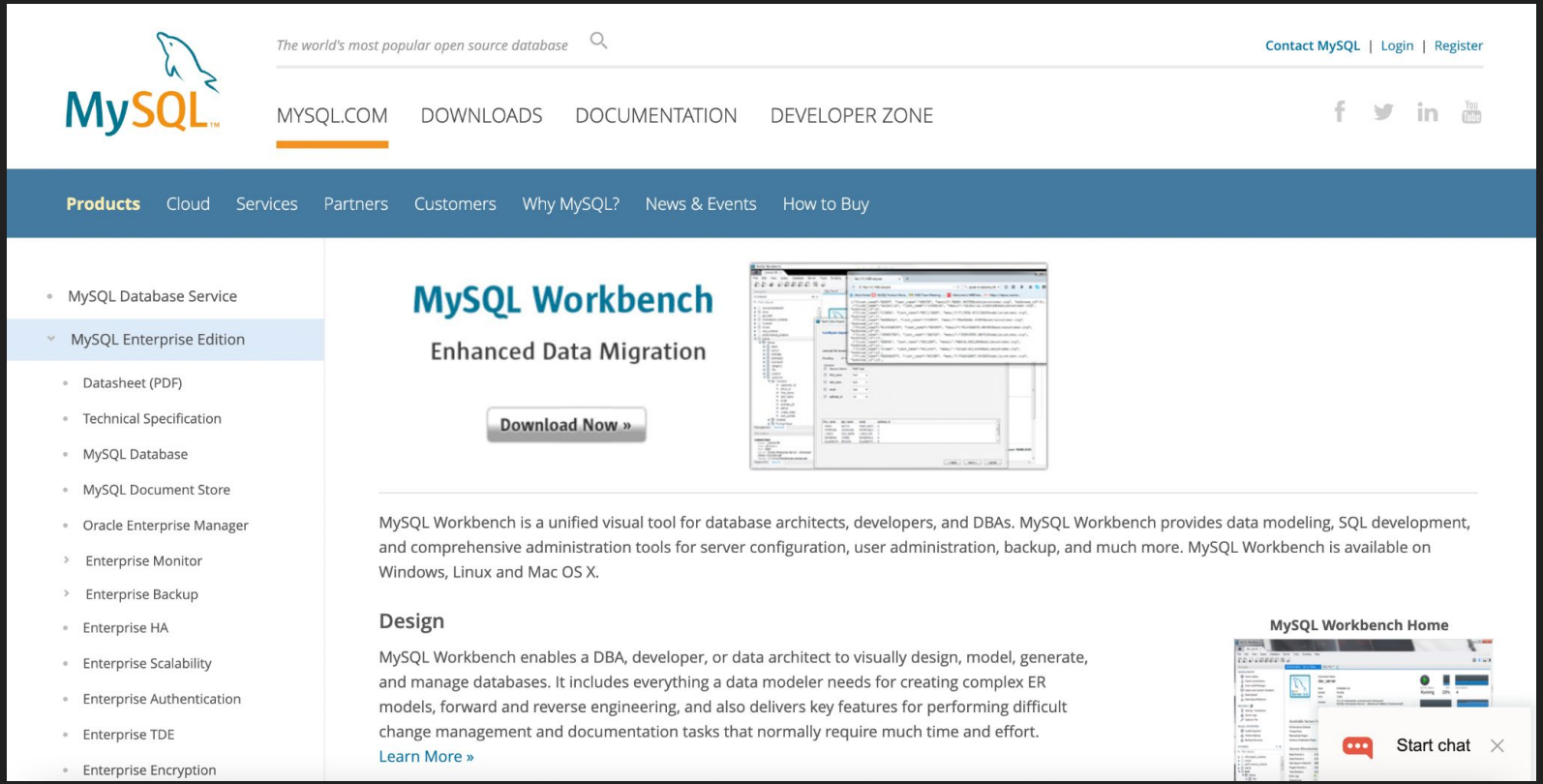
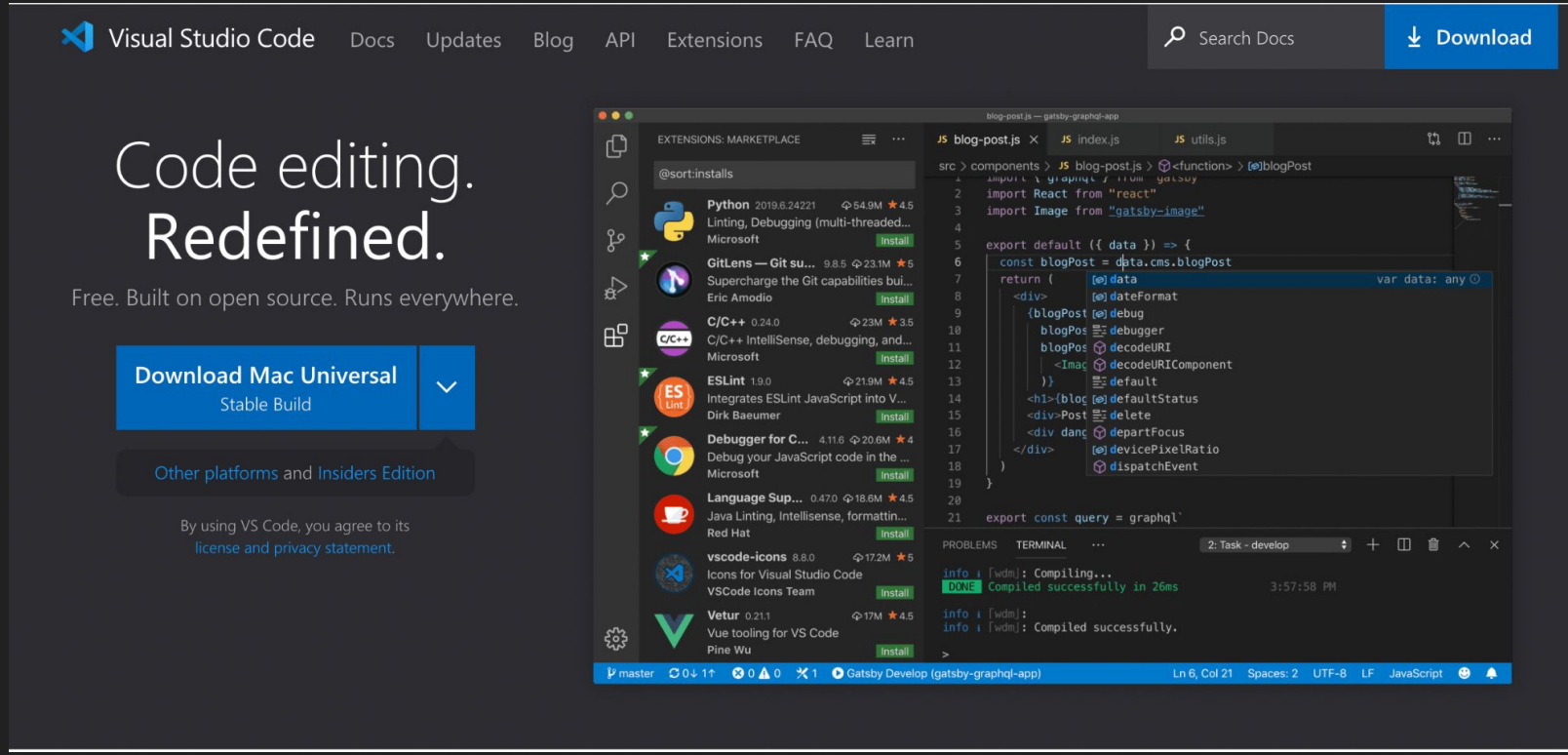
# Aplicação do modelo com .NET Core MVC

