

1º DAW Y 1º DAM

# PRESENTACIÓN

F I N A L 1 º D U A L

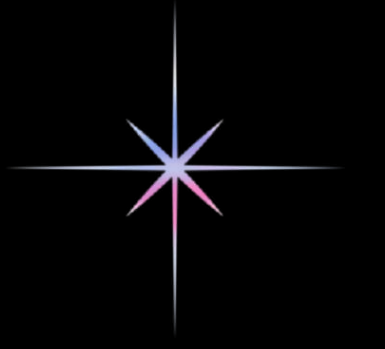


**IUITLM**

Instituto Universitario de Investigación  
de Tecnologías Lingüísticas Multilingües  
UNIVERSIDAD DE MÁLAGA



# Nuestros Integrantes



Javier  
Ballesteros

1ºDAM  
Mañana

Alberto  
Maldonado

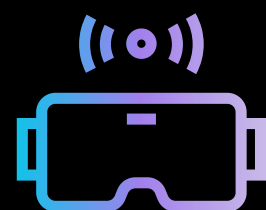
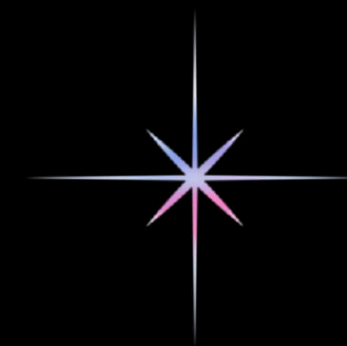
1ºDAM  
Mañana

Rubén  
Sánchez

1º DAW  
Mañana

Mateo R.  
Mayger

1ºDAW  
Mañana



# INTRODUCCIÓN

## IUITLM

El Instituto Universitario de Investigación de Tecnologías Lingüísticas Multilingües (IUITLM) es un laboratorio de investigación afiliado a la Universidad de Málaga.

Este se enfoca en I+D+i dirigidos a la aplicación de las tecnologías lingüísticas en investigación dentro de diversos ámbitos y ramas del conocimiento.





# Actividades Realizadas

Al ser un equipo de pocas personas nos han dado formación colectiva (hemos hecho las mismas actividades a modo de aprendizaje), por lo tanto hemos utilizado las mismas tecnologías

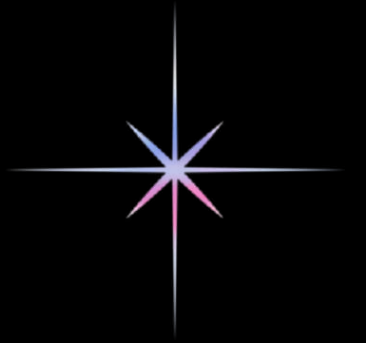
Conforme avanzábamos nos hemos enfocando en diferentes cosas, pero por lo general hemos seguido el mismo camino







# 1º Semana



## ACTIVIDADES REALIZADAS

Introducción a las tecnologías de la empresa.  
Como Docker, Python, NodeJS y PHP.

## ASIGNATURAS RELACIONADAS

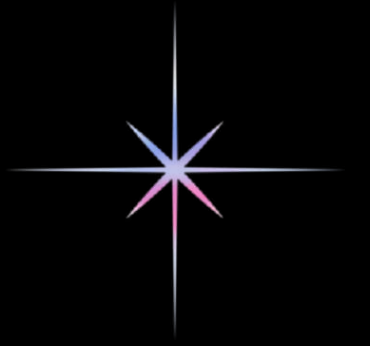
Programación, Sistemas, Lenguajes de Marcas,  
Base de datos y Entorno de Desarrollo.

## PROYECTOS REALIZADOS

Inicio del documento de introducción a la  
empresa.



# Tecnologías Usadas



Python



Docker



Css 3



Javascript



Php



Github



Nodejs

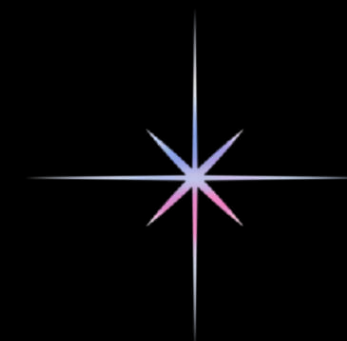


Htмл 5





# 2º Semana



## ACTIVIDADES REALIZADAS

Creacion de pares de llaves publico/privada  
Crear primeros “endpoints” con Python + Flask y  
PHP servidos con Docker.  
Conseguir conectar la API con una BBDD.

## ASIGNATURAS RELACIONADAS

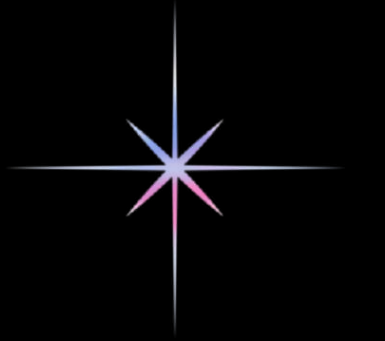
Sistemas, Programacion, Lenguajes de Marcas,  
BBDD y Entornos de desarrollo

