



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**título del TFG**



Presentado por Álvaro Villar Val  
en Universidad de Burgos — 8 de abril de 2024  
Tutor: Carlos Cambra Baseca







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Carlos Cambra, profesor del departamento de Digitalización, área de Ciencia de la Computación e Inteligencia Artificial.

Expone:

Que el alumno D. Álvaro Villar Val, con DNI 71303408G, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado desarrollo de aplicación de base de datos para central meteorologica.

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, 8 de abril de 2024

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Carlos Cambra Baseca

D. Ángel Arroyo Puente





## **Resumen**

Programar una aplicación que convierta los datos obtenidos de una central meteorológica a una base de datos. Luego la misma aplicación procesara los datos y los almacenara en una nueva base de datos. Para esta segunda base de datos se programará también una interfaz de usuario que permitirá que los usuarios soliciten los datos disponibles en un periodo de tiempo determinado, y los ofrezca en diferentes formatos.

## **Descriptores**

Bases de datos, UI, Analisis de datos, Datos meteorologicos, Programación de aplicacion, Python, Postgres, Graficos, Procesado de datos ...

### **Abstract**

Program an application that converts the data obtained from a weather station to a database. Then the same application will process the data and store it in a new database. For this second database we will also program a user interface that will allow users to request the data available in a given period of time, and offer them in different formats.

### **Keywords**

Databases, UI, Data Analysis, Meteorological Data, Application Programming, Python, Postgres, Graphics, Data Processing, Data Processing



---

# Índice general

---

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
2. Objetivos del proyecto	3
3. Conceptos teóricos	5
3.1. Secciones . . . . .	5
3.2. Referencias . . . . .	5
3.3. Imágenes . . . . .	6
3.4. Listas de items . . . . .	6
3.5. Tablas . . . . .	7
4. Técnicas y herramientas	9
4.1. Python . . . . .	10
4.2. PostgreSQL . . . . .	10
4.3. Pandas . . . . .	10
4.4. NumPy . . . . .	11
4.5. Matplotlib . . . . .	11
4.6. tkinter . . . . .	11
4.7. pysolar . . . . .	11
4.8. sqlalchemy . . . . .	12
4.9. psycopg2 . . . . .	12

4.10. Git . . . . .	12
4.11. GitHub . . . . .	12
4.12. Scrum . . . . .	13
4.13. Conclusiones . . . . .	13
<b>5. Aspectos relevantes del desarrollo del proyecto</b>	<b>15</b>
<b>6. Trabajos relacionados</b>	<b>17</b>
<b>7. Conclusiones y Líneas de trabajo futuras</b>	<b>19</b>
<b>Bibliografía</b>	<b>21</b>

---

# Índice de figuras

---

3.1. Autómata para una expresión vacía . . . . .	6
--	---

---

# Índice de tablas

---

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	7
---	---

---

# 1. Introducción

---

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



---

## 2. Objetivos del proyecto

---

En este proyecto se busca desarrollar una aplicación de escritorio, que permita al equipo Swift de la Universidad de Burgos, gestionar, visualizar y descargar los datos que se generan en la central meteorológica de la Universidad de Burgos. Para ello, se plantean los siguientes objetivos:

1. Implementación de una base de datos que permita almacenar los datos meteorológicos.
2. Creación de una aplicación de escritorio capaz de actualizar los datos en la base de datos, visualizarlos y descargarlos.
3. Creación de una interfaz gráfica que permita al usuario interactuar con la aplicación de forma sencilla.

Con todo esto se busca que el equipo Swift pueda gestionar los datos de la central meteorológica de forma más eficiente, sencilla y cómoda. Además, se busca aprender como se trabaja en un proyecto de software real, aplicando los conocimientos adquiridos en la carrera y adquiriendo nuevos conocimientos que nos permitan afrontar problemas reales en el futuro.





---

## 3. 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` [3]. Para citar webs, artículos o libros [2], si se desean citar más de uno en el mismo lugar [1, 2].

