



Universidad  
Rey Juan Carlos

INGENIERÍA EN TECNOLOGÍAS DE LA  
TELECOMUNICACIÓN

Curso Académico 2024/2025

Trabajo Fin de Grado

ANÁLISIS DE PATRONES DE ACTIVIDAD DE  
USUARIOS EN UNA PLATAFORMA WEB  
PARA DESARROLLO DE PROYECTOS DE  
ROBÓTICA

Autor/a : Alejandro Aguilera López

Tutor/a : José Felipe Ortega Soto



# Trabajo Fin de Grado

Título del Trabajo con Letras Capitales para Sustantivos y Adjetivos

**Autor/a :** Alejandro Aguilera López

**Tutor/a :** José Felipe Ortega Soto

La defensa del presente Proyecto Fin de Grado/Máster se realizó el día 3 de  
de 20XX, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Móstoles/Fuenlabrada, a de de 20XX



*Aquí normalmente  
se inserta una dedicatoria corta*



# Agradecimientos

Aquí vienen los agradecimientos...

Hay más espacio para explayarse y explicar a quién agradeces su apoyo o ayuda para haber acabado el proyecto: familia, pareja, amigos, compañeros de clase...

También hay quien, en algunos casos, hasta agradecer a su tutor o tutores del proyecto la ayuda prestada...

## *AGRADECIMIENTOS*



# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Índice general



# Índice de figuras

## ÍNDICE DE FIGURAS



# Índice de fragmentos de código

## *ÍNDICE DE FRAGMENTOS DE CÓDIGO*

# Capítulo 1

## Introducción

Debido al avance de las tecnologías, la robótica se encuentra en constante evolución, lo que hace que cada vez la podamos ver en más sectores, desde el mundo laboral hasta en nuestro día a día. Sin embargo, el aprendizaje de la robótica es un terreno complejo, ya que para poder empezar en este campo se requiere la instalación y configuración de distintas herramientas y entornos de programación, los cuales pueden resultar bastante enrevesados, en especial a nuevos usuarios que quiere empezar en el mundo de la robótica.

Otro gran problema del aprendizaje de la robótica es el coste de un robot en el cual ir probando nuestros códigos, ya que a diferencia de otras materias la robótica es un campo que su aprendizaje es a base de prueba y error.

Este trabajo de fin de grado gira entorno a UNIBOTICS una plataforma web, que nace como solución a estos problemas. UNIBOTICS permite a los usuarios acceder a ejercicios interactivos de robótica, en los cuales podrán programar robots y poder simularlos en escenarios 3D sin la necesidad de tener el robot de forma física y sin tener que instalar ningún tipo de programa.

Sin embargo, a pesar de sus ventajas, aún existen áreas de UNIBOTICS que podrían mejorarse. Actualmente, todos los datos de usuarios desde los códigos de programación, tiempo de actividad, tiempo de resolución de actividades y muchos otros datos más, se recogen en una base de datos del tipo PostgreSQL, pero estos datos únicamente se almacenan y no se les saca todo el potencial que podrían tener.

Dada esta situación, este proyecto quiere proporcionar una API que permita extraer y procesar estos datos. Además, se diseñarán unas Dashboards interactivas, donde se podrán ver de forma clara y detallada toda la información obtenida a través de la API.

Con la implementación de estas Dashboards se busca que los profesores puedan ver de

forma rápida y sencilla el trabajo realizado por sus alumnos, lo cual les podrá facilitar la enseñanza a los alumnos de robótica.

## 1.1 Objetivos

Esto es una sección, que es una estructura menor que un capítulo.

Por cierto, a veces me comentáis que no os compila por las tildes. Eso es un problema de codificación. Al guardar el archivo, guardad la codificación de “ISO-Latin-1” a “UTF-8” (o viceversa) y funcionará.

### 1.1.1 Objetivo general

Mi Trabajo Fin de Grado consiste en crear de una herramienta que permita extraer datos de la plataforma web UNIBOTICS y crear unos dashboards interactivos para representar esa información

### 1.1.2 Objetivos específicos

Los objetivos específicos se pueden entender como las tareas en las que se ha desglosado el objetivo general. Y, sí, también vienen en infinitivo.

Lo mejor suele ser utilizar una lista no numerada, como sigue:

- Un objetivo específico.
- Otro objetivo específico.
- Tercer objetivo específico.
- ...

## 1.2 Planificacion

Rellenar alfinal

## 1.3 Estructura del documento

Es conveniente que incluyas una descripción de lo que te ha llevado realizar el trabajo. Hay gente que añade un diagrama de GANTT. Lo importante es que quede claro cuánto tiempo has consumido en realizar el TFG/TFM (tiempo natural, p.ej., 6 meses) y a qué nivel de esfuerzo (p.ej., principalmente los fines de semana).

## 1.4 Estructura de la memoria

Por último, en esta sección se introduce a alto nivel la organización del resto del documento y qué contenidos se van a encontrar en cada capítulo.

