



Visual Studio

Herramientas de desarrollo

Descargas

Comprar

Suscripciones

Visual Studio gratuito

Todo Microsoft

Buscar

Sign in



Descargas



Visual Studio 2022 |

El IDE más completo para desarrolladores de .NET y C++ en Windows para crear aplicaciones web, en la nube, de escritorio, móviles, servicios y juegos.

Versión preliminar

Obtenga acceso anticipado a las características más recientes que aún no están en la versión principal

[Obtener más información →](#)

Community

IDE eficaz, gratuito para estudiantes, colaboradores de código abierto e individuos

[Descarga gratuita](#)

Professional

IDE profesional más adecuado para equipos

[Prueba gratuita](#)

Enterprise

Escalable, solución integral para equipos de cualquier tamaño

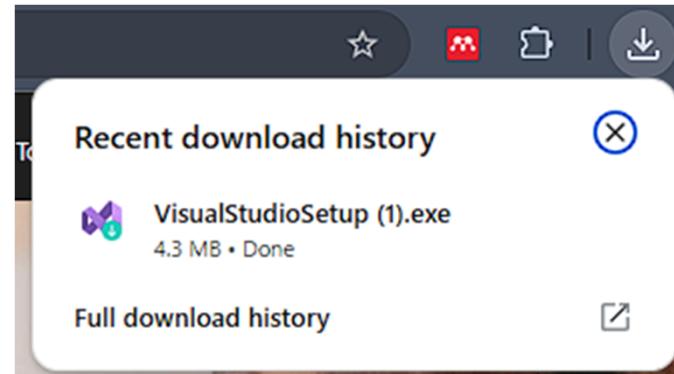
[Prueba gratuita](#)

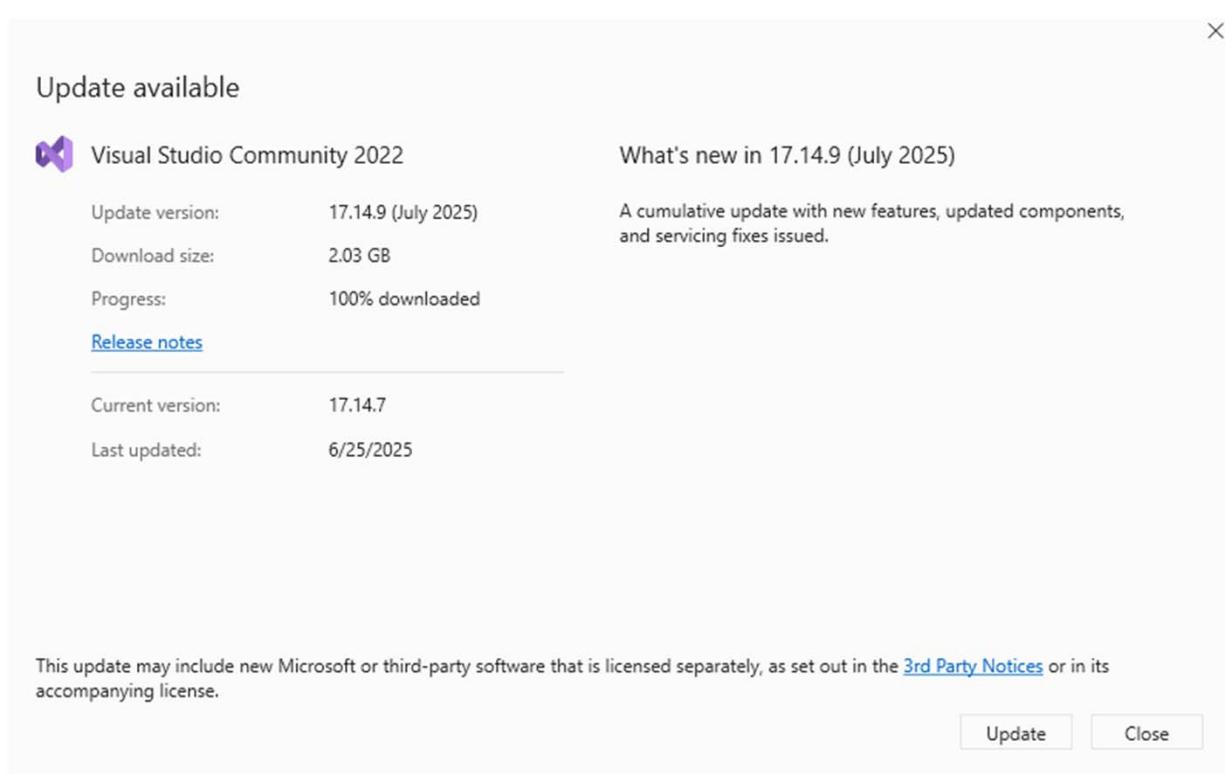
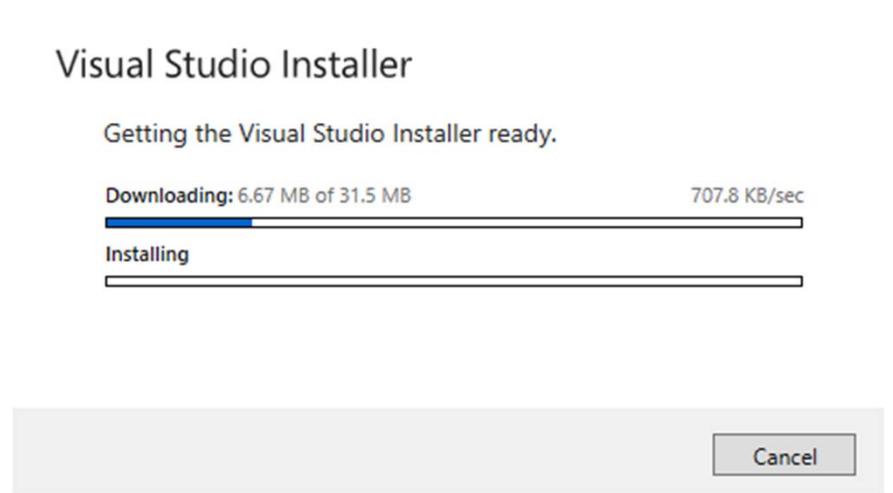
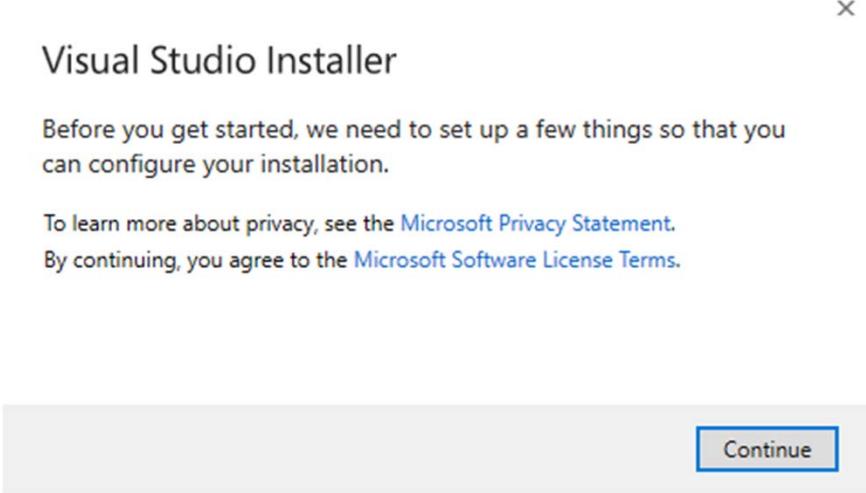
[Notas de la versión →](#)