## PROYECTOS REALIZADOS

Continuar con el kit de introducción de la  
empresa



# Tecnologías Usadas



Python



Docker



Css 3



Flask



Php



SSH



Nodejs



Htмл 5



Github

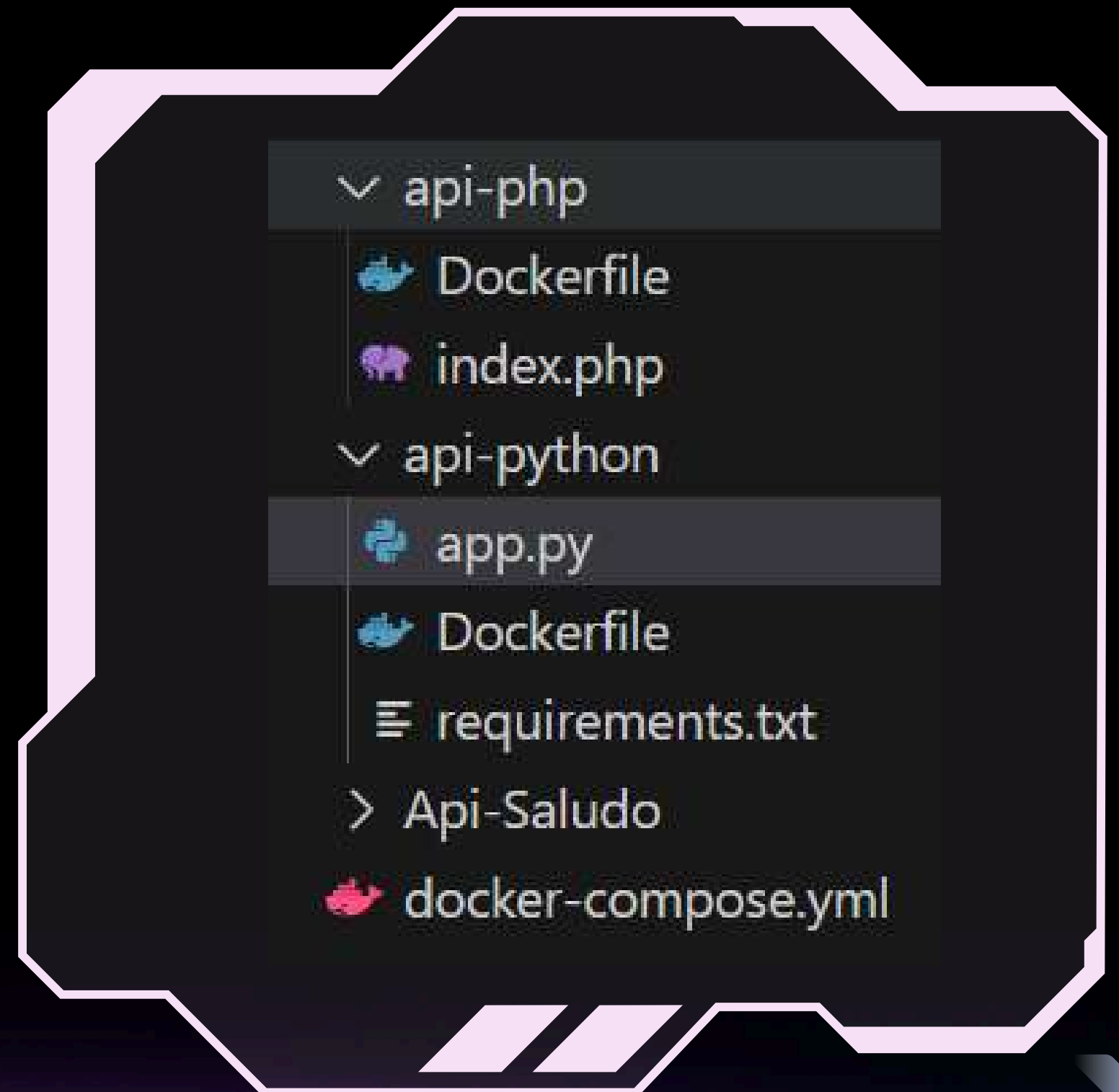




# Actividades Realizadas

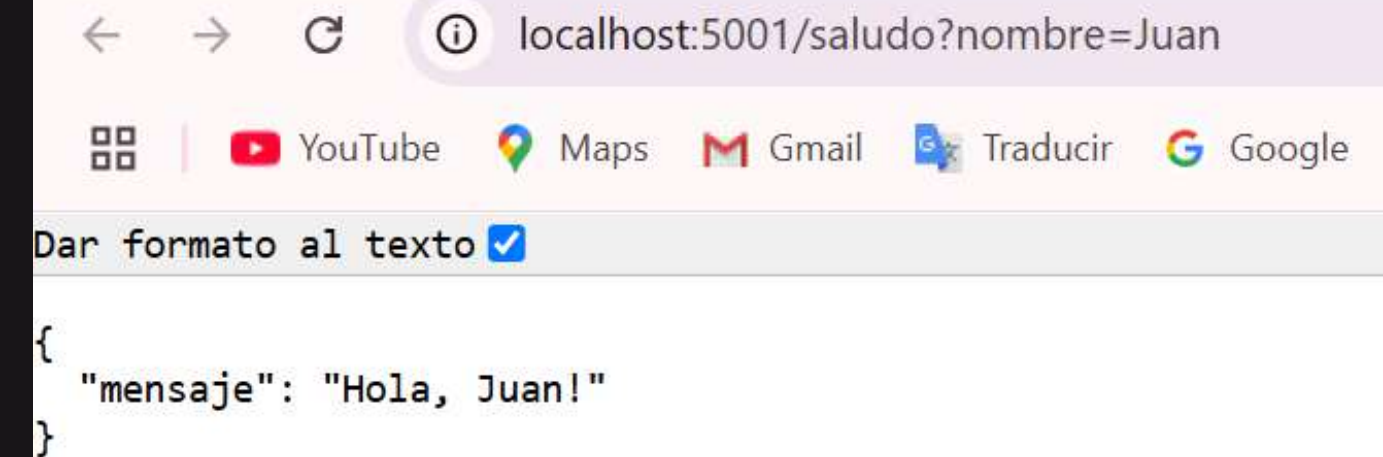
Empezamos a hacer proyectos con API's tanto en Python como en PHP, todo esto unidos con un Docker-Compose.

En tecnologías como ssh creamos un par de claves, tanto con Putty como por CMD.



# Resultado de la Actividad

Esta API lo que realiza es una contestación al usuario, funciona mandando un mensaje de parte del usuario para que la API te responde con un simple saludo a modo de prueba

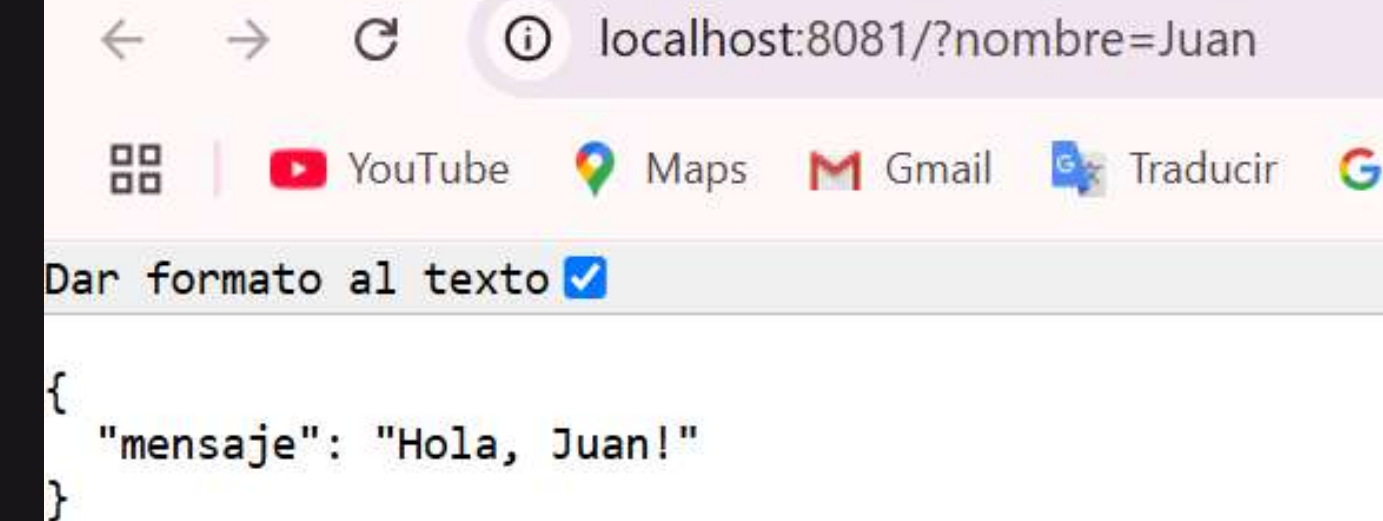


← → ↻ ⓘ localhost:5001/saludo?nombre=Juan

⌵ | YouTube Maps Gmail Traducir Google

Dar formato al texto ☒

```
{  
  "mensaje": "Hola, Juan!"  
}
```



← → ↻ ⓘ localhost:8081/?nombre=Juan

⌵ | YouTube Maps Gmail Traducir Google

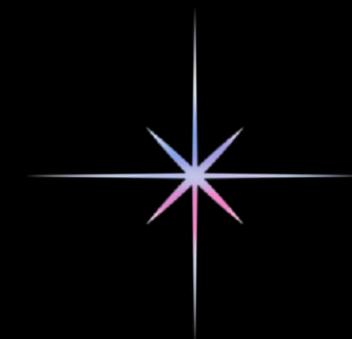
Dar formato al texto ☒

```
{  
  "mensaje": "Hola, Juan!"  
}
```



# 3°

# Semana



ACTIVIDADES REALIZADAS

Continuar creando API's tanto Python y PHP  
Crear logica backend de Login, metodos  
GET,POST etc... para obtener un CRUD  
Web y JS basico para el frontend  
Expandir el uso de Docker usando volumenes

ASIGNATURAS  
RELACIONADAS

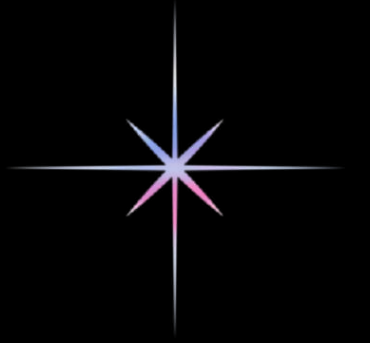
Programacion, Lenguaje de Marcas, Sistemas y  
Entorno de Desarrollo.

PROYECTOS REALIZADOS

Web basica para probar las API's



# Tecnologías Usadas



Python



Docker



Css 3



Nodejs



Php



Html 5





# Actividades Realizadas

Seguimos haciendo proyectos con API's tanto en python como en PHP, y empezamos a hacer CRUD's usando HTML y JS para recibir y enviar los datos de entrada, todo esto servido por un Docker-Compose.

