Asignación: Evolución del Generador de Reportes V1

Introducción

¡Felicidades! Has trabajado con una aplicación web funcional ("Generador de Reportes") que permite a los usuarios solicitar la creación de reportes CSV basados en datos de una API externa (actualmente, PokeAPI). La aplicación cuenta con una interfaz moderna, una API robusta, un proceso de fondo desacoplado y utiliza servicios en la nube de Azure, como se muestra en la arquitectura de referencia en los videos compartidos.

El objetivo de esta asignación es extender y adaptar esta aplicación base, añadiendo nuevas funcionalidades y demostrando tu habilidad para modificar y mejorar un sistema existente que debe permanecer desplegado y funcional en Azure.

Contexto Tecnológico

- **Frontend:** Next.js (React)
- **Backend:** API Python con FastAPI desplegada en Azure App Service
- **Proceso Asíncrono:** Azure Functions (Python Queue Trigger)
- **Base de Datos:** Azure SQL Database
- **Almacenamiento:** Azure Blob Storage (para los CSV), Azure Queue Storage (para mensajes)
- **API Externa:** PokeAPI
- **Infraestructura como Código (IaC):** Terraform (implícito por la mención del repositorio)

Tareas a Realizar (Total: 90 Puntos)

Deberás implementar las siguientes tres funcionalidades principales, asegurándote de que la aplicación completa siga siendo desplegable y funcione correctamente en Azure:

Tarea 1: Implementar Eliminación Completa de Reportes (25 Puntos)

Actualmente, los reportes generados permanecen en el sistema. Debes añadir la capacidad de eliminarlos por completo.

- **Frontend (Next.js):**
  - Añade un botón o ícono de "Eliminar" a cada fila de la tabla de reportes.
  - Implementa una confirmación (ej. un diálogo modal) antes de proceder con la eliminación.
  - Al confirmar, realiza una llamada a la API backend para solicitar la eliminación.
  - Actualiza la lista de reportes en la UI después de una eliminación exitosa.
- **Backend (API FastAPI):**
  - Crea un nuevo endpoint en la API (ej. `DELETE /api/report/{report_id}`).
  - Este endpoint debe recibir el ID del reporte a eliminar.
  - Implementa la lógica para:
    - Verificar que el reporte existe en Azure SQL DB.
    - **Eliminar el archivo CSV correspondiente** del contenedor en Azure Blob Storage. Construye el nombre esperado del blob y utiliza la SDK de `azure-storage-blob` para borrarlo.
    - Eliminar el registro del reporte de la tabla en Azure SQL DB.
    - Manejar posibles errores (reporte no encontrado, error al eliminar blob, error de base de datos).

Tarea 2: Enriquecer Reporte con Detalles del Pokémon (40 Puntos)

Actualmente, el reporte CSV generado solo incluye el nombre y la URL de cada Pokémon del tipo seleccionado. Debes modificar el proceso para incluir información más detallada sobre cada Pokémon en el reporte, utilizando siempre la PokeAPI.

- **Objetivo:** Obtener y añadir las **estadísticas base** (ej. HP, Ataque, Defensa, Ataque Especial, Defensa Especial, Velocidad) y las **habilidades** de cada Pokémon al reporte CSV. (Si alguna de esas variables tiene como valor un lista retorna solo uno, lo dejo a tu creatividad que tanta mas información vas a querer agregar)
- **Worker (Azure Function - Python):** Este es el componente con los cambios principales:
  - Después de obtener la lista inicial de Pokémon para el tipo seleccionado (que contiene `name` y `url`), deberás **iterar sobre esa lista**.
  - Para **cada Pokémon** en la lista, realiza una **llamada adicional a la PokeAPI** usando la URL específica de ese Pokémon (que obtuviste en el paso anterior, ej. `https://pokeapi.co/api/v2/pokemon/{nombre_o_id}/`) para obtener sus detalles completos.
  - De la respuesta detallada de cada Pokémon, extrae: Las **estadísticas base** (HP, Attack, Defense, Special Attack, Special Defense, Speed). Usualmente están en una lista dentro del JSON de respuesta.
  - Maneja posibles errores si la llamada a la URL de un Pokémon específico falla.
  - Modifica la función que genera el CSV (`generate_pokemon_csv_pandas`):
    - Añade nuevas columnas al DataFrame y al CSV para cada estadística base (ej. 'HP', 'Attack', 'Defense', etc.).
    - Añade una columna para las habilidades. Decide cómo representar la lista de habilidades en una sola celda del CSV (ej. como una cadena separada por comas: "ability1, ability2") o selecciona una.
- **Base de Datos (Azure SQL):** No requiere cambios.
- **Almacenamiento (Blob Storage):** El archivo CSV resultante tendrá más columnas.

