Bruno Emilio Sánchez Rey

16/12/24  Modelado Avanzado de la información

Desarrollo de una Pokédex interactiva

Índice

[Introducción 1](#_Toc185272505)

[Colecciones 2](#_Toc185272506)

[Pokémon 2](#_Toc185272507)

[Movimientos 3](#_Toc185272508)

[Ítems 3](#_Toc185272509)

[Usuarios 3](#_Toc185272510)

[Índices 4](#_Toc185272511)

[Pokémon 4](#_Toc185272512)

[Movimientos 5](#_Toc185272513)

[Ítems 6](#_Toc185272514)

[Usuarios 7](#_Toc185272515)

[Explicación de archivos 8](#_Toc185272516)

[app.py 8](#_Toc185272517)

[db.py 11](#_Toc185272518)

[Funcionalidades de la página 14](#_Toc185272519)

[Búsqueda de Pokémons 14](#_Toc185272520)

[Detalle de Pokémon 16](#_Toc185272521)

[Búsqueda de movimientos 17](#_Toc185272522)

[Búsqueda de ítems 19](#_Toc185272523)

[Register / Log in 19](#_Toc185272524)

[Conclusión 21](#_Toc185272525)

# Introducción

Este proyecto tiene como objetivo desarrollar una **Pokédex interactiva** que permita a los usuarios explorar información detallada sobre **Pokémon, movimientos y objetos**, utilizando filtros avanzados para realizar búsquedas. La aplicación está diseñada para ser intuitiva y funcional, permitiendo acceder a datos concretos a cualquier tipo de usuario.

La aplicación está creada en **Python** con el framework **Flask**, conectada a una base de datos **MongoDB**, que almacena la información estructurada en diferentes colecciones.

Los datos que se utilizan fueron obtenidos del siguiente GitHub: <https://github.com/fanzeyi/pokemon.json>

# Colecciones

Los datos están organizados en **cuatro colecciones principales** dentro de MongoDB:

## Pokémon

**Columnas/Claves:**

* id: Identificador único del Pokémon.
* name: Objeto que contiene los nombres en varios idiomas, donde name.english se utiliza para el nombre en inglés.
* type: Lista que contiene los tipos del Pokémon (por ejemplo, ["Electric"], ["Water"]).
* base: Objeto que almacena las estadísticas base del Pokémon:
  + HP: Puntos de vida.
  + Attack: Ataque físico.
  + Defense: Defensa física.
  + Sp. Attack: Ataque especial.
  + Sp. Defense: Defensa especial.
  + Speed: Velocidad.

Patrón de fondo

Descripción generada automáticamente con confianza baja

## Movimientos

**Columnas/Claves:**

* name: Nombre del movimiento.
* type: Tipo del movimiento (por ejemplo, "Fire").
* power: Poder del movimiento.
* category: Categoría del movimiento (Físico, Especial o Estado).
* pp: Power Points (cantidad de usos).

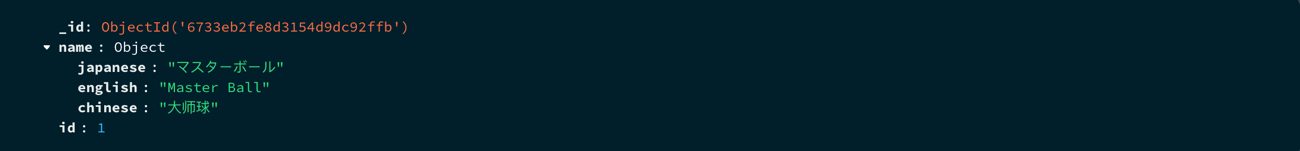
Patrón de fondo

Descripción generada automáticamente

## Ítems

**Columnas/Claves:**

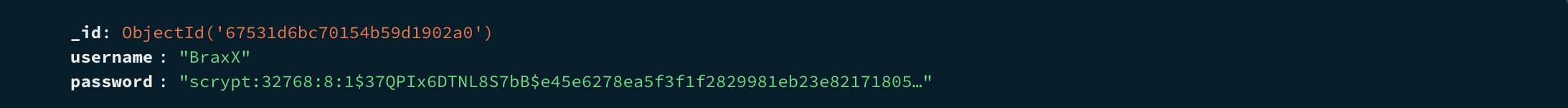
* id: Identificador único del objeto.
* name: objeto que contiene los nombres del item en diferentes idiomas (usaremos el nombre en inglés)



## Usuarios

**Columnas/Claves:**

* username: Nombre de usuario único.
* password: Contraseña cifrada.