htdocs

✓ api\_anios

🐘 api\_php.php

# estilo.css

<> form-php.html

<> form-python.html

✓ api\_numeros

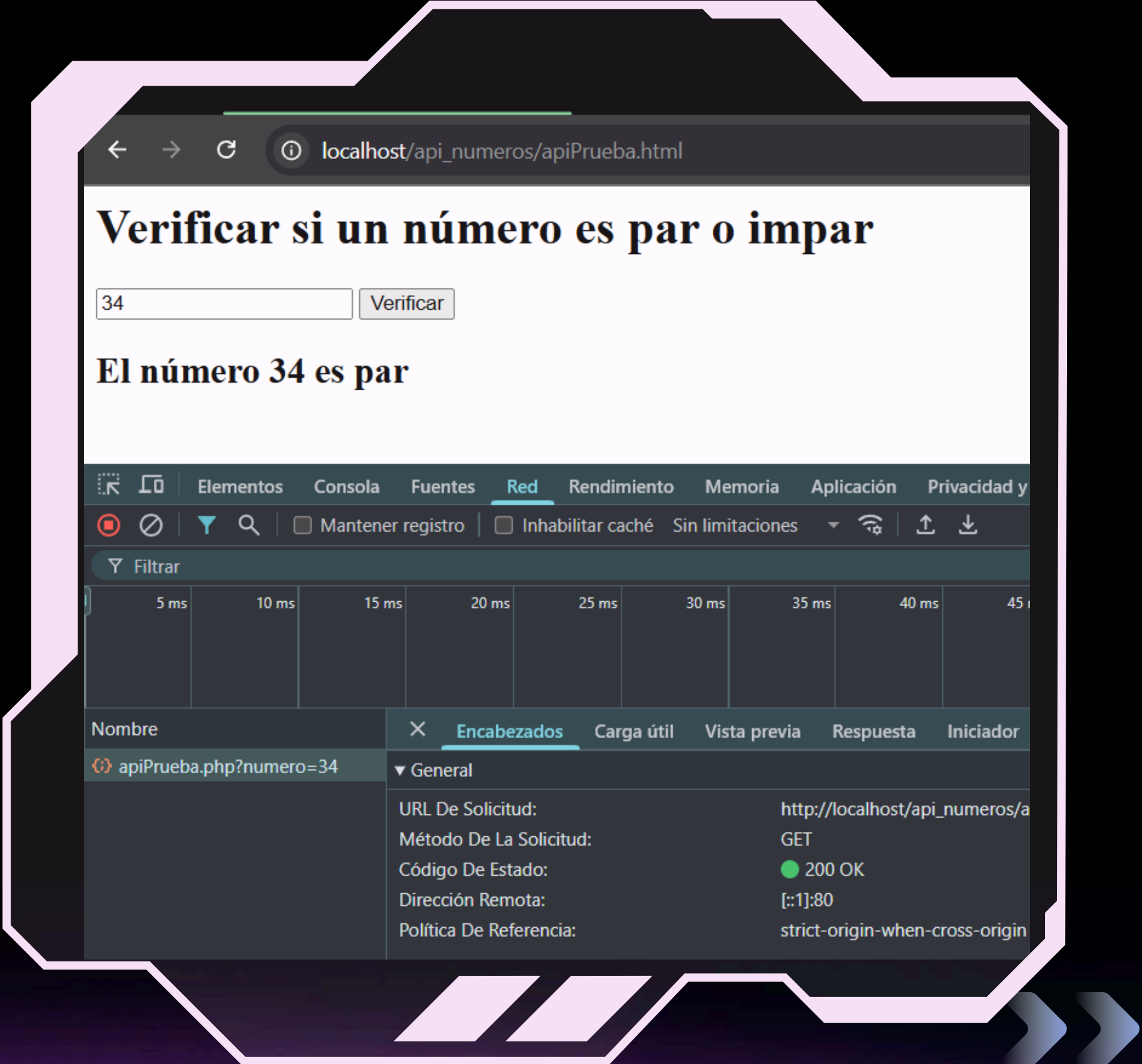
<> apiPrueba.html


🐘 apiPrueba.php

M

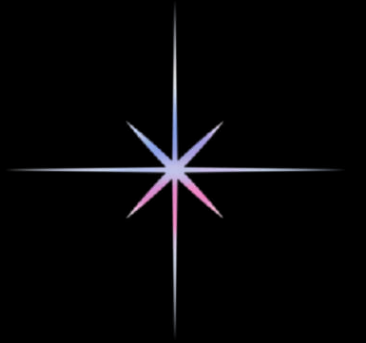
M

# Resultado de la Actividad





# 4<sup>o</sup> Semana



## ACTIVIDADES REALIZADAS

Desarrollo de CRUD sencillo tanto en Backend como Frontend, APIS's con Flask, Frontend con JS Manipulacion de BBDD a traves de llamadas a API

## ASIGNATURAS RELACIONADAS

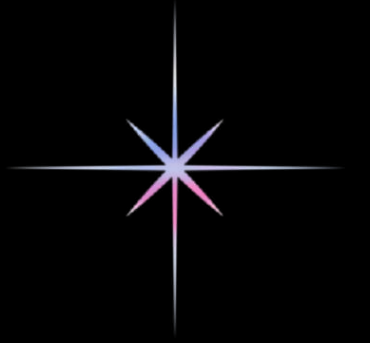
Lenguaje de Marcas, Programacion, BBDD y Entorno de Desarrollo

## PROYECTOS REALIZADOS

Ejercicios CRUD sencillos con persistencia



# Tecnologías Usadas



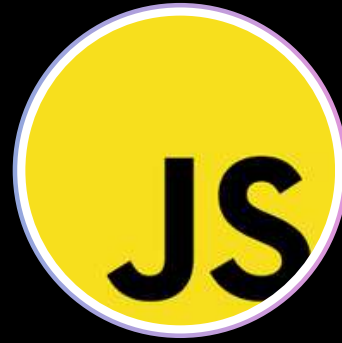
Python



Docker



Css 3



Javascript



Php



MySQL  
Workbench



Nodejs



Html 5






# Actividades Realizadas

Aquí ya empezamos a comprobar las API's con Bruno y Postman, seguimos haciendola en Php y Python, ya las unimos a una base de datos utilizando mysql haciendo que tenga persistencia.


Seguimos desplegando todo con el docker-compose.



```
▼ LIBROS-API
  > Api-Libros
  ▼ php-libros
    Dockerfile
    index.php
  ▼ python-libros
    app.py
    Dockerfile
    requirements.txt
    docker-compose.yml
    init.sql
    Libros_DB.sql
```

# Ejemplo Bruno

Este es el método Post, donde le añadimos el nombre, autor e isbn, ya que el id se autoincrementa solo.



The screenshot shows the Bruno API client interface. On the left, a sidebar lists API collections: 'Api-Libros' (expanded), 'PHP-Libros', and 'Python-Libros'. Under 'PHP-Libros', the 'POST Añadir libros' method is selected. The main panel displays the request body as a JSON object:

```
1 {  
2   "nombre": "La vuelta al mundo",  
3   "autor": "Pedro quevedo",  
4   "isbn" : "19283765"  
5 }
```

# Ejemplo Bruno

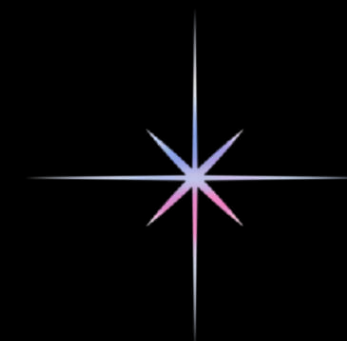
Este es el Get que hacemos para que nos dé los libros almacenados después de hacer el Post.

```
{
  "libros": [
    {
      "id": "1",
      "nombre": "La vuelta al mundo",
      "autor": "Pedro quevedo",
      "isbn": "19283765"
    }
  ]
}
```



# 5°

# Semana



## ACTIVIDADES REALIZADAS

Implementamos un login con usuarios en una BBDD, que lo utilizamos para poder iniciar sesión para poder ver las funciones de la API una vez iniciada la sesión.

## ASIGNATURAS RELACIONADAS

Programación, BBDD, Entornos, Lenguajes de Marcas.

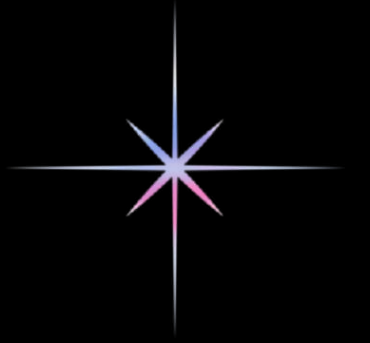
## PROYECTOS REALIZADOS

Aplicación web con login, añadir, eliminar usuarios, creación de operaciones CRUD en la API.





