

- Arquitectura Web
- Capas
- Entorno de trabajo
- Herramientas
- Visual Studio
- Repos-Git
- Objetos en C#
  - Modelos
  - Propiedades
  - Herencia y Relaciones

- Framework ASP .Net Core
  - MVC - Introducción
  - **HTML / CSS / JavaScript**
  - Vistas y Controladores
  - Validaciones y otros
- LinQ
- Entity Framework
- Persistencia | Inicializador DB

## Vistas y Controladores

## Validaciones y otros

- Built-in / Remote / Custom
- ModelState
- Mensajes de error
- ViewModels

## • Routing

- ¿Qué son las vistas? → información PARA GENERAL lo que ve el usuario
  - Mecanismos para representar información al cliente, las vistas se hacen con RAIZOR
- ¿Qué son los controladores? → funcionalidad → reglas de negocio
  - manejar las rutas, escucha la ruta a donde intentó entrar entonces procesa y devuelve una respuesta
- ¿Qué son los modelos? → información
  - modelos de información se van a representar como tablas en la base de datos
- ¿Cómo se valida?
- Qué son los Routing
- ¿Qué es linQ? consultas dentro de objetos en c#

-¿Qué es Entity framework? conectividad de la app con la base de datos

-Como se persisten los datos en la base de datos?

-¿Qué es y cómo se usa Scaffolding? Mecanismo que nos provee el framework ASP .NET Core que nos permite crear los ABM “alta, baja, modificación” automáticamente

-Como se hace la autenticación?

=Cuando vea cómo se hacen las validaciones en ASP MVC, las implemente tanto en front como en back y DEBO SABER EL POR QUE ES ASI (LO VAN A PREGUNTAR)

Request - Response → C# .net core 3.1

REVISAR EL PROYECTO DE CSHARP-INTRO EN EL GIT DEL PROFESOR

-Los tipos de datos pueden comportarse de dos formas, por valor y por referencia : (los objetos se pasan por referencia) - lo que se pasa por valor no afecta al objeto si no a una copia

-Todas las clases se manejan por referencia, si genero un objeto de esa clase se manejan y pasan por referencia cada vez que pase un parámetro a un método se pasa por referencia

-Los datos primitivos se manejan por valor

**HTML** no es un lenguaje de programación, es un lenguaje de marcado que define la estructura de un determinado contenido.

**Interpolación de string:** para lograr esto se pone un \$ delante de la cadena de caracteres, esto nos permite poner variables dentro de la cadena de caracteres que se van a resolver antes de que el string se muestre {clase.Metodo}

### Tipos de datos nulables:

Los tipos nulables por naturaleza son los que se envían por referencia sintaxis `Nullable<int> nullableX = null; O int? nullableY = null;`

-el “?” quiere decir que el tipo de dato acepta null

-Que sea nutable me permite tener dos propiedades utilizables -> **HasValue**: se utiliza para evaluar si la variable tiene un valor asignado o tiene null -> **Value**: devuelve el tipo de dato sin el nutable

**Operación Ternaria:** lo que hace es evaluar una condición

#### sintaxis

```
string resultado = nullableX?.ToString();
// si nullableX tiene null, resultado = null
// si nullableX no tiene null, resultado = "0" por ejemplo

// forma 2 - ternarias
resultado = nullableX.HasValue ? nullableX.ToString() : null;

// forma 3
if (nullableX.HasValue)
{
    resultado = nullableX.ToString();
}

// forma 1
resultado = nullableX?.ToString() ?? "null";

// forma 2
resultado = nullableX.HasValue ? nullableX.ToString() : "null";

// forma 3
if (nullableX.HasValue)
{
    resultado = nullableX.ToString();
}
else
{
    resultado = "null";
}
```

variable = (**condición**) Si se cumple se ejecuta lo que está después del signo ? **valorVerdadero** : si es falso entonces se ejecuta lo que está después de los dos puntos **valorFalso**;

variable = (**condición**) ? **valorVerdadero** : **valorFalso**

La podemos leer como “Si la condición se cumple guardar en variable el valor verdadero, si no se cumple asignar el valor falso”

la forma 1 solo se usa para cosas nullables y la forma 2 para cualquier IF que se quiera simplificar en una línea

