



Information Technology

.Net Framework Concepts:





Tecnologías de la información

Conceptos de .NET Framework

VOCABULARY

WORDS	PALABRAS
✓ .Net's main features	√Características principales de .Net
✓ .Net's common language infrastructure	√Infraestructura de lenguaje común de .Net
✓ Assemblies	✓ Asambleas
✓ Metadata	✓ Metadatos
✓ Base class library	✓ Biblioteca de clases base





Objective

Recognize concepts .Net framework for users.

Content:

- Introduction.
- .Net Framework Definition.
- History of .Net.
- .Net's Main Features.
- .Net's Common Language Infrastructure.
 - .Net's Programming Languages.
 - Conclusions.



Objetivo

Reconocer conceptos de .Net framework para usuarios.

Contenido:

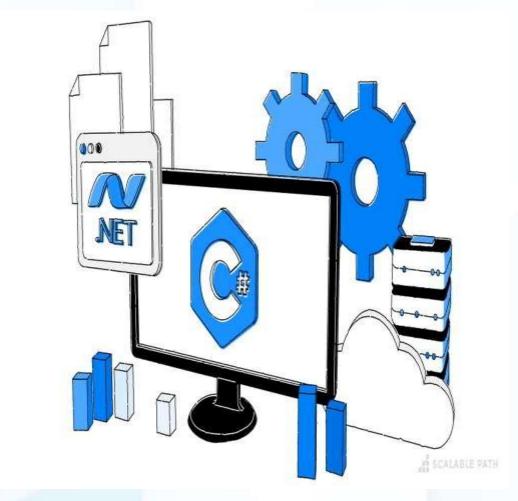
- Introducción.
- Definición de .Net Framework.
- Historia de .Net.
- Características principales de . Net.
- Infraestructura de lenguaje común de .Net.
- Lenguajes de programación de .Net.
- Conclusiones.

Introduction



Back in the early 2000s, the applications built with .NET were aimed exclusively for Windows desktop and the framework itself was strictly developed and maintained in a closed source format.

However, with the introduction of .NET Core, Microsoft made a complete shift and revolutionized software development.





Introducción



A principios de la década de 2000, las aplicaciones creadas con .NET estaban destinadas exclusivamente al escritorio de Windows y el marco en sí se desarrollaba y mantenía estrictamente en un formato de código cerrado.

Sin embargo, con la introducción de .NET Core, Microsoft hizo un cambio completo y revolucionó el desarrollo de software.





Introduction



Applications could now be developed for a broader set of operating systems and platforms including Linux and macOS.

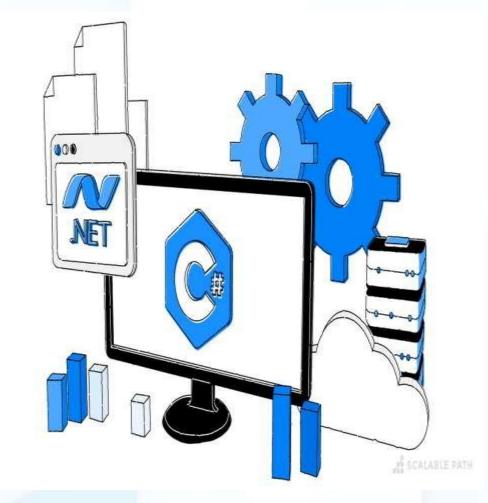




Introducción



Ahora se podrían desarrollar aplicaciones para un conjunto más amplio de sistemas operativos y plataformas, incluidos Linux y macOS.



