



# TECNOLOGÍA SUPERIOR UNIVERSITARIA EN DESARROLLO DE SOFTWARE

DESARROLLO DE APLICACIONES CON PYTHON

CARLOS LUNA

24 SEPTIEMBRE 2024

# PROYECTO FINAL PYTHON

## 1. Descripción del proyecto

El presente proyecto tiene como objeto que el Instituto Superior Universitario Tecnológico del Azuay presente un recorrido virtual interactivo de su campus, diseñado para promover su oferta académica. Durante el recorrido, los visitantes pueden explorar aulas, talleres y áreas administrativas mientras un agente recopila sus preferencias mediante preguntas e interacciones con el usuario. Al final, el sistema utiliza IA para sugerir la carrera ideal en base a los datos obtenidos. Además, se proporcionarán informes técnicos para el uso del instituto y se almacenarán datos del usuario para personalizar la experiencia y ofrecer seguimiento adecuado.

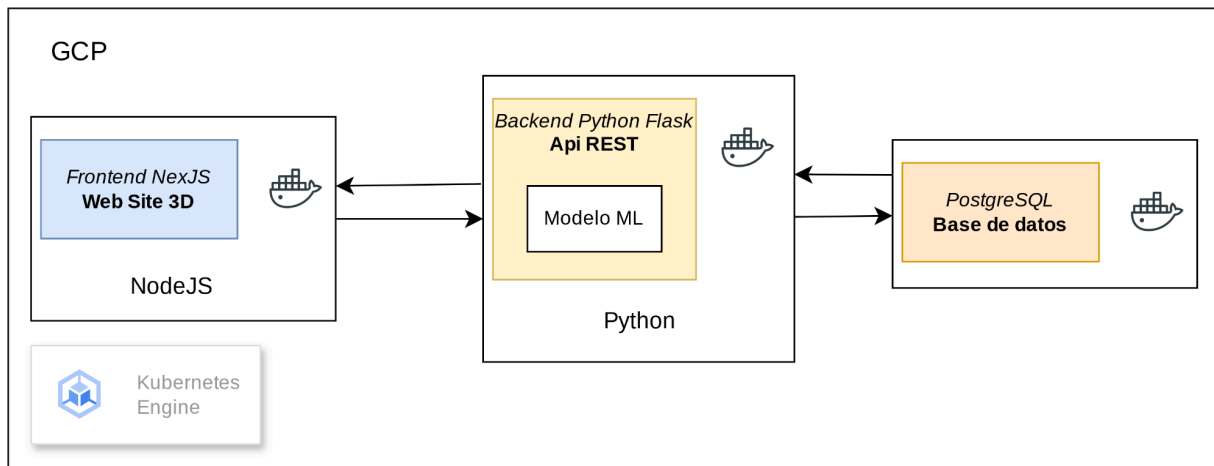
### a. Objetivo General

Desarrollar un recorrido virtual interactivo del campus del Instituto Superior Universitario del Azuay, que permita a los usuarios explorar las instalaciones y recibir recomendaciones personalizadas de carreras mediante un chatbot inteligente basado en inteligencia artificial, mejorando así la orientación vocacional y la promoción de los programas académicos.

### b. Objetivos específicos

- Crear un entorno virtual 3D detallado del campus que incluya aulas, talleres, auditorio, áreas administrativas y estacionamientos, utilizando herramientas como Blender y Three.js.
- Implementar un entorno interactivo y fluido para el recorrido virtual, asegurando una navegación intuitiva y optimizada para todos los usuarios.
- Desarrollar e integrar un chatbot inteligente que interactúe con los usuarios durante el recorrido, recolectando sus preferencias y sugiriendo carreras adecuadas.
- Implementar un sistema de gestión de datos de usuario que almacene información de interacción, asegurando la protección y privacidad de los datos personales.
- Generar informes técnicos detallados que analicen el uso del recorrido virtual y el rendimiento del chatbot, para optimizar la experiencia del usuario y mejorar el sistema.
- Crear documentación y capacitar al personal del instituto en el uso y mantenimiento del sistema, asegurando su correcta implementación y soporte continuo.

### c. Diagrama



## 1. Propuestas de Mejora

El sistema en cuanto a backend se refiere, estará desarrollado usando la última versión de Python disponible actualmente (3.12.x), actualmente el proyecto aún se encuentra en desarrollo temprano por lo que se empezaría implementando buenas prácticas desde el inicio para evitar problema a futuro y minimizar la refactorización cuando el proyecto ya esté avanzado.

Una de las prácticas a utilizar será la división por módulos aplicando una arquitectura limpia que nos permite segmentar las funcionalidades de la aplicación para obtener la menor cohesión posible en cuanto a tecnologías externas a las reglas de negocio, de esta manera el esfuerzo se iniciará por resolver la lógica necesaria para el funcionamiento y a posterior la implementación de capas externas de comunicación con otros servicios.

