EXCELENTE INVESTIGACIč N

30%

Instituto Tecnológico Superior de San Andrés Tuxtla

Ingeniería Mecatrónica

Exposición:

Tecnología .NET.

Docente:

Mtro. Roberto Esteban Guerrero Hernández

Presenta:

SERRANO VELAZQUEZ ESMERALDA
Grupo: 311 B

San Andrés Tuxtla a Noviembre de 2023

¿Qué es .NET?

.NET es una plataforma de código abierto para crear aplicaciones de escritorio, web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo. El sistema .NET incluye herramientas, bibliotecas y lenguajes que admiten el desarrollo de software moderno, escalable y de alto rendimiento. Una comunidad de desarrolladores activa mantiene y apoya la plataforma .NET.

En términos simples, la plataforma .NET es un software que puede realizar estas tareas:
Traducir el código del lenguaje de programación .NET en instrucciones que un dispositivo de computación pueda procesar.

Proporcionar utilidades para un desarrollo de software eficiente. Por ejemplo, puede encontrar la hora actual o imprimir texto en la pantalla.

Defina un conjunto de tipos de datos para almacenar información como texto, números y fechas en el equipo.

¿Qué es una implementación de .NET?

Varias implementaciones de .NET habilitan que el código .NET se ejecute en diferentes sistemas operativos, como Linux, macOS, Windows, iOS, Android y muchos otros.
.NET Framework
.NET Framework es la implementación de .NET original. Admite la ejecución de sitios web, servicios, aplicaciones de escritorio y más en Windows. Microsoft lanzó .NET Framework a principios de la década de 1990.
.NET Core

Microsoft lanzó .NET Core a finales de 2014 para permitir el soporte multiplataforma para los desarrolladores de .NET. La compañía lanzó la versión más reciente de .NET Core, .NET 5.0, en noviembre de 2020 y la renombró .NET. El término .NET en este artículo se refiere a .NET 5.0. .NET Core es de código abierto en GitHub.
.NET Standard

.NET Standard es una especificación formal de diferentes funciones (denominadas API). Las diferentes implementaciones de .NET pueden reutilizar el mismo código y las mismas bibliotecas. Cada implementación utiliza tanto las API estándar de .NET como las API únicas específicas de los sistemas operativos en los que se ejecuta.

¿Por qué elegir .NET?

Facilidad de desarrollo

A los desarrolladores les gusta usar .NET porque incluye muchas herramientas que facilitan su trabajo. Por ejemplo, con el paquete Visual Studio, los desarrolladores pueden escribir código más rápido, colaborar, probar y corregir su código de manera eficiente. La capacidad de reutilizar el código entre implementaciones reduce el costo de desarrollo.

Aplicaciones de alto rendimiento

Las aplicaciones .NET proporcionan tiempos de respuesta más rápidos y requieren menos potencia de computación. Tienen medidas sólidas de seguridad incorporadas y realizan de manera eficiente tareas del lado del servidor, como el acceso a la base de datos.

Comunidad y ayuda

.NET es de código abierto, lo que significa que cualquiera puede tener acceso para usarlo, leerlo y modificarlo libremente. Una comunidad activa de desarrolladores mantiene y mejora el software de .NET. .NET Foundation es una organización independiente sin fines de lucro establecida para ayudar a la comunidad .NET. Proporciona recursos de aprendizaje, proyectos .NET de código abierto y varios eventos para los desarrolladores de .NET.

¿Cuáles son los componentes de la arquitectura de .NET?

.NET tiene una arquitectura modular y optimizada. Los usuarios pueden elegir diferentes componentes para cumplir con sus requisitos de desarrollo de software.

Estos son los tres componentes principales de .NET:

Lenguajes .NET

Marcos de modelo de aplicación

Tiempo de ejecución de .NET

Los desarrolladores utilizan lenguajes de programación y marcos de modelo de aplicación .NET para crear sus aplicaciones .NET. A continuación, el tiempo de ejecución de .NET los ejecuta.

¿Qué son los lenguajes de programación .NET?