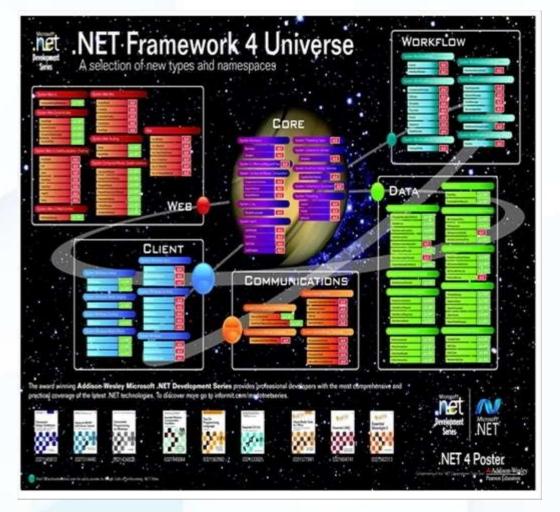


.Net Framework Definition



To put it simply, .NET is an open-source software development platform built by Microsoft.

It allows us to build cross-platform software products that can run on Windows, Linux, macOS, iOS, Android, etc.



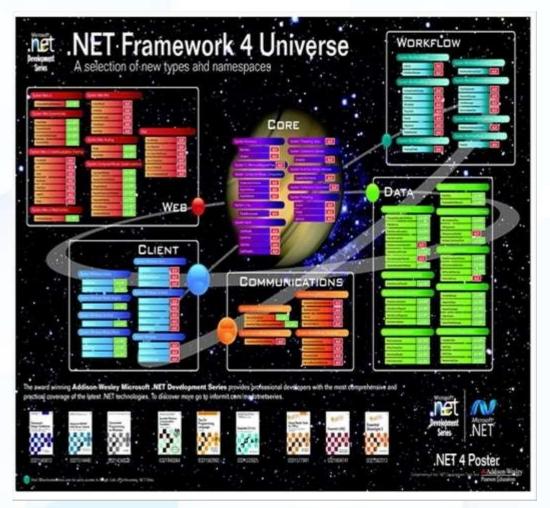


Definición de .NET Framework



En pocas palabras, .NET es una plataforma de desarrollo de software de código abierto creada por Microsoft.

Nos permite crear productos de software multiplataforma que pueden ejecutarse en Windows, Linux, macOS, iOS, Android, etc.

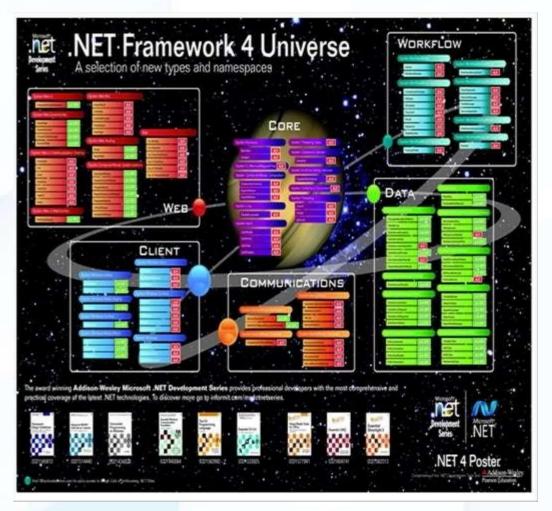




.Net Framework Definition



It's used for creating many different types of applications and services and provides the necessary programming environment for all software development phases.

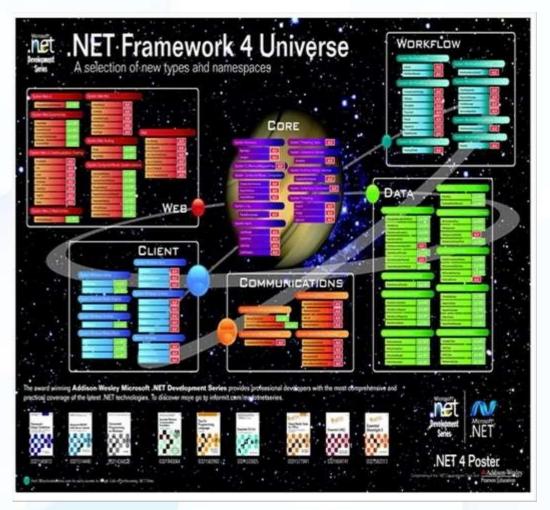




Definición de .NET Framework



Se utiliza para crear muchos tipos diferentes de aplicaciones y servicios y proporciona el entorno de programación necesario para todas las fases de desarrollo de software.