[Comparar ediciones →](#)

[Cómo instalarlo sin conexión →](#)

[Términos de licencia →](#)





[Workloads](#) [Individual components](#) [Language packs](#) [Installation locations](#) Need help choosing what to install? [More info](#)

Web & Cloud (4)



ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...



Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....



Python development

Editing, debugging, interactive development and source control for Python.



Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)



.NET Multi-platform App UI development

Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.



.NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

Location

C:\Program Files\Microsoft Visual Studio\2022\Community [Change...](#)

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Installation details

.NET desktop development

Included

- ✓ .NET desktop development tools
- ✓ .NET Framework 4.7.2 development tools
- ✓ C# and Visual Basic

Optional

- ✓ Development tools for .NET
- ✓ .NET Framework 4.8 development tools
- ✓ Entity Framework 6 tools
- ✓ .NET profiling tools
- ✓ IntelliCode
- ✓ Just-In-Time debugger
- ✓ Live Share
- ✓ ML.NET Model Builder
- ✓ Blend for Visual Studio
- F# desktop language support
- PreEmptive Protection - Dotfuscator
- .NET Framework 4.6.2-4.7.1 development t...
- .NET Portable Library targeting pack

[Remove out-of-support components](#)

Total space required 8.97 GB

[Install while downloading](#) [Install](#)

Visual Studio 2022

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

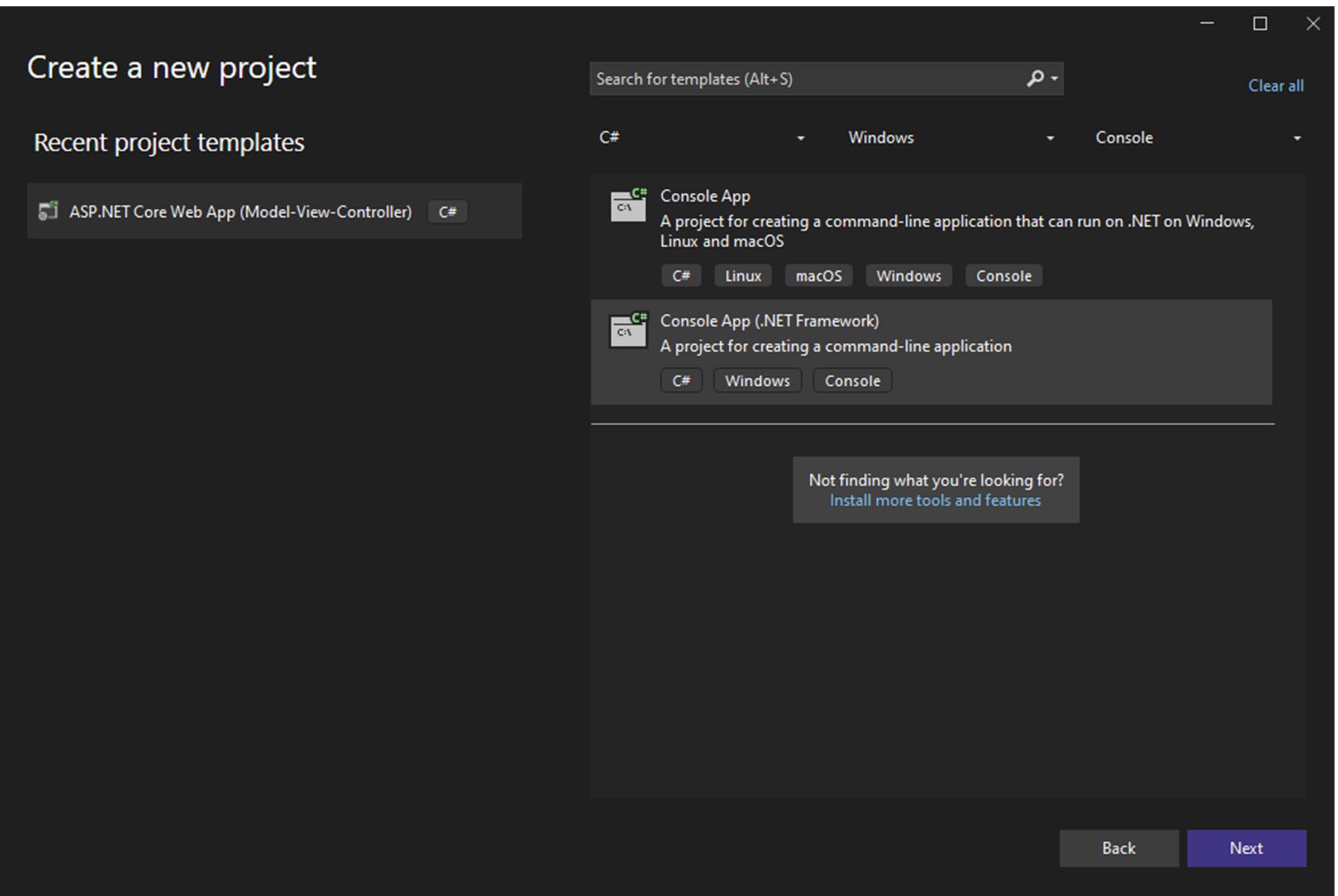
Navigate and edit code within any folder

