**Informe del Proyecto:**

**Consumo de APIs - Pokémon y Gatos Aleatorios**

**Objetivo del Proyecto**

El objetivo de este proyecto fue desarrollar una aplicación de escritorio utilizando **Windows Forms (WinForms)** en C#. La aplicación permite al usuario buscar información sobre distintos Pokémon y visualizar un gato aleatorio, integrando dos APIs públicas: **PokeAPI** y **The Cat API**.

**APIs Utilizadas**

Para la implementación, utilicé dos servicios web:

* **PokeAPI:** Permite obtener una lista de Pokémon y datos detallados como imágenes y habilidades.
* Obtiene una lista de Pokémon desde <https://pokeapi.co/api/v2/pokemon?limit=50>.
* Muestra información detallada de un Pokémon seleccionado, incluyendo su imagen.
* **The Cat API:** Proporciona imágenes aleatorias de gatos.
* Obtiene una imagen aleatoria de un gato desde https://api.thecatapi.com/v1/images/search.

Las peticiones se realizaron mediante HttpClient y se procesaron de manera asíncrona para evitar bloqueos en la interfaz.

**Dificultades y Soluciones**

Durante el desarrollo, enfrenté algunos desafíos:

* **Bloqueo de la interfaz al realizar peticiones:** Lo resolví implementando programación asíncrona con async y await.
* **Errores al cargar imágenes desde URLs incorrectas:** Añadí validaciones para asegurar que las imágenes se cargaran correctamente.
* **Problemas con el manejo de hilos al actualizar la interfaz:** Utilicé métodos seguros para actualizar los componentes gráficos después de recibir la respuesta de las APIs.

**Aprendizajes Adquiridos**

Este proyecto me permitió entender mejor el consumo de APIs REST y la programación asíncrona en C#. Además, fortalecí mis conocimientos en la construcción de interfaces gráficas y el manejo de excepciones para mejorar la experiencia del usuario.

**Apoyo de la Inteligencia Artificial (AI)**

Durante el desarrollo, me apoyé en herramientas de Inteligencia Artificial para:

* Resolver dudas sobre el manejo de hilos y peticiones asíncronas.
* Optimizar el código y organizar mejor la estructura de la aplicación.
* Aprender buenas prácticas en el manejo de excepciones y diseño de interfaces.