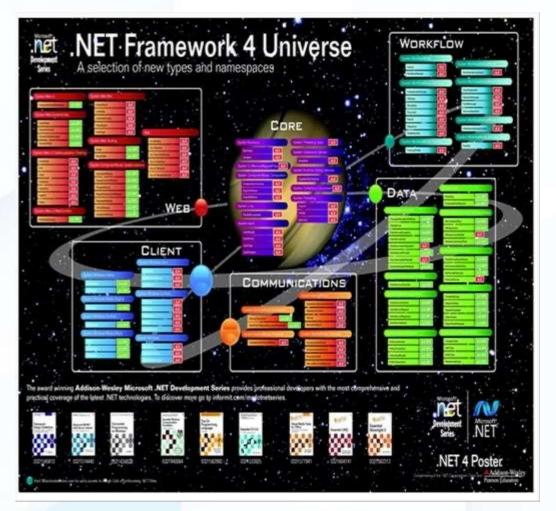




.Net Framework Definition



Its surrounding ecosystem is a great fit for businesses and individuals that want to develop a wide range of software products, from desktop and mobile apps to web apps and cloud-powered services.

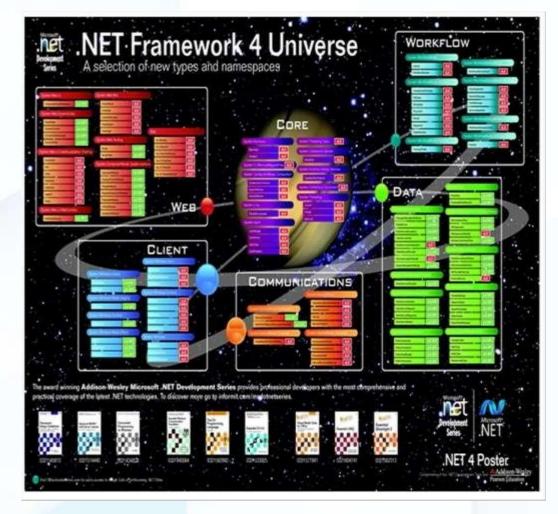




Definición de .NET Framework



Su ecosistema circundante es ideal para empresas e individuos que desean desarrollar una amplia gama de productos de software, desde aplicaciones de escritorio y móviles hasta aplicaciones web y servicios basados en la nube.





The History of .Net



In the late 90s, Microsoft created .NET hoping to compete with Sun's Java, which was making a huge impact in the software development world.

Envisioning a cross-platform development paradigm, Microsoft started incorporating the best brains in the market to develop its own software development ecosystem, going from desktop to the Web with Active Server Pages (ASP) and later on with ASP.NET, finally making the shift from on-premises to the cloud in recent years with Azure.

Cross-platform is now something that we can take for granted when employing the .NET platform.



La historia de .Net



A finales de los años 90, Microsoft creó .NET con la esperanza de competir con Java de Sun, que estaba teniendo un enorme impacto en el mundo del desarrollo de software

Visualizando un paradigma de desarrollo multiplataforma, Microsoft comenzó a incorporar los mejores cerebros del mercado para desarrollar su propio ecosistema de desarrollo de software, pasando del escritorio a la Web con Active Server Pages (ASP) y más tarde con ASP.NET, para finalmente hacer el cambio de local a la nube en los últimos años con Azure.

La multiplataforma es ahora algo que podemos dar por sentado cuando empleamos la plataforma .NET.



.Net's Main Features



The following list depicts .NET Framework main features:



Características principales de .Net



La siguiente lista describe las características principales de .NET Framework

1.- One Framework, multiple languages



.NET introduces a Common Type System (CTS) that defines all possible data types and programming constructs supported and how these may interact with each other.