Usar decimal si es un número con punto

**DateTime** es un tipo de dato para usar fechas

### Validación de modelos

ASP.NET Core MVC admite la validación decorando el objeto de modelo con los atributos de validación de anotación de datos. Los atributos de validación se comprueban en el lado cliente antes de que los valores se registren en el servidor, y también en el servidor antes de llamar a la acción del controlador.

```
C# Copiar
using System.ComponentModel.DataAnnotations;
public class LoginViewModel
{
    [Required]
    [EmailAddress]
    public string Email { get; set; }

    [Required]
    [DataType(DataType.Password)]
    public string Password { get; set; }

    [DisplayName("Remember me?")]
    public bool RememberMe { get; set; }
}
```

donde se hace la conexión a la base de datos (laun settig) y cuando se crea y se inicializa

2

- .la semilla hace que cada vez que se inicialice la base de datos se inicializan los datos tales
- .y donde se crea cada cosa en la base de datos
- .donde estan las validaciones (entro al archivo y le muestro)
- .en el controlador donde se define y se valida que sea o que hace cada rol
- . ppp mail, queria saber como puedo rendir el final, si son preguntas teóricas o tengo que presentar de nuevo el proyecto

STARTUP → se configura todos los servicios que usa la aplicación

- scaffolding : nos permite crear automáticamente un objeto con sus propiedades y nos da un adm ???
  - scaffolding: lo que hace es crear controladores y vistas asociadas que permiten interactuar entre los modelos y la base de datos usando el DBCONTEXT
- El constructor de dbContext se usa en la clase STARTUP Aquí se configura la creación de la base de datos una sola vez

- identity management : validar seguridad
- \* etn “va de punta a punta

Arquitectura de la WWW (World Wide Web)

Arquitectura lógica de Internet:

- Normalmente en Internet se usa la [arquitectura cliente-servidor](#).
- Todas todas las computadoras que están en la red, unos ofrecen servicios (los llamados **servidores**) y otros usan esos servicios (los denominados **clientes**).
- Cuando el cliente está viendo páginas, está accediendo a un servicio que le ofrece el **servidor de páginas WEB**.

DOM - Modelo de objetos de documento

- Podemos cambiar todos los atributos HTML de la página
- Podemos cambiar todos los estilos de la página.
- Podemos agregar nuevos elementos y atributos.
- Podemos eliminar todos los elementos y atributos.
- Podemos reaccionar a los eventos de los elementos.
- Podemos crear nuevos eventos en la página.

Roles en la Arquitectura

- Interfaz de usuario (Experiencia de usuario)
- Lógica de negocio (que hay que resolver, cómo lo resolvemos)
- Acceso a datos (como persisto la información)

Este patrón es conocido y se llama Modelo de capas

La separación de roles en tres capas hace más fácil reemplazar o modificar una capa sin afectar a las otras.

*“Es una separación conceptual, hace referencia a como una solución es segmentada desde el punto de vista lógico”*

PRESENTACIÓN	NEGOCIO	DATOS
<ol style="list-style-type: none"> <li>1. Recupera la información del usuario y la envía al servidor</li> <li>2. Reciben los resultados de la capa de proceso</li> <li>3. Visualiza la presentación al usuario.</li> <li>4. Consistencia de datos.</li> </ol>	<ol style="list-style-type: none"> <li>1. Recibe la entrada de datos de la capa de presentación</li> <li>2. Interactúa con la capa de acceso a datos para realizar operaciones</li> <li>3. Manda los resultados procesados a la capa de presentación</li> </ol>	<ol style="list-style-type: none"> <li>1. Almacena datos</li> <li>2. Recupera datos</li> </ol>

programación del lado del servidor c# y del lado del cliente js

## Proyecto

Marchese

### Temas – Ejemplos:

1. Agenda turnos
2. Historias clínicas
3. Carrito compras
4. Lista de compras
5. Reserva recursos
6. Manejo de licencias
7. Instituto educativo
8. Sistema ERP básico

### Requerimientos básicos:

- Al menos 4 modelos
- ABM full para cada modelo en BE
- ABM requeridos en FE. Al menos para 2 modelos
- Relaciones al menos una de cada una
  - 1 a 1
  - 1 a muchos
  - Muchos a muchos
- Persistencia (Base de datos)

que es ABM?

arquitectura web:

cliente, servidor, que se comunican por requets y response

Todos los objetos son únicos, cuando se crea un objeto se guarda un espacio de memoria

//todas las clases se manejan por referencia

//namespace es un concepto administrativo

pull push commit branch

MVC

back C#

Framework ASP .NET Core

VIEW son las vistas y cada vista es un documento HTML (actúan con el modelo) lenguaje reysol