- En el primer capítulo se hace una breve introducción al proyecto, se describen los objetivos del mismo y se refleja la planificación temporal.
- En el siguiente capítulo se describen las tecnologías utilizadas en el desarrollo de este TFM/TFG (Capítulo 2).
- En el capítulo 3 Se describe el proceso de desarrollo de la herramienta ...
- En el capítulo 4 Se presentan las principales pruebas realizadas para validación de la plataforma/herramienta...(o resultados de los experimentos efectuados).
- Por último, se presentan las conclusiones del proyecto así como los trabajos futuros que podrían derivarse de éste (Capítulo ??).



# Capítulo 2

## Tecnologías

En este capítulo describiré las tecnologías utilizadas para este proyecto.

### 2.1 Python

Python es un lenguaje de programación de alto nivel, interpretado, dinámico y fuertemente tipado. Python es un lenguaje multiparadigma, lo que quiere decir que permite desarrollar software utilizando distintos enfoques como la programación orientada a objetos, la programación funcional y la programación imperativa.

El principal motivo por el cual he elegido Python como lenguaje de programación para este trabajo es que la plataforma UNIBOTICS está desarrollada en este lenguaje, por lo que usar otro sería ineficiente y una complicación innecesaria. Además, Python destaca por su simplicidad y legibilidad, lo que facilita el desarrollo y mantenimiento del código, su gran comunidad que ofrece soporte constante y su gran variedad de bibliotecas en las cuales se apoya este proyecto. A continuación se enumeran y se detallan el funcionamiento de cada una:

#### 2.1.1 Pandas

Pandas es una de las herramientas más poderosas para la manipulación y análisis de datos en Python. Esta biblioteca se diseñó para hacer que la limpieza, transformación y análisis de datos sean rápidos y eficientes.

Una de las razones por las que he elegido pandas en este proyecto es su capacidad para

manipular datos de manera flexible y eficiente. Sus estructuras principales, Series y DataFrame, facilitan la organización y el acceso a la información de manera intuitiva. La Series actúa como un array unidimensional con etiquetas asociadas, mientras que el DataFrame es una tabla bidimensional que permite realizar operaciones similares a las de bases de datos o herramientas como Excel.

En el contexto de este proyecto, pandas es la mejor opción porque la plataforma en la que se desarrollará ya está basada en Python y requiere una gestión eficiente de datos tabulares. Además, como se menciona en *Python for Data Analysis, 3rd Edition*, pandas no solo facilita la manipulación de datos, sino que también nos permite enfocarnos en la interpretación y visualización de la información, en lugar de perder tiempo en tareas repetitivas de procesamiento.

### 2.1.2 Dash

Añadir cuando empiece con las dashboard

## 2.2 PostgreSQL

Añadir cuando tenga acceso a la base de datos

## 2.3 Entorno de desarrollo: PyCharm

PyCharm es un Integrated Development Enviroment (Entorno de Desarrollo Integrado) (IDE) dedicado concretamente a la programación en Python y desarrollado por la compañía checa JetBrains.

Proporciona análisis de código, un depurador gráfico, una consola de Python integrada, control de versiones y, además, soporta desarrollo web con Django. Todas estas características lo convierten en un entorno completo e intuitivo, idóneo para el desarrollo de proyectos académicos como el que nos ocupa.



## 2.4 Redacción de la memoria: LaTeX/Overleaf

LaTeX es un sistema de composición tipográfica de alta calidad que incluye características especialmente diseñadas para la producción de documentación técnica y científica. Estas características, entre las que se encuentran la posibilidad de incluir expresiones matemáticas, fragmentos de código, tablas y referencias, junto con el hecho de que se distribuya como software libre, han hecho que LaTeX se convierta en el estándar de facto para la redacción y publicación de artículos académicos, tesis y todo tipo de documentos científico-técnicos.

Por su parte, Overleaf es un editor LaTeX colaborativo basado en la nube. Lanzado originalmente en 2012, fue creado por dos matemáticos que se inspiraron en su propia experiencia en el ámbito académico para crear una solución satisfactoria para la escritura científica colaborativa.

Además de por su perfil colaborativo, Overleaf destaca porque, pese a que en LaTeX el escritor utiliza texto plano en lugar de texto formateado (como ocurre en otros procesadores de texto como Microsoft Word, LibreOffice Writer y Apple Pages), éste puede visualizar en todo momento y paralelamente el texto formateado que resulta de la escritura del código fuente.



# Capítulo 3

## Arquitectura

Aquí viene todo lo que has hecho tú (tecnológicamente). Puedes entrar hasta el detalle. Es la parte más importante de la memoria, porque describe lo que has hecho tú. Eso sí, normalmente aconsejo no poner código, sino diagramas.

### 3.1 Arquitectura general

Como ya dije antes, UNIBOTICS es una plataforma web diseñada con el fin de facilitar el aprendizaje práctico de la robótica, proporcionando a los estudiantes un entorno en el que pueden encontrar ejercicios y escenarios interactivos sin la necesidad de instalar o configurar entornos complejos de software.