This allows .NET to support multiple application development programming languages, including C#, F# or Visual Basic.





1.- Un Framework, múltiples lenguajes



. NET presenta un Sistema de tipos comunes (CTS) que define todos los tipos de datos y construcciones de programación posibles admitidos y cómo pueden interactuar entre sí.

Esto permite que .NET admita múltiples lenguajes de programación de desarrollo de aplicaciones, incluidos C#, F# o Visual Basic.





2.- CLR – Common Language Runtime



All the programming Languages in .NET are compiled into an intermediate language known as Common Intermediate Language (CIL).

This intermediate language is not interpreted but compiled to native code in a way known as Just In Time compilation (JIT).

The combination of these is called Common Language Infrastructure (CLI).





2.- CLR - Tiempo de ejecución del lenguaje común



Todos los lenguajes de programación en .NET se compilan en un lenguaje intermedio conocido como Lenguaje Intermedio Común (CIL).

Este intermedio idioma no es interpretado pero compilado en código nativo de una manera conocida como compilación Just In Time (JIT).

La combinación de estos se llama Infraestructura de Lenguaje Común (CLI).





3.- Interoperability



Often, interaction between new and old applications is required.

The .NET Framework provides a way to access features that are implemented in programs that run outside the .NET environment.





3.- Interoperabilidad



A menudo se requiere interacción entre aplicaciones nuevas y antiguas.

.NET Framework proporciona una manera de acceder a las características que se implementan en programas que se ejecutan fuera del entorno .NET.



4.- Base Class Library



The Base Class Library (BCL) is a type library available for all languages that use .NET.

The BCL provides classes that encapsulate many common functions, including reading and writing files, image design, database interaction, XML and JSON manipulation, etc.





4.- Biblioteca de clases base



La biblioteca de clases base (BCL) es una biblioteca de tipos disponible para todos los lenguajes que utilizan . NET.

El BCL proporciona clases eso encapsular muchas funciones comunes, incluida la lectura y escritura de archivos, diseño de imágenes, interacción con bases de datos, manipulación de XML y JSON, etc.





5.- Security



.NET allows code to run in different levels of security without the use of a separate security mechanism.



5.- Seguridad



. NET permite que el código se ejecute en diferentes niveles de seguridad sin el uso de un mecanismo de seguridad separado.



6.- Automatic Resource Management



.NET CLR automatically manages memory, network, database connections, etc.

It invokes built-in functions to allocate and deallocate the memory used by objects during runtime.

This removes the burden of managing memory from the developer.





6.- Gestión Automática de Recursos



. NET CLR administra automáticamente la memoria, la red, las conexiones de base de datos, etc.

Invoca funciones integradas para asignar y desasignar la memoria utilizada por los objetos durante el tiempo de ejecución.

Esto elimina la carga de administrar la memoria del desarrollador.





7.- Portability



Applications written in any .NET language are portable.

That means that when the source code of a program written in a CLR compliant language compiles, it generates machineindependent, intermediate code.

It's also possible to mix and match code written in the languages supported by the framework.





7.- Portabilidad



Las aplicaciones escritas en cualquier lenguaje .NET son portables.

Esto significa que cuando se compila el código fuente de un programa escrito en un lenguaje compatible con CLR, genera un código intermedio independiente de la máquina.

También es posible mezclar y combinar código escrito en los lenguajes soportados por el marco.





8.- Easy and rich profiling and debugging support



Integrated Development Environments (IDEs) like Visual Studio or Visual Studio Code provide a place to develop and debug .NET applications.

The framework provides rich debugging and profiling information which is highly integrated within these IDEs.

This helps us to check the application performance and track down runtime exceptions.





8.- Soporte fácil y rico para creación de perfiles y depuración



Los entornos de desarrollo integrados (IDE) como Visual Studio o Visual Studio Code proporcionan un lugar para desarrollar y depurar aplicaciones .NET.

