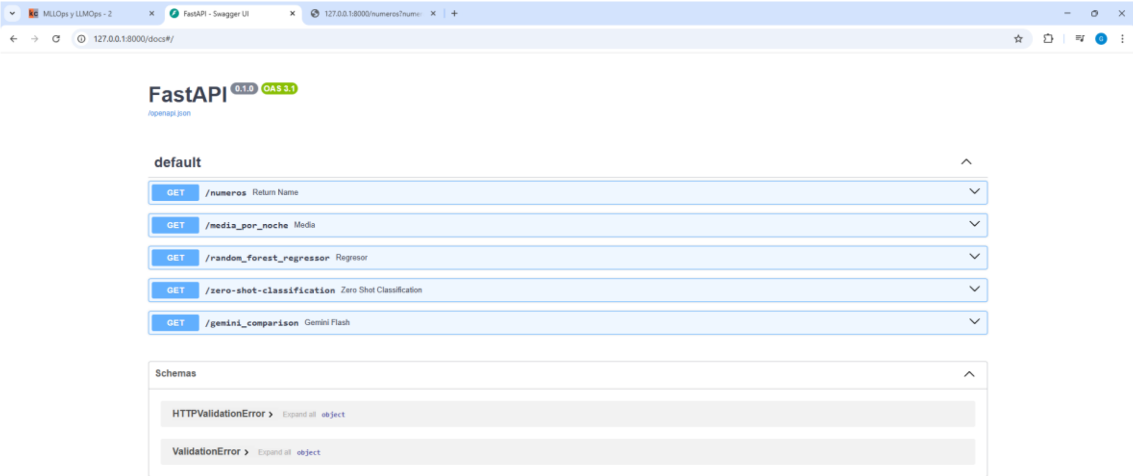
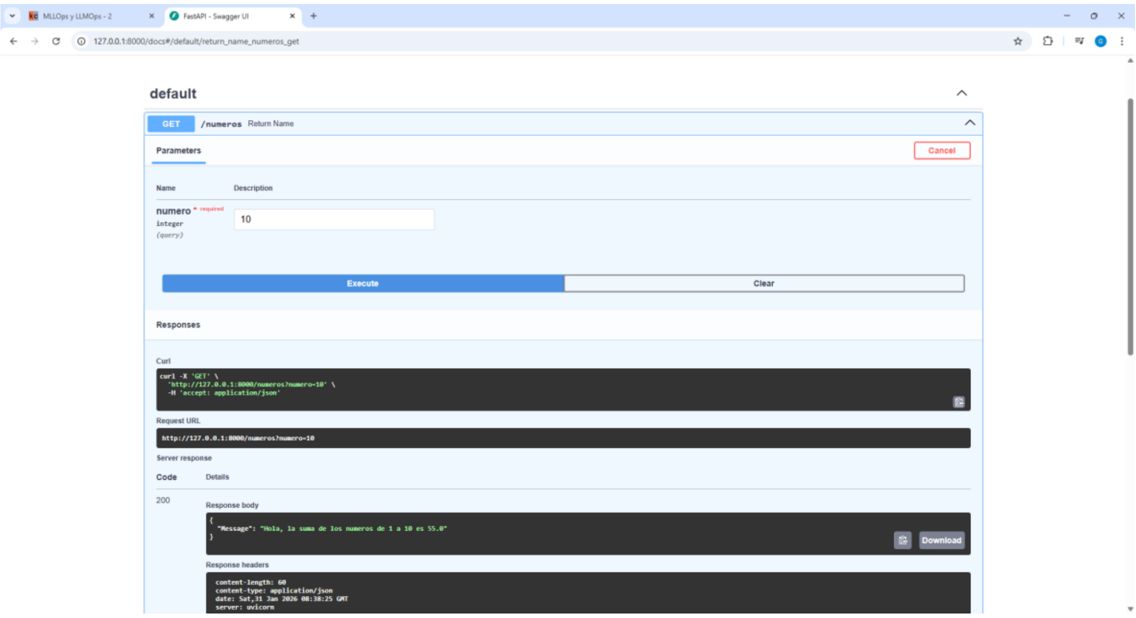
Capturas de pantallas de FastAPI

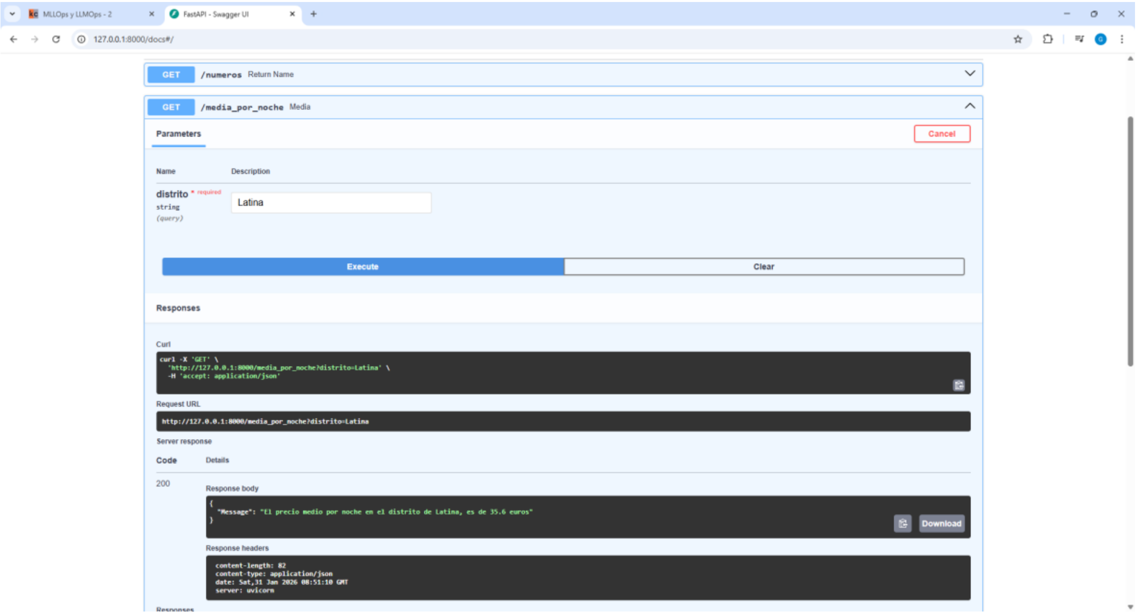
Comenzamos con los 5 módulos, que van creciendo en dificultad. Ahora mostraremos las capturas