Algunos patrones de diseño a implementar serían **Facade**, que nos permite definir las acciones del sistema y su interacción por medio de interfaces sencillas que luego se implementarán.

Se hará uso de **inyección de dependencias** para los diferentes servicios a usar ya que el proyecto será modular, y de esta manera se puede organizar el código y las diferentes implementaciones de las interfaces definidas anteriormente, en conjunto con esto también se haría uso del patrón **Repository** para manejar de mejor manera el acceso a la capa de persistencia, de igual manera hacer uso de instancias **Singleton** podría ser útil en este escenario.

El patrón **Observer** también podría ser utilizado para generar notificaciones de cuando un estudiante ha elegido una carrera universitaria en la aplicación.

El tipado en un lenguaje de programación lo considero muy importante, siempre he trabajado con lenguajes de este tipo como Java o Typescript, de esta manera permite especificar de manera explícita qué tipo de información es la entrada de una función y cuál es su salida, así evitaremos y reduciremos los

problemas con tipos de datos en tiempo de ejecución, para este punto hacer uso de herramientas que nos permitan chequear el código al momento de escribirlo es importante por lo que se implementará **Pylint**, una de las tecnologías más utilizadas para este propósito junto con **Pyright** que verifica el tipado en el código.

### Planes de Pruebas Completos:

Se realizarán pruebas unitarias de los diferentes casos de uso con la librería unittest que viene por defecto instalada en Python, esta librería es suficiente para cubrir las necesidades de testing del proyecto.

```
import unittest

def suma(a, b):
    return a + b

class TestSuma(unittest.TestCase):
    def test_suma(self):
        self.assertEqual(suma(2, 3), 5)
        self.assertEqual(suma(-1, 1), 0)

if __name__ == '__main__':
    unittest.main()
```

La herramienta de Postman también será incluida para realizar pruebas de integración, se plantea automatizar estas pruebas creando un flujo CI/CD con Github Actions.

### Mejoras en el Pipeline de CI/CD:

Se tiene planteado seguir este ejemplo de pipeline para desplegar y ejecutar los tests con Github Actions:

```
# .github/workflows/ci-cd.yml
name: CI/CD ISTA Tour API

# Ejecuta el workflow en push y pull request a la rama principal
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
```

```

build:
  runs-on: ubuntu-latest

  steps:
    - name: Checkout repository
      uses: actions/checkout@v3

    - name: Build Docker image
      run: |
        docker build -t api-rest:latest .

    - name: Run API container
      run: |
        docker run -d -p 5000:5000 --name api-container api-rest:latest

    - name: Run unit tests
      run: |
        docker exec api-container pytest

    - name: Install Newman
      run: |
        npm install -g newman

    - name: Run Postman integration tests
      run: |
        newman run postman/collection.json

    - name: Clean up
      run: |
        docker stop api-container
        docker rm api-container

```

### **Acceso a Datos:**

Para la capa de acceso a datos y persistencia se haría uso de SQLAlchemy que se puede integrar fácilmente con la estructura del proyecto planteada anteriormente.

### **Empaquetado y Despliegue:**

Para el despliegue del proyecto se hará uso de recursos en la nube con **Google Cloud Platform**, haciendo uso de contenedores y el **Artifact Registry** para las imágenes del proyecto, también se implementarán los dockers con Kubernetes para la interacción de estos.

## 2. Tecnologías y Prácticas No Utilizadas

Una de las tecnologías que se suelen usar para crear proyectos backend en python suele ser Django si embargo aunque ofrece un entorno con varias herramientas integradas, no se consideró puesto que requiere de cierta curva de aprendizaje y comprende una estructura predeterminada, por lo que se optó por el tiempo de desarrollo del proyecto hacer uso de Flask e integrar otras librerías según se requieran.

## 3. Conclusión

El proyecto propuesto para el Instituto Superior Universitario Tecnológico del Azuay tiene como objetivo implementar un recorrido virtual interactivo del campus, acompañado de un sistema de recomendación de carreras basado en IA. Las propuestas se centran en adoptar buenas prácticas desde el inicio, incluyendo el uso de una arquitectura modular y limpia, implementación de patrones de diseño como Facade, Repository, Singleton y Observer, y el uso de tipado estático con herramientas como Pylint y Pyright.

Además, se prioriza la automatización del flujo de pruebas y despliegue continuo con GitHub Actions, Docker y Kubernetes para optimizar el desarrollo. Se utilizarán SQLAlchemy para la gestión de acceso a datos y GCP para el despliegue en la nube. Aunque no se opta por Django debido a su curva de aprendizaje, se implementará Flask para mayor flexibilidad y control sobre las librerías utilizadas. La integración de pruebas unitarias con unittest y de integración con Postman garantiza una sólida validación del sistema.