

```
    if self.mirror_operation == "MIRROR_X" +  
        mirror_mod.use_x = True  
        mirror_mod.use_y = False  
        mirror_mod.use_z = False  
  
    if self.mirror_operation == "MIRROR_Y" +  
        mirror_mod.use_x = False  
        mirror_mod.use_y = True  
        mirror_mod.use_z = False  
  
    if self.mirror_operation == "MIRROR_Z" +  
        mirror_mod.use_x = False  
        mirror_mod.use_y = False  
        mirror_mod.use_z = True  
  
    #selection at the end -add back the deselected  
    mirror_ob.select= 1  
    modifier_ob.select=1  
    bpy.context.scene.objects.active = modifier_ob  
    print("Selected" + str(modifier_ob)) # modifier ob  
    #mirror_ob.select = 0  
    bpy.context.selected_objects[0]  
    bpy.context.selected_objects[0].select = True
```

Back end nivel intermedio

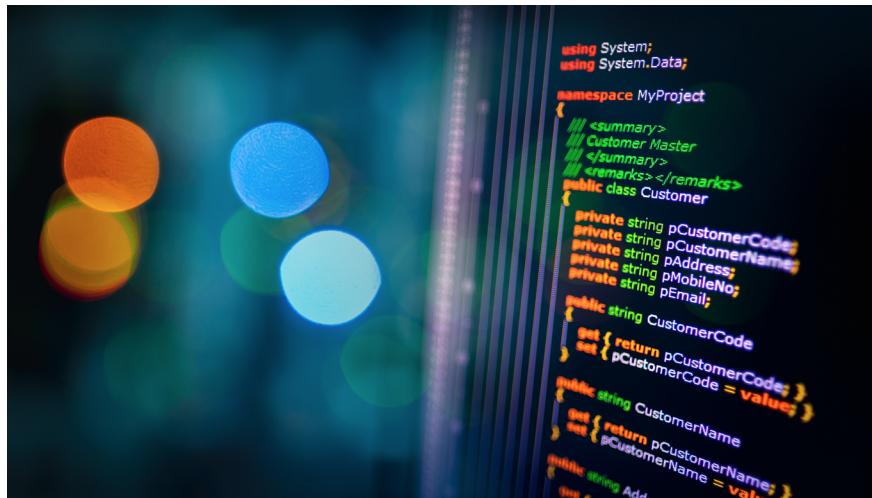
C# en Back End

Dentro del front end y el back end es de suma importancia que ambos sean fáciles de programar, de comprender su lógica y el tiempo de desarrollo sea reducido, con un costo no tan alto.

Debido a esta necesidad surge C#, lenguaje de programación orientado a objetos con el fin de brindar un fácil acceso a su conocimiento, una fácil implementación de las aplicaciones, así como una reducción de costos en su desarrollo, al igual que en el tiempo invertido para su implementación.



Introducción C#



C# fue creado en 2002 por Microsoft, este lenguaje está basado en objetos y se ejecuta desde el framework de .NET.

C# es mejor conocido como C-Sharp y lo puedes utilizar para realizar lo siguiente:

- Aplicaciones de escritorio.
- Aplicaciones móviles.
- Aplicaciones web.
- Servicios web.
- Sitios web.
- Videojuegos.
- Aplicaciones para bases de datos.

Ahora bien, ¿cuáles son las ventajas que tiene C# sobre los demás lenguajes?

- Es uno de los lenguajes de programación más populares.
- Es muy fácil de aprender y de usar.
- Es fácil obtener soporte.
- Permite reutilizar código, lo cual provocará que el costo del desarrollo web se vea disminuido.

Para el caso de back end puedes utilizar C# junto con ASP.NET, el cual es un framework que permitirá realizar la programación desde el lado del servidor, al igual, establecer las conexiones con las diferentes bases de datos en SQL.

A continuación, verás como realizar la instalación de ASP.NET para comenzar a programar tu back end:

Puedes instalarlo en los siguientes sistemas operativos:

Windows

Linux

MacOS

Para Windows realiza los siguientes pasos:

1. Ingresa a la siguiente liga:

Microsoft. (2022g). *Download and install*. Recuperado de
<https://dotnet.microsoft.com/en-us/learn/aspnet/hello-world-tutorial/install>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

2. Presiona el botón que dice “Download .NET 6 SDK (64-bit)”

The screenshot shows the Microsoft .NET website with the 'Windows' tab selected. Below it, a vertical list of links includes 'Intro', 'Download and install' (which is highlighted in blue), 'Create your app', 'Run your app', 'Edit your code', and 'Next steps'. A large heading 'Download and install' is centered below the list. Below the heading, a text block says 'To start building .NET apps, download and install the .NET SDK (Software Development Kit)'. A prominent blue button at the bottom of the section is labeled 'Download .NET 6 SDK (64-bit)'.