---

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

### 3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de  $\text{\LaTeX}$ , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

### 3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

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

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

■

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

### 3.5. Tablas

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



---

## 4. Técnicas y herramientas

---

En este proyecto he usado:

- Python: Lenguaje de programación en el que se ha desarrollado la aplicación.
- PostgreSQL: Sistema de gestión de bases de datos relacional en el que se almacenan los datos meteorológicos.
- Pandas: Librería de Python que proporciona estructuras de datos y herramientas de análisis de datos.
- NumPy: Librería de Python que proporciona soporte para arrays y matrices multidimensionales.
- Matplotlib: Librería de Python que permite generar gráficos a partir de datos.
- tkinter: Librería de Python que permite crear interfaces gráficas de usuario.
- pysolar: Librería de Python que permite calcular la posición del sol en un determinado lugar y momento.
- sqlchemy: Librería de Python que permite interactuar con bases de datos SQL.
- psycopg2: Librería de Python que permite interactuar con bases de datos PostgreSQL.

Ademas de todo esto he usado Git como sistema de control de versiones y GitHub como plataforma de alojamiento de repositorios remotos.

Tambien he seguido un enfoque de desarrollo ágil, utilizando la metodología Scrum, que me ha permitido adaptar el proyecto a los cambios que han ido surgiendo a lo largo del desarrollo.

## 4.1. Python

Python es el lenguaje de programación que he decidido utilizar para desarrollar la aplicación, debido a que es un lenguaje sencillo y fácil de aprender, que cuenta con una gran cantidad de librerías que facilitan el desarrollo de aplicaciones de todo tipo. Además, Python es un lenguaje interpretado, lo que permite ejecutar el código sin necesidad de compilarlo previamente, lo que agiliza el proceso de desarrollo.

Además de esto Python es un lenguaje multiplataforma, lo que significa que la aplicación desarrollada en Python se puede ejecutar en cualquier sistema operativo que cuente con un intérprete de Python.

También lo he usado por que es un lenguaje muy utilizado en el ámbito de la ciencia de datos, lo que me ha permitido utilizar librerías como Pandas, NumPy y Matplotlib, que me han facilitado el tratamiento y visualización de los datos meteorológicos.

## 4.2. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional, que he utilizado para almacenar los datos meteorológicos. He elegido PostgreSQL por que es un sistema de gestión de bases de datos muy potente y fiable, que cuenta con una gran cantidad de funcionalidades que facilitan la gestión de los datos. Además, PostgreSQL es un sistema de gestión de bases de datos de código abierto, lo que significa que es gratuito y se puede modificar y distribuir libremente.

## 4.3. Pandas

Pandas es una librería de Python que proporciona estructuras de datos y herramientas de análisis de datos. He utilizado Pandas para leer los datos meteorológicos de la base de datos PostgreSQL y para realizar operaciones de limpieza y transformación de los datos. Pandas me ha permitido trabajar con los datos de forma más eficiente y sencilla, ya que proporciona una serie de funciones y métodos que facilitan el tratamiento de los datos.

## 4.4. NumPy

NumPy es una librería de Python que proporciona soporte para arrays y matrices multidimensionales. He utilizado NumPy para realizar operaciones matemáticas y estadísticas con los datos meteorológicos, ya que NumPy proporciona una serie de funciones y métodos que facilitan la manipulación de arrays y matrices. NumPy me ha permitido realizar cálculos más complejos con los datos de forma más eficiente y sencilla.

## 4.5. Matplotlib

Matplotlib es una librería de Python que permite generar gráficos a partir de datos. He utilizado Matplotlib para visualizar los datos meteorológicos en forma de gráficos, ya que Matplotlib proporciona una serie de funciones y métodos que facilitan la creación de gráficos. Matplotlib me ha permitido representar los datos de forma más clara y comprensible, lo que ha facilitado la interpretación de los datos.

## 4.6. tkinter

tkinter es una librería de Python que permite crear interfaces gráficas de usuario. He utilizado tkinter para desarrollar la interfaz gráfica de la aplicación, ya que tkinter proporciona una serie de widgets y métodos que facilitan la creación de interfaces gráficas. tkinter me ha permitido diseñar una interfaz gráfica sencilla e intuitiva, que ha facilitado la interacción del usuario con la aplicación.