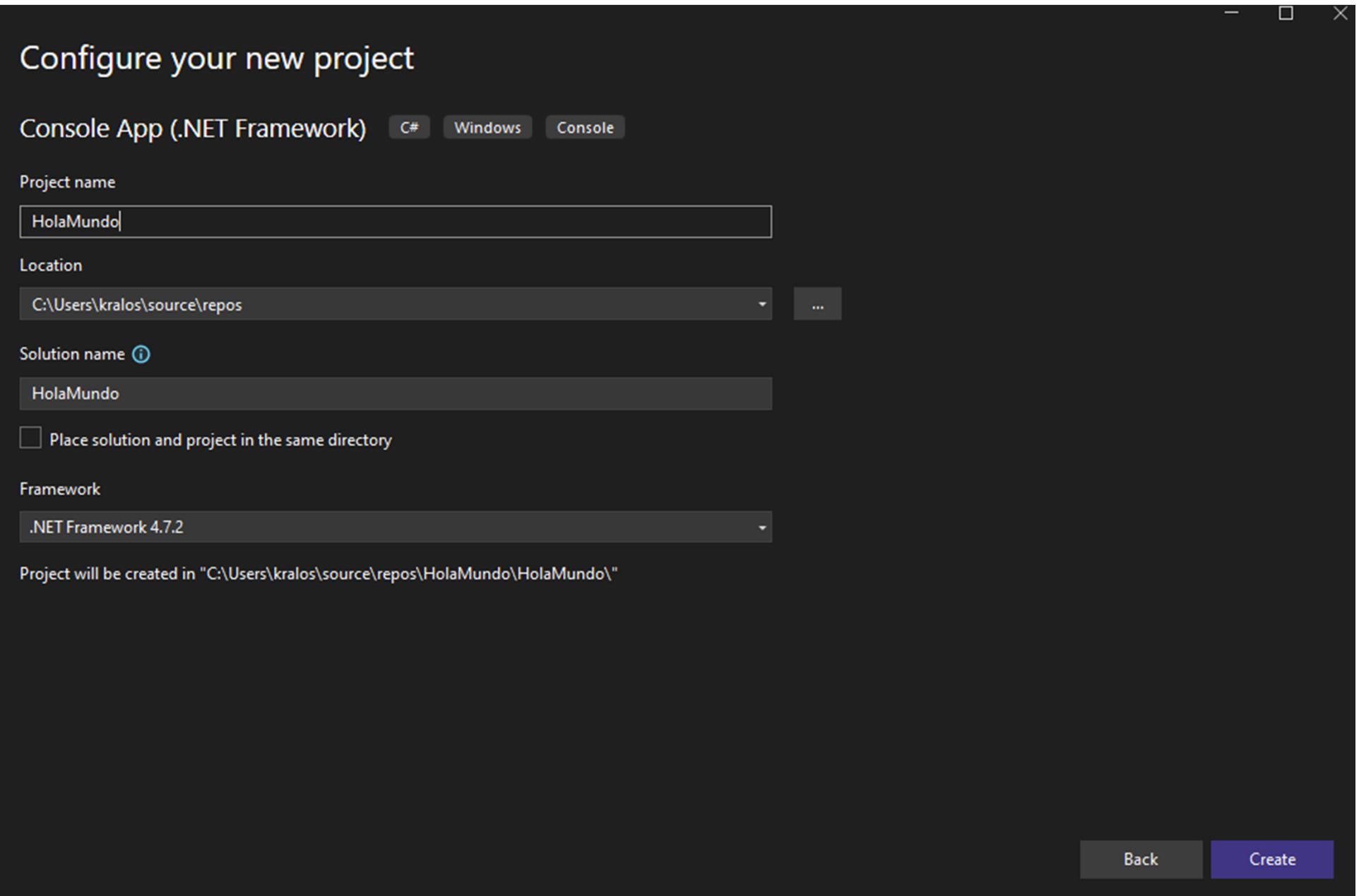


Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)





Configure your new project

Console App (.NET Framework)

C#

Windows

Console

Project name

HolaMundo

Location

C:\Users\kralos\source\repos

...

Solution name ⓘ

HolaMundo

Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Project will be created in "C:\Users\kralos\source\repos\HolaMundo\HolaMundo\"