Download and install

To start building .NET apps, download and install the .NET SDK (Software Development Kit).

[Download .NET 6 SDK \(64-bit\)](#)

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Después ejecuta el .exe y sigue los pasos del instalador.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

1. Ingresa a la liga:

Microsoft. (2022g). *Download and isntall*. Recuperado de
<https://dotnet.microsoft.com/en-us/learn/aspnet/hello-world-tutorial/install>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

2. Selecciona Linux.

3. Haz clic en el botón que dice “Install .Net 6 SDK instructions”

Windows	Linux	macOS
Intro		
	Download and install	
Create your app		
Run your app		
Edit your code		
Next steps		

Download and install

To start building .NET apps, download and install the .NET SDK (Software Development Kit).

Install .NET 6 SDK instructions

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora para fines educativos.

4. Sigue las instrucciones del instalador.
5. Una vez terminado, abre la línea de comandos e ingresa el siguiente comando:

`dotnet`

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

6. Si la instalación es correcta, observarás lo siguiente en la línea de comandos:

```
Command prompt

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
-h|--help      Display help.
--info         Display .NET information.
--list-sdks    Display the installed SDKs.
--list-runtimes Display the installed runtimes.

path-to-application:
The path to an application .dll file to execute.
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para realizar la instalación en MacOS debes seguir los siguientes pasos:

1. Ingresá a la siguiente liga:

Microsoft. (2022g). *Download and install.* Recuperado de <https://dotnet.microsoft.com/en-us/learn/aspnet/hello-world-tutorial/install>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considerá que debes apegarte a sus términos y condiciones.

2. Seleccioná MacOS.

3. Haz clic en el botón que dice: “Download .NET 6 SDK (64-bit).

The screenshot shows a web page from Microsoft's .NET learning center. At the top, there are three tabs: 'Windows', 'Linux', and 'macOS', with 'macOS' underlined. Below the tabs is a vertical list of links: 'Intro', 'Download and install' (which is highlighted with a blue border), 'Create your app', 'Run your app', 'Edit your code', and 'Next steps'. Underneath this list, the title 'Download and install' is bolded. A sub-instruction reads: 'To start building .NET apps, download and install the .NET SDK (Software Development Kit).'. Below this is a large blue button with the text 'Download .NET 6 SDK (64-bit)' in white.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Realiza los pasos del instalador.

5. Una vez finalizado, dirígete a la línea de comandos e ingresa el siguiente comando:

dotnet

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

6. Si la instalación se realizó de forma correcta, obtendrás lo siguiente en la línea de comandos:

```
Command prompt

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
-h|--help      Display help.
--info         Display .NET information.
--list-sdks    Display the installed SDKs.
--list-runtimes Display the installed runtimes.

path-to-application:
The path to an application .dll file to execute.
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

NOTA: Deberás tener instalado Visual Studio para poder programar en C#.

Para esto deberás realizar lo siguiente:

1. Ingresar a la liga:

Microsoft. (2022h). *Descarga Visual Studio*. Recuperado de <https://visualstudio.microsoft.com/es/vs/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

2. Haz clic en “Descargue Visual Studio” y guarda el archivo en tu equipo.

3. Después, ejecuta el programa de instalación.

En el siguiente video verás como realizar el típico ejemplo de “Hola Mundo”:

Aguilar Systems MX. (2019, 19 de abril). *Como hacer el Hola Mundo desde ASP.NET en Visual C# con Visual Studio Community con .NET [Archivo de video]*. Recuperado de <https://www.youtube.com/watch?v=Xjqq6Fpf-m4>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Clases, objetos, métodos, acciones y atributos

Como en todo lenguaje de programación orientado a objetos puedes crear clases, objetos, métodos acciones y atributos.

De acuerdo con Microsoft (2022b), las clases te permiten definir que instancias va a tener, es decir, los objetos que van a existir dentro de la misma clase. En otras palabras, una clase es como una plantilla, la cual contendrá todas las características de un objeto.

