

## Informe de Desarrollo - Proyecto Windows Forms

### 1. APIs utilizadas:

En el proyecto utilice dos APIs principales:

- **PokeAPI:** Para obtener información sobre los Pokémon, incluyendo sus nombres y sprites (imágenes oficiales). Esta API RESTful permite acceder a datos detallados de los Pokémon a través de solicitudes HTTP GET, que se procesan para mostrar el nombre y la imagen del Pokémon seleccionado en la aplicación.
- **The Cat API:** Esta API la use para obtener imágenes aleatorias de gatos, añadiendo un componente visual adicional al proyecto. La integración la utilice mediante una llamada HTTP GET, y la imagen se mostró en un PictureBox en la interfaz.

### 2. Dificultades encontradas y soluciones aplicadas:

- **Ubicación de las imágenes:** Inicialmente, las imágenes del Pokémon y del gato se superponían en el panel. Esto lo resolví ajustando las coordenadas de cada PictureBox y añadiendo un título para diferenciarlos.
- **Problemas de carga simultánea:** En ocasiones, al seleccionar un Pokémon y mostrar un gato al mismo tiempo, las imágenes no se actualizaban correctamente. Implementé el método RefreshAllImages() para forzar la actualización simultánea de ambos elementos visuales.
- **Errores en la conversión de datos de la API:** Durante la deserialización del JSON recibido, se presentaron errores de conversión. Esto lo solucione utilizando un casteo explícito y verificaciones adicionales para evitar excepciones nulas.

### 3. Aprendizajes obtenidos:

La tarea permitió profundizar en el manejo de APIs RESTful y su integración con aplicaciones de Windows Forms. Además, reforzó el uso de técnicas asíncronas en C# con async y await, optimizando la carga de datos sin bloquear el hilo principal de la interfaz.

### 4. Apoyo de la AI para resolver dudas:

La AI facilitó la identificación de errores y la optimización del código, proporcionando ejemplos claros y sugerencias sobre cómo manejar problemas específicos, como el manejo de JSON y la correcta actualización de imágenes en el formulario.

La asistencia fue útil para estructurar las consultas a las APIs y corregir problemas de interfaz de usuario que surgieron durante el desarrollo.