

## Documento como elabore mi Proyecto Final (1)

---

### 1. Objetivo del Proyecto

Desarrollar una aplicación de escritorio en C# (.NET Framework con Windows Forms) que utiliza la API gratuita de Mistral para realizar investigaciones automáticas, guardar resultados en una base de datos SQL Server y generar archivos Word y PowerPoint automáticamente.

### 2. Componentes Usados

- Lenguaje de programación: C#
- Framework: .NET Framework 4.7.2
- Entorno: Visual Studio 2022
- Librería: Newtonsoft.Json para manejo de JSON
- Base de datos: SQL Server
- Integración de Office: Microsoft Word 16.0 Object Library
- Librerías externas:
  - o **Newtonsoft.Json** (para manejar el formato JSON)
  - o **Microsoft.Office.Interop.Word** (para generar documentos Word)
  - o **Microsoft.Office.Interop.PowerPoint**(para generar presentaciones en PowerPoint)

### 3. API Utilizada

Se utilizó la API de OpenRouter.ai, específicamente el modelo:

mistralai/mistral-7b-instruct:free

Este modelo permite enviar preguntas mediante HTTP y recibir respuestas generadas por inteligencia. Se utiliza un endpoint de tipo POST con autorización mediante una API Key.

### 4. Funcionalidades Implementadas

- Consulta automática de temas usando Mistral API

- Almacenamiento de los resultados en SQL Server
- Generación automática de documentos Word con los resultados
- (En desarrollo: generación automática de presentaciones PowerPoint)

## 5. Ejemplo de código (Una parte nada mas)

Este método toma el resultado generado por Mistral y crea un documento Word:

```
private void GenerarDocumentoWord()
{
    string connectionString = @"Server=LAPTOP-
R27RH76P\SQLEXPRESS01;Database=Para_Proyecto_1;Trusted_Connection
=True;";
    string query = "SELECT TOP 1 * FROM Investigaciones ORDER BY Id
DESC";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        SqlCommand command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();

        if (reader.Read())
        {
            string tema = reader["Tema"].ToString();
            string descripcion = reader["Descripcion"].ToString();
            string resultado = reader["ResultadoIA"].ToString();

            var wordApp = new Word.Application();
            var documento = wordApp.Documents.Add();
            var parrafo = documento.Content.Paragraphs.Add();

            parrafo.Range.Text = "Reporte generado por IA";
            parrafo.Range.InsertParagraphAfter();

            parrafo.Range.Text = $"Tema: {tema}";
            parrafo.Range.InsertParagraphAfter();

            parrafo.Range.Text = $"Descripción: {descripcion}";
```

```

        parrafo.Range.InsertParagraphAfter();

        parrafo.Range.Text = $"Resultado generado por IA:\n{resultado}";
        parrafo.Range.InsertParagraphAfter();

        string carpeta = @"C:\Users\Public\ReportesIA";

        if (!Directory.Exists(carpeta))
        {
            Directory.CreateDirectory(carpeta);
        }

        string ruta = @"C:\Users\Public\ReporteIA.docx";
        documento.SaveAs2(ruta);
        documento.Close();
        wordApp.Quit();

        System.Diagnostics.Process.Start("explorer.exe", carpeta);

        System.Diagnostics.Process.Start("winword.exe", ruta);

        MessageBox.Show("¡Documento Word creado con éxito en:\n" + ruta +
"!");
    }
    else
    {
        MessageBox.Show("No se encontraron datos en la base.");
    }
}
}
}

```

## **6. Conclusión**

Este proyecto demuestra cómo se puede integrar una API de inteligencia artificial con una aplicación de escritorio, base de datos y generación de documentos. Es una solución ideal para tareas académicas o de oficina que requieren recopilación rápida de información.