C# (pronunciado “si sharp”), F# (pronunciado “ef sharp”) y Visual Basic son los tres lenguajes compatibles con Microsoft para el desarrollo de NET. Diferentes empresas y desarrolladores también han creado otros lenguajes que funcionan con la plataforma .NET.
C#
C# es un lenguaje de programación simple, moderno y orientado a objetos. Con una sintaxis similar a la familia de lenguajes C, C# resulta familiar para los programadores de C, C++, Java y JavaScript.
F#
F# tiene una sintaxis ligera y requiere muy poco código para crear el software. Es un lenguaje de código abierto que facilita la escritura de código sucinto, robusto y de alto rendimiento. También cuenta con un potente sistema de reglas de programación y una práctica biblioteca estándar para crear software de misión crítica, correcto, rápido y confiable.
Visual Basic

Visual Basic es un lenguaje de programación orientado a objetos desarrollado por Microsoft. El uso de Visual Basic facilita y agiliza la creación de aplicaciones .NET seguras para escribir. La seguridad de tipos es la medida en que un lenguaje de programación desalienta o previene errores de codificación lógica.

Lenguajes de infraestructura de lenguaje común (CLI)
Lenguajes como ClojuRecLR, Eiffel, IronPython y PowerBuilder, entre otros, también funcionan en la plataforma .NET. Esto se debe a que .NET implementa la infraestructura de lenguaje común (CLI). Imagine que CLI es una plantilla para crear lenguajes compatibles con .NET.

¿Qué es el tiempo de ejecución de .NET?

El tiempo de ejecución de .NET, también llamado Common Language Runtime (CLR), compila y ejecuta programas de .NET en diferentes sistemas operativos.
Compilación justo a tiempo

El CLR compila el código a medida que el desarrollador lo escribe. Durante la compilación, el CLR traduce el código al Common Intermediate Language (CIL). Por ejemplo, el código escrito en C# tiene palabras y sintaxis similares al inglés. .NET compila o traduce este código a CIL. El código CIL tiene un aspecto diferente porque es un lenguaje de código máquina de nivel inferior.

Ejecución

El tiempo de ejecución de .NET administra la ejecución del código CIL. CIL es compatible con varias plataformas y cualquier sistema operativo puede procesarlo. La compatibilidad multiplataforma se refiere a la capacidad de una aplicación para ejecutarse en varios sistemas operativos diferentes con modificaciones mínimas. Por ejemplo, una aplicación en C# se puede ejecutar en Windows, Linux o macOS sin ninguna modificación de código. Una aplicación de este tipo se denomina “aplicación multiplataforma”.

¿Qué son los marcos de modelo de aplicación .NET?

Los marcos de modelo de aplicación son una colección de herramientas y bibliotecas para desarrolladores que admiten el desarrollo rápido y eficiente de proyectos de .NET. Existen diferentes marcos para diferentes tipos de aplicaciones, como las que se enumeran a continuación.

Aplicaciones web

El marco ASP.NET amplía la plataforma para desarrolladores de .NET con el objetivo de crear aplicaciones basadas en web. Admite tecnologías web como API de REST, HTML, CSS y JavaScript. Proporciona una base de datos de usuarios integrada con autenticación externa y multifactor. ASP.NET admite protocolos de autenticación estándar del sector con un mecanismo de seguridad integrado para proteger sus aplicaciones .NET de los ciberataques.
Aplicaciones para móviles

Puedes usar Xamarin/Mono para ejecutar aplicaciones .NET en todos los principales sistemas operativos móviles, como iOS y Android. Xamarin incluye Xamarin.Forms, un marco de interfaz de usuario móvil de código abierto. Los desarrolladores de .NET utilizan Xamarin.Forms para crear una experiencia de usuario coherente en todas las plataformas móviles. Todas las aplicaciones .NET pueden tener el mismo aspecto, incluso en diferentes dispositivos móviles.

Aplicaciones de escritorio

Puede usar Xamarin para el desarrollo de aplicaciones de escritorio. Además, Universal Windows Platform amplía el desarrollo de aplicaciones de .NET de Windows 10. Windows Presentation Foundation y Windows Forms son otros marcos para el diseño de la interfaz de usuario en Windows.
Otras aplicaciones

Con ML.NET, puede desarrollar e integrar modelos personalizados de machine learning en sus aplicaciones .NET. Puede utilizar las bibliotecas .NET IoT para desarrollar aplicaciones en sensores y otros dispositivos inteligentes. Para cualquier solución que no esté disponible en los marcos, puede encontrar muchas bibliotecas de funciones específicas en el repositorio público de NuGet. Puede usar Nuget para crear, compartir y usar muchas bibliotecas .NET para casi cualquier propósito.