La estructura de una clase es la siguiente:

- Encabezado.
- Cuerpo.
- Declaraciones.

En estructura de código la clase se define de la siguiente manera:

```
class [nombre de la clase] {  
    [atributos o variables de la clase]  
    [métodos o funciones de la clase]  
    [main]  
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

A continuación, se muestra un ejemplo de una clase: Imagina que tu desarrollo necesita una clase para detectar si el usuario y la contraseña que se están colocando es la correcta, para lo anterior, se debe de implementar una clase que contenga lo siguiente:

```
Class usuario  
{  
    Private string usuario;  
    Private string contraseña;  
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Al momento que se conecta a la base de datos para verificar si el usuario y la contraseña es válido, podrás usar esta clase para realizar esta comprobación.

De acuerdo con Microsoft (2022a), un método es aquel que contiene una serie de instrucciones para que al momento que se ejecute, se ejecuten dichas instrucciones. Los métodos los puedes declarar dentro de una clase, struct o una interfaz y puedes brindarle un grado de seguridad cuando especifiques el nivel de acceso (público o privado).

Retomando el ejemplo anterior: dentro de la clase de usuario puedes definir un método que te permita verificar si el usuario y contraseña existen. Esto se realizaría de la siguiente forma:

```
Class usuario
{
    Private string usuario;
    Private string contraseña;
    Public void Revisar();
}
```

Imagen 11. Ejemplo de un método.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

De acuerdo con Microsoft (2022c), un objeto se encuentra dentro de una clase, dicho objeto se almacena como una variable y puede ser de tipo string, matriz, entero, etc.

En el caso del ejemplo de la clase usuario, las variables: usuario y contraseña, serían tus objetos y dichos objetos puedes utilizarlos para realizar las verificaciones pertinentes.

```
Class usuario
{
    Private string usuario;
    Private string contraseña;
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

De acuerdo con Microsoft (2022d), los atributos permiten dar información que necesiten de otros elementos dentro de tu desarrollo. Son guardados en los metadatos y pueden ser accesos desde el compilador. En otras palabras, los atributos guardan las características del objeto.

Para crear un atributo debes seguir la siguiente sintaxis:

```
public class nombreAttribute : Attribute
{
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Retomando el tema del usuario y de la contraseña podrías crear un atributo en donde el usuario tenga que cumplir con ciertas características como:

- Ser un correo electrónico existente.

Para la contraseña podrías crear un atributo donde se cumpla lo siguiente:

- Mínimo 8 caracteres.
- Mínimo una mayúscula.
- Mínimo una minúscula.
- Mínimo un carácter especial.
- Mínimo un número.

Si el usuario y la contraseña cumplen con los atributos podrías revisar si se encuentra en la base de datos.

Si lo anterior lo colocas en código se vería de la siguiente forma:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace login
{
    public partial class login : Form
    {
        public login()
        {
            InitializeComponent();
        }

        //Conexión con BD, envías los parámetros nombre del servidor o BD Nombre de la BD Seguridad
        SqlConnection con = new SqlConnection(@"Data Source=(nombre del servidor o BD);Initial Catalog=Nombre de la BD;Integrated Security=True");

        public void logear(string usuario, string contrasena)
        {
            try
            {
                con.Open();
                //Comando SQL que va a buscar a la BD
                SqlCommand cmd = new SqlCommand("SELECT Nombre, Tipo_Usuario FROM usuarios WHERE Usuario = @usuario AND Contraseña = @pass", con);
                //Agregamos parámetros a nuestro comando para usuario y contraseña
                cmd.Parameters.AddWithValue("usuario", usuario);
                cmd.Parameters.AddWithValue("pass", contrasena);

                //Nuevo objeto de SQL data adapter
                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                //Llenar el DataTable con datos
                sda.Fill(dt);

                //Evalua si la consulta devuelve filas, si devuelve existe el registro
                if(dt.Rows.Count == 1)
                {
                    //Comando que oculta el formulario
                    this.Hide();

                    //Verifica qué tipo de usuario está ingresando
                    //Accedemos al tipo de columna, 0 = Nombre, 1 = Tipo_Usuario
                    if(dt.Rows[0][1].ToString() == "Admin")
                    {
                        //Si es Admin, vamos a mostrar el formulario de Administrador
                        new Admin(dt.Rows[0][0].ToString()).Show();
                    }
                    else if(dt.Rows[0][1].ToString() == "Usuario")
                    {
                        //Si es Usuario, vamos a mostrar el formulario de Usuario
                        new Usuario(dt.Rows[0][0].ToString()).Show();
                    }
                }
            }
        }
    }
}
```