# Índices

Para optimizar las consultas en las colecciones, se han creado índices específicos ya que estos ayudan a evitar escanear completamente una colección con cada consulta y poder escalar los datos sin miedo a perder velocidad de búsqueda.

## Pokémon

En la colección “Pokémon”, se han implementado los siguientes índices para optimizar las búsquedas:

1. **name.english\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice se utiliza para las búsquedas por el nombre en inglés del Pokémon. Al ser un índice en un solo campo (name.english), mejora la eficiencia de las consultas que filtran Pokémon por su nombre.
   * **Propiedad**: UNIQUE
     + Garantiza que no existan nombres duplicados en la base de datos
2. **type\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice permite acelerar las búsquedas por el tipo de Pokémon. Es particularmente útil cuando se realizan búsquedas o filtrados por tipos, como en la página de búsqueda de Pokémon.
3. **base.HP\_1\_base.Attack\_1\_base.Speed\_1**
   * **Tipo**: Compuesto
   * **Uso**: Este índice compuesto optimiza las consultas que filtran por múltiples estadísticas a la vez: HP, Attack y Speed. Esto es útil en búsquedas complejas donde se buscan pokemon que sean atacantes físicos para combatir.
4. **base.HP\_1\_base.Sp. Attack\_1\_base.Speed\_1**
   * **Tipo**: Compuesto
   * **Uso**: Similar al anterior, este índice compuesto mejora la eficiencia en búsquedas por HP, Sp. Attack y Speed. La combinación de estas estadísticas es común en búsquedas avanzadas sobre todo cuando buscamos un buen atacante especial.

Captura de pantalla de un celular

Descripción generada automáticamente

## Movimientos

En la colección “movimientos”, se han implementado los siguientes índices:

1. **ename\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice facilita las búsquedas por el nombre en inglés de los movimientos (ename). Además, tiene la propiedad ‘unique’ ya que no existen dos movimientos con el mismo nombre.
2. **type\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice acelera las búsquedas por tipo de movimiento (e.g., "Fire", "Water").
3. **category\_1**
   * **Tipo**: Regular
   * **Uso**: El índice category\_1 permite optimizar las búsquedas por la categoría del movimiento (Físico, Especial o Estado).
4. **power\_1**
   * **Tipo:** Regular
   * **Uso:** este índice tiene como objetivo mejorar los tiempos de aquellas consultas que se basan en la fuerza de los ataques.

Captura de pantalla de un celular

Descripción generada automáticamente

## Ítems

En la colección “ítems”, se han implementado el siguiente índice:

1. **name\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice se aplica al campo name y permite realizar búsquedas eficientes por el nombre de los ítems. Es útil en la página de búsqueda de objetos, donde se filtran por nombre o ID.

Captura de pantalla de un celular

Descripción generada automáticamente

## Usuarios

En la colección usuarios, se han implementado los siguientes índices:

1. **username\_1**
   * **Tipo**: Regular
   * **Uso**: Este índice facilita las búsquedas por el nombre de usuario al iniciar sesión o verificar si un usuario ya existe.
   * **Propiedad**: UNIQUE
     + Garantiza que no haya nombres de usuario duplicados, lo cual es lo más importante a la hora de gestionar cuentas de usuarios ya que podría dar error si varios usuarios tienen el mismo nombre.

Captura de pantalla de un celular

Descripción generada automáticamente

# Explicación de archivos

## app.py

El archivo “app.py” actúa como la base principal de la aplicación web, encargándose de las rutas, el control de usuarios y la interacción con las plantillas HTML. Es la parte central del **backend** que coordina la lógica de la aplicación.

1. **Importación de librerías**:
   * Flask: Framework web para la creación de rutas y renderización de plantillas.
   * werkzeug.security: Se utiliza para la creación y validación de contraseñas seguras en este caso por hash.
   * **Base de Datos**: Se importan funciones y conexiones desde “db.py” para interactuar con MongoDB​.
2. **Autenticación de usuarios**:
   * Ruta **/login**:
     + Gestiona el registro y login del usuario.
     + POST para recibir datos del formulario y verificar si un usuario existe o crear uno nuevo.
     + Se almacenan los usuarios con contraseñas **hasheadas** mediante “generate\_password\_hash.session” se usa para mantener la sesión iniciada.
   * Ruta **/logout**:
     + Cierra la sesión eliminando los datos almacenados en “sesión”.