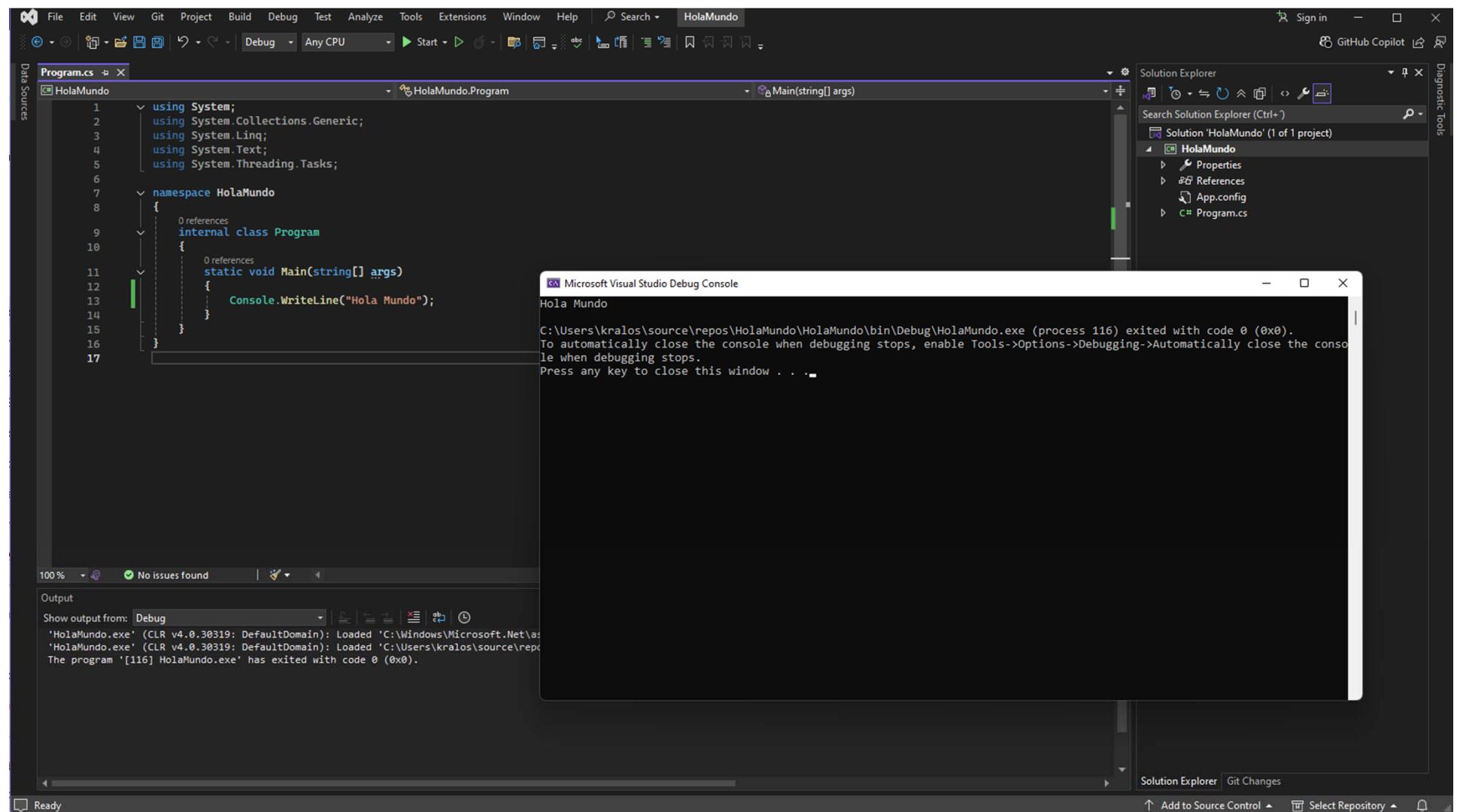
Back

Create

Sobre el código

```
using System;           // Importa funcionalidades básicas como Console
using System.Collections.Generic; // Colecciones genéricas como List<>
using System.Linq;        // Funciones LINQ para colecciones
using System.Text;        // Funciones para trabajar con texto
using System.Threading.Tasks; // Para tareas asincrónicas y concurrencia

namespace HolaMundo    // Define un espacio de nombres llamado HolaMundo
{
    internal class Program // Clase principal llamada Program
    {
        static void Main(string[] args) // Punto de entrada de la aplicación
        {
        }
    }
}
```



The screenshot shows a Windows File Explorer interface with two main panes and a PowerShell window at the bottom.

Top Left Pane: Shows the directory structure: kralos > source > repos > HolaMundo. The "HolaMundo" folder is selected.

Top Right Pane: Shows the contents of the selected "HolaMundo" folder. The table below lists the files and folders:

Name	Date modified	Type	Size
bin	7/28/2025 6:12 PM	File folder	
obj	7/28/2025 6:12 PM	File folder	
Properties	7/28/2025 6:12 PM	File folder	
App	7/28/2025 6:12 PM	Configuration Sou...	1 KB
HolaMundo	7/28/2025 6:12 PM	C# Project file	3 KB
Program	7/28/2025 6:14 PM	C# Source File	1 KB

Bottom Left Pane: Shows the directory structure: kralos > source > repos > HolaMundo > HolaMundo > bin > Debug. The "Debug" folder is selected.

Bottom Right Pane: Shows the contents of the selected "Debug" folder. The table below lists the files:

Name	Date modified	Type	Size
HolaMundo	7/28/2025 6:14 PM	Application	5 KB
HolaMundo.exe	7/28/2025 6:12 PM	Configuration Sou...	1 KB
HolaMundo.pdb	7/28/2025 6:14 PM	Program Debug D...	20 KB

PowerShell Window: Displays the following output:

```
PowerShell 7.5.2
Loading personal and system profiles took 3101ms.
[kralos]
[~\source\repos\HolaMundo\HolaMundo\bin\Debug]
> ls

Directory: C:\Users\kralos\source\repos\HolaMundo\HolaMundo\bin\Debug

Mode                LastWriteTime       Length Name
----                -              ----- 
-a---        7/28/2025  6:14 PM      4608  HolaMundo.exe
-a---        7/28/2025  6:12 PM       189  HolaMundo.exe.config
-a---        7/28/2025  6:14 PM     19968  HolaMundo.pdb

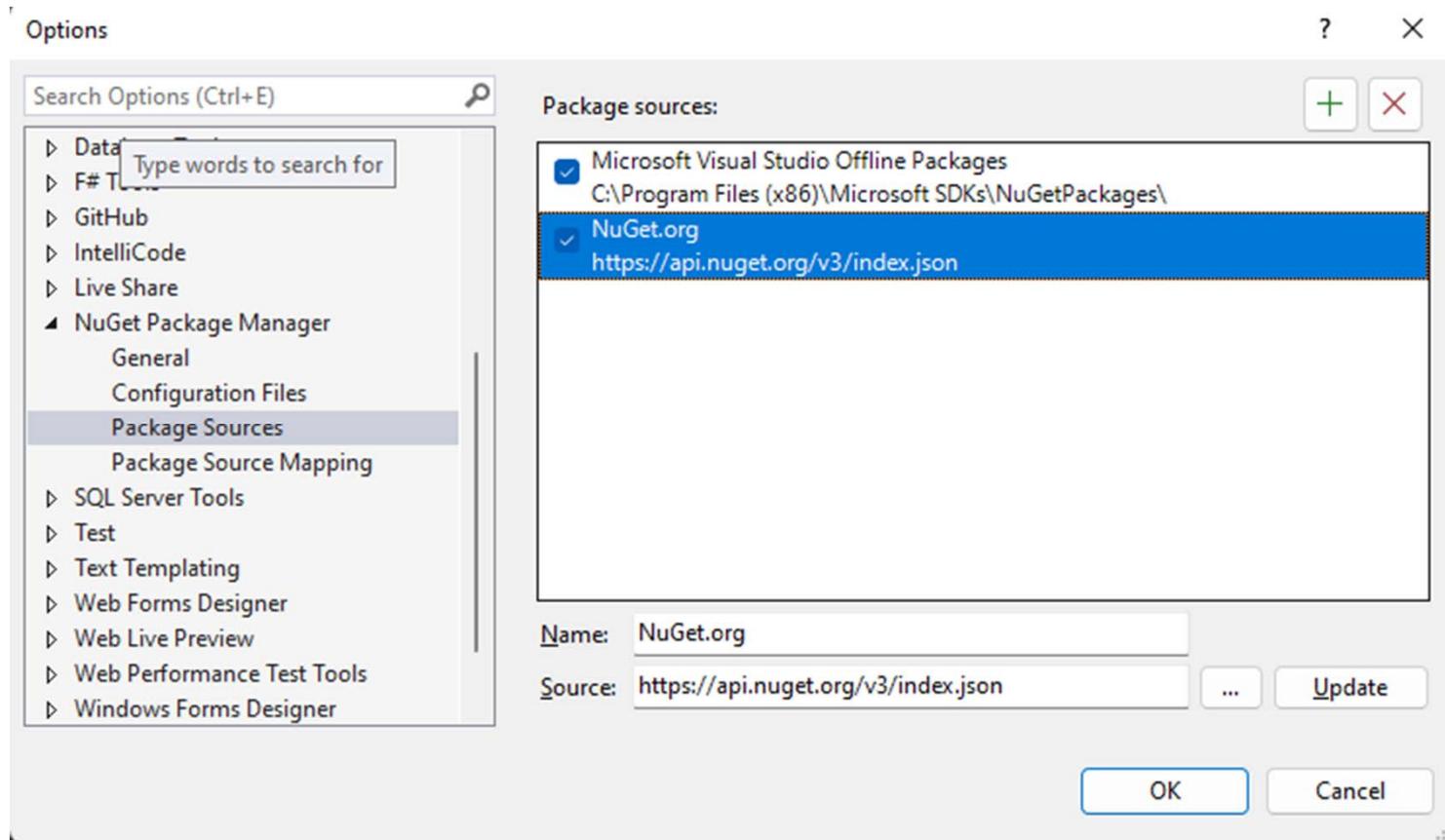
[kralos]
[~\source\repos\HolaMundo\HolaMundo\bin\Debug]
> .\HolaMundo.exe
Hola Mundo
```