El marco proporciona información rica de depuración y creación de perfiles que está altamente integrada dentro de estos IDE.

Esto nos ayuda a comprobar el rendimiento de la aplicación y rastrear excepciones en tiempo de ejecución.





9.- Deployment made easy



When it's time for application deployment, the developer has plenty of options to package the app depending on the desired deployment environment (desktop, mobile, cloud, etc.).





9.- Implementación fácil



Cuando llega el momento de implementar la aplicación, el desarrollador tiene muchas opciones para empaquetar la aplicación dependiendo del entorno de implementación deseado (escritorio, móvil, nube, etc.).



.Net's Common Language Infrastructure



The most important .NET component is located within the Common Language Infrastructure (CLI).

The CLI's purpose is to provide an agnostic language platform for application development and execution, including but not limited to components for exception handling, garbage collection, security, interoperability, etc.



Infraestructura de lenguaje común de .Net



El componente .NET más importante se encuentra dentro de la Infraestructura de lenguaje común (CLI).

El propósito de la CLI es proporcionar una plataforma de lenguaje agnóstico para el desarrollo y ejecución de aplicaciones, incluidos, entre otros, componentes para el manejo de excepciones, recolección de basura, seguridad, interoperabilidad, etc.



.Net's Common Language Infrastructure



The CLI implementation in .NET is called Common Language Runtime (CLR). The CLR is composed of four primary parts:

Common Type System (CTS)
Common Language Specification (CLS)
Just-in-Time Compiler (JIT)
Virtual Execution System (VES)



Infraestructura de lenguaje común de .Net



La implementación de CLI en .NET se denomina Common Language Runtime (CLR). El

CLR se compone de cuatro partes principales:

Sistema de tipos comunes (CTS),

Especificación de lenguaje común (CLS),

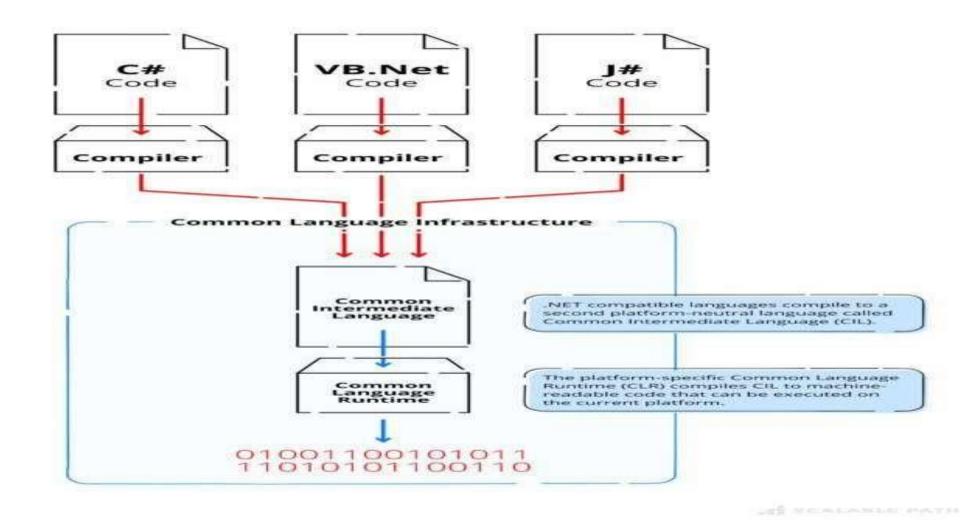
Compilador justo a tiempo (JIT)

Sistema de ejecución virtual (VES)



