

Documentación Trabajo – “Brawl Stars”

Índice:

1. Objetivos
2. Partes del proyecto
3. Manual de uso

1. Objetivos

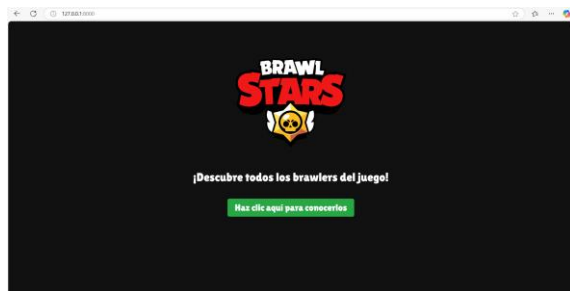
El objetivo principal de este proyecto es desarrollar una plataforma interactiva y visualmente atractiva que permita a los usuarios explorar y consultar información detallada sobre los brawlers del juego Brawl Stars, incluyendo sus cualidades y características más relevantes. Para ello, se ha diseñado una interfaz animada, divertida e interactiva, con el propósito de captar la atención del usuario y ofrecerle una experiencia inmersiva que refleje el estilo único del juego.

2. Partes del proyecto

Todo el proyecto se construyó utilizando Django como base principal, sirviendo tanto de framework como de estructura para integrar todas las funcionalidades desarrolladas. Django permitió gestionar de manera eficiente los modelos de datos, las vistas y las plantillas, lo que facilitó la implementación de las distintas herramientas del proyecto, como BeautifulSoup, Whoosh y el Sistema de Recomendación.

2.1. Página de inicio

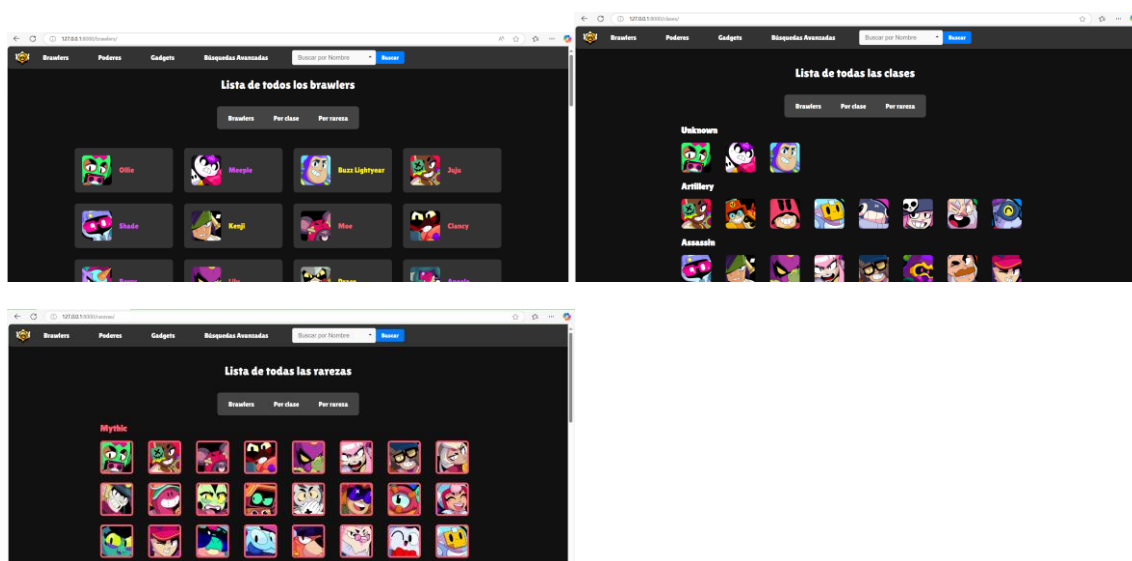
La página de inicio es la que se muestra al lanzar la aplicación. En ella se puede encontrar un botón que se encarga de popular la base de datos. La herramienta de BeautifulSoup se ha utilizado para el método que obtiene los datos, llamado “populateDatabase”, que obtiene los datos de las siguientes urls: “https://brawlify.com/es/brawlers/” para la gran mayoría de datos y, de “https://cube.brawltime.ninja/cubejs-api/v1/load?query=%7B%22measures%22%3A%5B%22map.winRateAdj_measure%22%2C%22map.us eRate_measure%22%2C%22map.picks_measure%22%5D%2C%22dimensions%22%3A%5B%22map.b rawler_dimension%22%5D%2C%22filters%22%3A%5B%7B%22member%22%3A%22map.season_dim ension%22%2C%22operator%22%3A%22gte%22%2C%22values%22%3A%5B%222024-12- 09%22%5D%7D%2C%7B%22member%22%3A%22map.mode_dimension%22%2C%22operator%22%3 A%22equals%22%2C%22values%22%3A%5B%22wipeout5V5%22%5D%7D%5D%2C%22order%22%3A %7B%22map.winRateAdj_measure%22%3A%22desc%22%7D%7D&queryType=multi”, que completa los valores de un par de propiedades de los brawlers.