de la API, y después las de las llamadas http.



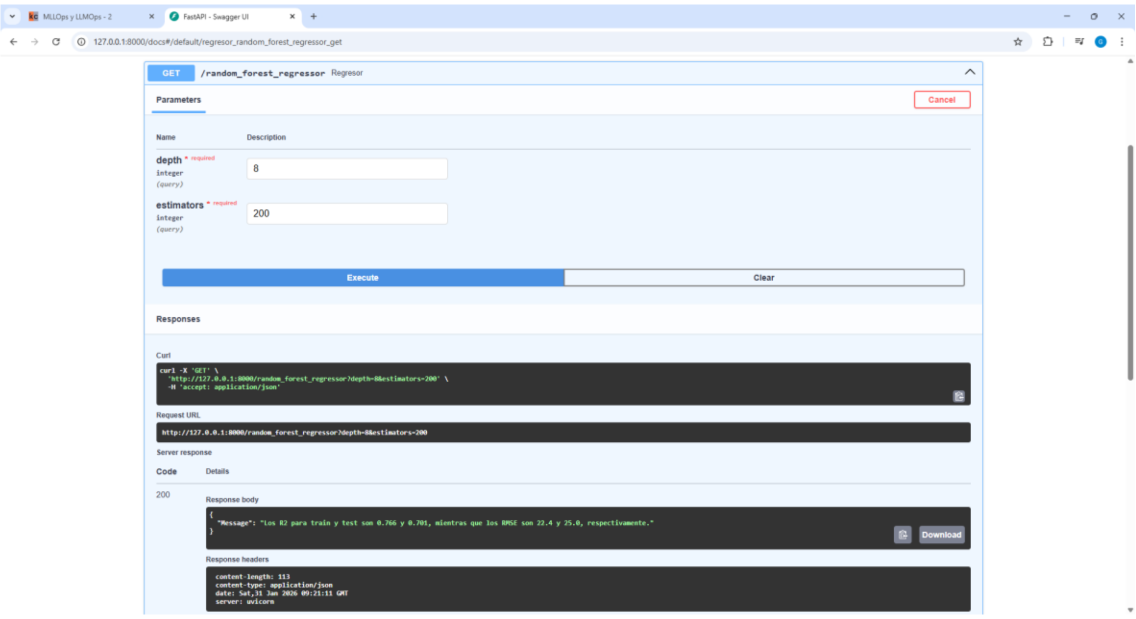
El primer módulo es simplemente que pide un número entero y devuelve la suma de los enteros desde 1 a ese número.



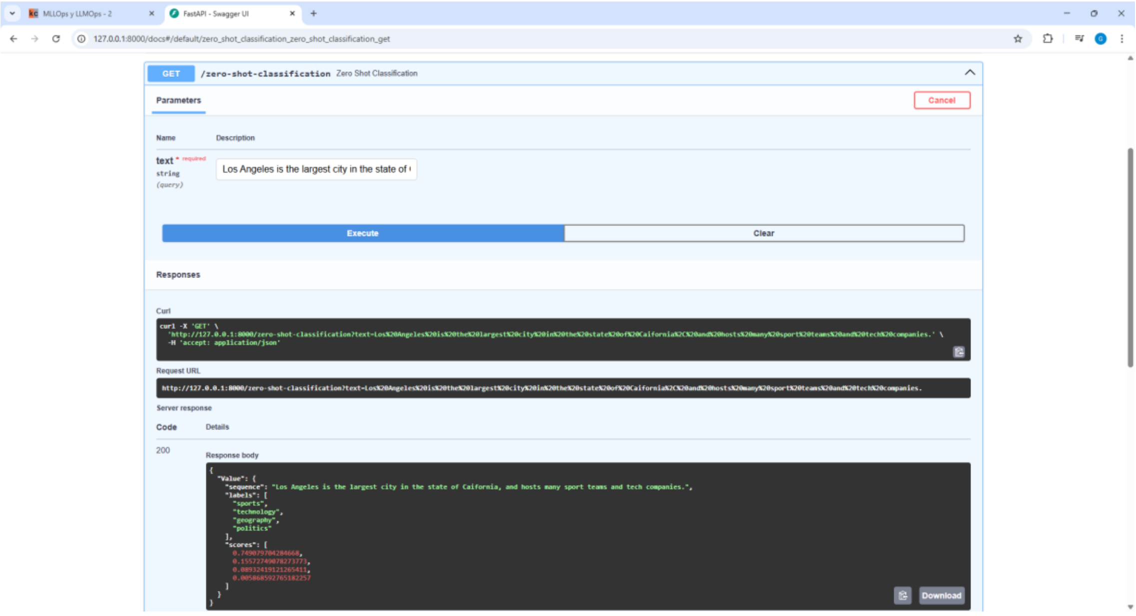
El siguiente módulo hace uso de los dataset que hemos utilizado en MFlow, así como de las funciones definidas extraídas del notebook de esa parte. El módulo pide el nombre de un distrito y devuelve el precio medio por noche en él de la oferta de Airbnb.