**31/03/2021**

para abrir un proyecto y que se pueda compilar es con “open a project or solution y abris el archivo “sln”

**14 /04/2021**

## Entity framework es una librería

ORM : es una tecnología que nos permiten abstraer las consultas

los Scaffolding nos permiten ejecutar acciones que se llaman CRUD: create, redet\*, update, delete, sobre una entidad en particular usando entity framework

```

[Required(ErrorMessage = "El campo {0} es requerido")]
[MaxLength(100, ErrorMessage = "{0} admite un máximo de {1} caracteres")]
[EmailAddress(ErrorMessage = "El campo {0} debe ser un email válido")]
12 references
public string Email { get; set; }

[MaxLength(100, ErrorMessage = "{0} admite un máximo de {1} caracteres")]
12 references
public string LinkedIn { get; set; }

[MaxLength(100, ErrorMessage = "{0} admite un máximo de {1} caracteres")]
12 references
public string Twitter { get; set; }

[MaxLength(100, ErrorMessage = "{0} admite un máximo de {1} caracteres")]
12 references
public string Instagram { get; set; }

[MaxLength(100, ErrorMessage = "{0} admite un máximo de {1} caracteres")]
12 references
public string Facebook { get; set; }

9 references
public Alumno Alumno { get; set; }

```

```

public Guid AlumnoId { get; set; }
9 references
public Alumno Alumno { get; set; }

```

[ForeignKey(nameOf(Alumno))]  
public Guid Alumnold

<https://www.entityframeworktutorial.net/efcore/cli-commands-for-ef-core-migration.aspx>

21/04/2001

```

namespace usando_entity_framework
{
    3 references
    public class Startup
    {
        0 references
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        1 reference
        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        0 references
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddDbContext<InstitutoDbContext>(options => options.UseSqlite("filename=instituto.db"));
            // services.AddDbContext<AlumnosDbContext>(options => options.UseSqlServer(Configuration.GetConnecti

            services.AddControllersWithViews();
        }
    }
}

```

Marchese Federico

escafordin de  
controladores  
y vistas me  
permiten crear  
un modelo en  
particular

```

// POST: Alumnos/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to, for
// more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(Guid id, Alumno alumno)
{
    if (id != alumno.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(alumno);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
        }
    }
}

```

28/04/2021

autorización y autenticación  
la forma en la que vamos a autenticar será a través de las cookies

- 
- ClaimsIdentity
  - FirstOrDefault
  - AddClaim
  - CookieAuthenticationDefault
  - ClaimTypes - Name, Role, NameIdentifier
  - HttpContext - SignInAsync (cuando hace eso el cliente ya esta autenticado)
  - ScaffoldColum

12/05/2021

- reglas de negocio
- asignación de salas
- borrador de cartelera

02/06/2021

el control es el mismo, controla todo el flujo, si la complejidad es grande hacer otro método action . la vista se puede reutilizar si es igual, si tiene muchas diferencias no



---

- que framework es el que usamos?

asp .net core

- que es razor y para qué sirve?

genera las vistas de los controladores

- ¿Cuál es el ciclo de vida de un request en .net?

el ciclo de vida no tiene estado, es un protocolo sin estado solo hay un request

- qué 3 cosas toma en cuenta el model binder

=> data enviada en el body del request, data enviada como parámetros querystring y data enviada como path params (en la url, por ejemplo /peliculas/5)

- que es un modelo de datos con tus palabras

1- son las entidades que quiero representar

2- es lo que se quiere representar, son las entidades con las que se va a trabajar, en una vista es la entidad con la que voy a trabajar

**-cómo funciona linq:** es para entity framework, es para escribir consultas de forma genérica, entity framework lo traduce en consultas, es una herramienta que tiene su propia consulta

**Entity** sirve como intermediario, genera los script para la base de datos  
cual es la responsabilidad de cada capa

-donde esta la definicion exacta de la base de datos?

→ esta en las configuraciones

[9:38 p. m., 13/7/2022] Federico Ort: bueno te va a preguntar cosas técnicas y del trabajo

[9:38 p. m., 13/7/2022] Federico Ort: es que puede ser cualquier cosa

[9:38 p. m., 13/7/2022] Federico Ort: cosa de los view

[9:38 p. m., 13/7/2022] Federico Ort: cosas de los input

de los actions cómo se redirige y demás → asp son métodos, es para indicar en que controlador va la petición y se le paso el parametro que se necesita

[9:39 p. m., 13/7/2022] Federico Ort: un poco de todo

cosas de la base de datos