.Net's Common Language Infrastructure

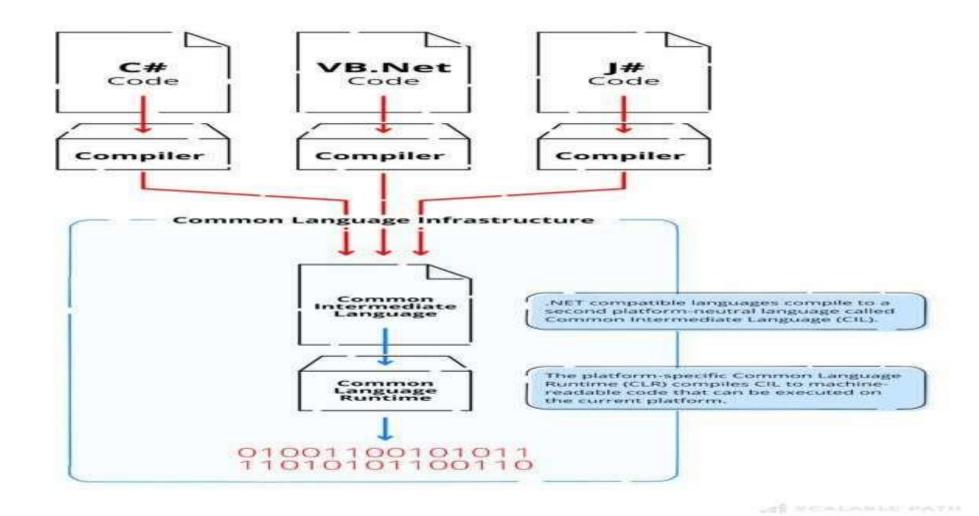






Infraestructura de lenguaje común de .Net





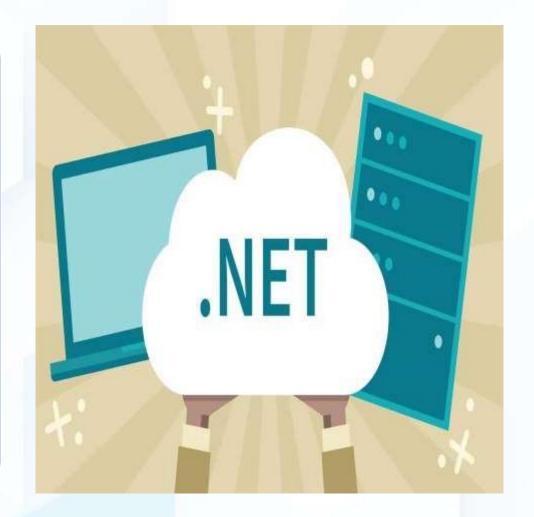


Assemblies



The Common Intermediate Language (CIL) is hosted in partially compiled code libraries called .NET assemblies, which for a Windows implementation means a portable executable (PE), a file (EXE) or a Dynamic Link Library (DLL).

These form the .NET distribution, version and security units.



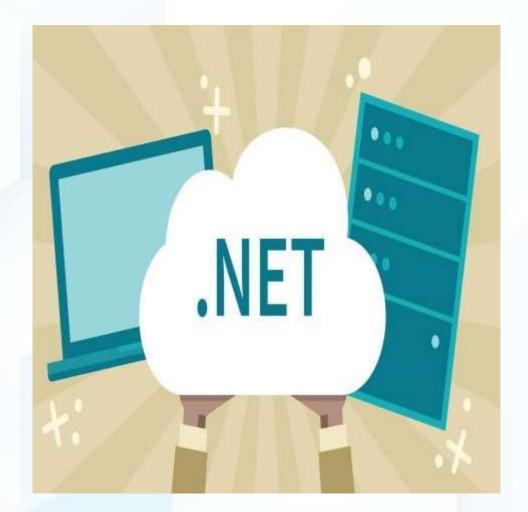


Asambleas



El lenguaje intermedio común (CIL) está alojado en bibliotecas de código parcialmente compiladas llamadas ensamblados .NET, lo que para una implementación de Windows significa un ejecutable portátil (PE), un archivo (EXE) o una biblioteca de vínculo dinámico (DLL).

Estos forman las unidades de distribución, versión y seguridad de .NET.