# Tecnologías Usadas



Python



Docker



Css 3



Javascript



Php



MySQL Workbench



Nodejs



Html 5



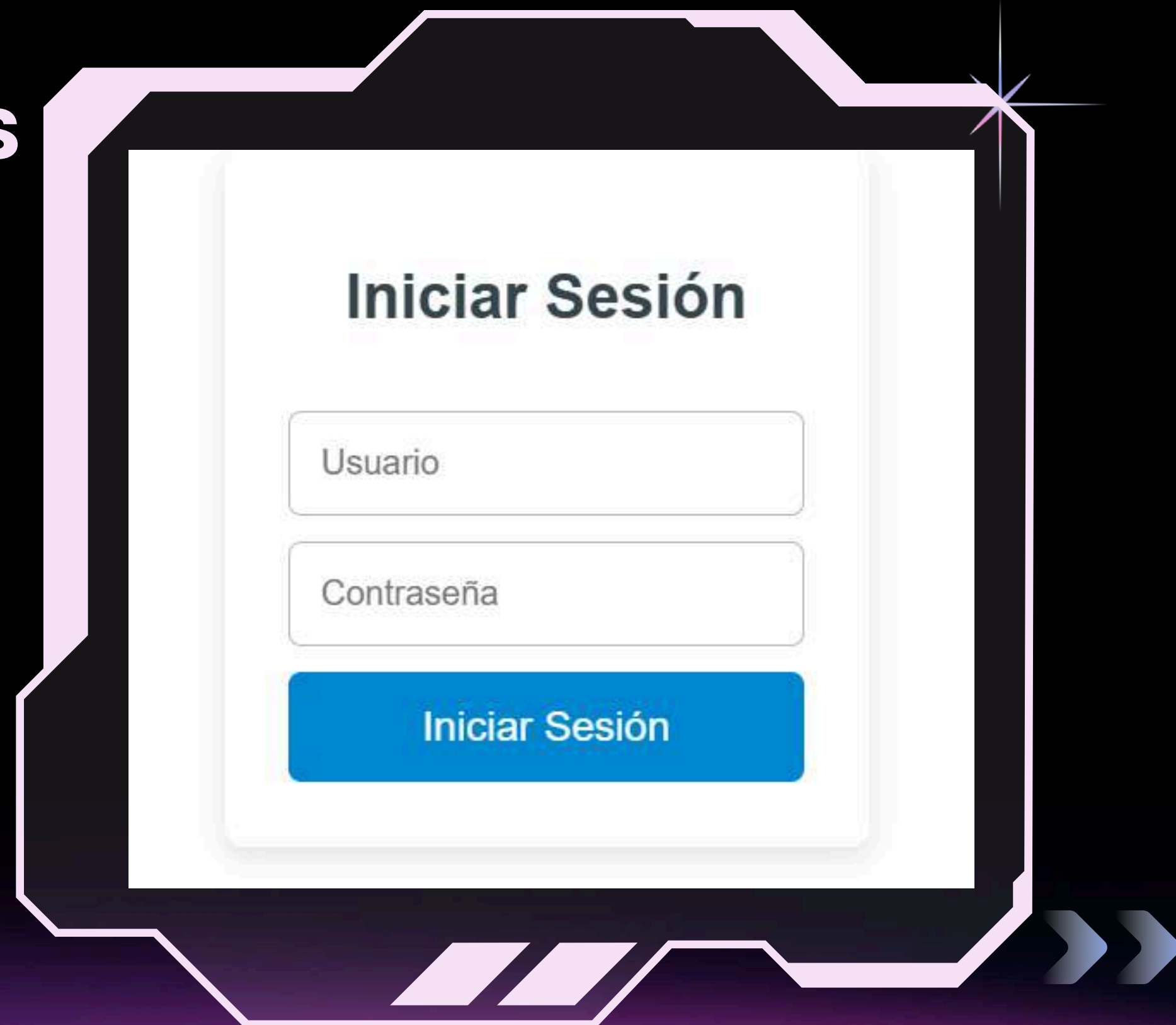
Github



# Actividades Realizadas

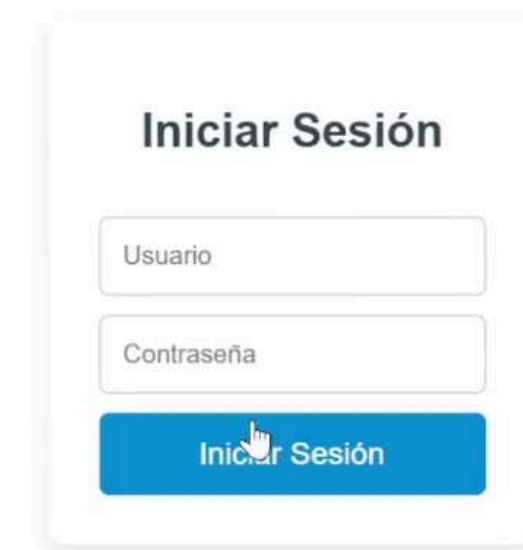
En esta API, realizamos una gestión de biblioteca, con varios lenguajes como python o php, además de conectar una web con una base de datos en tiempo real con un login, para poder hacer diferentes operaciones con los productos. Ya sea añadir, eliminar o modificar, esos productos.

Desplegando docker para poder arrancar todo desde un mismo contenedor.



# Resultado de la Actividad

Esta api una vez se ponga el usuario y la contraseña que esta en la base de datos, te muestra las diferentes operaciones con los libros. Además, cada actividad de esa API esta registrada en tiempo real en la base de datos para que quede un registro.




Iniciar Sesión

Usuario

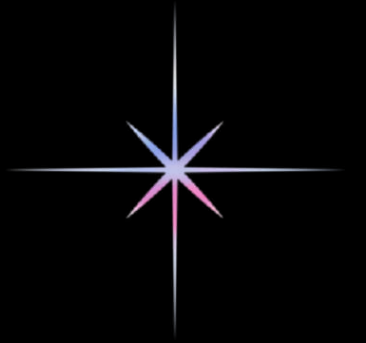
Contraseña

Iniciar Sesión





# 6<sup>o</sup> Semana



## ACTIVIDADES REALIZADAS

Mejora de la aplicacion de procesamiento de lenguaje natural, permitiendo el Español y uso del algoritmo de Gale-Church para el alineamiento de el texto

## ASIGNATURAS RELACIONADAS

Programacion, Lenguajes de Marcas, Sistemas y Entorno de desarrollo.

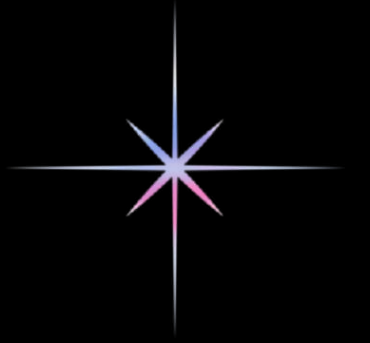
## PROYECTOS REALIZADOS

Apliacion web que tokeniza y etiqueta texto automaticamente en Ingles y Español





# Tecnologías Usadas



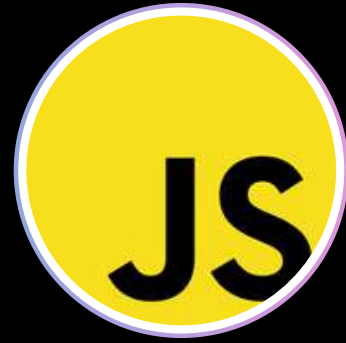
Python



Docker



Css 3



Javascript



Php



NLTK



Nodejs



Htмл 5

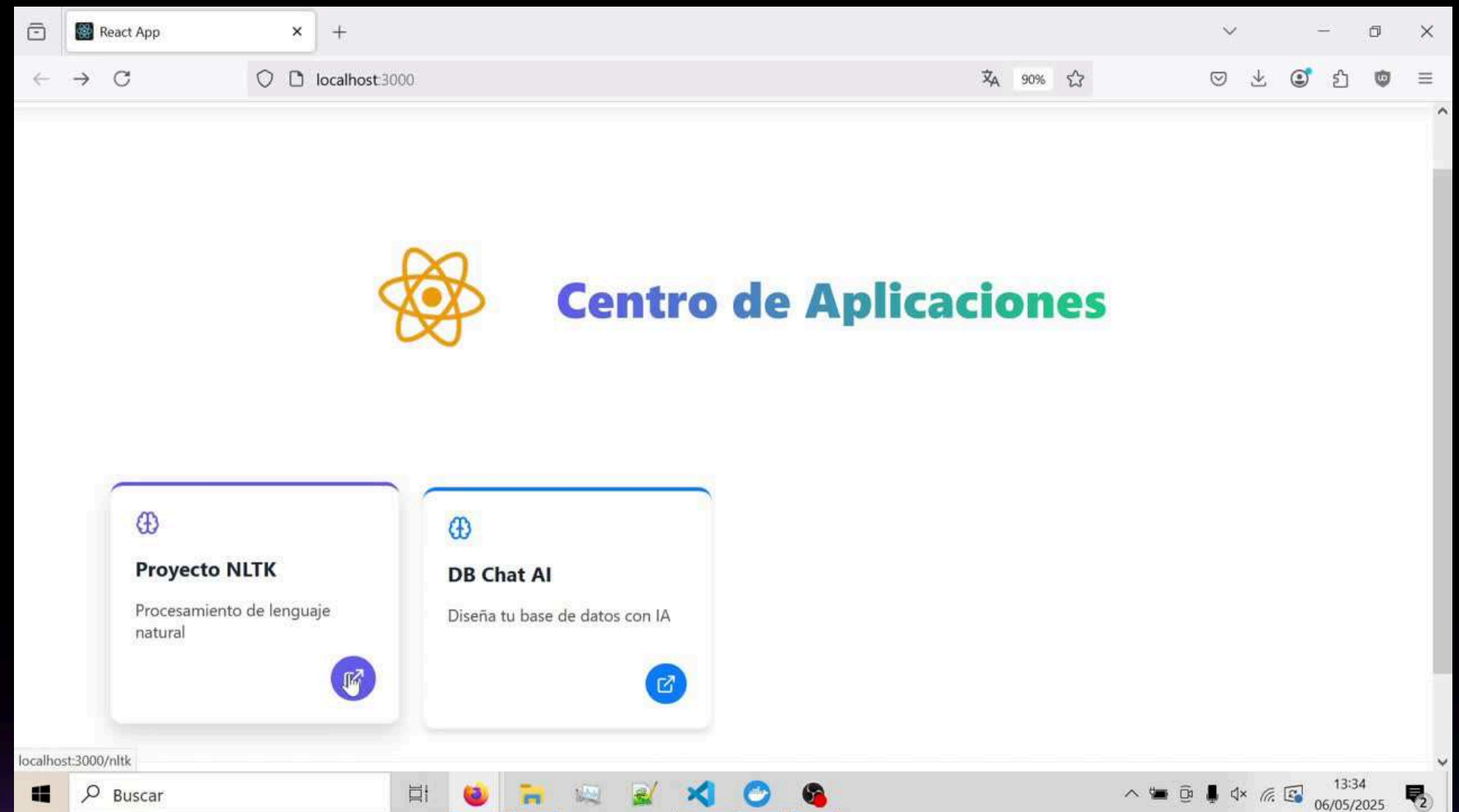


Spacy



# Actividades Realizadas

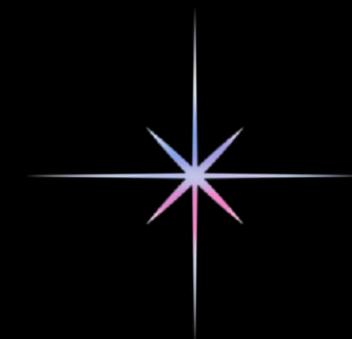
Aplicacion de procesamiento (Segmentado y etiquetado) de lenguaje natural en tiempo real, tanto en Ingles y Español y uso de el algoritmo de Gale-Church para el alineamiento de el texto paralelo.