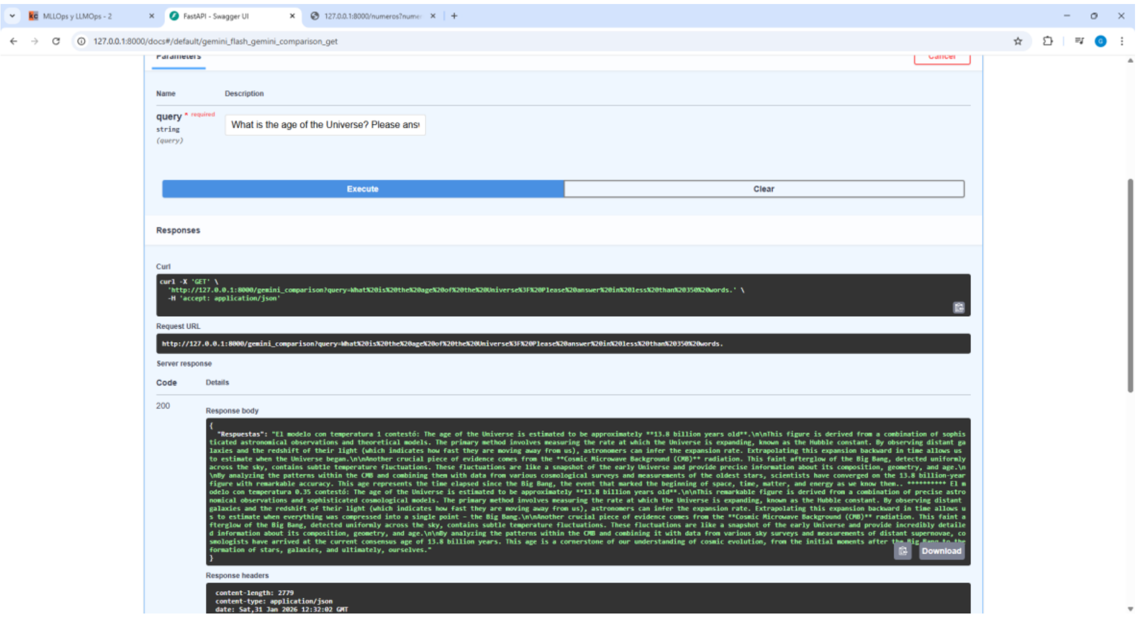
El tercer módulo profundiza algo más, y calcula el modelo de Random Forest y métricas de R2 y RMSE, tras pedir la profundidad y el número de estimadores, tanto para train como para test.



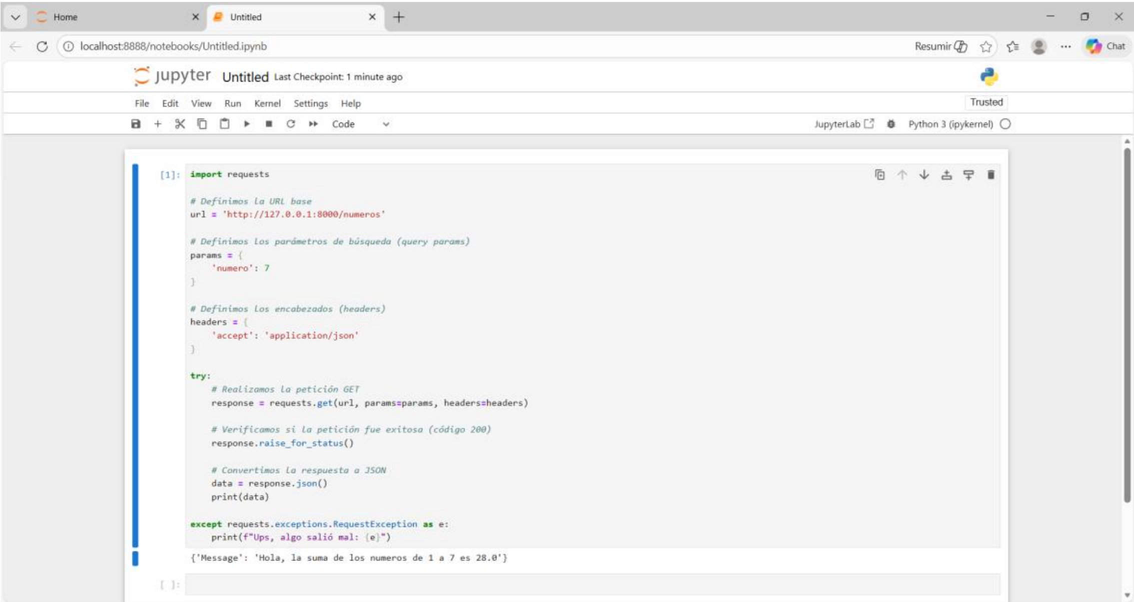
El cuarto módulo hace uso de la pipeline de Hugging Face Zero Shot Classification, a la que hay que introducir un texto. Intenté añadir otras pipelines como NER o traducción, pero me dio errores.

