



UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

Magic Personality Matcher (MPM)

**Grupo Onyrix
(ABP-2)**

DNI: 77014895Q
DNI: 44664969G
DNI: 53861988J
DNI: 39512816N

NOMBRE: GUIMERÁNS CORDEIRO, JESÚS
NOMBRE: GUIJARRO VILLAVICENCIO, TANIA
NOMBRE: VELOSO BOUBETA, JAVIER
NOMBRE: VILA COMESAÑA, MARIO

APRENDIZAJE BASADO EN PROYECTOS 2023-2024



Índice

Revisión	4
Glosario de términos	4
Descripción	4
Metodología	4
Sprint 1	4
Objetivos	4
Historias de usuario	4
Diseño de Arquitectura	5
Sprint 2	5
Objetivos	5
Tareas realizadas	5
Sprint 3	5
Objetivos	5
Tareas realizadas	5
Sprint 4 no realizado	5
Sprint 5	5
Tareas realizadas	5
Meta	5
Objetivos técnicos	6
Modelo de IA para análisis de personalidad	6
Aprendizaje de las valoraciones de los usuarios	6
Poblar una base de datos de personalidades de personajes de series	6
Desarrollo de una interfaz de usuario	6
Historias de usuario	7
Página principal	7
Botones de valoración	7
Selección de serie	8
Resultado de recomendación	8
Funcionalidades	8
Interfaz web	9
Modelo de IA para predecir personalidades de texto	9
Backend	9
Base de datos	9
El poblador de la base de datos	9
Diagrama de arquitectura	10
Diagrama de secuencia de sistema	11
Poblado de la Base de Datos	12
Marco de ampliación	12
Pruebas	13
Bibliografía	13
Anexo I - Versión inicial	14
Descripción	14

Meta	14
Objetivos técnicos	14
Recopilación de datos	14
Modelo de IA para análisis de personalidad	15
Poblar una base de datos de personalidades de personajes de series	15
Desarrollo de una interfaz de usuario	15
Historias de usuario	15
Inicio de sesión con X	15
Selección de serie	16
Resultado de recomendación	16
Funcionalidades	17
Interfaz web	17
Modelo de IA para predecir personalidades de texto	17
Backend	17
Base de datos	17
El poblador de la base de datos	17
Glosario de términos	18
Diagrama de arquitectura	18
Diagrama entidad-relación	18
Poblado de la Base de datos	19
Marco de ampliación	19
Manual de uso	21
Estructura del proyecto	21
Pasos previos	22
Modelo de IA	22
Poblado de la base de datos	22
Ejecución	22
Backend	23
Notebooks	23

Índice de figuras

Diagrama de arquitectura	10
Diagrama de secuencia de sistema	11
Diagrama de arquitectura (antiguo)	18
Diagrama entidad-relación	18

Revisión

El proyecto sufrió cambios integrales que forzaron un rediseño en la aplicación, redefiniendo objetivos, historias de usuario y el diagrama de arquitectura. Esto se debe a los recientes cambios la API de X (anteriormente Twitter), que pasó de permitir extraer posts hasta cierta cantidad de manera gratuita a requerir una suscripción mensual de al menos 100\$.

Glosario de términos

[Ver Glosario de términos](#) (no cambió desde la versión inicial).

Descripción

¿No sabes qué decir cuando te preguntan cómo eres? ¿Quieres saber con qué personas tienes mayor compatibilidad? Muchas personas se preguntan qué tipo de personalidad tienen, sin embargo, hoy en día es complicado conocer este hecho sobre nosotros mismos. Esto es un factor influyente a la hora de relacionarse con el entorno y con las personas que nos rodean, afectando en la manera en la que nos relacionamos con nuestros amigos o nuestra familia, o incluso con nosotros mismos.

No esperes más y haz nuestro test online gratis, 100% eficaz con resultados inmediatos, y te diremos con qué amigos te puedes llevar mejor, y cómo será tu futura pareja ideal.

El proyecto consiste en una aplicación capaz de clasificar un texto en una personalidad del índice de Myers-Briggs y ver qué personajes de series tienen esa misma personalidad. Ya que no es posible obtener posts de redes sociales, el usuario selecciona por sí mismo los textos a analizar.

Metodología

La metodología de trabajo utilizada para realizar el proyecto es SCRUM, por lo que el proyecto se realizará a base de incrementos (Sprints), en concreto se realiza con 5 sprints

Sprint 1

Objetivos

- Revisión y comprensión detallada de los requisitos del proyecto.
- Identificación de los objetivos técnicos y funcionales del proyecto.
- Establecimiento de los hitos y entregables clave para este script.

Historias de usuario

- Identificación de las historias de usuario para el proyecto.
- Establecimiento de criterios de aceptación para cada historia de usuario.

Diseño de Arquitectura

- Revisión y establecimiento de la arquitectura general del proyecto.

Sprint 2

Objetivos

- Desarrollo del modelo de Análisis de personalidad.
- Población de la base de datos de personalidades.

Tareas realizadas

- Entrenamiento del modelo de IA utilizando técnicas de análisis de texto.