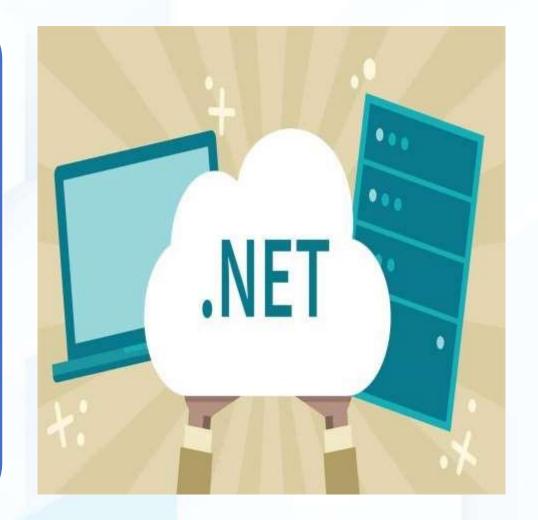
Metadata



Every CIL is self-described through metadata.

The CLR checks the metadata to ensure the correct method is called.

Metadata is usually generated by language compilers and also describes all the information about the assembly.





Metadatos



Cada CIL se autodescribe a través de metadatos.

El CLR verifica los metadatos para garantizar que se llame al método correcto.

Metadatos es generalmente generado por compiladores de lenguaje y también describe toda la información sobre el ensamblaje.



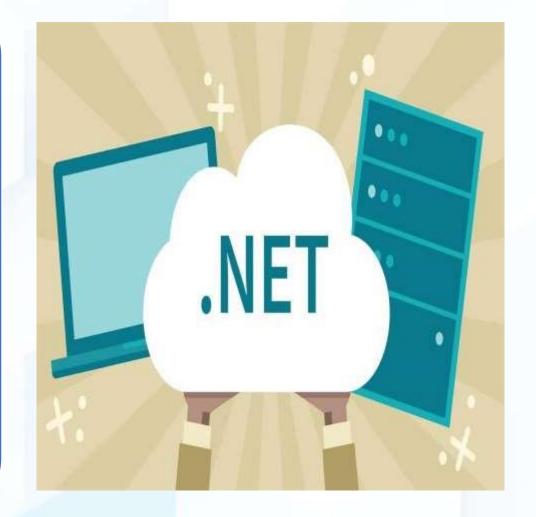


Base Class Library



The Base Class Library (BCL) is available for all .NET languages.

It provides encapsulation for the most diverse functions used by programs developed with the support of the .NET Framework.





Biblioteca de clases base



La biblioteca de clases base (BCL) está disponible para todos los lenguajes .NET.

Proporciona encapsulación para las más diversas funciones utilizadas por programas desarrollados con el soporte de .NET Framework.





.Net's Programming Languages



While it supports alternatives such as F# and Visual Basic, .NET's main programming language is C#.

C# is a modern, object-oriented, and type-safe programming language that enables developers to build many types of secure and robust applications.

C# has its roots in the C family of languages, which should make developers with a background in C, C++, Java, and JavaScript feel comfortable working with it.



Lenguajes de programación de .Net



Si bien admite alternativas como F# y Visual Basic, el lenguaje de programación principal de .NET es C#.

C# es un lenguaje de programación moderno, orientado a objetos y seguro en cuanto a tipos, que permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas.

C# tiene sus raíces en la familia de lenguajes C, lo que debería hacer que los desarrolladores con experiencia en C, C++, Java y JavaScript se sientan cómodos trabajando con él.

.Net's Programming Languages



C# has these great constructs built-in:

-Generics, partial classes, anonymous types, iterators, nullable types, static classes, delegate interface.

-Implicit types, object and collection initializers, auto-implemented properties, extension methods, query and lambda expressions, expression trees, partial methods.

-Dynamic binding, named and optional arguments, generic covariance and contravariance, embedded interop types.