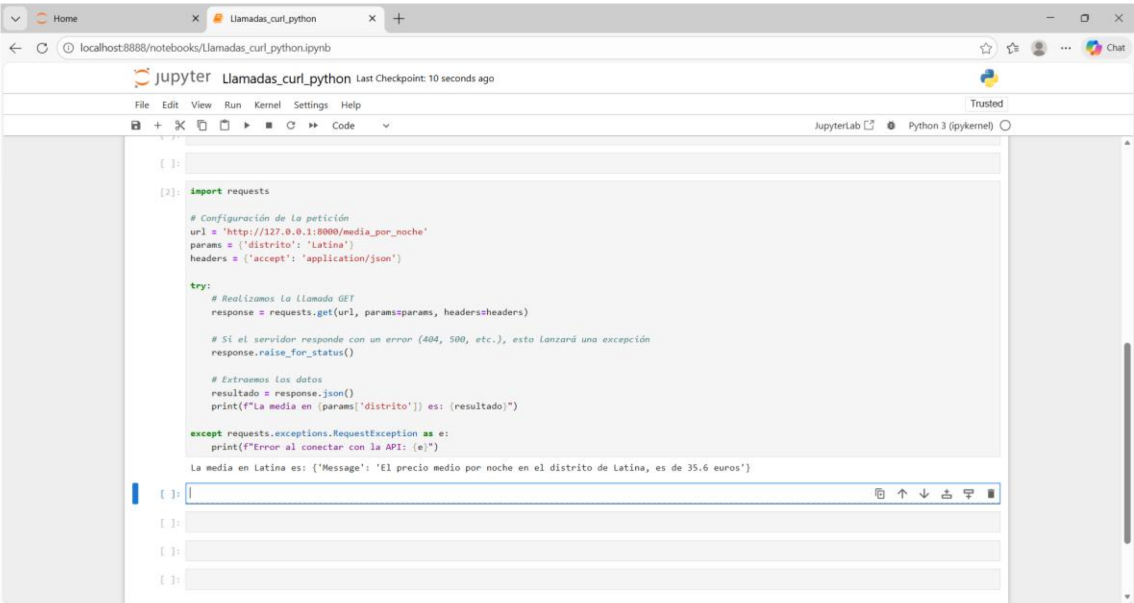


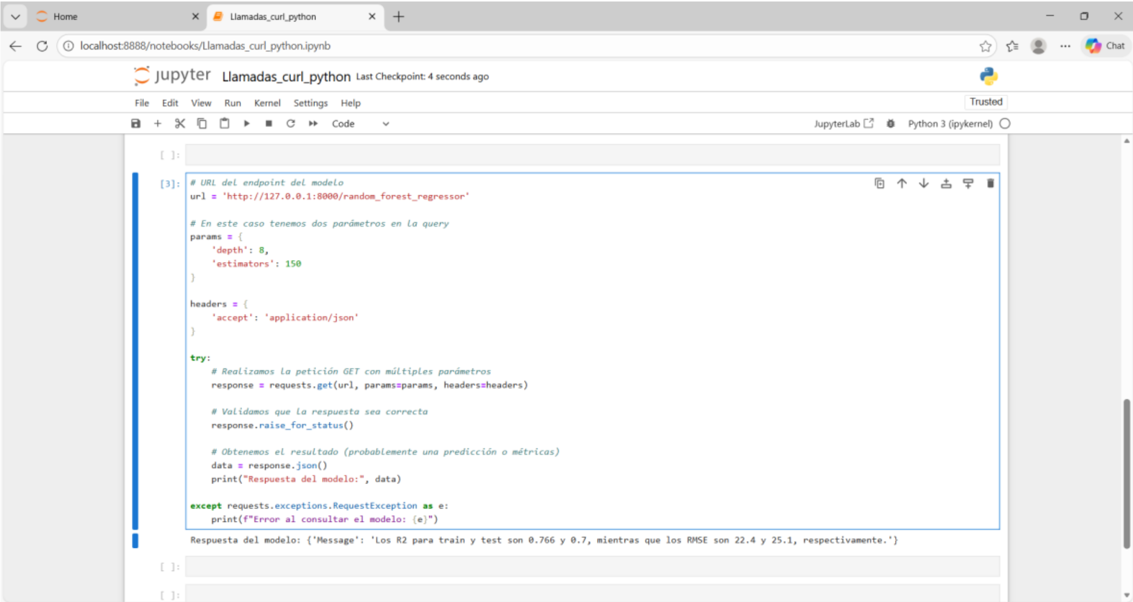
Por último, el quinto módulo ejecuta el modelo Gemini2.5-Flash, para un mismo texto dos veces, pero le variamos la temperatura en cada caso. Hubo que variar el código bastante, porque ahora la API funciona a base de Client().



A continuación, se muestran las capturas de pantalla de las peticiones http, en el mismo orden.

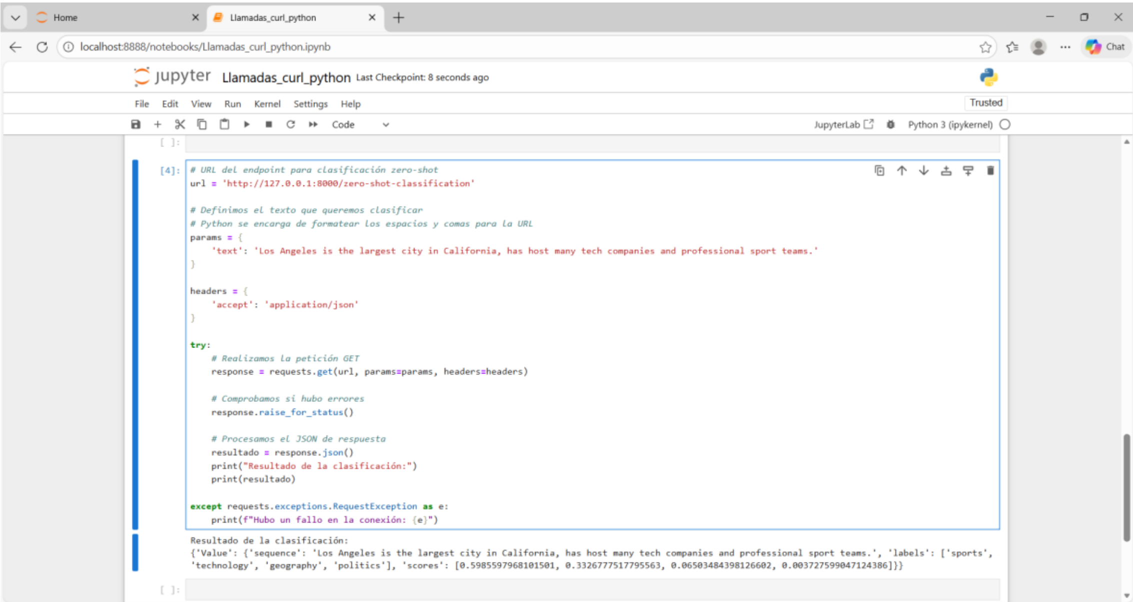






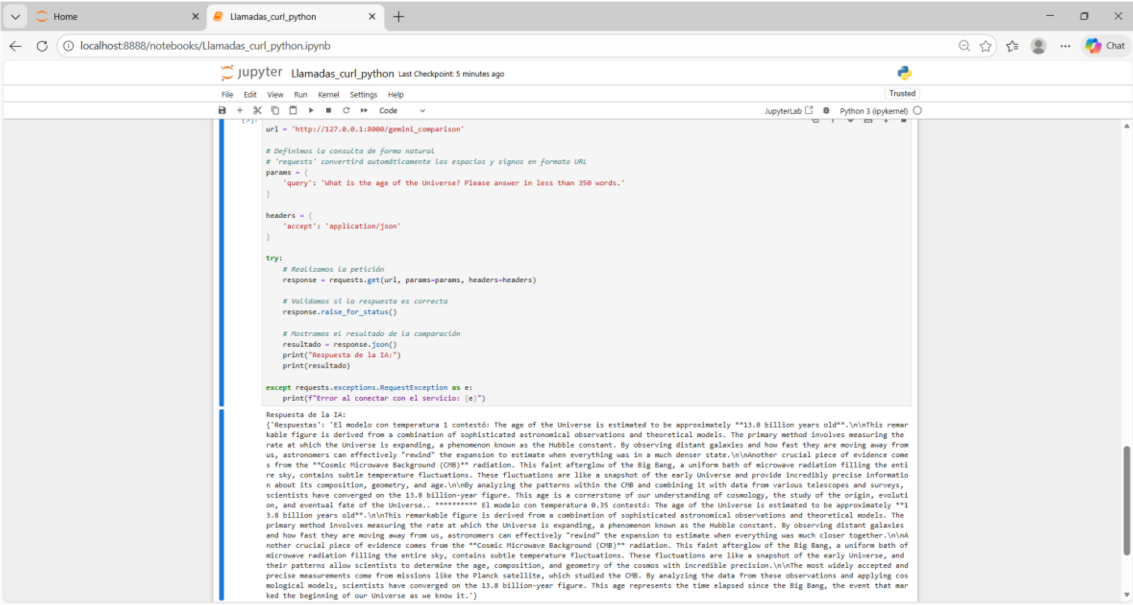
```python
# URL del endpoint del modelo
url = 'http://127.0.0.1:8000/random_forest_regressor'

# En este caso tenemos dos parámetros en la query
params = {
    'depth': 8,
    'estimators': 150
}

headers = {
    'accept': 'application/json'
}

try:
    # Realizamos la petición GET con múltiples parámetros
    response = requests.get(url, params=params, headers=headers)

    # Validamos que la respuesta sea correcta
    response.raise_for_status()

    # Obtenemos el resultado (probablemente una predicción o métricas)
    data = response.json()
    print("Respuesta del modelo:", data)

except requests.exceptions.RequestException as e:
    print(f"Error al consultar el modelo: {e}")
```

```
Respuesta del modelo: {'Message': 'Los R2 para train y test son 0.766 y 0.7, mientras que los RMSE son 22.4 y 25.1, respectivamente.'}
```



```python
# URL del endpoint para clasificación zero-shot
url = 'http://127.0.0.1:8000/zero-shot-classification'

# Definimos el texto que queremos clasificar
# Python se encarga de formatear los espacios y comas para la URL
params = {
    'text': 'Los Angeles is the largest city in California, has host many tech companies and professional sport teams.'