# 7º

# Semana



## ACTIVIDADES REALIZADAS

Se nos dio una charla de introducción al funcionamiento interno de los modelos de lenguaje y nos pusieron ejercicios en Jupyter

## ASIGNATURAS RELACIONADAS

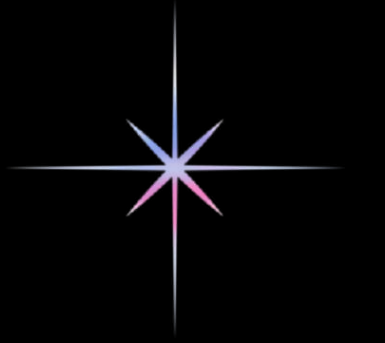
Sistemas, Programación y Entornos de Desarrollo

## PROYECTOS REALIZADOS

Ejercicios de entrenamiento de modelos de lenguaje



# Tecnologías Usadas



Python



Jupyter



SSH



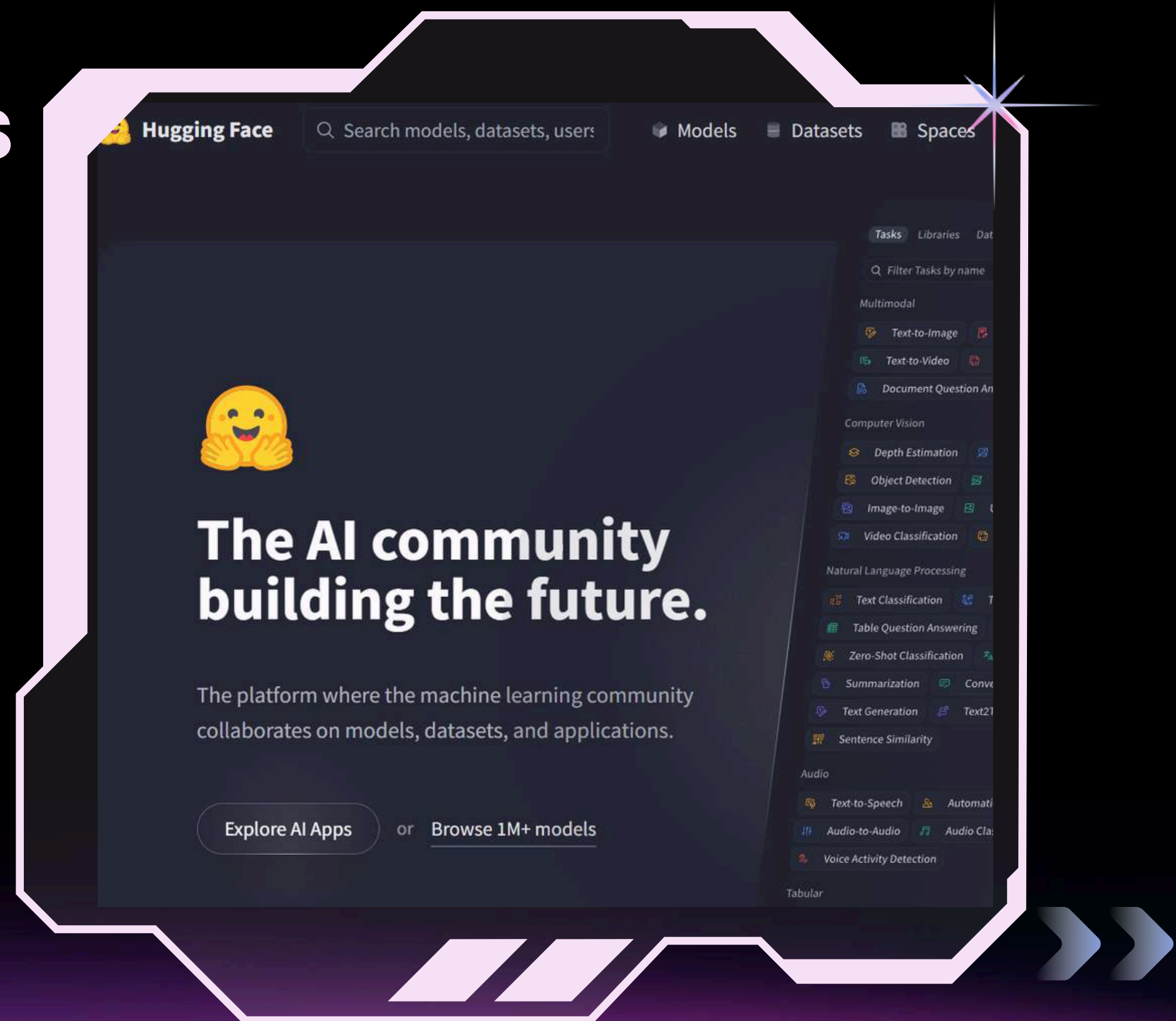
Visual Studio  
Code



# Actividades Realizadas

Nos pusieron ejercicios para familiarizarnos con el entorno Jupyter y la los modelos de IA.

El objetivo era corregir los errores en el código y conseguir entrenar a un modelo de reconocimiento de emociones y que las respuestas fueran consecuentes con el tono del texto introducido





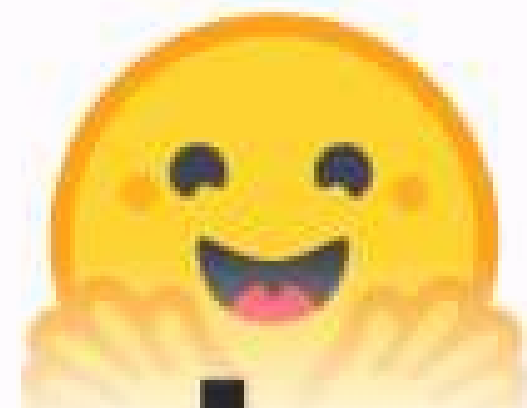
# Librerías utilizadas



**Transformers**  
Getting Started



**Datasets**  
[huggingface.co/datasets](https://huggingface.co/datasets)

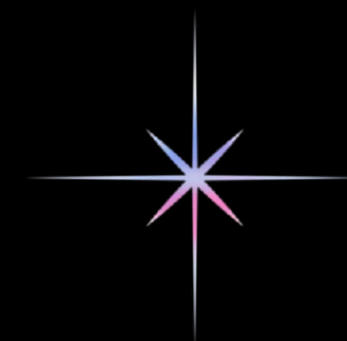


**Evaluate**  
Documentation





# Librerías de HuggingFace



TRANSFORMERS

Transformers es una biblioteca de modelos preentrenados de procesamiento de lenguaje natural, visión artificial, audio y multimodales para inferencia y entrenamiento

DATASETS

Datasets es una biblioteca que facilita el acceso y el intercambio de conjuntos de datos para tareas de audio, visión artificial y procesamiento del lenguaje natural (PLN).

EVALUATE

Una biblioteca para evaluar fácilmente modelos y conjuntos de datos de aprendizaje automático



# Resultado de la actividad

El modelo da una reseña que aumenta en estrellas en función de si cree que esta es positiva y disminuye si esta es negativa

## 5. Prueba

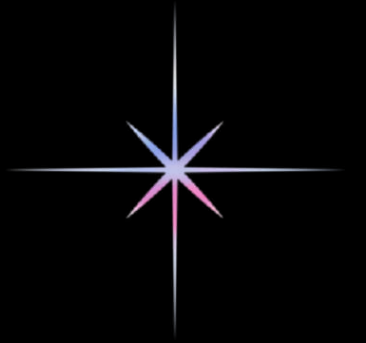
```
test_text = "The customer support was excelent. They treated me very well"
features = tokenizer(test_text, return_tensors="pt", padding="max_length",
input_values = features.input_ids.to(device)
attention_mask = features.attention_mask.to(device)
pred_logits = model(input_ids=input_values, attention_mask=attention_mask)
pred_results = torch.softmax(pred_logits, dim=1).tolist()[0]
classes = ["1 estrella", "2 estrellas", "3 estrellas", "4 estrellas", "5 e

for i in range(len(classes)):
    print(f"{classes[i]}: {int(round(pred_results[i] * 100))}%")

1 estrella: 0%
2 estrellas: 0%
3 estrellas: 1%
4 estrellas: 7%
5 estrellas: 91%
```



# 8<sup>o</sup> Semana



## ACTIVIDADES REALIZADAS

Desarrollamos una IA de STT en entorno Jupyter.  
Empezar la creacion de la presentacion.