Sprint 3

Objetivos

- Desarrollo de la Interfaz de Usuario.
- Implementación del prototipo de la aplicación.
- Implementación de una interfaz que permita conectarse con X(twitter).

Tareas realizadas

- Elaboración del manual de uso.
- Creación del webscraper de la web 'Personality Database' y población de la base de datos con información de las personalidades.
- Desarrollo de la sección de resultado que muestra los personajes recomendados, incluyendo sus nombres, imágenes y una breve descripción.

Sprint 4 no realizado

Sprint 5

Tareas realizadas

- Desarrollo de la interfaz web que permita conectarse con el servidor.
- Cambio de las historias de usuario relacionadas con X, por problemas externos.
- Implementación de un sistema de valoración.

Meta

El objetivo principal es crear un modelo capaz de clasificar un texto según la personalidad de su autor. Este debe ser capaz de reconocer patrones que permitan diferenciar las personalidades y así clasificar los textos.

Objetivos técnicos

Modelo de IA para análisis de personalidad

Se debe desarrollar un modelo de IA capaz de determinar la personalidad según el índice de Myers-Briggs, de forma que, reciba como entrada un texto y determine de los 16 tipos cuál corresponde al autor del texto.

Historias de usuario:

- [Resultado de recomendación](#)

Funcionalidades:

- [Modelo de IA para predecir personalidades de texto](#)
- [Backend](#)

Aprendizaje de las valoraciones de los usuarios

Se deben recoger valoraciones positivas y negativas de los usuarios sobre las clasificaciones realizadas que permitan mejorar el modelo de IA.

Historias de usuario:

- [Resultado de recomendación](#)

Funcionalidades:

- [Backend](#)
- [Interfaz web](#)

Poblar una base de datos de personalidades de personajes de series

Usando los datos de la web [Personality Database](#) para obtener información sobre las personalidades de algunas series populares.

Historias de usuario:

- [Selección de serie](#)
- [Resultado de recomendación](#)

Funcionalidades:

- [El poblador de la base de datos](#)
- [Base de datos](#)

Desarrollo de una interfaz de usuario

Debe tener una interfaz de usuario que permita acceder con una cuenta de X y muestre el resultado, dando la opción de compartirlo en X.

Historias de usuario:

- [Selección de serie](#)
- [Resultado de recomendación](#)

Funcionalidades:

- [Interfaz web](#)

Historias de usuario

Página principal

Como usuario, se puede introducir el texto de entrada y mediante un botón clasificarlo según su personalidad. Esta página también debe contener ejemplos con posts de X y su clasificación.

Criterios de Aceptación:

1. Cuadro de texto en el que el usuario pueda escribir
2. Botón de clasificar el texto
3. Ejemplos de clasificación de posts de X

Objetivos técnicos:

- [Modelo de IA para analisis de personalidad](#)
- [Desarrollo de una interfaz de usuario](#)

Funcionalidades:

- [Interfaz web](#)
- [Backend](#)

Botones de valoración

Como usuario, se debe poder valorar el resultado de la clasificación con botones de valoración positiva o “pulgar arriba” o valoración negativa “pulgar abajo”.

Criterios de Aceptación:

4. Debe haber un botón de pulgar arriba 👍
5. Debe haber un botón pulgar abajo 👎
6. Al hacer click se registran y se añaden al modelo

Objetivos técnicos:

- [Modelo de IA para analisis de personalidad](#)
- [Desarrollo de una interfaz de usuario](#)

Funcionalidades:

- [Interfaz web](#)
- [Backend](#)

Selección de serie

Como usuario, deseo tener la posibilidad de elegir una serie de mi interés de una lista proporcionada. Esto me permitirá obtener recomendaciones de personajes de la serie seleccionada.

Criterios de Aceptación:

1. Debe mostrarse una lista de series disponibles en la aplicación.
2. El usuario debe poder interactuar con la lista y seleccionar una serie de su elección.

Objetivos técnicos:

- [Poblar una base de datos de personalidades de personajes de series](#)
- [Desarrollo de una interfaz de usuario](#)

Funcionalidades:

- [Interfaz web](#)
- [Backend](#)
- [Base de datos](#)

Resultado de recomendación

Como usuario, quiero ver el resultado de mi análisis de personalidad y cómo se relaciona con los personajes de la serie que he elegido. También deseo poder compartir este resultado en X.

Criterios de Aceptación:

1. Al finalizar el análisis, la aplicación debe mostrar el resultado.
2. El resultado debe incluir el nombre del personaje recomendado, una imagen de este y una breve descripción de los índices de Myers-Briggs detectados.
3. Debe existir una opción para compartir el resultado en la plataforma X.

Objetivos técnicos:

- [Desarrollo de una interfaz de usuario](#)
- [Modelo de IA para análisis de personalidad](#)
- [Poblar una base de datos de personalidades de personajes de series](#)

Funcionalidades:

- [Interfaz web](#)
- [Modelo de IA para predecir personalidades de texto](#)
- [Base de datos](#)

Funcionalidades

El proyecto se divide en 5 componentes y se desplegarán usando contenedores Docker.