```
        else
        {
            MessageBox.Show("Usuario y/o Contraseña incorrecta");
        }

    }
    catch(Exception e)
    {
        MessageBox.Show(e.Message);
    }
    finally
    {
        con.Close();
    }
}

private void btnInicioSesion_Click(object sender, EventArgs e)
{
    logear(this.textBox1.Text, this.textBox2.Text);
}
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

De esta forma, si el usuario existe en la base de datos permitirá el acceso, en caso de que no. Le notificará al usuario que es incorrecto.

Arreglos y Colecciones

De acuerdo con Google (2022); un **arreglo** es una matriz donde puedes almacenar diferentes datos siempre y cuando, sean del mismo tipo (string, int, double, etc). Los arreglos ayudarán a que no tengas que crear muchas variables del mismo tipo.

La sintaxis de un arreglo es la siguiente:

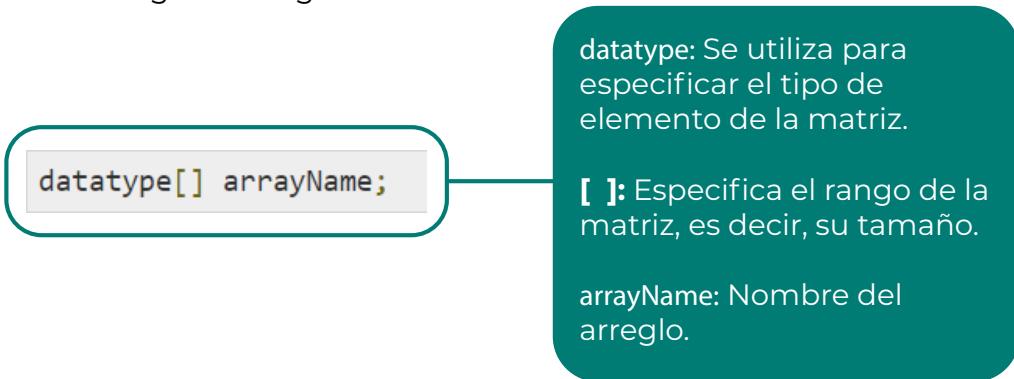


Tabla 1. Sintaxis de una matriz.

Imagina lo siguiente: dentro de tu desarrollo web necesitas obtener todos los libros relacionados con cierto autor. Al momento que obtienes la información de la base de datos, puedes guardar dicha información en un arreglo. Esto te permitirá solo crear una variable para guardar múltiples datos relacionados con un autor, por lo tanto, la respuesta será más rápida y el desarrollo web no utilizará tantos recursos.

