Resumen: Gestión de Archivos y Configuración Externa en C#

1. El Archivo App.config y la librería System.Configuration

En el desarrollo profesional, **nunca se deben escribir rutas de carpetas o cadenas de conexión a bases de datos directamente en el código fuente de C#** (práctica conocida como "hardcodear").

- **¿Qué es el App.config?** Es un archivo en formato XML que acompaña a nuestra aplicación. Funciona como un diccionario externo de configuraciones. La gran ventaja es que, si el programa se instala en la computadora de un cliente y la ruta de la carpeta cambia, solo se debe modificar este archivo de texto sin necesidad de abrir Visual Studio ni volver a compilar todo el proyecto.
- **¿Qué es System.Configuration?** Es la librería (referencia) que funciona como el traductor oficial de .NET. Le da a C# la capacidad de leer archivos XML de configuración.
- **¿Cómo se usa?**
    1. En el App.config, dentro de <appSettings>, se crea una clave: <add key="images-folder" value="D:\MiCarpeta\"/>
    2. En C#, se llama a esa clave usando: ConfigurationManager.AppSettings["images-folder"]

2. La librería System.IO (Input / Output)

Por defecto, una aplicación en C# se ejecuta aislada en la memoria RAM por motivos de seguridad. No tiene permiso ni capacidad para interactuar con el disco duro de la computadora.

- **¿Para qué sirve System.IO?** Es el espacio de nombres (namespace) que le otorga a nuestra aplicación las herramientas necesarias para leer, crear, copiar, mover y eliminar archivos y carpetas reales dentro del sistema operativo (Windows).
- Sin esta librería importada en la parte superior del código (using System.IO;), es imposible realizar operaciones físicas con las imágenes del usuario.

3. Selección y Copiado de Imágenes Locales

Para que el usuario pueda subir una imagen a nuestra aplicación, se utiliza una combinación de herramientas visuales y de entrada/salida de datos:

- **OpenFileDialog:** Es un componente nativo que invoca la clásica ventana de exploración de Windows ("Abrir archivo").
    - **Filtros de Seguridad:** Mediante la propiedad .Filter = "Archivos de imagen|*.jpg;*.png", bloqueamos la ventana para que el usuario no pueda seleccionar archivos no deseados (como PDFs o ejecutables).
    - **Validación de Acción:** Siempre se debe encerrar en un if (archivo.ShowDialog() == DialogResult.OK) para garantizar que el código de carga solo se ejecute si el usuario realmente seleccionó un archivo y no cerró la ventana.
- **Diferencia de Rutas:**
    - .FileName: Devuelve la ruta absoluta completa (Ej: C:\Fotos\pikachu.jpg).

    - .SafeFileName: Devuelve únicamente el nombre y extensión del archivo (Ej: pikachu.jpg).

- **El guardado local (File.Copy):** Es el método de System.IO que realiza el clonado del archivo. Exige dos parámetros estrictos:
    1. *Origen:* La ruta absoluta donde está la foto actualmente (archivo.FileName).
    2. *Destino:* La ruta de nuestra carpeta de proyecto (ConfigurationManager...) concatenada con el nombre del archivo original (archivo.SafeFileName).