[Ver diagrama de arquitectura](#)

Interfaz web

Se realizará usando Next.js. Será una página con 3 rutas, la página principal tendrá un cuadro de texto y ejemplos con posts de X, al hacer click en el botón de clasificar se pasará a la siguiente página que mostrará el resultado, permitirá realizar valoraciones y una lista de películas para ver en la tercera y última página los personajes con su personalidad de una serie concreta.

Modelo de IA para predecir personalidades de texto

Se realizará en un Notebook de Google Colab usando las librerías de Python *sckit-learn* y *NLTK* (Natural Language Toolkit). Los detalles de su implementación se encuentran en el notebook anexo.

Backend

Se encarga de responder a las peticiones de la interfaz web, ejecutando el modelo de IA y coteja el resultado con la base de datos para devolver al usuario los personajes su misma personalidad.

Este componente se programará en Python usando el framework [FastAPI](#) y para la comunicación con la base de datos usaremos el ORM [SQLAlchemy](#).

Base de datos

Será una base de datos relacional, con el sistema gestor de bases de datos MySQL.

[Ver diagrama de entidad relación](#) (no cambió desde la versión inicial)

El poblador de la base de datos

Será un programa en Python, que usando el entorno [Selenium WebDriver](#) para Python, se hará web scraping a la página web de [Personality Database](#) de la que obtendrá los datos de las distintas personalidades de diferentes personajes de series y películas.