## 4.7. pysolar

pysolar es una librería de Python que permite calcular la posición del sol en un determinado lugar y momento. He utilizado pysolar para calcular la posición del sol en la central meteorológica de la Universidad de Burgos, ya que pysolar proporciona una serie de funciones y métodos que facilitan el cálculo de la posición del sol. pysolar me ha permitido determinar la altura y el azimut del sol en un determinado momento, lo cual ha sido necesario

para poder aplicar los distintos filtros de calidad a los datos meteorológicos.

## 4.8. sqlalchemy

sqlalchemy es una librería de Python que permite interactuar con bases de datos SQL. He utilizado sqlalchemy debido a su compatibilidad con pandas, ya que a través de esta librería he sido capaz de insertar los csv que la central genera y los datos procesados que genera la aplicación de manera sencilla y eficiente.

## 4.9. psycopg2

psycopg2 es una librería de Python que permite interactuar con bases de datos PostgreSQL. He utilizado psycopg2 para conectarme a la base de datos PostgreSQL y realizar operaciones de lectura y escritura de los datos meteorológicos. psycopg2 me ha permitido interactuar con la base de datos de forma sencilla y eficiente, lo que ha facilitado la gestión de los datos.

## 4.10. Git

Git es un sistema de control de versiones distribuido, que he utilizado para llevar un control de los cambios realizados en el código fuente de la aplicación. He utilizado Git para crear un repositorio local en el que he ido guardando las distintas versiones del código fuente, y para sincronizar el repositorio local con un repositorio remoto alojado en GitHub. Git me ha permitido llevar un control de los cambios realizados en el código fuente, y revertir los cambios en caso de ser necesario.

## 4.11. GitHub

GitHub es una plataforma de alojamiento de repositorios remotos, que he utilizado para alojar el repositorio remoto de la aplicación. He utilizado GitHub para sincronizar el repositorio local con el repositorio remoto, lo cual me ha permitido trabajar tanto desde mi portátil como desde mo



ordendaor de sobremesa. A su vez GitHub me ha permitido tener una copia de seguridad del código fuente en la nube.

## 4.12. Scrum

Scrum es una metodología ágil de desarrollo de software, que he utilizado para gestionar el proyecto. He seguido un enfoque de desarrollo ágil, utilizando la metodología Scrum, que me ha permitido adaptar el proyecto a los cambios que han ido surgiendo a lo largo del desarrollo. Scrum me ha permitido dividir el proyecto en iteraciones cortas, llamadas sprints, y priorizar las tareas en función de su importancia y complejidad. Scrum me ha permitido llevar un control de los avances del proyecto, y tomar decisiones en función de los resultados obtenidos en cada sprint.

## 4.13. Conclusiones

En resumen, he utilizado Python, PostgreSQL, Pandas, NumPy, Matplotlib, tkinter, pysolar, sqlalchemy, psycopg2, Git, GitHub y Scrum para desarrollar la aplicación de gestión de datos meteorológicos. Estas herramientas y técnicas me han permitido desarrollar la aplicación de forma eficiente y sencilla, y adaptar el proyecto a los cambios que han ido surgiendo a lo largo del desarrollo. Gracias a estas herramientas y técnicas, he sido capaz de gestionar los datos meteorológicos de la central meteorológica de la Universidad de Burgos de forma más eficiente, sencilla y cómoda, y aprender cómo se trabaja en un proyecto de software real, aplicando los conocimientos adquiridos en la carrera y adquiriendo nuevos conocimientos que me permitirán afrontar problemas reales en el futuro.



---

## 5. 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.



---

## 6. 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.



---

## **7. 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.





---

## Bibliografía

---

- [1] Zachary J Bortolot and Randolph H Wynne. Estimating forest biomass using small footprint lidar data: An individual tree-based approach that incorporates training data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(6):342–360, 2005.
- [2] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [3] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].