Para agregar un valor a un arreglo; primero necesitas inicializarlo. Al momento que se inicializa, lograrás crear una instancia de dicho arreglo y podrás agregar valores. Siguiendo con el ejemplo de varios libros de un mismo autor, colocarás el siguiente código:

```
string [] libro= new string [50]
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Por otro lado, existen las colecciones, De acuerdo con Microsoft (2022e), una colección sirve para almacenar diferentes objetos que pueden ser utilizados o consultados en un futuro. En C# existen 2 tipos de colecciones: las genéricas y las no genéricas. Las genéricas son las más utilizadas, ya que al momento de ejecutarse a nivel compilador protegen mejor la información.

Vamos a utilizar nuevamente el ejemplo del usuario y contraseña; imagina que estás haciendo el módulo de alta de un cliente, dicho módulo contendrá los datos del usuario como: su nombre, edad, genero, usuario, contraseña, etc. Puedes utilizar una colección para almacenar toda esta información para cada usuario y a su vez, enviarla a la base de datos para que sea almacenada o mismo extraer una serie de datos de tu tabla o tablas de las bases de datos y almacenarlos en las colecciones.

A continuación, se presentan las colecciones que puedes crear:

Nombre de la colección	Comportamiento	Características
Queue (Queue<T>) Stack (Stack <T>)	Comportamiento first in first out.	Lista secuencial que descartan el elemento una vez que se recupera su valor.
ArrayList, String Collection, Hashtable, SortedList, ListDictionary, Name ObjectCollectionBase, KeyedCollection < TKey, TItem >	Te permite acceder a los elementos, gracias a un índice o por una clave.	Solo puedes acceder a ellas mediante un índice o una clave.
IList <T>, IDictionary< TKey >, TValue >	Cada elemento tendrá un valor junto con una clave y ambos se combinarán.	Necesitas la clave y el valor para acceder a ellos.
ListDictionary,Hashtable,Dictionary < TKey, TValue >	Realiza búsquedas y recupera de forma rápida la información.	Las búsquedas son más rápidas.
StringCollection y StringDictionary	Solo aceptan cadenas de caracteres.	Basadas en IList y IDictionary

Tabla 2. Tipos de colecciones.

ASP.NET

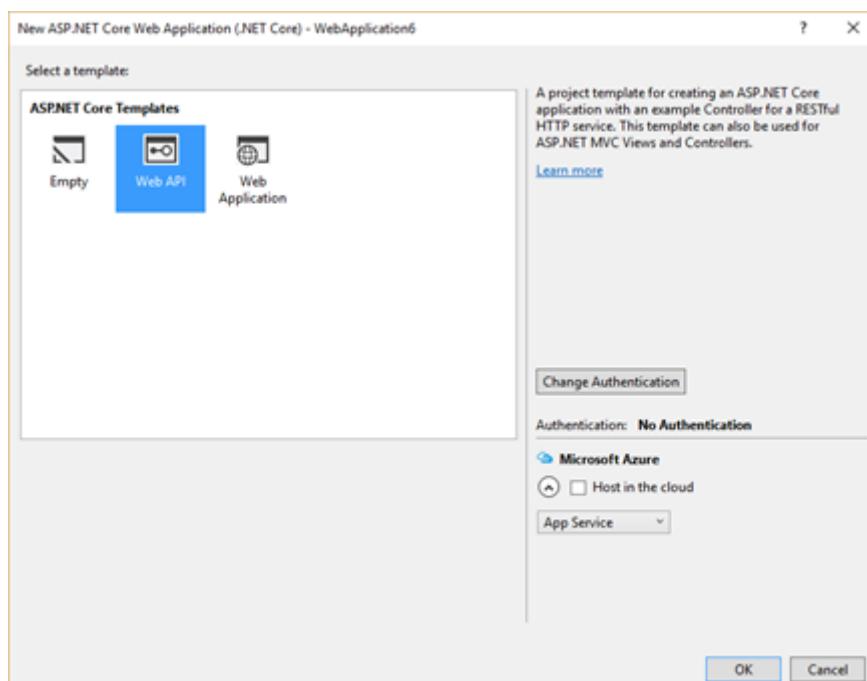
Como se comentó al principio del tema, ASP.NET permitirá realizar el back end mediante C#.

De acuerdo con Microsoft (2022f), ASP.NET permite hacer lo siguiente:

- Páginas web.
- API'S Rest.
- Microservicios.

A continuación, verás los pasos para realizar un back end para una aplicación móvil, en este escenario se va a hacer un back end para una agenda móvil:

1. Crea una nueva aplicación en Visual Studio.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Cambia el archivo “Startup.cs” para que se pueda recibir mediante el puerto 5000 tráfico mediante http a través del siguiente código:

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        // For mobile apps, allow http traffic.
        app.UseHttpsRedirection();
    }

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Después, crea una clase que represente los elementos que tendrá la lista de cosas por hacer a través del siguiente código:

```
using System.ComponentModel.DataAnnotations;

namespace TodoAPI.Models
{
    public class TodoItem
    {
        [Required]
        public string ID { get; set; }

        [Required]
        public string Name { get; set; }

        [Required]
        public string Notes { get; set; }

        public bool Done { get; set; }
    }
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Para los métodos API, se requieren trabajar con los datos, para esto usarás la interfaz original de Xamarin, la cual se crea a través del siguiente código:

```
using System.Collections.Generic;
using TodoAPI.Models;

namespace TodoAPI.Interfaces
{
    public interface ITodoRepository
    {
        bool DoesItemExist(string id);
        IEnumerable<TodoItem> All { get; }
        TodoItem Find(string id);
        void Insert(TodoItem item);
        void Update(TodoItem item);
        void Delete(string id);
    }
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

5. Luego, crea una colección privada para los elementos de la lista, utilizando el siguiente código:

```
using System.Collections.Generic;
using System.Linq;
using TodoAPI.Interfaces;
using TodoAPI.Models;

namespace TodoAPI.Services
{
    public class TodoRepository : ITodoRepository
    {
        private List<TodoItem> _todoList;

        public TodoRepository()
        {
            InitializeData();
        }

        public IEnumerable<TodoItem> All
        {
            get { return _todoList; }
        }

        public bool DoesItemExist(string id)
        {
            return _todoList.Any(item => item.ID == id);
        }

        public TodoItem Find(string id)
        {
            return _todoList.FirstOrDefault(item => item.ID == id);
        }

        public void Insert(TodoItem item)
        {
            _todoList.Add(item);
        }
    }
}
```

```
public void Update(TodoItem item)
{
    var todoItem = this.Find(item.ID);
    var index = _todoList.IndexOf(todoItem);
    _todoList.RemoveAt(index);
    _todoList.Insert(index, item);
}

public void Delete(string id)
{
    _todoList.Remove(this.Find(id));
}

private void InitializeData()
{
    _todoList = new List<TodoItem>();

    var todoItem1 = new TodoItem
    {
        ID = "6bb8a868-dba1-4f1a-93b7-24ebce87e243",
        Name = "Learn app development",
        Notes = "Take Microsoft Learn Courses",
        Done = true
    };

    var todoItem2 = new TodoItem
    {
        ID = "b94afb54-a1cb-4313-8af3-b7511551b33b",
        Name = "Develop apps",
        Notes = "Use Visual Studio and Visual Studio for Mac",
        Done = false
    };

    var todoItem3 = new TodoItem
    {
        ID = "ecfa6f80-3671-4911-aabe-63cc442c1ecf",
        Name = "Publish apps",
        Notes = "All app stores",
        Done = false,
    };

    _todoList.Add(todoItem1);
    _todoList.Add(todoItem2);
    _todoList.Add(todoItem3);
}
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

6. Después, configura su implementación en el archivo “Startup.cs” usando el siguiente código:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<ITodoRepository, TodoRepository>();
    services.AddControllers();
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

7. Crea el controlador para poder manejar los requerimientos del usuario usando el siguiente código:

```
[ApiController]
[Route("api/[controller]")]
public class TodoItemsController : ControllerBase
{
    private readonly ITodoRepository _todoRepository;

    public TodoItemsController(ITodoRepository todoRepository)
    {
        _todoRepository = todoRepository;
    }
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Ya que realizaste la configuración podrás establecer la conexión con la base de datos para obtener la información solicitada por el front end. En el siguiente video podrás revisar como se realiza:

Aguilar Systems MX. (2019, 4 de mayo). *Como hacer una Conexión a Base de Datos de SQL Server desde ASP.NET Visual C# con Web.config* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=1FB_X3adKpQ

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Para leer los elementos puedes utilizar el siguiente código (método GET):

```
public IActionResult List()
{
    return Ok(_todoRepository.All);
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para añadir elementos puedes usar el siguiente código (método POST)

```
public IActionResult Create([FromBody]TodoItem item)
{
    try
    {
        if (item == null || !ModelState.IsValid)
        {
            return BadRequest(ErrorCode.TodoItemNameAndNotesRequired.ToString());
        }
        bool itemExists = _todoRepository.DoesItemExist(item.ID);
        if (itemExists)
        {
            return StatusCode(StatusCodes.Status409Conflict, ErrorCode.TodoItemIDInUse.ToString());
        }
        _todoRepository.Insert(item);
    }
    catch (Exception)
    {
        return BadRequest(ErrorCode.CouldNotCreateItem.ToString());
    }
    return Ok(item);
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para actualizar algún elemento, puedes utilizar el siguiente código (método PUT):

```
public IActionResult Edit([FromBody] TodoItem item)
{
    try
    {
        if (item == null || !ModelState.IsValid)
        {
            return BadRequest(ErrorCode.TodoItemNameAndNotesRequired.ToString());
        }
        var existingItem = _todoRepository.Find(item.ID);
        if (existingItem == null)
        {
            return NotFound(ErrorCode.RecordNotFound.ToString());
        }
        _todoRepository.Update(item);
    }
    catch (Exception)
    {
        return BadRequest(ErrorCode.CouldNotUpdateItem.ToString());
    }
    return NoContent();
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Lo anterior lo puedes adaptar a cualquier desarrollo que estés realizando ya sea web o móvil.