Diagrama de arquitectura

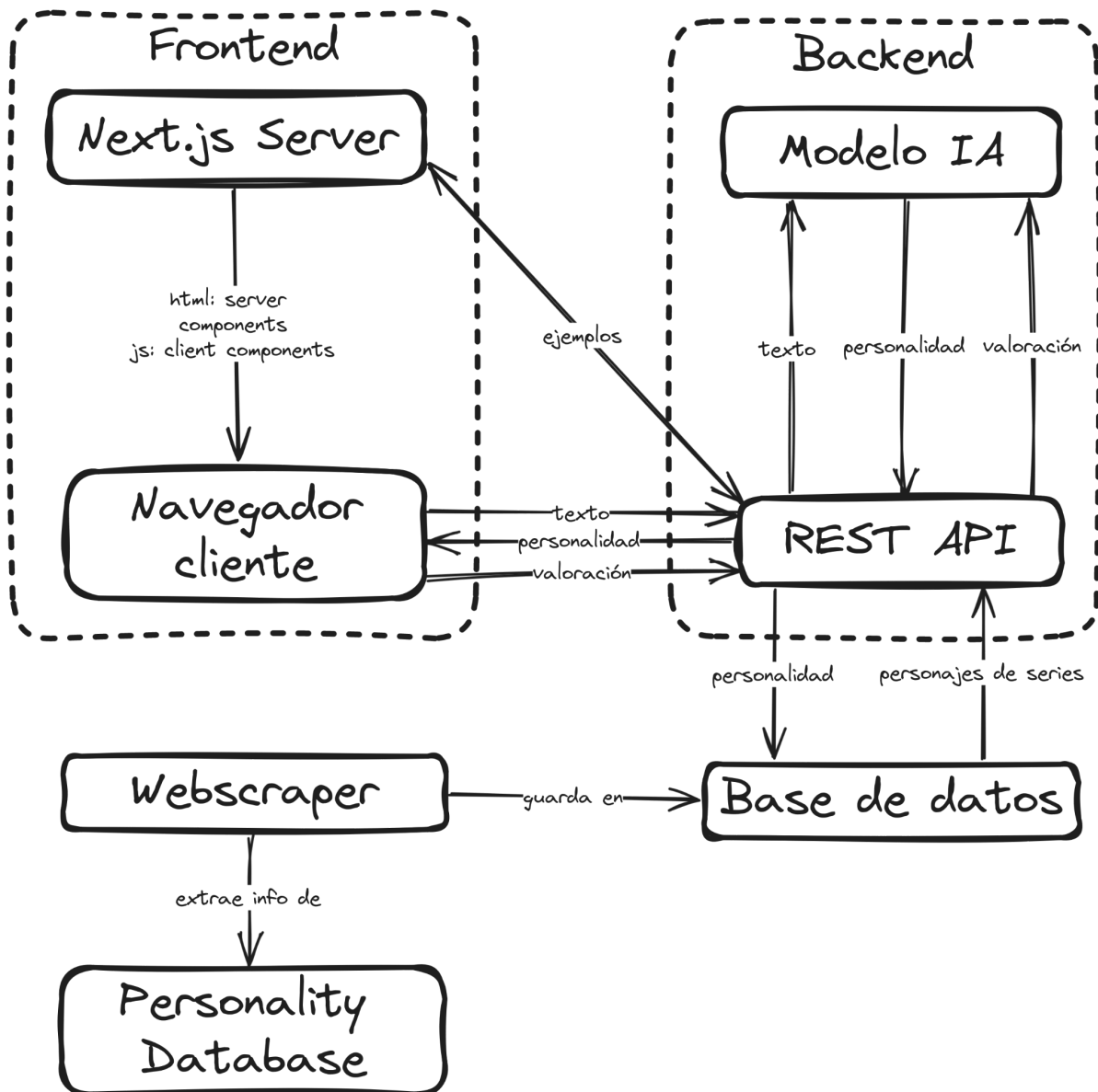
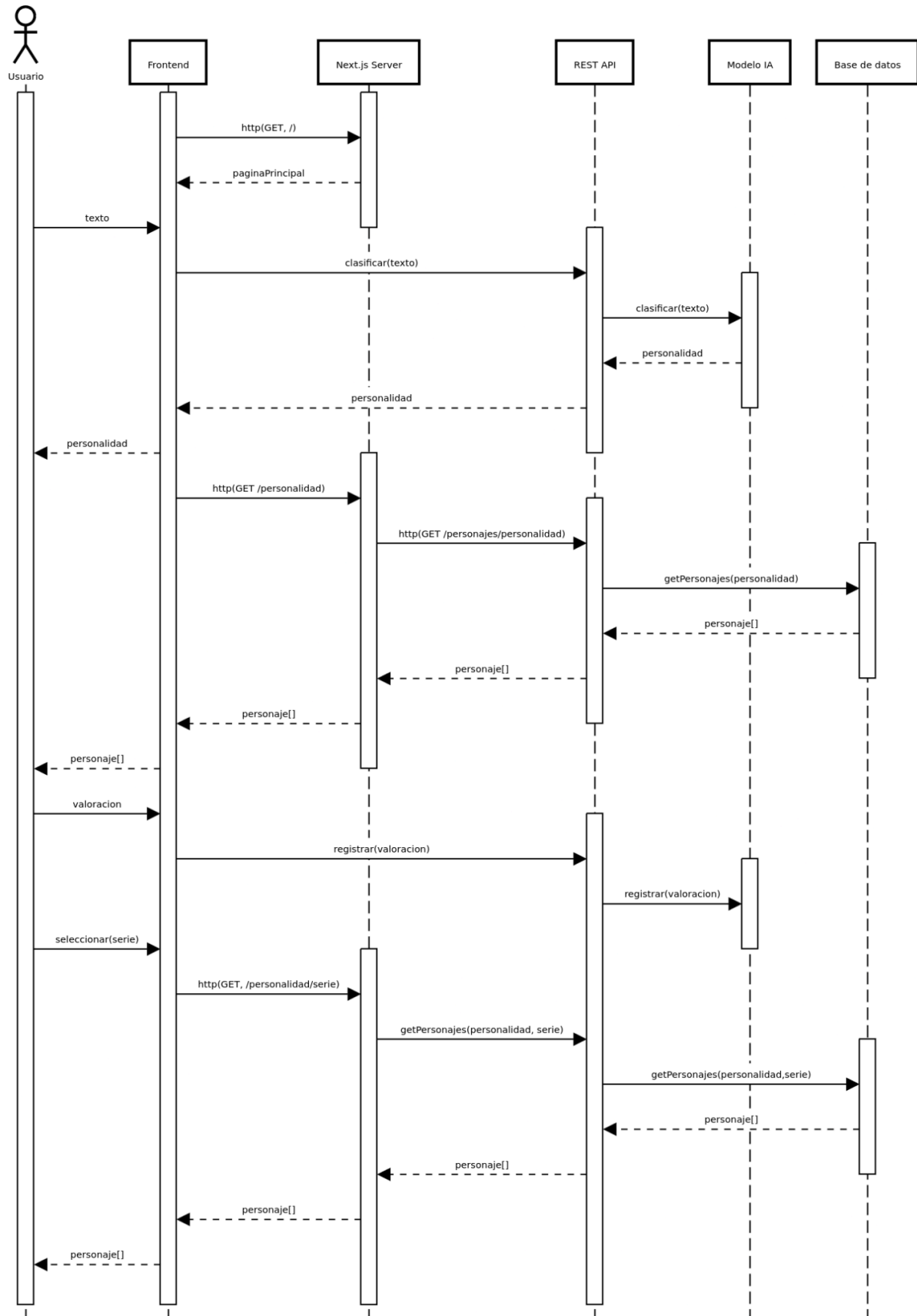


Diagrama de secuencia de sistema



Poblado de la Base de Datos

[Ver Poblado de la Base de Datos](#) (no cambió desde la versión inicial)

Marco de ampliación

Como una posible ampliación del proyecto, se podría aprovechar la información ya recopilada de ciertas imágenes relacionadas con el personaje en cuestión para hacer un modelo que coteje las imágenes con la foto de perfil del usuario para saber cuál es el personaje más parecido al usuario físicamente .

Dentro de las posibles expansiones del proyecto, se vislumbra la oportunidad de utilizar la información previamente recopilada de imágenes asociadas al personaje en consideración. Este recurso ofrece la base idónea para desarrollar un modelo avanzado capaz de comparar las imágenes obtenidas con la fotografía de perfil del usuario. La idea central sería identificar, mediante un algoritmo de reconocimiento facial, cuál de los personajes catalogados comparte similitudes físicas más cercanas con el usuario, ofreciendo así una experiencia interactiva y personalizada.

Este enfoque innovador permitiría no solo vincular la personalidad del usuario con la de un personaje, sino también establecer un vínculo visual y físico, generando un nivel de empatía y conexión aún más profundo. La aplicación de técnicas de inteligencia artificial y aprendizaje automático en este proceso de comparación de imágenes proporcionaría resultados precisos y relevantes.