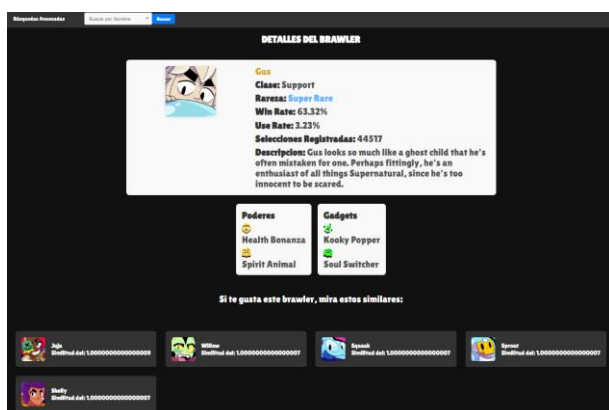
Tras cargar todos los datos se accede al listado de brawlers automáticamente.

2.2. Listado de brawlers

En esta página se observa el listado de brawlers cargados gracias a la página anterior. Se pueden agrupar por clase o por rareza según se prefiera. Se cambia la url según la ordenación elegida y nos muestra el listado de clases/rarezas y sus brawlers asociados.



Al seleccionar en la imagen del brawler en cualquiera de las páginas anteriores nos lleva a los detalles de cada uno de los brawlers.

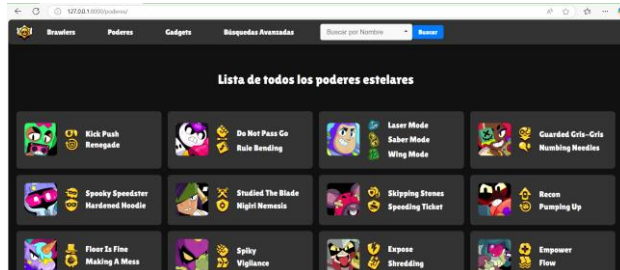


En esta página encontramos los siguientes detalles: Nombre, Clase, Rareza (según la rareza se muestra de un color u otro), Win Rate, Use Rate, Selecciones Registradas, Descripción, Poderes y Gadgets, además de la imagen. Por otro lado, en la parte inferior, podemos encontrar un listado de 5 brawlers más similares al brawler detallado, de los que también podemos acceder a sus detalles pinchando en el contenedor.

Esto se obtiene gracias a la función llamada “recomendaciones_brawlers_detalle” que llama a la función “topMatches” del archivo recommendations.py dado en la asignatura.

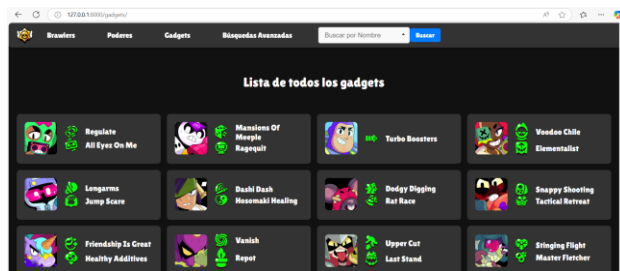
2.3. Listado de poderes

En esta página se muestra el listado de los poderes con sus brawlers asociados. Si se selecciona el contenedor del brawler se puede acceder a sus detalles.