Tarea 3: Reportes con Muestreo Aleatorio (25 Puntos)

Permite al usuario especificar un número máximo de registros aleatorios a incluir en el reporte.

- **Frontend (Next.js):**
  - Añade un campo de entrada numérica en el formulario (ej. "Número Máximo de Registros"). Valida que sea un entero positivo.
- **Backend (API FastAPI):**
  - Modifica el endpoint de creación para aceptar el nuevo parámetro (`sample_size`). Actualiza el modelo Pydantic (opcional, `gt=0`).
  - Almacena este valor en una nueva columna en la tabla de reportes en Azure SQL DB (ej. `SampleSize INT NULL`).
- **Worker (Azure Function - Python):**
  - Obtén el `sample_size` para el reporte que se está procesando.
  - Después de obtener la lista *completa* de resultados de la API externa (PokeAPI), si se especificó un `sample_size` válido y menor al total de resultados, usa `random.sample()` para seleccionar aleatoriamente los registros.
  - Genera el CSV usando la lista resultante (completa o muestreada).

Entregables

1. **Código Fuente: Cinco (5) repositorios Git** actualizados con todo el código modificado y funcional:
   - Frontend (Next.js)
   - API (FastAPI)
   - Azure Functions (Python)
   - Base de Datos (SQL Scripts para tablas, SPs, etc.)
   - Infraestructura como Código (Terraform) Los enlaces a estos **cinco** repositorios deben estar incluidos en tu entrada de portafolio.
2. **Entrada de Portafolio (10 puntos):** Crea una entrada en tu portafolio personal (o en una plataforma como GitHub Pages, LinkedIn, blog personal, etc.)

describiendo este proyecto y las mejoras realizadas. **Esta entrada debe incluir obligatoriamente:**

- Una descripción general del proyecto y su propósito.
- La arquitectura utilizada (puedes basarte en el diagrama de referencia).
- Las tecnologías clave (Next.js, FastAPI, Azure Functions, Azure SQL, Blob Storage, Terraform, etc.).
- Un resumen de las funcionalidades que implementaste en esta asignación (Eliminar, **Enriquecer Reporte**, Muestreo).
- Los desafíos encontrados y cómo los resolviste.
- **Los enlaces a los cinco repositorios de código fuente (Entregable 1).**
- **Las URLs para acceder a la UI y la API desplegadas en Azure (Entregable 3).**

3. **Aplicación Desplegada:** La aplicación completa **debe estar desplegada y funcional en Azure** (App Service, Azure Functions, SQL DB, Storage). Las URLs de acceso se incluirán en la entrada de portafolio.
4. **(Opcional)** Breve demostración en video.

Criterios de Evaluación (Ejemplo)

- Funcionalidad completa y correcta de las 3 tareas en el entorno desplegado.
- Calidad y claridad del código en todos los repositorios.
- Correcta interacción y manejo de errores entre los servicios de Azure.
- Seguridad básica y manejo adecuado de configuraciones/secrets.
- Calidad y profesionalismo de la entrada de portafolio, incluyendo los enlaces requeridos.

¡Mucha suerte extendiendo las capacidades de esta aplicación en la nube y documentando tu trabajo!