Además, esta extensión del proyecto podría tener un impacto significativo en la experiencia del usuario, al ofrecer una forma lúdica y novedosa de identificación con un personaje determinado. Esta característica adicional potenciaría la interacción del usuario con la plataforma, brindándole una experiencia más inmersiva y atractiva.

Otra posible expansión del proyecto podría consistir en la integración de recomendaciones de playlists personalizadas basadas en la personalidad del usuario. Para llevar a cabo esta extensión se podría utilizar la base de datos proporcionada en <https://www.kaggle.com/datasets/xtrnglc/spotify-mbti-playlists>, que asigna personalidades a diferentes playlist de Spotify.

Este enfoque se centraría en analizar la personalidad del usuario y una vez determinada, se podría vincular con playlists específicas de la base de datos mencionada, sugiriendo aquellas que coincidan con la personalidad del usuario.

La implementación de este componente adicional no sólo enriquecería la experiencia del usuario al proporcionar recomendaciones musicales personalizadas, sino que también crearía una conexión más profunda entre la identificación del usuario con el personaje y sus preferencias musicales. Esta sinergia entre la personalidad del personaje y la música recomendada contribuiría a una experiencia más completa y atractiva para el usuario.

Pruebas

Se han hecho varias pruebas, en el manual de uso se encuentra un ejemplo de uso en el que dado un texto, nos generará una personalidad. Se encuentra también un ejemplo similar en el notebook del sistema de recomendación y valoraciones.

Bibliografía

- Mazlumi, A. (2018). MBTI Personality Type Twitter Dataset [Dataset]. Kaggle. <https://www.kaggle.com/datasets/mazlumi/mbti-personality-type-twitter-dataset>
- datasnaek. (2017). MBTI Type [Dataset]. Kaggle. <https://www.kaggle.com/datasets/datasnaek/mbti-type>
- Khalid, Z. (2018). MBTI Personality Types 500 Dataset [Dataset]. Kaggle. <https://www.kaggle.com/datasets/zeyadkhalid/mbti-personality-types-500-dataset>
- Personality Database. (n.d.). Personality Types. <https://www.personality-database.com/>
- modAL. (n.d.). GitHub Repository. <https://github.com/modAL-python/modAL>
- scikit-learn. (n.d.). API Reference. <https://scikit-learn.org/stable/modules/classes.html>
- Brownlee, J. (2022, January 10). Histogram-Based Gradient Boosting Ensembles. Machine Learning Mastery. <https://machinelearningmastery.com/histogram-based-gradient-boosting-ensembles/>
- Analytics Vidhya. (2022, January 6). Histogram Boosting Gradient Classifier. <https://www.analyticsvidhya.com/blog/2022/01/histogram-boosting-gradient-classifier/>
- Dolly. (2022, January 10). Personality Prediction using Myers-Briggs Type Indicator. Medium. <https://medium.com/@dolly19304/personality-prediction-using-myers-briggs-type-indicator-56888416e87c>
- Brownlee, J. (2022, January 5). Singular Value Decomposition for Dimensionality Reduction in Python. Machine Learning Mastery. <https://machinelearningmastery.com/singular-value-decomposition-for-dimensionality-reduction-in-python/>
- Elldora. (2021, December 29). How to Plot Feature Importance using Truncated SVD. DEV Community. <https://dev.to/elldora/how-to-plot-feature-importance-using-truncated-svd-166n>

Anexo I - Versión inicial

Descripción

¿No sabes qué decir cuando te preguntan cómo eres? ¿Quieres saber con qué personas tienes mayor compatibilidad? Muchas personas se preguntan qué tipo de personalidad tienen, sin embargo, hoy en día es complicado conocer este hecho sobre nosotros mismos. Esto es un factor influyente a la hora de relacionarse con el entorno y con las personas que nos rodean, afectando en la manera en la que nos relacionamos con nuestros amigos o nuestra familia, o incluso con nosotros mismos.

No esperes más y haz nuestro test online gratis, 100% eficaz con resultados inmediatos, y te diremos con qué amigos te puedes llevar mejor, y cómo será tu futura pareja ideal.

El proyecto consiste en un programa de recomendación de personajes de series, basado en los posts de X del usuario. Se utilizarán distintos indicadores como el indicador Myers-Briggs, utilizado en los procesos de selección de algunas empresas. Es también utilizado en populares aplicaciones como Tinder para catalogar las personalidades.

Meta

El objetivo principal es permitir a los usuarios obtener una mejor comprensión de su personalidad utilizando sus posts en X, y aplicando el modelos de análisis de personalidad de Myers-Briggs, para proporcionar información detallada sobre sus rasgos de personalidad.

La aplicación busca recomendar personajes de series de televisión que compartan similitudes en términos de personalidad con el usuario.

Objetivos técnicos

Recopilación de datos

Se deben recopilar los posts de X del usuario para usarlos en el modelo de análisis de personalidad.