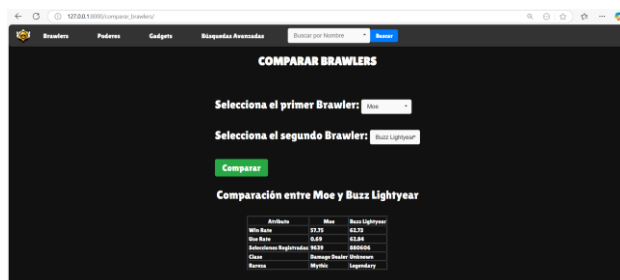
2.4. Listado de gadgets

En esta página se muestra el listado de los gadgets con sus brawlers asociados. Si se selecciona el contenedor del brawler se puede acceder a sus detalles.

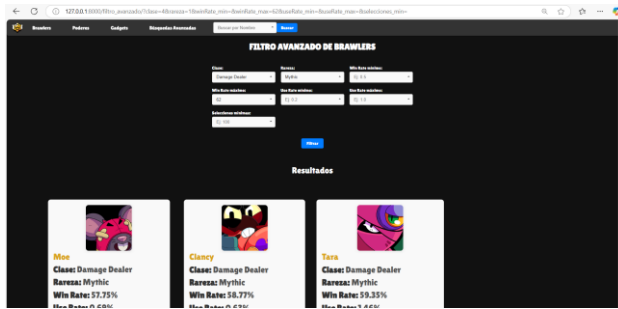


2.5. Búsquedas avanzadas

- ¿Quieres comparar dos brawlers?: Esta opción de la barra de navegación nos lleva a una página para comparar dos brawlers, en la que se utiliza la función “comparar_brawlers”, de la que se emplea el formulario “CompararBrawlersForm”.



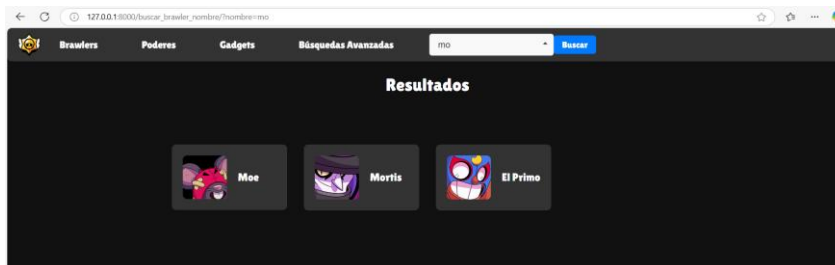
- Filtro avanzado: Esta opción de la barra de navegación nos lleva a una página en la que se puede realizar un filtro para encontrar brawlers según los campos rellenados. Se utiliza el formulario “FiltroAvanzadoForm” en la función “filtro_avanzado”.



Si se selecciona el contenedor del brawler se puede acceder a sus detalles.

2.6. Buscador de brawler por nombre

Este buscador que encontramos en la barra de navegación nos encuentra todos los brawlers que empiecen o contengan la secuenciación de letras escritas en el buscador. Al poblar la base de datos se crea un índice, utilizando la herramienta Whoosh. Se crea una función de Whoosh llamada “buscar_brawler_por_nombre”, ésta se utiliza en la función “buscar_brawler_nombre”. Si se selecciona el contenedor del brawler se puede acceder a sus detalles.



3. Manual de uso

Para el correcto funcionamiento del proyecto hay que tener instaladas las siguientes dependencias: django, whoosh, bs4, lxml. Para ello he creado un archivo “requirements.txt” para llevarlas a cabo, solo es necesario ejecutar el siguiente comando: “pip install -r requirements.txt”.

Para lanzar la aplicación es necesario ejecutar los comandos: “python manage makemigrations”, “python manage migrate” y “python manage.py runserver”, preferiblemente en Visual Studio Code (programa utilizado para el desarrollo). Tras ello, en el navegador, Chrome por ejemplo, accedemos a <http://127.0.0.1:8000/> y podremos visualizar la aplicación correctamente.

Funcionalidades:

- Poblado de la base de datos: pinchando en el botón verde de la página inicial (tarda unos minutos).
- Listado de brawlers.
- Listado de clases.
- Listado de rarezas.
- Listado de poderes.
- Listado de gadgets.
- Comparar brawlers.
- Filtro avanzado.
- Búsqueda de brawlers por nombre.

En caso de que el zip no funcionase correctamente, se puede descargar el proyecto en el siguiente enlace: <https://github.com/SoniaRM/trabajo-aii-2425.git>