C# namespace vs Java package

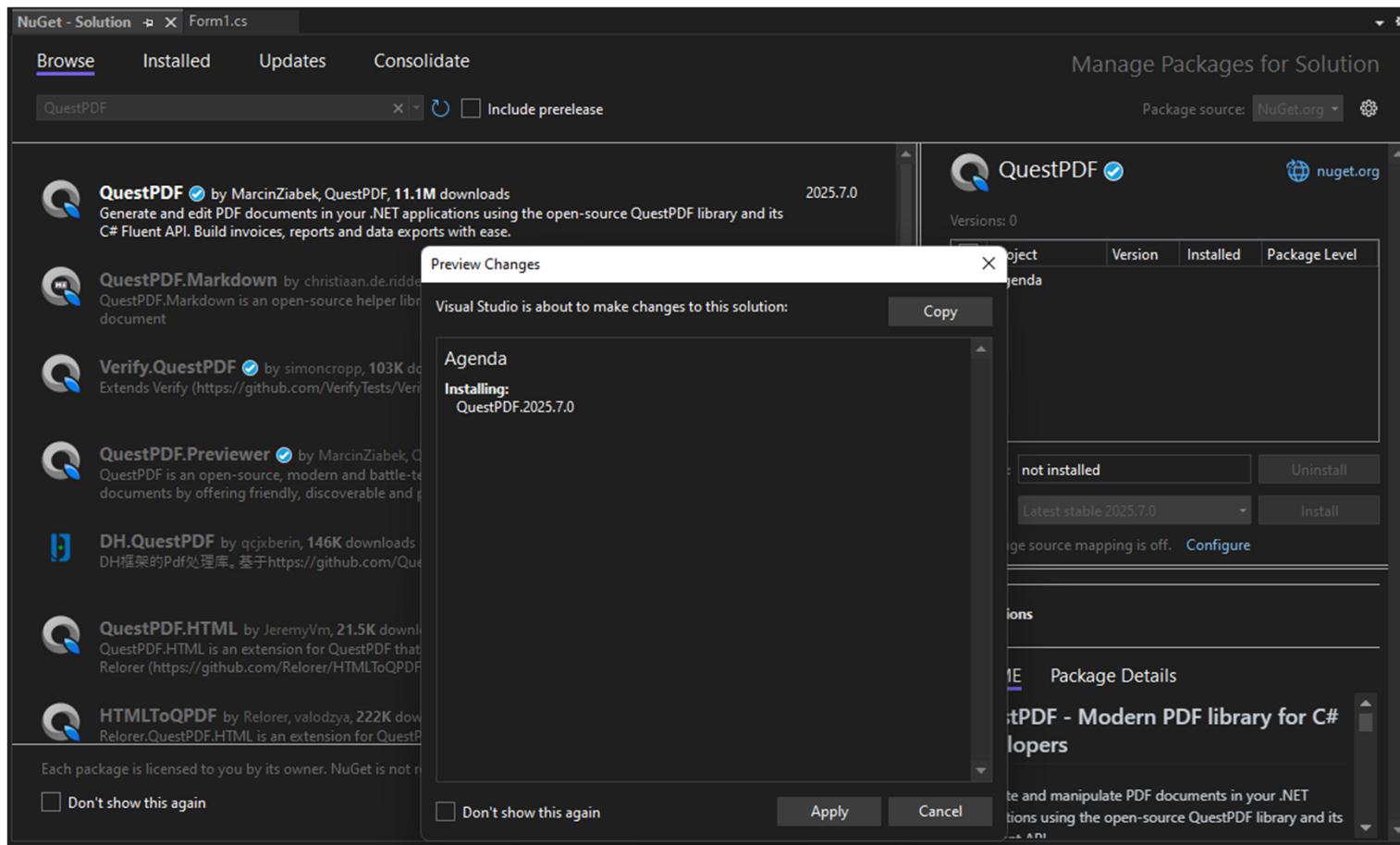
Característica	namespace en C#	package en Java
Propósito	Organizar clases, evitar conflictos	Igual: organizar y evitar conflictos
Sintaxis	<code>namespace MiAplicacion.Utilidades { ... }</code>	<code>package miaplicacion.utilidades;</code>
Relación con estructura de carpetas	No obligatoria, pero recomendable	Obligatoria: debe coincidir con estructura
Importación	<code>using MiAplicacion.Utilidades;</code>	<code>import miaplicacion.utilidades.*;</code>
Accesibilidad predeterminada	<code>internal</code> para clases sin modificador	<code>package-private</code> (sin modificador)
Nombre de archivo	Puede diferir del nombre de la clase	El archivo debe llamarse como la clase pública
Alias de importación	Sí, con <code>using alias = namespace;</code>	No directamente, se usa <code>static import</code> en algunos casos
Compilación	No requiere estructura de carpetas específica	Requiere que la carpeta refleje el <code>package</code>

