto) . . . . .	8
Aspectos relevantes del desarrollo del proyecto	9
Trabajos relacionados	10
Conclusiones y Líneas de trabajo futuras	11

---

# Índice de figuras

---

3.1. Autómata para una expresión vacía . . . . .	4
--	---



---

# Índice de tablas

---

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto .	5
---	---

---

# Introducción

---

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

---

# Objetivos del proyecto

---

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

---

# Conceptos teóricos

---

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L<sup>A</sup>T<sub>E</sub>X<sup>1</sup>.

## 3.1. Secciones

Las secciones se incluyen con el comando `section`.

### Subsecciones

Además de secciones tenemos subsecciones.

### Subsubsecciones

Y subsecciones.

## 3.2. Referencias

Las referencias se incluyen en el texto usando `cite [?]`. Para citar webs, artículos o libros `[?]`.

## 3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de L<sup>A</sup>T<sub>E</sub>X, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

---

<sup>1</sup>Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz



Figura 3.1: Autómata para una expresión vacía

### 3.4. Listas de ítems

Existen tres posibilidades:

- primer ítem.
- segundo ítem.

1. primer ítem.
2. segundo ítem.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

**Primer ítem** más información sobre el primer ítem.

**Segundo ítem** más información sobre el segundo ítem.

■

### 3.5. Tablas

Igualmente se pueden usar los comandos específicos de  $\text{\LaTeX}$ o bien usar alguno de los comandos de la plantilla.

---

# Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

## 4.1. Herramientas Candidatas (Aplicación Web)

Este apartado detallará las herramientas que inicialmente fueron consideradas para la programación de la aplicación web de forma compatible con el uso de Python.

Los principales criterios a tener en cuenta para quedarnos con una de las alternativas han sido: la flexibilidad que ofreciera esa herramienta para trabajar con archivos de tipo PDF, la facilidad de distribución de la propia aplicación y el hecho de que adaptarse a ella no suponga un esfuerzo excesivo al programador.

### PyQt

Permite crear aplicaciones de escritorio de forma sencilla, pudiendo crear elementos de interfaz sencillos como botones, cuadros de texto... etc(más aún si se usa su editor de GUI propio).

No obstante, las opciones gráficas no eran tan atractivas como podía parecer en un primer momento, y además no es adecuada si la intención es crear

una aplicación web.

## Kivy

Es una alternativa interesante ya que permite hacer aplicaciones de escritorio con código Python con la ventaja de que es “Cross-platform”, es decir, que las aplicaciones que se creen usando esta herramienta se pueden distribuir en distintos sistemas operativos. No obstante, no se tuvo especialmente en cuenta ya que para cuando se estaba investigando esta opción, la opción de hacer una aplicación web en vez de una de escritorio estaba prácticamente decidida.

## Flask

Permite combinar código HTML y Python de forma simple. La característica que más nos interesaba es su facilidad de combinar ficheros HTML y códigos escritos puramente en Python, además de poder incluir de forma sencilla librerías con estilos para la interfaz. No obstante, a la hora de practicar con la herramienta (siguiendo el mega-tutorial de [miguelgrinberg.com](http://miguelgrinberg.com)), la estructura de ficheros daba numerosos problemas al intentar ejecutar aplicaciones, unas veces relacionadas con el intérprete de Python, y otras por errores al combinar los ficheros. Aunque finalmente se completó la guía, las sensaciones finales eran que con Jupyter se desarrollaban aplicaciones sencillas con más rapidez.

## Jupyter

En esta herramienta, y con ayuda de alguna librería de terceros (`wand`, `display`...) es especialmente sencillo combinar código HTML y Python en un mismo fichero. De hecho, haciendo uso de los llamados “ipywidgets” (o dicho de otra manera, los Widgets que ofrece IPython), se puede encapsular el código HTML de forma muy cómoda en un script normal de Python, especialmente el `ipywidget HTML`, que permite introducir una estructura HTML como si se tratara de una cadena de texto normal.

Además, la curva de aprendizaje ha sido bastante sencilla y conseguir un código inicial que muestre documentos PDF ha llevado poco tiempo. Las posibilidades de combinarlo con otros elementos, como Flask o MongoDB (PyMongo) hacen de esta herramienta una de las mejores alternativas. De hecho, fue la elegida para implementar la aplicación web.

## 4.2. Herramientas candidatas (Bases de Datos de texto)

De forma similar al apartado previo, en las fases iniciales del proyecto se



planteaban varias opciones para elegir cómo almacenar los datos necesarios para que el usuario pudiera obtener noticias y a la vez poder acceder a los datos para realizar la minería de los mismos.

## **PyLucene**

## **PyMongo**

Es la extensión de MongoDB para Python. Su principal característica es que es una base de datos no relacional, o en otras palabras, no SQL. De modo que funciona con un modelo distinto al que es más habitual ver en una base de datos convencional. Por ejemplo, en vez de estructurarse como tablas con relaciones entre ellas, la forma de almacenar los datos se asemeja mucho más a lo que se puede ver en JSON: los datos se almacenan en una estructura clave-valor similar a un diccionario. No obstante, aunque no se puedan usar relaciones, estas estructuras son muy flexibles, y se puede, por ejemplo, introducir unas estructuras como campos de otras, o introducir cualquier tipo de dato como valor.

## **SQLAlchemy**

---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

---

## Trabajos relacionados

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.