}

headers = {
    'accept': 'application/json'
}

try:
    # Realizamos la petición GET
    response = requests.get(url, params=params, headers=headers)

    # Comprobamos si hubo errores
    response.raise_for_status()

    # Procesamos el JSON de respuesta
    resultado = response.json()
    print("Resultado de la clasificación:")
    print(resultado)

except requests.exceptions.RequestException as e:
    print(f"Hubo un fallo en la conexión: {e}")
```

```
Resultado de la clasificación:
{'Value': {'sequence': 'Los Angeles is the largest city in California, has host many tech companies and professional sport teams.', 'labels': ['sports', 'technology', 'geography', 'politics'], 'scores': [0.5985597968101501, 0.3326777517795563, 0.06503484398126602, 0.003727599047124386]}}
```



```python
url = 'http://127.0.0.1:8000/gemini_comparison'

# Definimos la consulta de forma natural
# 'requests' convertirá automáticamente los espacios y signos en formato URL
params = {
    'query': 'What is the age of the Universe? Please answer in less than 350 words.'
}

headers = {
    'accept': 'application/json'
}

try:
    # Realizamos la petición
    response = requests.get(url, params=params, headers=headers)

    # Validamos si la respuesta es correcta
    response.raise_for_status()

    # Mostramos el resultado de la comparación
    resultado = response.json()
    print("Respuesta de la IA:")
    print(resultado)

except requests.exceptions.RequestException as e:
    print(f"Error al conectar con el servicio: {e}")
```

```
Respuesta de la IA:
{'Respuestas': 'El modelo con temperatura 1 contestó: The age of the Universe is estimated to be approximately **13.8 billion years old**.\n\nThis remarkable figure is derived from a combination of sophisticated astronomical observations and theoretical models. The primary method involves measuring the rate at which the Universe is expanding, a phenomenon known as the Hubble constant. By observing distant galaxies and how fast they are moving away from us, astronomers can effectively "rewind" the expansion to estimate when everything was in a much denser state.\n\nAnother crucial piece of evidence comes from the **Cosmic Microwave Background (CMB)** radiation. This faint afterglow of the Big Bang, a uniform bath of microwave radiation filling the entire sky, contains subtle temperature fluctuations. These fluctuations are like a snapshot of the early Universe and provide incredibly precise information about its composition, geometry, and age.\n\nBy analyzing the patterns within the CMB and combining it with data from various telescopes and surveys, scientists have converged on the 13.8 billion-year figure. This age is a cornerstone of our understanding of cosmology, the study of the origin, evolution, and eventual fate of the Universe.. ********** El modelo con temperatura 0.35 contestó: The age of the Universe is estimated to be approximately **13.8 billion years old**.\n\nThis remarkable figure is derived from a combination of sophisticated astronomical observations and theoretical models. The primary method involves measuring the rate at which the Universe is expanding, a phenomenon known as the Hubble constant. By observing distant galaxies and how fast they are moving away from us, astronomers can effectively "rewind" the expansion to estimate when everything was much closer together.\n\nAnother crucial piece of evidence comes from the **Cosmic Microwave Background (CMB)** radiation. This faint afterglow of the Big Bang, a uniform bath of microwave radiation filling the entire sky, contains subtle temperature fluctuations. These fluctuations are like a snapshot of the early Universe, and their patterns allow scientists to determine the age, composition, and geometry of the cosmos with incredible precision.\n\nThe most widely accepted and precise measurements come from missions like the Planck satellite, which studied the CMB. By analyzing the data from these observations and applying cosmological models, scientists have converged on the 13.8 billion-year figure. This age represents the time elapsed since the Big Bang, the event that marked the beginning of our Universe as we know it.'}
```