Historias de usuario:

- [Resultado de recomendación](#)

Funcionalidades:

- [Backend](#)
- [Interfaz web](#)

Modelo de IA para análisis de personalidad

Se debe desarrollar un modelo de IA capaz de determinar la personalidad según el índice de Myers-Briggs, de forma que, reciba como entrada un texto y determine de los 16 tipos cuál corresponde al autor del texto.

Historias de usuario:

- [Inicio de sesión con X](#)
- [Resultado de recomendación](#)

Funcionalidades:

- [Modelo de IA para predecir personalidades de texto](#)
- [Backend](#)

Poblar una base de datos de personalidades de personajes de series

Usando los datos de la web [Personality Database](#) para obtener información sobre las personalidades de algunas series populares.

Historias de usuario:

- [Selección de serie](#)
- [Resultado de recomendación](#)

Funcionalidades:

- [El poblador de la base de datos](#)
- [Base de datos](#)

Desarrollo de una interfaz de usuario

Debe tener una interfaz de usuario que permita acceder con una cuenta de X y muestre el resultado, dando la opción de compartirlo en X.

Historias de usuario:

- [Inicio de sesión con X](#)
- [Selección de serie](#)
- [Resultado de recomendación](#)

Funcionalidades:

- [Interfaz web](#)

Historias de usuario

Inicio de sesión con X

Como usuario de la aplicación, quiero poder iniciar sesión con mi cuenta de X para acceder a una experiencia personalizada basada en mis posts de X.

Criterios de Aceptación:

7. Debe haber un botón "Iniciar sesión con X" en la pantalla de inicio.
8. Al hacer clic en el botón, aparecerá un diálogo que autorice a la aplicación el acceso a su histórico de posts.

Objetivos técnicos:

- [Modelo de IA para analisis de personalidad](#)
- [Desarrollo de una interfaz de usuario](#)

Funcionalidades:

- [Interfaz web](#)
- [Backend](#)

Selección de serie

Como usuario, deseo tener la posibilidad de elegir una serie de mi interés de una lista proporcionada. Esto me permitirá obtener recomendaciones de personajes de la serie seleccionada.

Criterios de Aceptación:

3. Debe mostrarse una lista de series disponibles en la aplicación.
4. El usuario debe poder interactuar con la lista y seleccionar una serie de su elección.

Objetivos técnicos:

- [Poblar una base de datos de personalidades de personajes de series](#)
- [Desarrollo de una interfaz de usuario](#)

Funcionalidades:

- [Interfaz web](#)
- [Backend](#)
- [Base de datos](#)

Resultado de recomendación

Como usuario, quiero ver el resultado de mi análisis de personalidad y cómo se relaciona con los personajes de la serie que he elegido. También deseo poder compartir este resultado en X.

Criterios de Aceptación:

4. Al finalizar el análisis, la aplicación debe mostrar el resultado.
5. El resultado debe incluir el nombre del personaje recomendado, una imagen de este y una breve descripción de los índices de Myers-Briggs detectados.
6. Debe existir una opción para compartir el resultado en la plataforma X.

Objetivos técnicos:

- [Desarrollo de una interfaz de usuario](#)
- [Modelo de IA para análisis de personalidad](#)
- [Recopilación de datos](#)
- [Poblar una base de datos de personalidades de personajes de series](#)

Funcionalidades:

- [Interfaz web](#)
- [Modelo de IA para predecir personalidades de texto](#)
- [Base de datos](#)

Funcionalidades

El proyecto se divide en 5 componentes y se desplegarán usando contenedores Docker.

[Ver diagrama de arquitectura](#)

Interfaz web

Se realizará usando usando Next.js. Será una página con 3 rutas, la primera tendrá la información del proyecto y el botón de iniciar sesión con X, que pasará a la segunda página, que realizará la petición al backend para analizar al usuario y con la respuesta de este se pasará a la última página, que mostrará el resultado y permitirá compartirlo en X.

Modelo de IA para predecir personalidades de texto

Se realizará en un Notebook de Google Colab usando las librerías de Python *scikit-learn* y *NLTK* (Natural Language Toolkit).

Backend

Se encarga de responder a las peticiones de la interfaz web, ejecutando el modelo de IA y coteja el resultado con la base de datos para devolver al usuario los personajes su misma personalidad.

Este componente se programará en Python usando el framework [FastAPI](#) y para la comunicación con la base de datos usaremos el ORM [SQLAlchemy](#).

Base de datos

Será una base de datos relacional, con el sistema gestor de bases de datos MySQL.

[Ver diagrama de entidad relación](#)

El poblador de la base de datos

Será un programa en Python, que usando el entorno [Selenium WebDriver](#) para Python, se hará web scraping a la página web de [Personality Database](#) de la que obtendrá los datos de las distintas personalidades de diferentes personajes de series y películas.

Glosario de términos

Índices	Significado
I/E	Introverso/Extroverso
S/N	Sensitivo/Intuitivo
T/F	Pensativo/Sentimental
J/P	Juicioso/Perceptivo

Estos 4 pares de índices se combinan para formar los 16 tipos de personalidades.