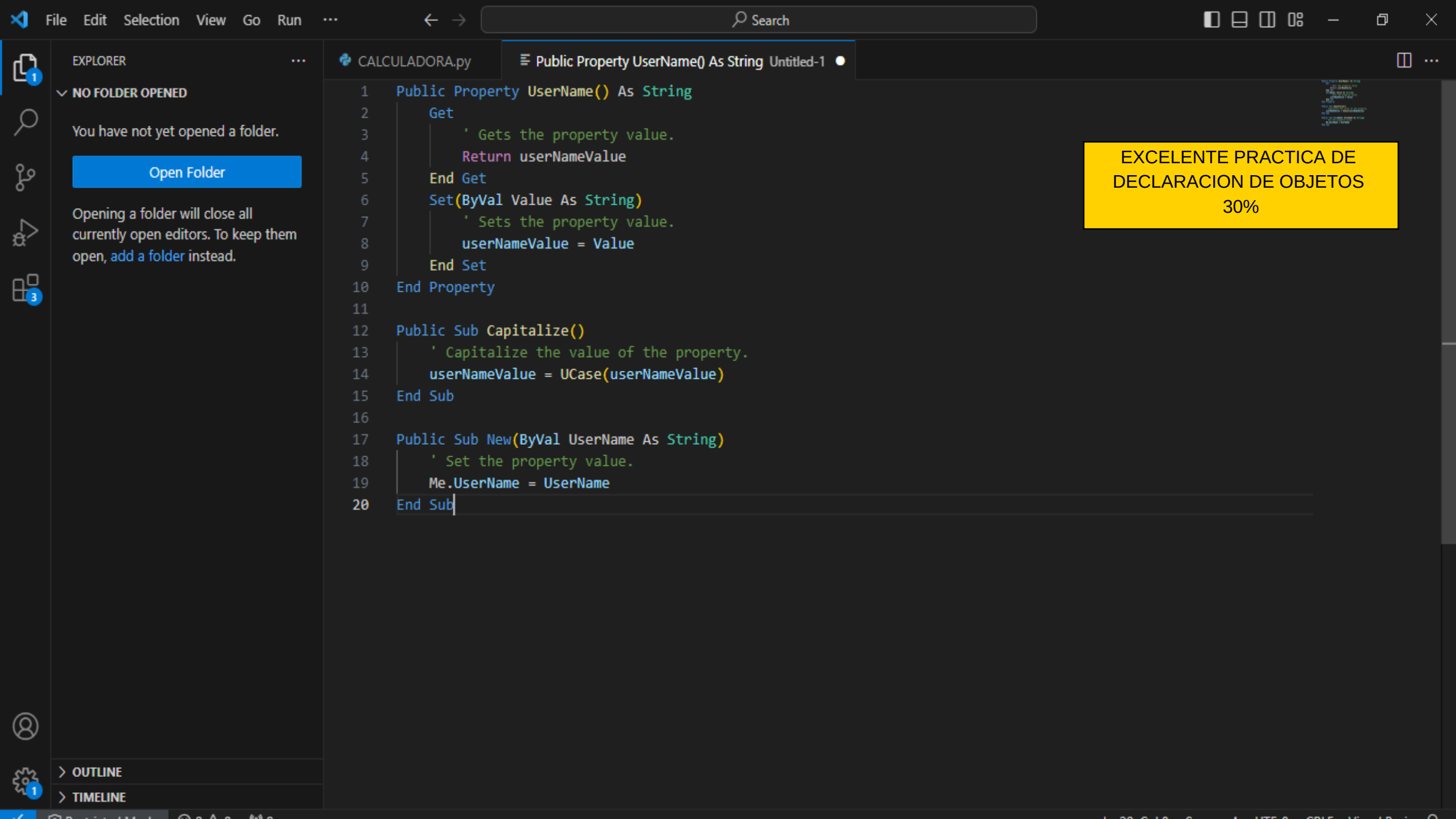
¿Cómo puede ayudar AWS a los desarrolladores de .NET?