Aspecto	C#	Java
Sintaxis principal	<code>Console.WriteLine(...)</code>	<code>System.out.println(...)</code>
Estructura del programa	Usa <code>namespace</code>	Usa <code>class</code> , opcionalmente <code>package</code>
Método principal	<code>static void Main(string[] args)</code>	<code>public static void main(String[] args)</code>
Nombre del archivo	No necesita coincidir con la clase	Debe coincidir con el nombre de la clase
Compilación	<code>csc archivo.cs</code> o <code>dotnet run</code>	<code>javac archivo.java</code> y luego <code>java</code>
Requiere entorno	.NET SDK o Visual Studio	JDK (Java Development Kit)

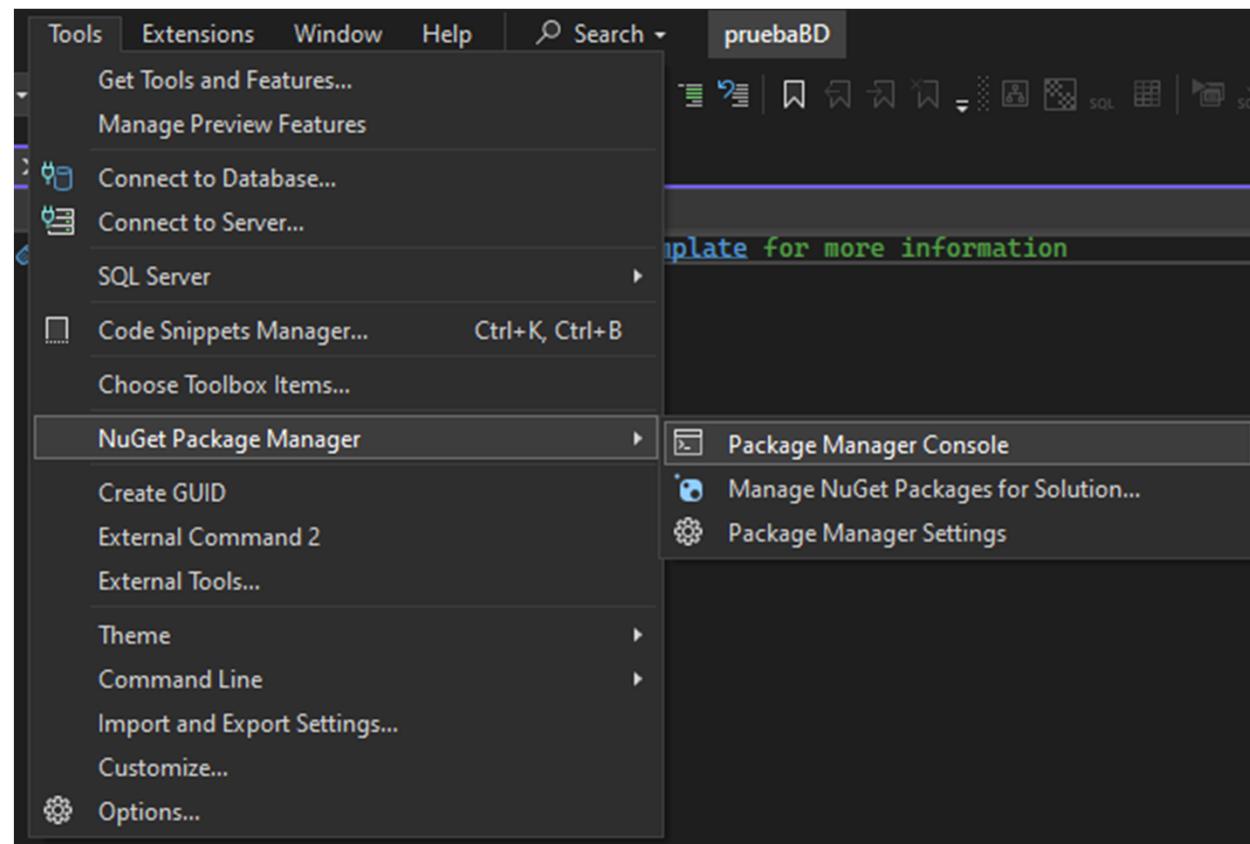
Instalar QuestPDF en .NET 8



Instalar QuestPDF en .NET 8



Instalar MySql.Data



Instalar MySql.Data

```
Package Manager Console
Package source: All | Default project: pruebaBD | X | Y
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses
dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine
Package Manager Console Host Version 6.14.1.1
Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package MySql.Data
Restoring packages for C:\Users\kralos\source\repos\pruebaBD\pruebaBD\pruebaBD.csproj...
GET https://api.nuget.org/v3-flatcontainer/mysql.data/index.json
OK https://api.nuget.org/v3-flatcontainer/mysql.data/index.json 116ms
GET https://api.nuget.org/v3-flatcontainer/mysql.data/9.4.0/mysql.data.9.4.0.nupkg
OK https://api.nuget.org/v3-flatcontainer/mysql.data/9.4.0/mysql.data.9.4.0.nupkg 61ms
GET https://api.nuget.org/v3-flatcontainer/bouncycastle.cryptography/index.json
=====
Successfully installed 'System.Runtime.Loader 4.3.0' to pruebaBD
Successfully installed 'System.Security.Cryptography.ProtectedData 8.0.0' to pruebaBD
Successfully installed 'System.Security.Permissions 8.0.0' to pruebaBD
Successfully installed 'System.Text.Encoding 4.3.0' to pruebaBD
Successfully installed 'System.Text.Encoding.CodePages 8.0.0' to pruebaBD
Successfully installed 'System.Text.Encodings.Web 9.0.0' to pruebaBD
Successfully installed 'System.Text.Json 9.0.0' to pruebaBD
Successfully installed 'System.Threading.Tasks 4.3.0' to pruebaBD
Successfully installed 'System.Threading.Tasks.Extensions 4.5.4' to pruebaBD
Successfully installed 'System.Windows.Extensions 8.0.0' to pruebaBD
Successfully installed 'ZstdSharp.Port 0.8.5' to pruebaBD
Executing nuget actions took 790 ms
Time Elapsed: 00:00:17.8329857
PM>
```

XAMPP

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	7840 14512	80, 443	Stop	Admin	Config	Logs
	MySQL	7812	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

