**ClosedXML**

**ObtenerEncabezados**

// Función para obtener los encabezados de un archivo Excel.

// Esta función abre el archivo, lee la primera fila y devuelve una lista de los nombres de las columnas.

public List<string> ObtenerEncabezados(string filePath)

{

var encabezados = new List<string>();

// Abre el archivo Excel en la ruta especificada.

using (var workbook = new XLWorkbook(filePath))

{

// Obtiene la primera hoja del archivo.

var worksheet = workbook.Worksheet(1);

// Encuentra la última columna utilizada en la primera fila para determinar cuántos encabezados hay.

int lastColumnUsed = worksheet.LastColumnUsed().ColumnNumber();

// Recorre las columnas en la primera fila y obtiene el nombre de cada columna.

for (int col = 1; col <= lastColumnUsed; col++)

{

// Obtiene el valor de la celda como texto.

string encabezado = worksheet.Cell(1, col).GetValue<string>();

encabezados.Add(encabezado);

}

}

return encabezados;

}

**ObtenerDatos**

// Función para obtener los datos de un archivo Excel y mapearlos a una lista de objetos de MyExcelClass.

// Esta función abre el archivo, lee cada fila y mapea los valores a las propiedades de MyExcelClass.

public List<MyExcelClass> ObtenerDatos(string filePath)

{

var dataList = new List<MyExcelClass>();

// Abre el archivo Excel en la ruta especificada.

using (var workbook = new XLWorkbook(filePath))

{

// Selecciona la primera hoja del archivo.

var worksheet = workbook.Worksheet(1);

// Encuentra la última fila utilizada en la hoja.

int lastRowUsed = worksheet.LastRowUsed().RowNumber();

// Recorre cada fila de datos, comenzando desde la segunda fila (omitiendo los encabezados).

for (int row = 2; row <= lastRowUsed; row++)

{

// Crea una instancia de MyExcelClass y asigna valores a sus propiedades desde las celdas de la fila actual.

var dataItem = new MyExcelClass

{

ID = worksheet.Cell(row, 1).GetValue<string>(), // Lee el valor de la columna 1

Nombre = worksheet.Cell(row, 2).GetValue<string>(), // Lee el valor de la columna 2

Edad = worksheet.Cell(row, 3).GetValue<int>() // Lee el valor de la columna 3

};

// Agrega el objeto mapeado a la lista de resultados.

dataList.Add(dataItem);

}

}

return dataList;

}

**InsertarDatos**

// Función para insertar nuevos datos en un archivo Excel existente.

// Esta función abre el archivo, encuentra la última fila utilizada e inserta los nuevos datos a partir de esa posición.

public void InsertarDatos(string filePath, List<MyExcelClass> nuevosDatos)

{

// Abre el archivo Excel en la ruta especificada.

using (var workbook = new XLWorkbook(filePath))

{

// Selecciona la primera hoja del archivo.

var worksheet = workbook.Worksheet(1);

// Encuentra el número de la última fila utilizada en la hoja.

int lastRowUsed = worksheet.LastRowUsed().RowNumber();

// Recorre la lista de objetos y los inserta como nuevas filas en el archivo Excel.

foreach (var item in nuevosDatos)

{

lastRowUsed++; // Mueve a la siguiente fila disponible

// Inserta los valores en las columnas correspondientes de la nueva fila.

worksheet.Cell(lastRowUsed, 1).Value = item.ID;

worksheet.Cell(lastRowUsed, 2).Value = item.Nombre;

worksheet.Cell(lastRowUsed, 3).Value = item.Edad;

}

// Guarda los cambios en el archivo.

workbook.Save();

}

Console.WriteLine("Datos insertados exitosamente en el archivo Excel.");

}

}

**ActualizarDatosPorId**

// Función para actualizar datos en el archivo Excel según ID

public void ActualizarDatosPorId(string filePath, MyExcelClass datosActualizados)

{

using (var workbook = new XLWorkbook(filePath))

{

var worksheet = workbook.Worksheet(1); // Selecciona la primera hoja

int lastRowUsed = worksheet.LastRowUsed().RowNumber();

bool encontrado = false;

// Buscar la fila donde el ID coincide y actualizar los datos

for (int row = 2; row <= lastRowUsed; row++)

{

string idActual = worksheet.Cell(row, 1).GetValue<string>();

if (idActual == datosActualizados.ID)

{

worksheet.Cell(row, 2).Value = datosActualizados.Nombre;

worksheet.Cell(row, 3).Value = datosActualizados.Edad;

encontrado = true;

break;

}

}

if (encontrado)

{

workbook.Save(); // Guarda los cambios en el archivo

Console.WriteLine("Datos actualizados exitosamente en el archivo Excel.");

}

else

{

Console.WriteLine("No se encontró un registro con el ID especificado.");

}

}

}