Texto

Descripción generada automáticamente

1. **Ruta principal (Home)**:
   * Muestra un saludo personalizado si el usuario ha iniciado sesión:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. **Búsqueda de Pokémon**:
   * Ruta **/search/pokemon** (GET/POST):
     + Recibe filtros como name, id, type, hp, attack, etc.
     + Llama a la función leer\_pokemon() de **db.py** para buscar Pokémon en la base de datos.
     + Renderiza pokemon.html con los resultados.

Texto

Descripción generada automáticamente

1. **Detalles del Pokémon**:
   * Ruta **/pokemon/<name>**:
     + Muestra detalles específicos de un Pokémon al buscarlo por su nombre en inglés:

Texto

Descripción generada automáticamente

1. **Búsqueda de movimientos**:
   * Ruta **/search/moves**:
     + Permite buscar movimientos por name, type, power, category, y pp.
     + Invoca la función leer\_movimientos() de **db.py**.

Texto

Descripción generada automáticamente

1. **Búsqueda de ítems**:
   * Ruta **/search/items**:
     + Busca ítems usando name o item\_id a través de la función leer\_items().

Texto

Descripción generada automáticamente

## db.py

El archivo “db.py” centraliza las funciones de consulta a la base de datos. Aquí se definen las conexiones y funciones para hacer todas las consultas posibles.

1. **Conexión a MongoDB**:
   * Se establece una conexión a la base de datos utilizando MongoClient con una URI específica:

Texto

Descripción generada automáticamente

* + Se accede a las colecciones:
    - collection\_pokemon: Pokémon.
    - collection\_items: Ítems.
    - collection\_moves: Movimientos.
    - collection\_users: Usuarios.
    - collection\_types: Tipos (esta colección al final no será usada).

1. **Funciones de búsqueda**:
   * **leer\_pokemon()**:
     + Construye una consulta dinámica basada en los filtros enviados desde la ruta **/search/pokemon**.
     + Ejemplo de filtro por nombre y estadísticas:
     + Devuelve los resultados coincidentes:

Texto

Descripción generada automáticamente

* + **leer\_movimientos()**:
    - Consulta los movimientos utilizando filtros como ename, type, category y pp.
    - Responde a búsquedas precisas según las columnas de la colección movimientos.

Texto

Descripción generada automáticamente

* + **leer\_items()**:
    - Busca ítems por name o id en la colección items:

Texto

Descripción generada automáticamente

1. **Gestión de usuarios**:
   * **find\_user()**:
     + Busca un usuario por username en la colección usuarios.
   * **create\_user()**:
     + Inserta un nuevo usuario con nombre y contraseña **hasheada**.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

# Funcionalidades de la página

Tras acceder al buscador, lo primero que se nos presenta son las tres páginas de búsqueda que contiene nuestra web: pokemon, movimientos y objetos.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

## Búsqueda de Pokémons

La página de **búsqueda de Pokémons** ofrece una interfaz amigable con filtros avanzados que permiten a los usuarios encontrar Pokémons específicos de la base de datos. Las opciones de búsqueda incluyen:

* **Nombre**: Permite buscar un Pokémon por su nombre en inglés (parcial o completo).
  + Ejemplo: Buscar "Pi" devuelve resultados como Pikachu, Pichu o Caterpie.
  + Se utiliza una búsqueda con **expresiones regulares** ($regex) para facilitar búsquedas parciales
* **Número de Pokédex**: Filtra el Pokémon por su número en la Pokédex.
* **Tipos**: Ofrece una selección visual de tipos de Pokémon mediante íconos interactivos. El usuario puede seleccionar uno o varios tipos para refinar la búsqueda.
* **Estadísticas**: Permite filtrar los resultados por atributos como:
  + **HP (Vida)**
  + **Attack (Ataque)**
  + **Defense (Defensa)**
  + **Sp. Attack (Ataque Especial)**
  + **Sp. Defense (Defensa Especial)**
  + **Speed (Velocidad)**

Cada estadística se filtra utilizando un rango mínimo (por ejemplo, HP ≥ 80)

**Interfaz de resultados**:

* Los resultados aparecen como tarjetas individuales que muestran el nombre del Pokémon, sus tipos (con íconos) y su número de Pokédex.
* Cada entrada es **clicable**, permitiendo al usuario acceder a los detalles específicos del Pokémon.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Detalle de Pokémon

La página de **detalles del Pokémon** proporciona información completa y visualmente atractiva sobre un Pokémon seleccionado.

