



# Desenvolvimento Web com ASP.NET MVC e ASP.NET Core

Carga Horária: 80

## Pré-requisitos

Para o melhor aproveitamento do Curso Desenvolvimento Web com ASP.NET MVC e ASP.NET Core, é imprescindível ter conhecimento da linguagem C# e de tecnologias Front-End como HTML5, CSS3, JavaScript jQuery, além de conhecimentos de acesso a dados com SQL Server.

# Conteúdo Programático

## Fundamentos de aplicações Web

Arquitetura cliente-servidor.

O servidor Web.

O protocolo HTTP: Requisições (Request); Respostas (Response).

Tecnologias utilizadas nas páginas Web: Exemplo de uma página retornada do servidor contendo HTML, CSS e JavaScript.

Tecnologias utilizadas no servidor.

IIS - Internet Information Services: Habilitando o IIS no Windows; IIS - Criando uma aplicação ASP.NET.

O IIS Express e o Visual Studio.

Criando um projeto Web.

## Web Pages com Razor

O mecanismo Razor: Criando um projeto com uma Web Page usando Razor; Estruturas de repetição; Estruturas condicionais; Render e layout (RenderPage(), RenderBody(), RenderSection()); Request e Form; HTML Helpers.

## ASP.NET MVC - Controllers, Models e Views

Criando um projeto MVC: Conhecendo as partes de um projeto ASP.NET MVC.

Rotas: Iniciando a rota.

Controllers e Actions: Arquitetura MVC (Model - View - Controller); Criando o controller; Criando um action; Retornos comuns de actions; Criando Views.

Os objetos ViewBag e ViewData.

Models - Views fortemente tipadas: Definindo o Model; Consumindo o Model no Controller; Consumindo o Model na View.

Elaboração de formulários: HTML Helpers; Formulário não tipado; Formulário fortemente tipado.

Validação e filtros: Validação; Data Annotation (ModelState, Required, StringLength); Regular Expression; Range; Remote; Compare (Display, DisplayFormat).

Layout: Fluxo de execução do MVC com o layout.

## Entity Framework

Preparando o ambiente.

Database First: Usando o modelo criado.



Model First: Adicionando uma Entity (entidade); Adicionando propriedades; Adicionando associações; Criando o banco de dados; Usando o modelo criado.

Code First: Modelo de domínio; Contexto da conexão; Mapeamento; Mapeamento por código (Método OnModelCreating); Database\_INITIALIZER.

## **Segurança - ASP.NET Identity**

ASP.NET Identity.

OWIN.

Implementações do Visual Studio: Incluindo um menu de acesso; Criando os models e os actions; Escrevendo o código para criar um novo usuário (Definindo a string de conexão, Implementação do cadastro de usuários, Implementação do Login, Implementação do Logout); Executando a aplicação.

## **Serviços - Web API**

Introdução à Web API.

REST na prática.

Usando Web API: Estrutura Web API; Criando um serviço REST; Definindo actions para o verbo HTTP GET; Testando o método GET com parâmetros; Definindo actions para os verbos HTTP POST, PUT e DELETE.

Criando um projeto para consumir o serviço.

## **ASP.NET Core**

NET Core e seus projetos.

Comandos do .NET Core: Criando aplicativos por linhas de comandos; Usando o Visual Studio Code; Usando o Visual Studio; Aplicação ASP.NET Core.

Fluxo de execução de uma aplicação ASP.NET Core.

Controllers e Actions: Adicionando um controller ao projeto.

Microsoft.AspNetCore.Mvc.TagHelpers.

## **Entity Framework Core**

Entity Framework Core (EF Core).

Database Providers.

Usando o Entity Framework Core: Definindo a entidade; Definindo o contexto; Adicionando o DbSet ao contexto; Conectando o EF Core ao

Provider SQL Server; Definindo o componente responsável por criar o banco de dados; Configurando a aplicação para a injeção de dependência; Preparando o controller para incluir e listar registros.

## **ASP.NET Core Identity**

Configuração do ASP.NET Core Identity: Definindo o projeto; Definindo a classe de usuário; Definindo o contexto do banco de dados; Configurando o Identity Core na classe Startup.

Utilização do ASP.NET Core Identity na aplicação: Definindo o controller; Definindo os modelos; Definindo as operações de login e de registro; Definindo o componente com os itens de cadastro, login e logout.

Implementando autorização.

## **Web API Core**

Representação de um Web service REST.

Media Types.

Desenvolvimento de serviços baseados em Web API Core.

Criação do projeto Web API: Definição dos actions (Actions com parâmetros, Valores de retorno de actions, Objetos como parâmetros); Consumindo o Web API Core no cliente.





# IMPACTA

Avenida Paulista, 1009  
11 3254 2200

[atendimento@impacta.com.br](mailto:atendimento@impacta.com.br)