## ASIGNATURAS RELACIONADAS

Programación, digitalización.

## PROYECTOS REALIZADOS

IA capaz de convertir el habla en texto.





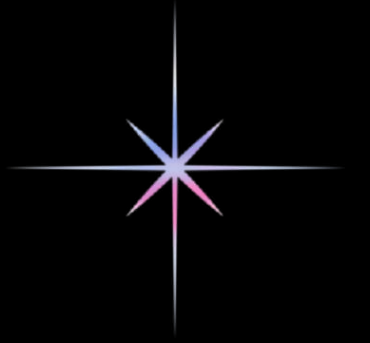
M  
U  
E  
S  
T  
R  
A

E  
N  
T  
O  
R  
N  
O





# Tecnologías Usadas



Python



Docker



HuggingFace



jupyter

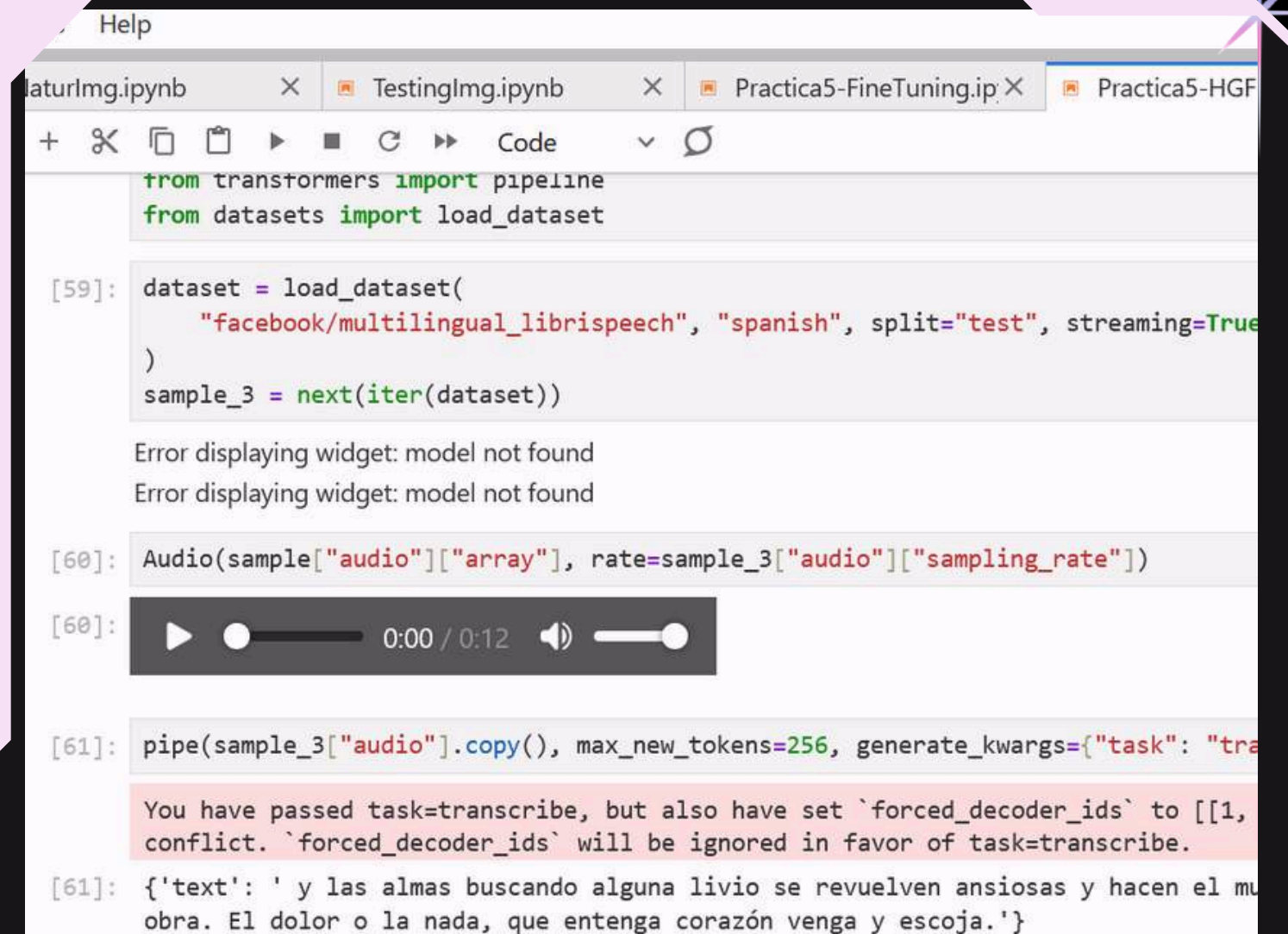


Whisper



# Actividades Realizadas

La actividad, descarga un modelo pre-entrenado con miles de datos, pero afinado con conjunto de datos para cada idioma, en el ejemplo, mostramos la ejecución de un modelo afinado para el español, le pasamos un audio y lo transcribe.



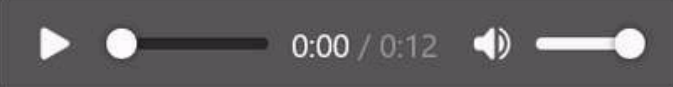
The screenshot shows a Jupyter Notebook with several tabs: 'aturlmg.ipynb', 'TestingImg.ipynb', 'Practica5-FineTuning.ip', and 'Practica5-HGF'. The active tab is 'Practica5-FineTuning.ip'. The code in the notebook is as follows:

```
from transformers import pipeline
from datasets import load_dataset

[59]: dataset = load_dataset(
        "facebook/multilingual_librispeech", "spanish", split="test", streaming=True
    )
    sample_3 = next(iter(dataset))

Error displaying widget: model not found
Error displaying widget: model not found

[60]: Audio(sample["audio"]["array"], rate=sample_3["audio"]["sampling_rate"])

[60]: 

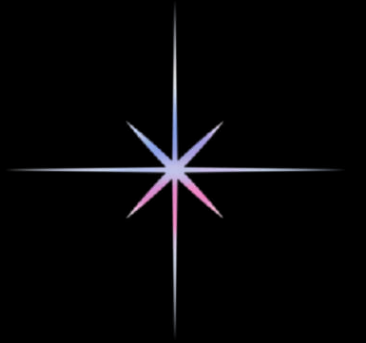
[61]: pipe(sample_3["audio"].copy(), max_new_tokens=256, generate_kwargs={"task": "tra

You have passed task=transcribe, but also have set `forced_decoder_ids` to [[1,
conflict. `forced_decoder_ids` will be ignored in favor of task=transcribe.

[61]: {'text': ' y las almas buscando alguna livio se revuelven ansiosas y hacen el mu
obra. El dolor o la nada, que entenga corazón venga y escoja.'}
```



# 9<sup>o</sup> Semana



## ACTIVIDADES REALIZADAS

Desarrollamos una aplicacion web de alineacion y edicion de corpus.

## ASIGNATURAS RELACIONADAS

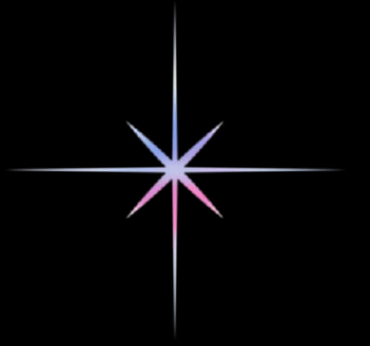
Programacion, lenguaje de marcas,  
digitalizacion.

## PROYECTOS REALIZADOS

Aplicacion web para alineación, edición de  
corpus, herramienta integrada en la empresa.