Modules

Config Netstat Shell Explorer Services Help Quit

```
4:13:26 PM [main] All prerequisites found
4:13:26 PM [main] Initializing Modules
4:13:26 PM [main] Starting Check-Timer
4:13:26 PM [main] Control Panel Ready
4:13:30 PM [mysql] Attempting to start MySQL app...
4:13:31 PM [mysql] Status change detected: running
4:23:55 PM [Apache] Attempting to start Apache app...
4:23:56 PM [Apache] Status change detected: running
```

phpMyAdmin

The screenshot shows the phpMyAdmin interface running on a local host at 127.0.0.1. The left sidebar lists databases: New, information_schema, mysql, performance_schema, phpmyadmin, sistore, and test. The main content area is divided into several sections:

- General settings:** Server connection collation is set to utf8mb4_unicode_ci.
- Appearance settings:** Language is English, and the theme is pmahomme.
- Database server:** Details include:
 - Server: 127.0.0.1 via TCP/IP
 - Server type: MariaDB
 - Server connection: SSL is not being used
 - Server version: 10.4.32-MariaDB - mariadb.org binary distribution
 - Protocol version: 10
 - User: root@localhost
 - Server charset: UTF-8 Unicode (utf8mb4)
- Web server:** Details include:
 - Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
 - Database client version: libmysql - mysqlnd 8.2.12
 - PHP extension: mysqli curl mbstring
 - PHP version: 8.2.12
- phpMyAdmin:** Version information: 5.2.1, latest stable version: 5.2.2. Links include Documentation, Official Homepage, Contribute, Get support, List of changes, and License.

phpMyAdmin

The screenshot shows the phpMyAdmin interface running on a local server at 127.0.0.1. The left sidebar displays a tree view of databases and tables, including 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'sistore', and 'test'. The 'sistore' database is currently selected, showing its 'usuarios' table. The main panel features a toolbar with tabs for 'Databases', 'SQL', 'Status', 'User accounts', 'Export', and 'Import', with 'SQL' being the active tab. Below the toolbar is a search bar labeled 'Run SQL query/queries on server "127.0.0.1":'. The SQL query editor contains the following code:

```
1 CREATE DATABASE prueba;
2 USE prueba;
3
4 CREATE TABLE usuarios (
5     id INT AUTO_INCREMENT PRIMARY KEY,
6     nombre VARCHAR(100)
7 );
8
9 INSERT INTO usuarios (nombre) VALUES ('Carlos'), ('Ana'), ('Luis');
10
```

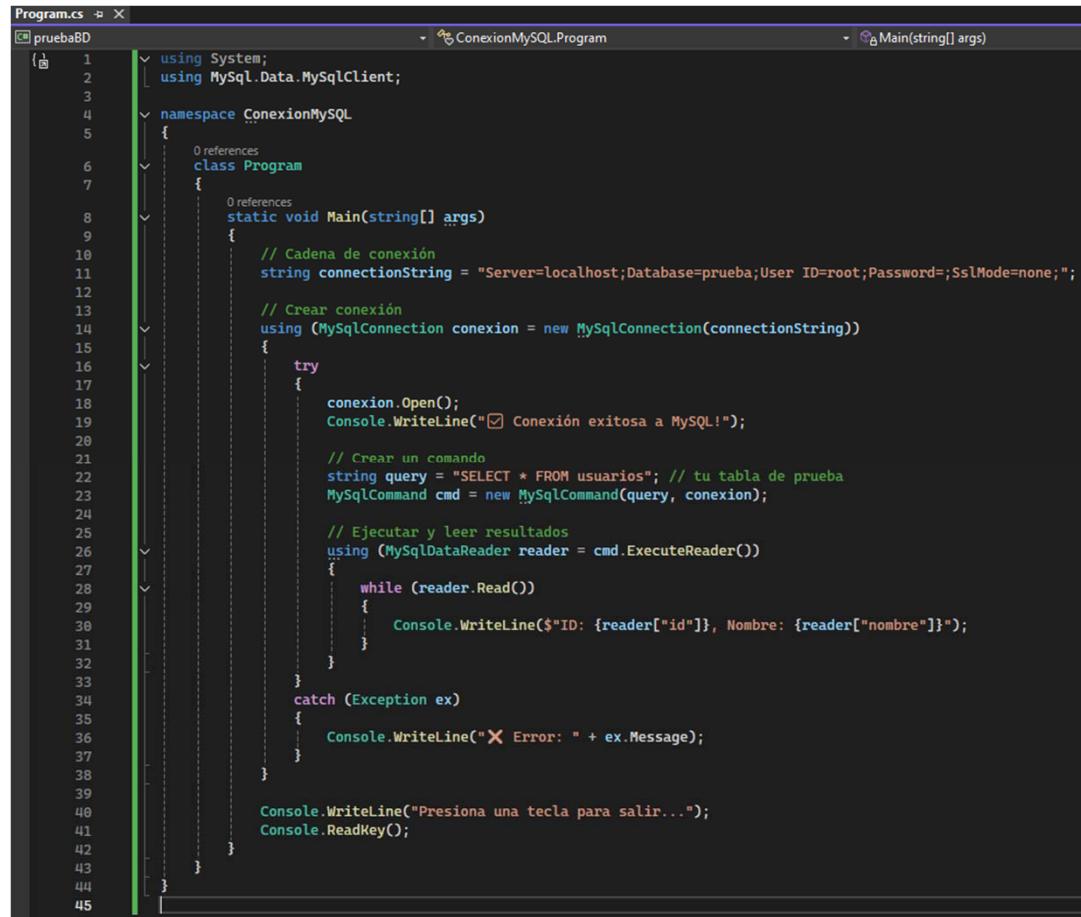
phpMyAdmin

The screenshot shows the phpMyAdmin interface running on a local server at 127.0.0.1. The current database selected is 'prueba'. The main window displays the 'Structure' tab for the 'usuarios' table. The table has 3 rows and is defined as InnoDB with utf8mb4_general_ci collation. The table structure includes columns for id, nombre, and apellido.

id	nombre	apellido
1	John	Doe
2	Jane	Doe
3	John	Doe

The left sidebar shows a tree view of databases: information_schema, mysql, performance_schema, phpmyadmin, prueba, sistore, and test. The 'prueba' database is expanded, showing its tables: New, usuarios, and New.