Diagrama de arquitectura

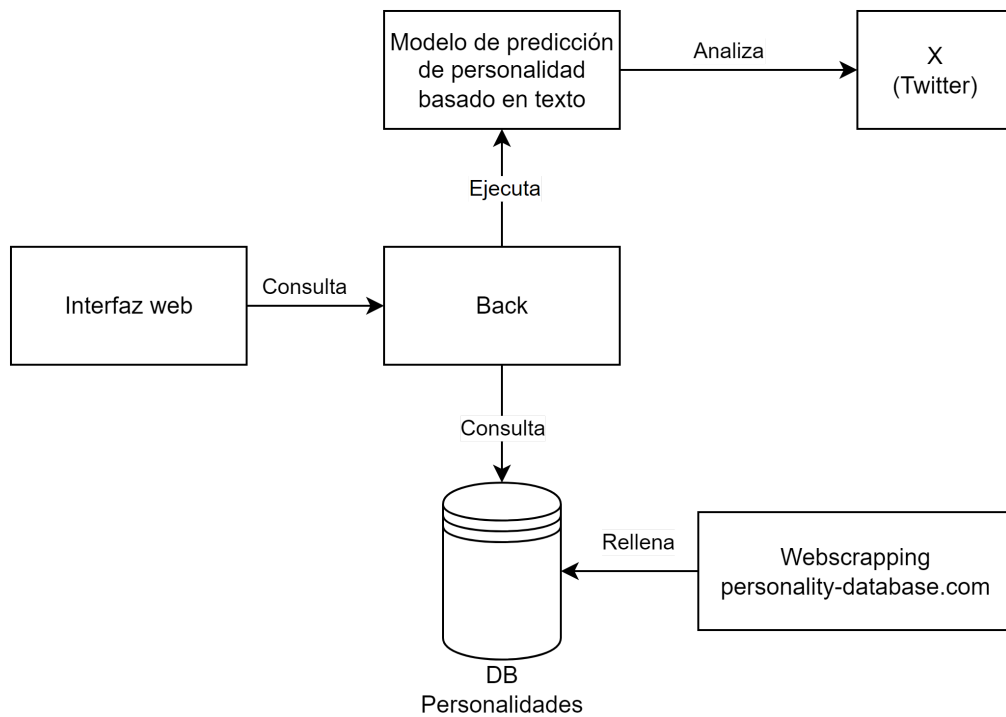
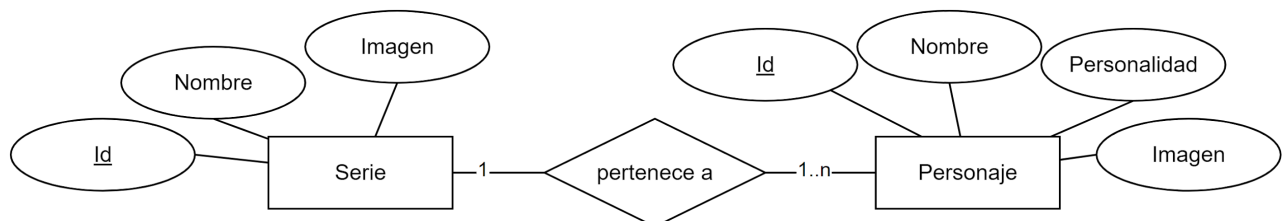


Diagrama entidad-relación



Poblado de la Base de datos

La población de la base de datos se llevó a cabo mediante el uso de un web scraper diseñado para extraer los valores de personalidad de la plataforma "Personality Database". Esta página web alberga información detallada sobre las personalidades de miles de personajes de series y personas reales. Dado que los datos de entrada consisten en texto plano proporcionado por el usuario, y no se almacena en la base de datos ningún detalle relacionado con la identidad del usuario, las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) no son consideradas como necesarias en este contexto.

La naturaleza del dataset, proveniente de un web scraper, implica que los datos son estáticos y se cargan en la base de datos durante el proceso de inicialización. Al no haber una interacción directa y continua con los datos por parte de los usuarios, la actualización o modificación de registros específicos no se considera esencial. Además, la información recopilada es estable y se centra en las personalidades de los personajes, evitando la necesidad de realizar actualizaciones frecuentes.

Por lo tanto, aunque las operaciones CRUD tradicionales no son una prioridad en este contexto, se enfoca en garantizar la integridad y coherencia de los datos durante el proceso de carga inicial. La simplicidad de este enfoque se alinea con la naturaleza estática y de solo lectura de la base de datos, proporcionando eficiencia y estabilidad en el manejo de la información recopilada de "Personality Database".

Marco de ampliación

Como una posible ampliación del proyecto, se podría aprovechar la información ya recopilada de ciertas imágenes relacionadas con el personaje en cuestión para hacer un modelo que coteje las imágenes con la foto de perfil del usuario para saber cuál es el personaje más parecido al usuario físicamente .

Dentro de las posibles expansiones del proyecto, se vislumbra la oportunidad de utilizar la información previamente recopilada de imágenes asociadas al personaje en consideración. Este recurso ofrece la base idónea para desarrollar un modelo avanzado capaz de comparar las imágenes obtenidas con la fotografía de perfil del usuario. La idea central sería identificar, mediante un algoritmo de reconocimiento facial, cuál de los personajes catalogados comparte similitudes físicas más cercanas con el usuario, ofreciendo así una experiencia interactiva y personalizada.

