

DOM: Es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

HTML4: Es una versión anterior de html5, por lo que tiene menos funciones en comparación con html5, No proporciona un mejor manejo de errores como html5.

HTML5: Es una extensión de html4. Es la quinta versión de HTML, Html proporciona consistencia en documentos mal formados y tiene un mejor manejo de errores.

CSS3: Es la última evolución del lenguaje CSS y tiene como objetivo extender CSS2. Trae muchas novedades, como esquinas redondeadas, sombras, degradados , transiciones o animaciones , así como nuevos diseños como múltiples columnas , cajas flexibles o diseños de cuadrícula

Bootstrap: Es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

SPA: Son las siglas de Single Page Application. Es un tipo de aplicación web donde todas las pantallas las muestra en la misma página, sin recargar el navegador.

JQuery: es una biblioteca de JavaScript que simplifica la forma de desarrollar aplicaciones web.

Javascript: Es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web.

Responsividad: Es adaptar el tamaño de la pantalla, a diferentes dispositivos

Front-end: no es solo lo visual, también involucra js.

UI: User Interface

UX: User Experience

ASP.Net --> viejito propiedad de Microsoft
ASP.Net CORE en Visual Studio -> abierto

Snippets: código de C# embebido en el html

Json: tipo de archivo que se estructura con clave valor

IIS: El servidor web de .Net

Java: Apache, Tomcat, JBoss, etc.

Render: Procesar y cargar (en pantalla)

onhover

AJAX: Asynchronous JavaScript and XML, lo que me permite hacer llamadas asincrónicas al backend desde

HTML= HyperText Markup Language

JS= JavaScript

CSS= Hoja de estilo en cascada

DEV- QA/UAT - PRO (live)

SPA= Single Page Application

Customizar = personalizar

landing page= página principal

onhover/hover= Cambie de color cuando se presione

Json: Tipo de archivo que se estructura tipo clave valor

nuggets: Manejador de dependencias

Render/Renderizar: Procesar y cargar en pantalla (imagenes, codigo etc)

Sql : lenguajes

SqlServer management studio -> es el motor

Hard coded: Código quemado

popular =llenar

signatura del método: modificador de acceso, nombre, tipo de retorno (si los hay), y parámetros si los hay

Modificador acceso: public, protected

DTO: DATA TRANSFER OBJECT

EF: Entity Framework

Arquitecto Software:
Web stack: Plantilla

Controller: CRUD

CRUD= CREATE,READ,UPDATE,DELETE

ActionResult: Interface a nivel de clase

Full Stack developer: front end / back end

API conectar backend-front-end

REST-> Es una arquitectura para mandar y recibir información se usa mas JSON
arquitectura de presentación de datos

SOAP: Arqui dividida por niveles

rest vs soap -- Examen ojo sale

soap el formato es XML para ser transmitidos, para ser puestos en BD a través de una api
Rest es una arquitectura de presentación de datos en tipo json que es la mas comun

Protocolo HTTP: Protocolo de transferencia de hipertexto

servicio, back end donde está la lógica de negocios

servicio web, -> este servicio debe correr a través de la red
api no es un SW porque diay el api puede que no esté en la web

Vistas Razor: permite escribir el código de nuestras **vistas** de una manera mucho
más ágil y limpia

STARTUP-> archivo configuracion

ojo si quiero llamarlo o invocar a otro metodo, puedo poner rutas
ejemplo:

```
[Route("[action]")]  
[HttpGet]
```

de esa forma, cuando recibo el url, puedo poner el nombre del metodo que quiero consumir,
sin necesidad de dejar el por defecto GET y no especificarlo en el URL

```
[Route("[metodoName]/{parametros}")]  
[HttpGet]
```

ejemplo

```
[Route("[action]/{date}")]  
[HttpGet]
```