# Tecnologías Usadas



Python



Docker



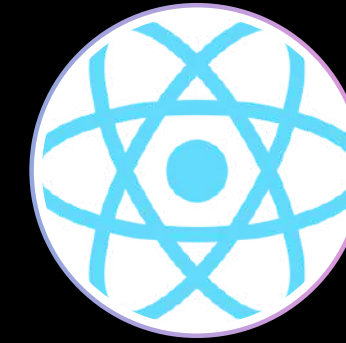
Css 3



Javascript



Hugging  
Face



React



Nodejs



Html 5



npm

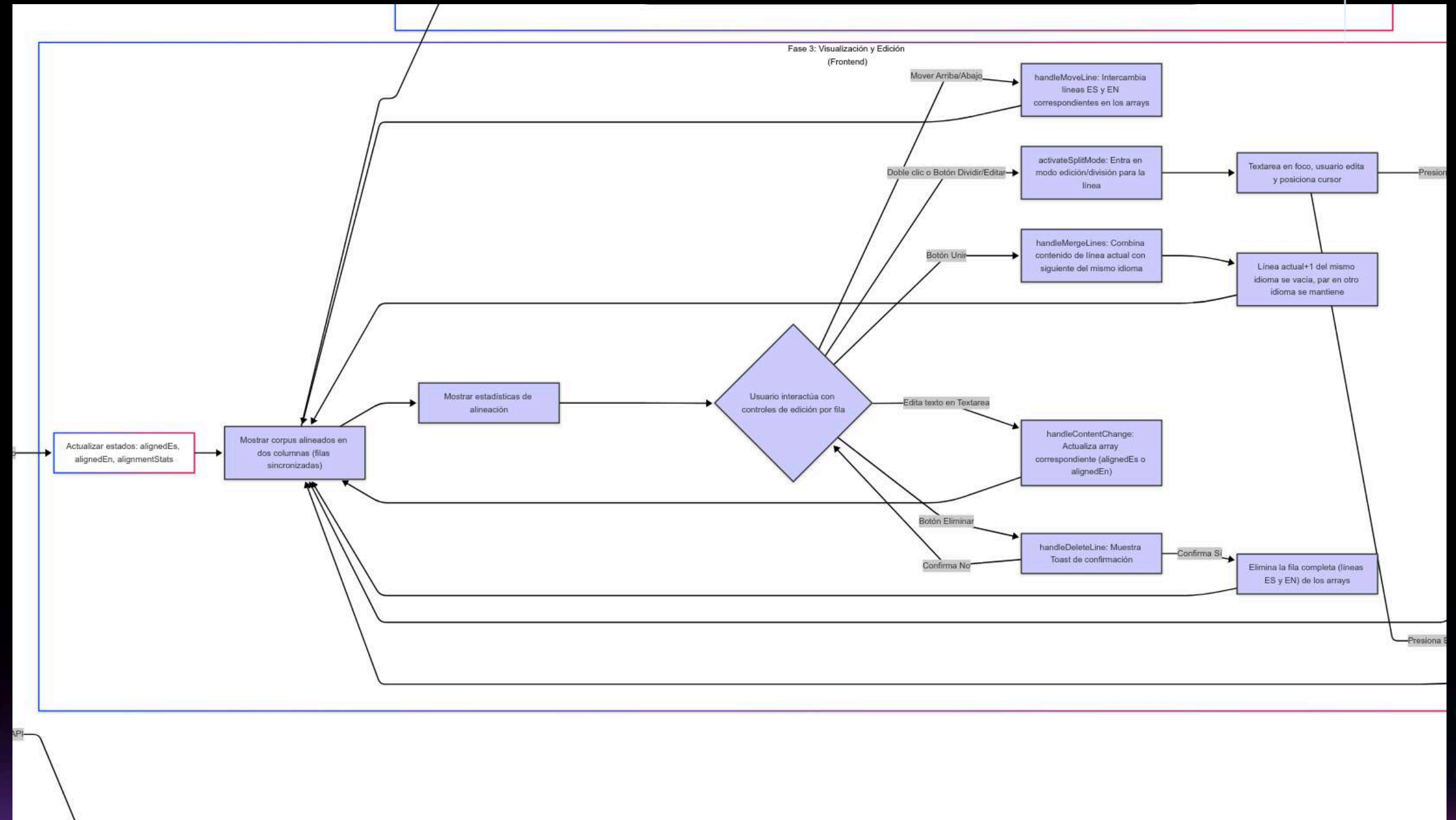




# Actividades Realizadas

Diagrama para la página de el alineador de corpus usando React, la API de alineamiento de corpus esta echa con Python.

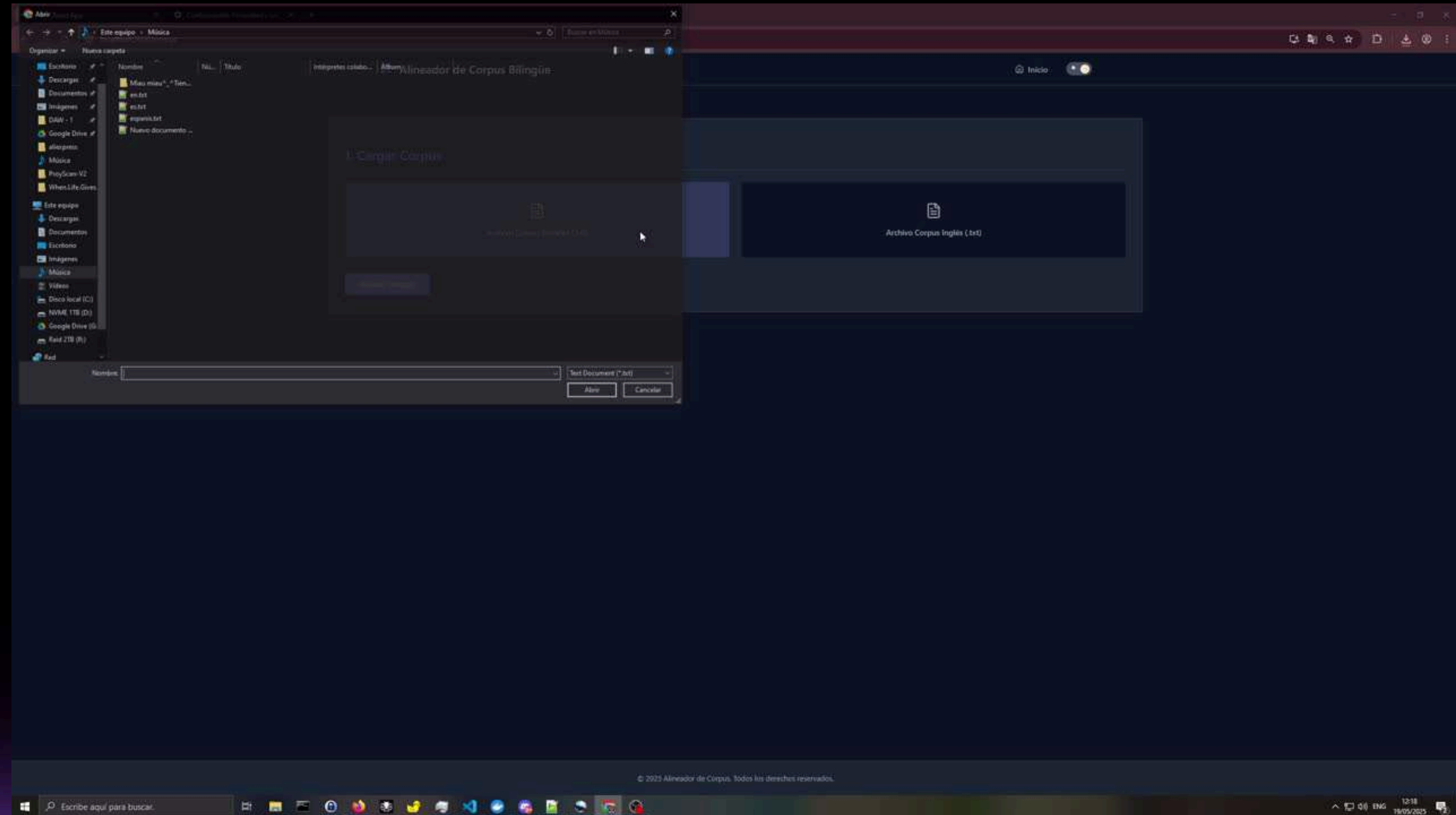
Luego se decora la pagina con css y algunas funciones de React para las animaciones o iconos.





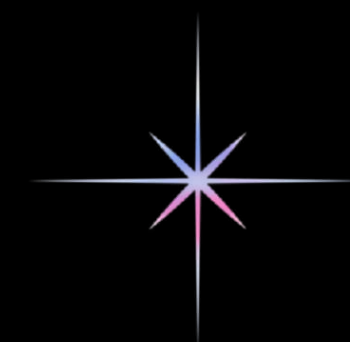
# Resultado de la Actividad

Alienador de corpus con modelo de IA, permite la edición en vivo de los corpus





# 10<sup>o</sup> Semana



## ACTIVIDADES REALIZADAS

Implementar una IA de análisis de sentimiento multilinguaje (Chino) con calificación de 1 a 5 estrellas

## ASIGNATURAS RELACIONADAS

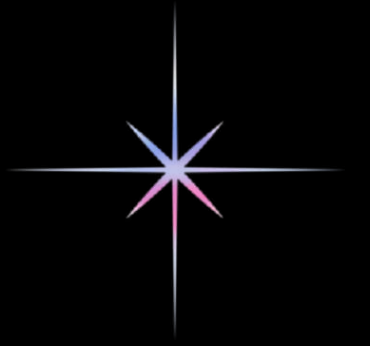
Programación, lenguaje de marcas (no framework), digitalización

## PROYECTOS REALIZADOS

Aplicación de análisis de sentimiento multilinguaje.