* **Elementos mostrados**:
  + **Nombre e ID**: Centrados en la parte superior con un estilo llamativo.
  + **Imagen**: Mostrada en gran tamaño.
  + **Tipos**: Representados con íconos visuales de los tipos.
  + **Estadísticas**:
    - Cada estadística se presenta con una **barra de progreso dinámica** cuyo tamaño se ajusta en función de su valor (máximo 252):
* **Botón Volver**:
  + Permite regresar a la página de búsqueda de Pokémon.

Esta página utiliza una combinación de **HTML dinámico** con datos servidos desde Flask y MongoDB.

Interfaz de usuario gráfica

Descripción generada automáticamente

## Búsqueda de movimientos

La página de **búsqueda de movimientos** permite encontrar ataques y habilidades específicas mediante varios filtros:

* **Nombre**: Permite buscar movimientos por su nombre en inglés.
* **Tipo**: Similar a la búsqueda de Pokémons, permite seleccionar un tipo único mediante íconos visuales.
* **Categoría**: Incluye tres opciones con imágenes representativas:
  + **Físico**
  + **Especial**
  + **Estado**

Estos valores corresponden a las columnas 物理, 特殊, y 变化 en la base de datos, siendo gestionados en el backend.

* **Poder (Power)**: Filtra los movimientos según su daño mínimo.
* **PP (Puntos de Poder)**: Filtra por cantidad de usos del movimiento.

**Interfaz de resultados**:

* Las tarjetas muestran:
  + Nombre del movimiento.
  + Tipo (ícono).
  + Categoría (ícono).
  + Poder y PP.
* Las entradas están bien distribuidas horizontalmente para aprovechar el espacio.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Búsqueda de ítems

La página de **búsqueda de ítems** es la más simple y está optimizada para facilitar el acceso rápido a los datos de la base de datos:

* **Filtros**:
  + **Nombre**: Permite buscar ítems por su nombre en inglés.
  + **ID**: Filtra ítems según su identificador único.

**Interfaz de resultados**:

* Cada resultado muestra únicamente el nombre del ítem acompañado de su **ID**
* Se usa la lógica de búsqueda basada en la columna name:

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

## Register / Log in

El sistema de **autenticación de usuarios** permite el registro e inicio de sesión, cumpliendo con la funcionalidad básica **CRUD** para usuarios.

* **Registro**:
  + Cuando un usuario introduce un nombre y contraseña por primera vez, se guarda en la colección **usuarios**.
  + La contraseña se almacena utilizando **hashing seguro** para proteger los datos:
* **Inicio de sesión**:
  + Si el usuario ya existe, el sistema verifica la contraseña usando:
  + Si la verificación es exitosa, la sesión se inicia y aparece un saludo en la interfaz superior derecha

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

* **Logout**:
  + El usuario puede cerrar sesión y el sistema elimina sus datos de sesión.

**Interfaz**:

* Al no estar autenticado, se muestra un botón para acceder al formulario
* Al iniciar sesión, aparece el mensaje:
  + **"Hola, [nombre\_usuario]"**.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

# Conclusión

El objetivo principal de este proyecto era desarrollar una aplicación web interactiva que permitiera la búsqueda, visualización y cambios en datos almacenados en una base de datos de **MongoDB**, aplicando filtros avanzados y mostrando resultados de manera eficiente y atractiva.

1. **Desarrollo de Habilidades en MongoDB**:

Se implementaron consultas usando **expresiones regulares** y **filtrado por campos específicos**, optimizando las búsquedas mediante la creación de **índices** en las colecciones.

1. **Interfaz Web Funcional y Amigable**:

Se creó una interfaz web limpia y accesible que permite interactuar con los datos de manera sencilla. Permite buscar entre diferentes categorías con gran precisión y los resultados se muestran en tarjetas visualmente organizadas que aprovechan el espacio.

1. **Implementación de CRUD Básico**:

Se añadió una funcionalidad de **autenticación de usuarios** con registro y login, cumpliendo con los requerimientos CRUD básicos. La seguridad se gestionó mediante el **hashing** de contraseñas.

En conclusión, el proyecto logra **cumplir con los objetivos propuestos**: desarrollar una aplicación que permita la interacción con datos almacenados en una base de MongoDB mediante una interfaz web funcional y eficiente. Además, la inclusión de funcionalidades avanzadas, como la autenticación de usuarios o el hasheo de contraseñas, añade un valor significativo al proyecto.