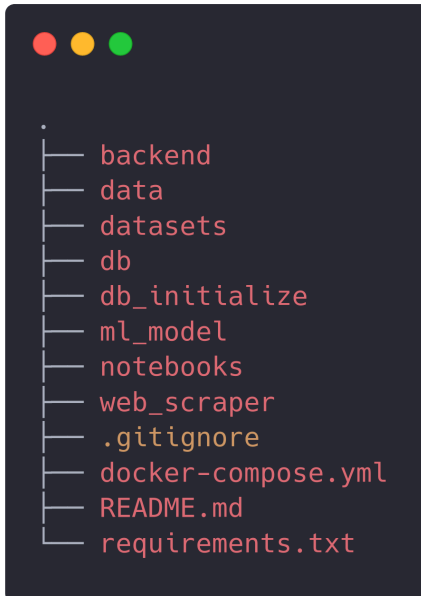
Este enfoque innovador permitiría no solo vincular la personalidad del usuario con la de un personaje, sino también establecer un vínculo visual y físico, generando un nivel de empatía y conexión aún más profundo. La aplicación de técnicas de inteligencia artificial y aprendizaje automático en este proceso de comparación de imágenes proporcionaría resultados precisos y relevantes.

Además, esta extensión del proyecto podría tener un impacto significativo en la experiencia del usuario, al ofrecer una forma lúdica y novedosa de identificación con un personaje determinado. Esta característica adicional potenciaría la interacción del usuario con la plataforma, brindándole una experiencia más inmersiva y atractiva.

Manual de uso

Estructura del proyecto

El proyecto se puede descargar desde el [repositorio de Github](#). Una vez descargado la estructura de ficheros es la siguiente:



- [backend](#): contiene el backend.
- [data](#): reservado para los datos de la base de datos [MySQL](#), es un volumen de `/var/lib/mysql` del contenedor.
- [db](#): contiene el modelo y ORM de la base de datos como un módulo Python.
- [db_initialize](#): módulo Python para conexión e inicialización de la base de datos.
- [ml_model](#): carpeta para almacenar el resultado del entrenamiento con los notebooks.
- [notebooks](#): Jupyter Notebooks correspondientes a preprocesamiento y entrenamiento.
- [web_scraper](#): módulo Python encargado del web scraping de [Personality Database](#) y poblado de la base de datos.

Pasos previos

Antes de levantar los servicios principales es necesario preparar el modelo de IA y realizar el poblado de la base de datos.

Modelo de IA

Antes de poner en marcha la web es necesario entrenar el modelo de IA o descargar sus ficheros en la carpeta `ml_model`.

Se genera con los notebooks:

1. [Crear dataset](#): genera `dataset_definitivo.csv`
2. [Preprocesado](#): genera `X_train.lzma`, `y_train.lzma`, `X_test.lzma`, `y_test.lzma`, `bow.lzma`.
3. [Entrenamiento](#): genera `model.lzma`

Poblado de la base de datos

Para descargar información de las series usando el `web_scraper` hay que volcar los links de las páginas a utilizar en `web_scraper/links.txt`. Luego se ejecuta el script usando Docker con `docker-compose --profile dev --profile init up --build`. Este script crea la base de datos, las tablas e inserta la información obtenida del web scraping.

Las webs son propensas a cambios, por tanto es muy probable que el script requiera modificaciones para adaptarse a cambios del HTML incluso a corto plazo. En nuestro caso, hicimos el script y realizamos la obtención necesaria para evitar tener que adaptarlo.

Ejecución

Para controlar la ejecución y servicios activos se usa Docker Compose con distintos perfiles para controlar qué acciones se realizan:



```
# Primer inicio del proyecto con webscraping y phpmyadmin
$ docker-compose --profile dev --profile init up

# Inicio con datos existentes
$ docker-compose up
```

- Siempre: backend, mysql
- dev: phpmyadmin
- init: web_scraper

Backend

Ofrece al frontend una REST API para realizar todas las operaciones necesarias para la conexión con la base de datos, además permite también la conexión con el modelo, que, recibe como entrada un texto y la salida consiste en una de las 16 posibles personalidades.

Los endpoints que ofrece la API REST son los siguientes:

MÉTODO HTTP	ENDPOINT	EXPLICACIÓN	PARÁMETROS
GET	/api/series	Devuelve una lista con todas las series disponibles	
GET	/api/series/{serie_id}/characters/{personality}	Devuelve los personajes de una serie que tengan una determinada personalidad	Header: - ID de la serie - personalidad
POST	/api/personality	Predice la personalidad de un texto	Body: - "text": texto a predecir

Notebooks

El proceso se ha dividido en varios notebooks adjuntos al entregable para simplificarlo. De esta manera, el preprocesamiento se realiza por separado del entrenamiento. Cada parte es independiente, lo que facilita comprender y ejecutar cada paso individualmente. Además, se guarda el resultado tanto del preprocesamiento para utilizar los datos en el entrenamiento, como del modelo entrenado para su uso en el backend.

También hicimos un notebook que explica como funciona el web scraper, aunque lo ejecutamos usando Docker para realizar la conexión a base de datos fácilmente.