Prueba de conexión



The screenshot shows a Microsoft Visual Studio interface with the following details:

- Project Explorer:** Shows a project named "pruebaBD".
- Code Editor:** Displays the file "Program.cs" with the following code:

```
1  using System;
2  using MySql.Data.MySqlClient;
3
4  namespace ConexionMySQL
5  {
6      class Program
7      {
8          static void Main(string[] args)
9          {
10             // Cadena de conexión
11             string connectionString = "Server=localhost;Database=prueba;User ID=root;Password=;SslMode=None";
12
13             // Crear conexión
14             using (MySqlConnection conexion = new MySqlConnection(connectionString))
15             {
16                 try
17                 {
18                     conexion.Open();
19                     Console.WriteLine("✅ Conexión exitosa a MySQL!");
20
21                     // Crear un comando
22                     string query = "SELECT * FROM usuarios"; // tu tabla de prueba
23                     MySqlCommand cmd = new MySqlCommand(query, conexion);
24
25                     // Ejecutar y leer resultados
26                     using (MySqlDataReader reader = cmd.ExecuteReader())
27                     {
28                         while (reader.Read())
29                         {
30                             Console.WriteLine($"ID: {reader["id"]}, Nombre: {reader["nombre"]}");
31                         }
32                     }
33                 }
34             catch (Exception ex)
35             {
36                 Console.WriteLine("✖ Error: " + ex.Message);
37             }
38
39             Console.WriteLine("Presiona una tecla para salir...");
40             Console.ReadKey();
41
42         }
43     }
44 }
45 }
```

- Output Window:** Titled "Microsoft Visual Studio Debug Console", it displays the following output:

```
? Conexión exitosa a MySQL!
ID: 1, Nombre: Carlos
ID: 2, Nombre: Ana
ID: 3, Nombre: Luis
Presiona una tecla para salir...
```

Instala paquete para SQLite

```
Package Manager Console
Package source: All | Default project: pruebaSQLite
dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 6.14.1.1
Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package System.Data.SQLite
Restoring packages for C:\Users\kralos\source\repos\pruebaSQLite\pruebaSQLite\pruebaSQLite.csproj...
GET https://api.nuget.org/v3-flatcontainer/system.data.sqlite/index.json
OK https://api.nuget.org/v3-flatcontainer/system.data.sqlite/index.json 212ms
GET https://api.nuget.org/v3-flatcontainer/system.data.sqlite/2.0.1/system.data.sqlite.2.0.1.nupkg
OK https://api.nuget.org/v3-flatcontainer/system.data.sqlite/2.0.1/system.data.sqlite.2.0.1.nupkg 293ms
Installed System.Data.SQLite 2.0.1 from https://api.nuget.org/v3/index.json to C:\Users\kralos\.nuget\packages\system.data.sqlite\2.0.1 with content hash x1WBV7ADvWQD7vvupe+S4b7MTj3YMR14iysDvgclt+i1AZDdpIox0WsIplvzK+eapmegQYuKV8lQhf4o8b5Q==.
CACHE https://api.nuget.org/v3/vulnerabilities/index.json
CACHE https://api.nuget.org/v3-vulnerabilities/2025.07.31.17.40.39/vulnerability.base.json
CACHE https://api.nuget.org/v3-vulnerabilities/2025.07.31.17.40.39/2025.08.19.17.18.38/vulnerability.update.json
Installing NuGet package System.Data.SQLite 2.0.1.
Writing assets file to disk. Path: C:\Users\kralos\source\repos\pruebaSQLite\pruebaSQLite\obj\project.assets.json
Restored C:\Users\kralos\source\repos\pruebaSQLite\pruebaSQLite\pruebaSQLite.csproj (in 1.17 sec).
Successfully installed 'System.Data.SQLite 2.0.1' to pruebaSQLite
Executing nuget actions took 550 ms
Time Elapsed: 00:00:02.4492401
PM>

PM> Install-Package System.Data.SQLite.Core
Restoring packages for C:\Users\kralos\source\repos\pruebaSQLite\pruebaSQLite\pruebaSQLite.csproj...
GET https://api.nuget.org/v3-flatcontainer/system.data.sqlite.core/index.json
OK https://api.nuget.org/v3-flatcontainer/system.data.sqlite.core/index.json 119ms
GET https://api.nuget.org/v3-flatcontainer/system.data.sqlite.core/1.0.119/system.data.sqlite.core.1.0.119.nupkg
OK https://api.nuget.org/v3-flatcontainer/system.data.sqlite.core/1.0.119/system.data.sqlite.core.1.0.119.nupkg 221ms
GET https://api.nuget.org/v3-flatcontainer/stub.system.data.sqlite.core.netstandard/index.json
OK https://api.nuget.org/v3-flatcontainer/stub.system.data.sqlite.core.netstandard/index.json 223ms
```

Prueba Sqlite

The screenshot shows a Windows desktop environment with three main windows:

- Code Editor (Program.cs):** Displays C# code for connecting to an SQLite database and performing basic operations like creating a table and inserting data.
- File Explorer:** Shows the file path: This PC > Local Disk (C:) > sqlite, with a file named "mi_base" listed.
- DB Browser for SQLite:** A database management tool showing the "usuarios" table with one record: id 1 and nombre Carlos.

```
Program.cs
1  // Ruta al archivo de la base de datos
2  string connectionString = "Data Source=C:\\sqlite\\mi_base.db;Version=3;";
3
4  using (SQLiteConnection conexion = new SQLiteConnection(connectionString))
5  {
6      try
7      {
8          conexion.Open();
9          Console.WriteLine("Conexión exitosa a SQLite!");
10
11         // Crear tabla si no existe
12         string sqlTabla = "CREATE TABLE IF NOT EXISTS usuarios (id INTEGER PRIMARY KEY AUTOINCREMENT, nombre TEXT)";
13         SQLiteCommand cmdTabla = new SQLiteCommand(sqlTabla, conexion);
14         cmdTabla.ExecuteNonQuery();
15
16         // Insertar un dato de prueba
17         string sqlInsert = "INSERT INTO usuarios (nombre) VALUES ('Carlos')";
18         SQLiteCommand cmdInsert = new SQLiteCommand(sqlInsert, conexion);
19         cmdInsert.ExecuteNonQuery();
20
21         // Leer datos
22         string sqlSelect = "SELECT * FROM usuarios";
23         SQLiteCommand cmdSelect = new SQLiteCommand(sqlSelect, conexion);
24
25         using (SQLiteDataReader reader = cmdSelect.ExecuteReader())
26         {
27             while (reader.Read())
28             {
29                 Console.WriteLine($"ID: {reader["id"]}, Nombre: {reader["nombre"]}");
30             }
31         }
32     }
33     catch (Exception ex)
34     {
35         Console.WriteLine("Error: " + ex.Message);
36     }
37
38     Console.WriteLine("Presiona una tecla para salir...");
39     Console.ReadKey();
40
41 }
42
43
44
45
46
47
48
49
50
51
52
53 }
```