Esta plataforma proporciona diferentes herramientas para trabajar con robots. La herramienta que gestiona el software robótico requerido, es RADI (Robotics Academy Docker Image). Se trata de unos contenedores Docker especiales en los que se llevan a cabo la ejecución de ejercicios y las simulaciones robóticas.

Estos contenedores integran ROS2 y Gazebo, además de otras dependencias necesarias para poder ejecutar el código de los usuarios. ROS2 (Robot Operating System 2) ofrece un middleware estándar para la programación robótica, permitiendo comunicar diferentes nodos y controlar así el flujo de datos entre los distintos sensores. Gazebo aporta un entorno de simulación física realista que proporciona entornos 3D donde se puede simular el comportamiento de los robots con el código previamente programado por el usuario.

Para comunicar el contenedor RADI con el navegador, se utiliza otra herramienta llamada RAM (Robot Application Manager), que actúa como puente entre el navegador y el contenedor. De esta forma, cuando el usuario modifica o ejecuta código, RAM lo recibe, lo

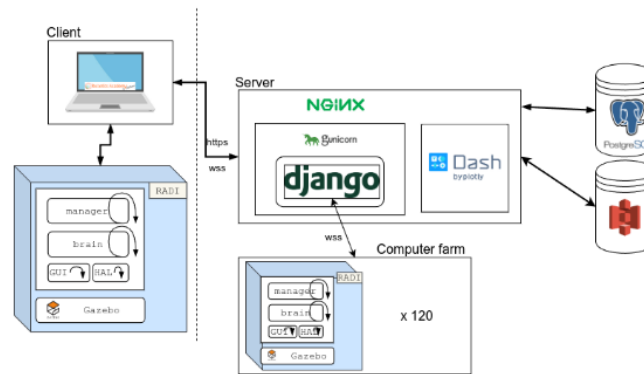


Figura 3.1: Estructura de Unibotics.

ejecuta en el contenedor y devuelve los resultados correspondientes. Estos resultados pueden ser desde imágenes del entorno y la posición del robot hasta cualquier dato relevante para el ejercicio.

La Figura 3.1 ilustra esta arquitectura, mostrando la relación entre los distintos componentes de la plataforma, desde la interacción del usuario a través del navegador hasta la gestión de simulaciones y datos en los servidores.

Para almacenar toda esta cantidad de datos que ofrece cada simulación, además de otros datos de Unibotics como los usuarios, datos estadísticos de ejercicios y datos del desempeño del usuario, se emplea una base de datos relacional del tipo PostgreSQL, lo que facilita la gestión y el análisis futuro de la información, de la cual obtendremos los datos a través de nuestra API para así poder representarlos en los dashboards.

Respecto a la parte del usuario, la interacción con esta plataforma se realiza a través de un navegador web, mediante el cual el usuario accede a una interfaz construida con React, donde se encuentran los ejercicios y escenarios interactivos.

Para implementar todas estas herramientas, UNIBOTICS se apoya en Django, un framework de alto nivel orientado al desarrollo web en Python, especialmente a la creación de aplicaciones web complejas. Gracias a Django, es posible estructurar de forma clara la lógica interna de la plataforma, lo que permite relacionar usuarios con ejercicios y almacenar información sobre las sesiones de trabajo.

Además de Django, emplea Nginx y Gunicorn, herramientas que mejoran la eficiencia de la plataforma. Estas permiten gestionar un gran volumen de usuarios de manera simultánea, asegurando que el sistema opere sin interrupciones.

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación. asíque lo añadire cuando tenga acceso a esa base de datos

# Capítulo 4

## Experimentos y validación

**Atención:** Este capítulo se introdujo como requisito en 2019.

Describe los experimentos y casos de test que tuviste que implementar para validar tus resultados. Incluye también los resultados de validación que permiten afirmar que tus resultados son correctos.

### 4.1 Incorporación de código en la memoria

Es bastante habitual que se reproduzcan fragmentos de código en la memoria de un TFG/TFM. Esto permite explicar detalladamente partes del desarrollo que se ha realizado que se consideren de especial interés. No obstante, tampoco es conveniente pasarse e incluir demasiado código en la memoria, puesto que se puede alargar mucho el documento. Un recurso muy habitual es subir todo el código a un repositorio de un servicio de control de versiones como GitHub o GitLab, y luego incluir en la memoria la URL que enlace a dicho repositorio.

Para incluir fragmentos de código en un documento  $\text{\LaTeX}$  se pueden combinar varias herramientas:

- El entorno `kbegin+nblistign+nb[]...kend+nblistign+nb` permite crear un marco en el que situar el fragmento de código (parecido al generado cuando insertamos una tabla o una figura). Podemos insertar también una descripción (*caption*) y una etiqueta para referenciarlo luego en el texto.
- Dentro de este entorno, se puede utilizar el paquete `minted`<sup>1</sup>, que utiliza el paquete

---

<sup>1</sup>[https://es.overleaf.com/learn/latex/Code\\_Highlighting\\_with\\_minted](https://es.overleaf.com/learn/latex/Code_Highlighting_with_minted)