Proyecto de Migración - Liga Nacional de Básquet

Fernandez, Andrés
ACT4AV

Introducción

Este documento describe el proceso de desarrollo y migración realizado para el segundo parcial de la materia Análisis y Metodologías. A partir de la documentación generada en el primer parcial, el objetivo principal fue construir una aplicación funcional utilizando Flask, integrando tecnologías externas y aplicando buenas prácticas de organización, modularización y manejo de sesiones.
La prioridad durante el desarrollo fue lograr un sistema estable y coherente antes de avanzar con características adicionales o detalles estéticos. Esto permitió mantener un flujo de trabajo claro y evitar desviaciones que pudieran comprometer la funcionalidad principal.

Estructura General del Proyecto

El proyecto está organizado siguiendo una estructura modular que facilita su comprensión y mantenimiento. La aplicación se divide de la siguiente manera:

```
LNB-Migracion/
  Documentación/
  Instance/
  static/
    img/
    ...
  templates/
    ...
  venv/
  .env
  .gitignore
  models.py
  requirements.txt
  summ_utils.py
  app.py
```

Descripción de las carpetas y módulos

- **app.py** contiene toda la lógica de la aplicación. Las rutas y sus elementos, la importación de librerías, archivos y otros recursos necesarios para el correcto funcionamiento del proyecto.

- **models.py** incluye los modelos migrados desde la base de datos del proyecto original en java, utilizados con SQLAlchemy.

- **Instance/** contiene la base de datos importada del proyecto anterior.
- **summ_utils.py** contiene la función auxiliar del uso del modelo de IA para resumir texto.
- **templates/** almacena todas las vistas del proyecto conectadas a un base.html, que dicta la estética general y la distribución visual de los componentes de la aplicación.
- **static/** contiene las imágenes usadas en las vistas.
- **requirements.txt** lista todas las dependencias necesarias.

Librerías y dependencias utilizadas

Todas las dependencias están detalladas en requirements.txt. Las más importantes son:

- **Flask**: Framework principal utilizado para desarrollar la aplicación.
- **Flask Login**: Manejo de sesiones y autenticación.
- **Flask SQLAlchemy**: ORM para la interacción con la base de datos.
- **Flask Bcrypt**: Hash seguro de contraseñas.
- **Authlib**: Base para autenticación externa.
- **python-dotenv**: Manejo de variables de entorno.
- **transformers**: Modelo de IA utilizado para resumir texto de artículos.
- **torch**: Backend necesario para ejecutar los modelos de transformers.
- **requests**: Utilizado para llamadas externas.

Cada una de estas librerías fue instalada dentro de un entorno virtual para evitar conflictos con otras configuraciones locales.

Instrucciones de ejecución

Para ejecutar el proyecto correctamente en cualquier equipo se deben seguir estos pasos:

1. Instalar Python 3.

2. Crear el entorno virtual:

   ```
   python -m venv env
   ```

3. Activarlo:

   Windows:

   ```
   env\Scripts\activate
   ```

   Mac/Linux:

   ```
   source env/bin/activate
   ```

   Instalar dependencias:

   ```
   pip install -r requirements.txt
   ```

4. Ejecutar la aplicación:

   ```
   python app.py
   ```

5. La aplicación iniciará en http://127.0.0.1:5000.

Diagrama de flujo y arquitectura (explicado textual)

El flujo principal de la aplicación se organiza de la siguiente forma:

1. El usuario accede a la página principal.
2. Si el usuario no está logueado, puede registrarse o iniciar sesión, o navegar por las secciones públicas como artículos, eventos y equipos.
3. Una vez autenticado, accede a las secciones de perfil, mi jugador, y juegos.
4. Si el usuario tiene el rol de administrador, se habilitan las funciones de creación y edición y eliminación de los elementos de cada entidad.
5. Cada vez que se crea un artículo, el sistema envía el contenido al modelo de IA que genera un resumen automático.

6. La base de datos almacena cada instancia creada y modificada por el administrador en su panel o características de perfil modificadas por un aficionado.

Resultados obtenidos

El desarrollo del proyecto se centró en construir una base sólida y funcional antes de avanzar con mejoras secundarias. La aplicación final incluye autenticación tanto de Flask como de Google, manejo de sesiones, distinción de roles, panel administrativo, carga de contenido y la integración con un modelo de IA para generar resúmenes de manera automática.

Desafíos enfrentados

Uno de los desafíos más importantes fue reorganizar el proyecto para adaptarlo correctamente a Flask, manteniendo una estructura clara sin perder funcionalidad. Las rutas, los templates y el manejo de sesiones requirieron una reorganización completa respecto al proyecto original.
La integración del modelo de IA también trajo complicaciones, especialmente por el tamaño de las dependencias y la necesidad de asegurar que funcionara dentro del entorno virtual sin errores.

Mejoras futuras

Si bien el proyecto entrega una versión funcional, existen varias mejoras que pueden incorporarse en una etapa siguiente:

- Optimizar lo que ya está implementado, especialmente en algunas secciones donde se podría mejorar la eficiencia o simplificar el código.

- Mejorar el diseño visual y darle una identidad más consistente a la interfaz.

- Implementar almacenamiento de imágenes en AWS S3, como estaba previsto originalmente.

- Completar la carga de datos reales: jugadores, artículos y eventos, para que el proyecto refleje más fielmente su propósito.

- Implementar el sistema de puntos, que es uno de los elementos centrales del proyecto, pero no llegó a hacerse por una cuestión de tiempos. La estructura previa ya está pensada, así que su integración futura sería natural y sin necesidad de rehacer la base.

En resumen, la base del proyecto ya está construida de forma firme. La continuación lógica es mejorar, pulir y expandir lo existente, sin necesidad de replantear el sistema desde cero.

Conclusión

El proyecto cumple con los objetivos planteados para el parcial: migración sólida, aplicación funcional, integración con una tecnología externa y documentación completa. El proceso de trabajo se enfocó en priorizar lo esencial, garantizando una base estable y clara antes de sumar características adicionales.
El resultado es una aplicación ordenada, modular y lista para seguir creciendo con las mejoras planificadas

Fernandez, Andrés
ACT4AV