En el siguiente video podrás observar cómo se crea un desarrollo back end que permita el registro de clientes en un desarrollo web:

BaezStone Creators. (2021, 29 de junio). *Desarrollo BackEnd en Asp.net Core (FullStack Parte 1)* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=RHzlnfpsvwo>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Recuerda, que existen diferentes lenguajes de programación para que realices el back end, antes de tomar una decisión acerca de cuál vas a utilizar, debes tener en cuenta lo siguiente:

1. Experiencia del equipo de desarrolladores.
2. Presupuesto.
3. Tiempo para elaborar el proyecto.
4. Equipo de cómputo con el que se contará para el proyecto.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

C# es uno de los lenguajes de programación más utilizados hoy en día tanto para front end como back end. Su fácil entendimiento y su estructura orientada a objetos te permitirá programar (con la ayuda de ASP.NET) un back end fácil de implementar, seguro, y con menor presupuesto a comparación de otros lenguajes.

¿Consideras que C# es mejor para front end o para back end?

¿Consideras ASP.NET lo suficientemente efectivo como para soportar el back end?

Referencias bibliográficas

- Aguilar Systems MX. (2019, 19 de abril). *Como hacer el Hola Mundo desde ASP.NET en Visual C# con Visual Studio Community con .NET Framework* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=Xjqq6Fpf-m4>
- Aguilar Systems MX. (2019, 4 de mayo). *Como hacer una Conexión a Base de Datos de SQL Server desde ASP.NET Visual C# con Web.config* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=1FB_X3adKpQ
- BaezStone Creators. (2021, 29 de junio). *Desarrollo BackEnd en Asp.net Core (FullStack Parte 1)* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=RHz1nfpsvwo>
- Google (2022). *Programación en .NET C# Arreglos*. Recuperado de <https://sites.google.com/site/programacionnetc/arreglos>
- Microsoft. (2022a). *Métodos*. Recuperado de <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/classes-and-structs/methods>
- Microsoft. (2022b). *Tipos y miembros de C#*. Recuperado de <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/types>
- Microsoft. (2022c). *Objetos: creación de instancias de tipos*. Recuperado de <https://docs.microsoft.com/es-es/dotnet/csharp/fundamentals/object-oriented/objects>
- Microsoft. (2022d). *Usar atributos en C#*. Recuperado de <https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/attributes>
- Microsoft. (2022e). *Collections (C#)*. Recuperado de <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/collections>

Para saber más

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Lecturas

Para conocer más acerca de **C# en back end**, te sugerimos leer lo siguiente:

- Microsoft. (s.f.). *ASP.NET Documentation*. Recuperado de https://docs.microsoft.com/en-us/aspnet/core/?WT.mc_id=dotnet-35129-website&view=aspnet-core-6.0
- W3Schools. (s.f.). *ASP.NET Tutorial*. Recuperado de <https://www.w3schools.com/asp/>

Videos

Para conocer más acerca de **C# en back end**, te sugerimos revisar lo siguiente:

- Billy Tech. (2020, 19 de mayo). *ASP.NET Core Basics #1 | Full Stack Tutorial* [Archivo de video] Recuperado de <https://www.youtube.com/watch?v=m5UzOL7lfeI>
- InfoToolsSV. (2021, 08 de mayo). *ASP.NET C || LOGIN básico con procedimientos almacenados y Sesiones desde cero (SQL, CSS, JS) [2021]* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=iOmC2JufnnI>

Checkpoints

Asegúrate de:

- Comprender la utilidad de C# en el back end para poder recibir los requerimientos del front end y regresar lo solicitado.
- Comprender como utilizar ASP.NET para programar el back end de un desarrollo web o móvil.
- Comprender la importancia de un lenguaje de programación orientado a objetos para programar el back end.

Requerimientos técnicos

- C#.
- Visual Studio.
- ASP.NET

Prework

- Deberás revisar el tema de Javascript.
- Deberás revisar el tema de bases de datos.
- Descarga ASP.NET programa de:

Microsoft. (s.f.). *Download .NET*. Recuperado de
<https://dotnet.microsoft.com/en-us/download>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

- Descarga Visual Studio programa de:

Microsoft. (2022h). *Descarga Visual Studio*. Recuperado de
<https://visualstudio.microsoft.com/es/vs/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

- Leer la siguiente información, para repasar cómo crear una base datos, tables etc. En MySQL.

MySQLTUTORIAL. (s.f.). *MySQL Node.js*. Recuperado de
<https://www.mysqltutorial.org/mysql-nodejs/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Descargar el txt de la actividad (Archivo adjunto)

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.