# Documentación del Proyecto: Generación de Archivos con OpenAI y OpenXML

## 1. Introducción

Este documento describe el desarrollo de una aplicación de escritorio en C# que permite realizar investigaciones automatizadas a través de la API de OpenAI, y generar automáticamente documentos de Word, presentaciones en PowerPoint y almacenar los resultados en una base de datos local (SQL Server).

## 2. Objetivo del Proyecto

Automatizar la investigación asistida por inteligencia artificial utilizando OpenAI, generando resultados que se guardan en documentos y presentaciones con soporte de OpenXML, y almacenarlos en una base de datos SQL.

## 3. Tecnologías y APIs Utilizadas

- API de OpenAI: Para generar texto con el modelo 'gpt-4o-mini'.  
- OpenXML SDK: Para crear documentos Word (.docx) y presentaciones PowerPoint (.pptx).  
- .NET Framework / Windows Forms: Para la interfaz gráfica del usuario.  
- SQL Server Express: Base de datos local para almacenar el historial de investigaciones.  
- System.Net.Http y System.Text.Json: Para consumir APIs REST y procesar respuestas JSON.

## 4. Flujo del Usuario

1. El usuario ingresa una consulta en un campo de texto.  
2. Al presionar el botón de búsqueda, se realiza la solicitud a la API de OpenAI.  
3. Se recibe la respuesta y se muestra en un cuadro de texto.  
4. Se crea una carpeta en el escritorio (si no existe) para guardar los archivos.  
5. Se generan automáticamente un documento Word y una presentación PowerPoint con el resultado.  
6. Finalmente, se guarda la consulta y respuesta en la base de datos local.

## 5. Librerías y Paquetes NuGet

- DocumentFormat.OpenXml  
- System.Net.Http  
- System.Text.Json  
- System.Data.SqlClient  
- System.Windows.Forms (interfaz gráfica)  
- Microsoft.Data.SqlClient (si se actualiza a .NET moderno)

## 6. Problemas y Soluciones

- \*\*Conflictos con clases 'Path'\*\*: Solucionado usando `System.IO.Path` para evitar ambigüedad.  
- \*\*Errores en creación de PowerPoint\*\*: Resuelto configurando correctamente los nodos de `SlideMaster`, `SlideLayout`, y `ShapeTree`.  
- \*\*Persistencia de la API Key\*\*: Se recomienda usar una variable de entorno o archivo seguro, aunque en esta versión está incrustada para pruebas.  
- \*\*Creación automática de carpetas\*\*: Manejado mediante verificación con `Directory.Exists`.  
- \*\*Inserción a base de datos\*\*: Controlado mediante uso de `SqlCommand` y parámetros para evitar SQL Injection.

## 7. Cómo se Elaboró

Con el apoyo de ChatGPT, se desarrolló una clase en C# con métodos organizados que permiten integrar de forma sencilla la funcionalidad de consulta a OpenAI, creación de archivos Word y PowerPoint, y conexión a base de datos. El asistente proporcionó soluciones a errores específicos en la manipulación de OpenXML y sugerencias de buenas prácticas para la implementación.