Lenguajes de programación de .Net



C# tiene estas excelentes construcciones incorporadas:

- Genéricos, clases parciales, tipos anónimos, iteradores, tipos que aceptan valores nulos, clases estáticas, interfaz delegada.
- Tipos implícitos, inicializadores de objetos y colecciones, propiedades implementadas automáticamente, métodos de extensión, expresiones de consulta y lambda, árboles de expresión, métodos parciales.
- Enlace dinámico, argumentos nombrados y opcionales, covarianza y contravarianza genéricas, tipos de interoperabilidad integrados.



.Net's Programming Languages



- -Asynchronous tasks with async and await (task asynchronous programming model).
- -Roslyn (compiler-as-a-service), exception filters, await in catch/finally block, auto property initializer, string interpolation, name of the operator, dictionary initializer, string interpolation, name of the operator and dictionary initializer.
- -Tuples, pattern matching, record types, local functions.
- -Using declarations, nullable reference types, async streams, null-coalescing assignments.



Lenguajes de programación de .Net



- Tareas asincrónicas con async y await (modelo de programación asincrónica de tareas).
- Roslyn (compilador como servicio), filtros de excepción, espera en el bloque catch/finally, inicializador automático de propiedades, interpolación de cadenas, nombre del operador, inicializador de diccionario, interpolación de cadenas, nombre del operador e inicializador de diccionario.
- Tuplas, coincidencia de patrones, tipos de registros, funciones locales.
- Uso de declaraciones, tipos de referencia que aceptan valores nulos, transmisiones asíncronas y asignaciones de fusión de valores nulos.

Help Documentation (Microsoft Docs)



Every developer needs to check documentation to see how a given piece of code works, how the language construct is written, etc.

Microsoft has been putting a tremendous amount of effort into building Microsoft Docs for these kinds of things.





Documentación de ayuda (Microsoft Docs)



Cada revelador necesidades a controlar documentación para ver cómo funciona un determinado fragmento de código, cómo está escrita la construcción del lenguaje, etc.

Microsoft ha dedicado un enorme esfuerzo a desarrollar Microsoft Docs para este tipo de cosas.





Help Documentation (Microsoft Docs)



.NET is really well covered with tons of documentation and videos (.NET YouTube channel and Channel 9, just to name a few) that explain how things work, from the most basic to the most advanced level.

Once we get to the main .NET doc page, we're presented with the necessary links to jump into .NET easily.





Documentación de ayuda (Microsoft Docs)



. NET está realmente bien cubierto con toneladas de documentación y videos (canal .NET de YouTube y Canal 9, solo por nombrar algunos) que explican cómo funcionan las cosas, desde el nivel más básico hasta el más avanzado.

Una vez que llegamos a la página principal de documentación de .NET, se nos presentan los enlaces necesarios para acceder a .NET fácilmente.







Conclusions:

When we talk about .NET for software development, we're speaking about a broad range of things.



Conclusiones:

Cuando hablamos de .NET para el desarrollo de software, hablamos de una amplia gama de cosas.



Conclusions:

Applications can be developed using one of the multiple available programming languages, which are very feature-rich, being on par or even surpassing related programming languages like Java, Python, and the likes.

So why would you choose .NET over any other framework?





Conclusiones:

Se pueden desarrollar aplicaciones utilizando uno de los múltiples lenguajes de programación disponibles, que son muy ricos en características y están a la par o incluso superan a lenguajes de programación relacionados como Java, Python y similares.

Entonces, ¿por qué elegiría .NET en lugar de cualquier otro marco?



Conclusions:

The most straightforward answer is that it provides everything needed and is highly integrated from start to end, allowing you to develop, deploy and maintain code easily.

Its cross-platform nature allows your business logic to be ported to many different kinds of platforms with no trouble.





Conclusiones:

La respuesta más sencilla es que proporciona todo lo necesario y está altamente integrado de principio a fin, lo que le permite desarrollar, implementar y mantener el código fácilmente.

Su naturaleza multiplataforma permite que su lógica empresarial se adapte a muchos tipos diferentes de plataformas sin problemas.