# Tecnologías Usadas



Python



Docker



Css 3



Javascript



Hugging  
Face



HTML 5



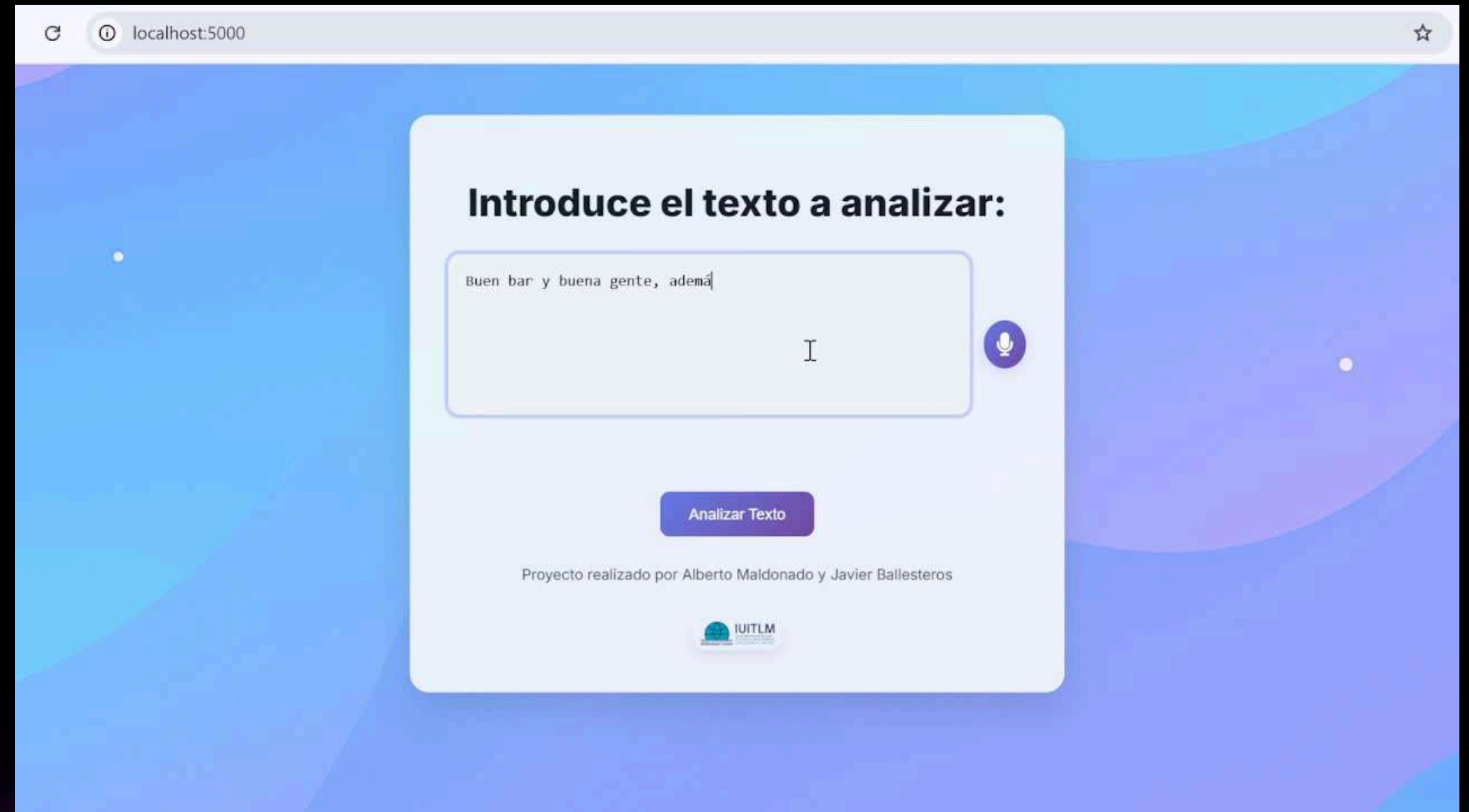
Nodejs



# Resultado de la Actividad

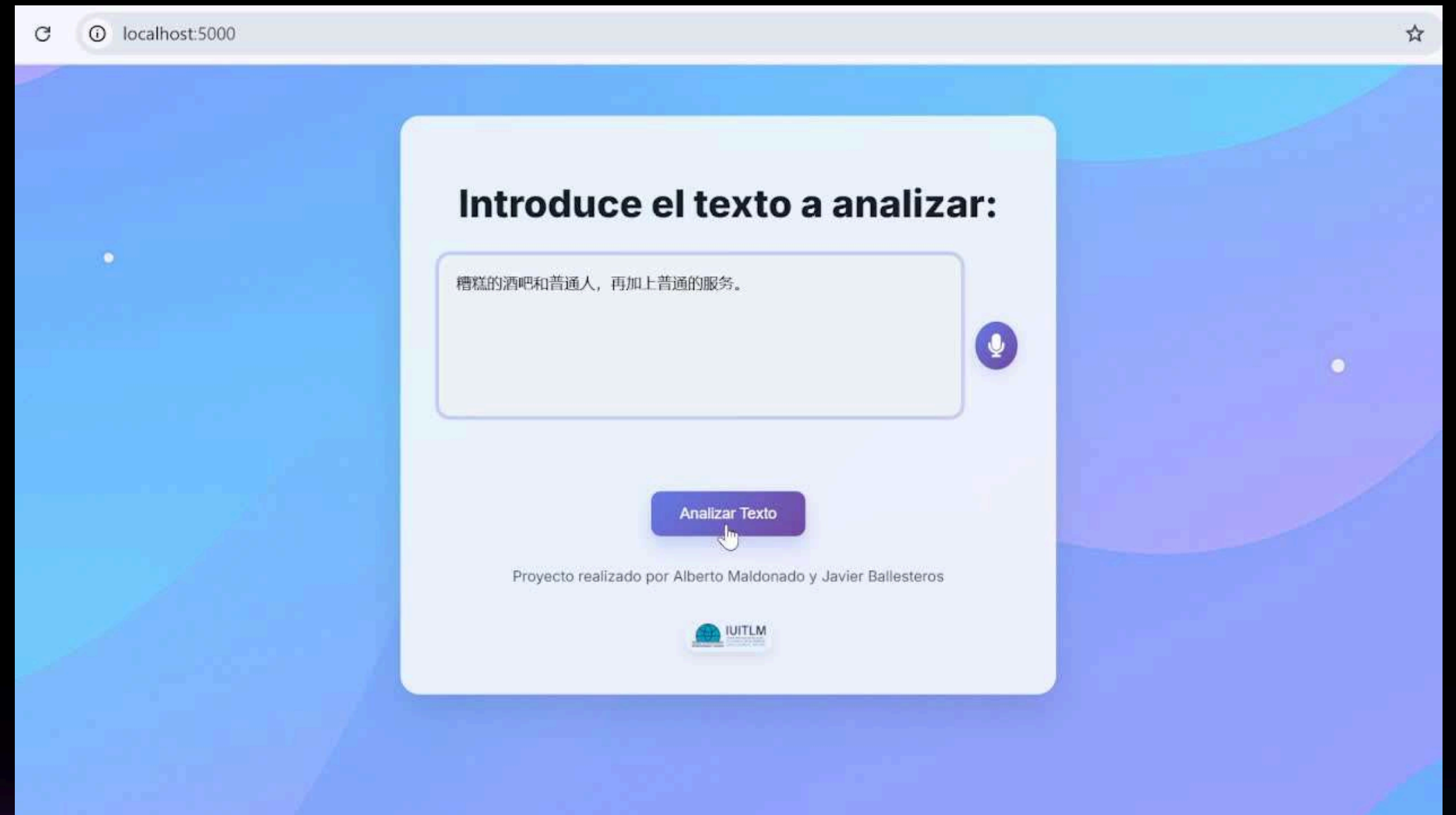
Esta API hace un análisis de sentimientos con valoración de 1 a 5 estrellas.

Es capaz de analizar texto en Español, Inglés y en Chino. Así como también es capaz de analizar grabaciones de audio en los diferentes idiomas nombrados anteriormente



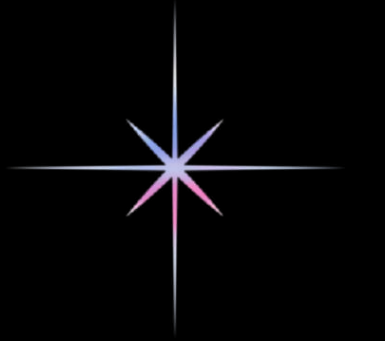
# Resultado de la Actividad

Mostración de análisis en Chino.





# Opinión de la empresa



## Opinión de la empresa

Desde nuestra experiencia, la empresa se ha mostrado como un lugar dinámico y con ganas de hacer las cosas bien. Hemos notado un ambiente de trabajo bastante positivo, donde la gente esta a gusto y colabora. Además, nos ha parecido que apuestan por dar oportunidades y eso es algo que valoramos mucho.

## Opinión de la tarea

En cuanto a las tareas que hemos realizado, nos ha parecido interesantes y variadas. Hemos tenido la oportunidad de poner en práctica lo que sabemos y también hemos aprendido cosas nuevas. Aunque a veces ha habido momentos de más presión, en general hemos sentido que nuestro trabajo era útil y tenía un sentido dentro del equipo.





# GRACIAS

JAVIER BALLESTEROS | ALBERTO MALDONADO  
RUBÉN SÁNCHEZ | MATEO R. MAYGER