Los desarrolladores de .NET pueden hacer que las aplicaciones sean más rápidas con .NET en AWS. Cuenta con un servicio para cada trabajo, por lo que puede crear rápidamente pruebas de concepto sin preocuparse por administrar la infraestructura. A continuación, se muestran algunos ejemplos de servicios de AWS para el desarrollo de .NET:

AWS Elastic Beanstalk gestiona el despliegue de aplicaciones y las tareas operativas.
Amazon EC2 ofrece capacidad de computación segura y de tamaño ajustable en la nube.
Amazon Aurora automatiza la administración de bases de datos.
Los desarrolladores de.NET también pueden usar las herramientas y bibliotecas de código abierto de AWS, como las siguientes:
El AWS SDK para .NET facilita a los desarrolladores de Linux y Windows la creación de aplicaciones .NET.

El repositorio de código de ejemplo de AWS sirve para acelerar su comprensión de las aplicaciones que funcionan con los servicios de AWS.
La Biblioteca digital de .NET contiene un archivo de videos, tutoriales, blogs y otros recursos para los desarrolladores de .NET.

Con el lanzamiento de .NET 6, los desarrolladores de .NET pueden aprovechar aún más el rendimiento y el ahorro de costes de AWS con Linux. Las aplicaciones de .NET 6 ya pueden usar muchos servicios de AWS sin trabajo adicional. En esta guía se describe la compatibilidad de .NET 6 que ofrecen los servicios y herramientas de AWS.


File Edit Selection View Go Run ···
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
OUTLINE
TIMELINE
CALCULADORA.py
Public Property UserName() As String Untitled-1
1 Public Property UserName() As String
2 Get
3 ' Gets the property value.
4 Return userNameValue
5 End Get
6 Set(ByVal Value As String)
7 ' Sets the property value.
8 userNameValue = Value
9 End Set
10 End Property
11
12 Public Sub Capitalize()
13 ' Capitalize the value of the property.
14 userNameValue = UCase(userNameValue)
15 End Sub
16
17 Public Sub New(ByVal UserName As String)
18 ' Set the property value.
19 Me.UserName = UserName
20 End Sub
EXCELENTE PRACTICA DE DECLARACION DE OBJETOS 30%

File Edit Selection View Go Run ··· Search

EXPLORER

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

OUTLINE

TIMELINE

CALCULADORA.py | Public Class Customer Untitled-1

```
Public Class Customer
    Public Property AccountNumber As Integer
End Class
Dim nextCustomer As Customer
Dim nextCustomer As New Customer
nextCustomer.AccountNumber = lastAccountNumber + 1
warningLabel.Text = "Data not saved"
Dim warningWidth As Integer = warningLabel.Width
warningLabel.ForeColor = System.Drawing.Color.Red
Dim safetyTimer As New System.Windows.Forms.Timer
safetyTimer.Start()

Dim secondForm As New System.Windows.Forms.Form
secondForm.Show()

Console.WriteLine("This computer is called " & Environment.MachineName)
Public Sub ExamineTimeZone()
    Dim tz As System.TimeZone = System.TimeZone.CurrentTimeZone
    Dim s As String = "Current time zone is "
    s &= CStr(tz.GetUtcOffset(Now).Hours) & " hours and "
    s &= CStr(tz.GetUtcOffset(Now).Minutes) & " minutes "
    s &= "different from UTC (coordinated universal time)"
    s &= vbCrLf & "and is currently "
    If tz.IsDaylightSavingTime(Now) = False Then s &= "not "
    s &= "on ""summer time""."
    Console.WriteLine(s)
End Sub
Public Class ReversibleButton
    Inherits System.Windows.Forms.Button
        Public Sub ReverseColors()
            Dim saveColor As System.Drawing.Color = Me.BackColor
            Me.BackColor = Me.ForeColor
            Me.ForeColor = saveColor
```

EXAMEN

25 %

Restricted Mode ⊗ 0 ⚠ 0 0

Ln 35, Col 10 Spaces: 4 UTF-8 CRLF Visual Basic