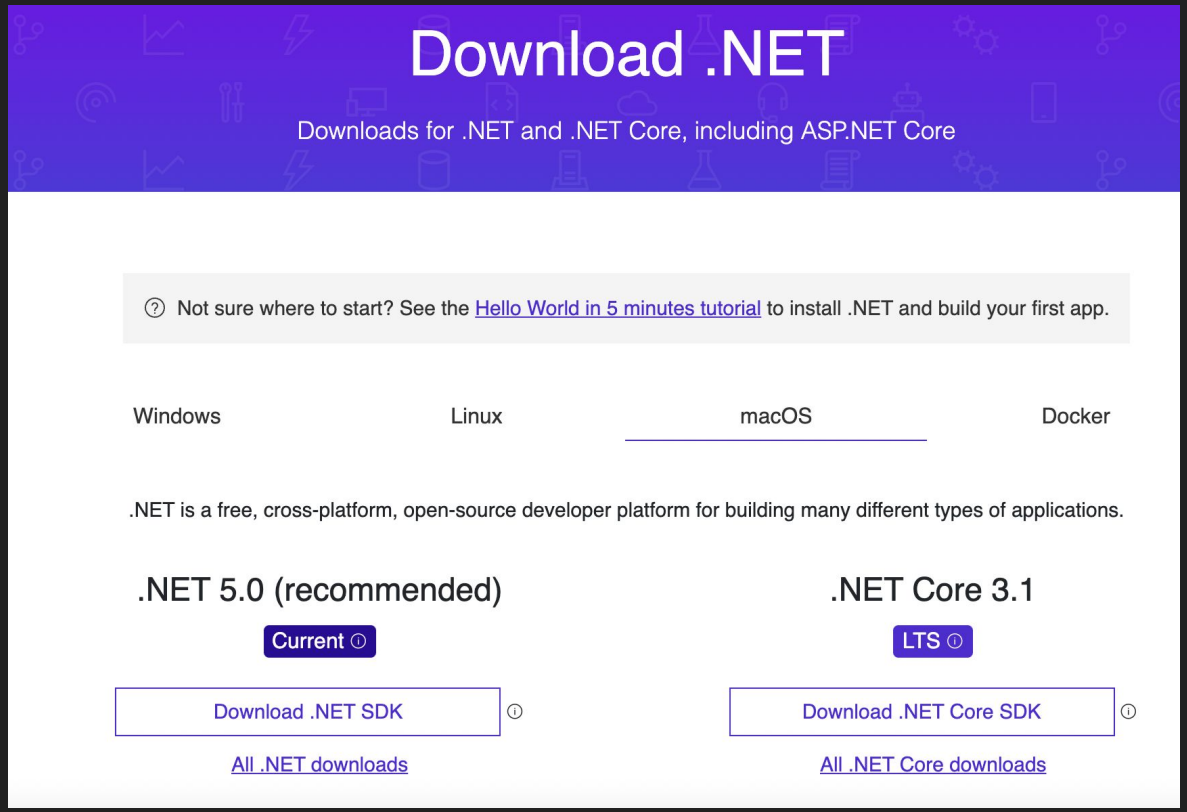
# Instalação MySQL Workbench



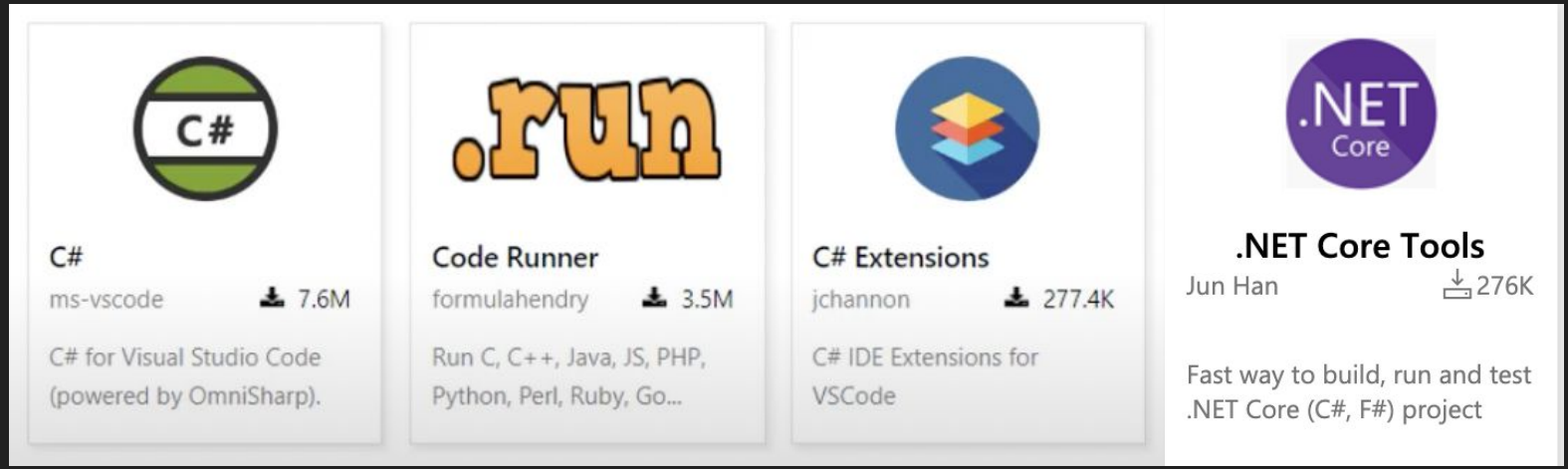
# Instalação Visual Code



# Instalação .NET Core SDK



# Instalação dos package no Visual Code



# Iniciar o projeto

- Criar a pasta do projeto
  - mkdir <pasta>
- Acesse a pasta do projeto
  - cd <pasta>
- Criar o projeto
  - dotnet new mvc --no-https
- Restaurar o projeto
  - dotnet restore
- Compilar o projeto
  - dotnet build
- Executar o projeto
  - dotnet run

# Instalar os pacotes do projeto: **- -version 6.0.1**

- dotnet add package Microsoft.EntityFrameworkCore
- dotnet add package Microsoft.EntityFrameworkCore.Design
- dotnet add package Microsoft.EntityFrameworkCore.Tools
- dotnet add package Microsoft.EntityFrameworkCore.SqlServer
- dotnet add package Pomelo.EntityFrameworkCore.MySql
- dotnet add package Microsoft.EntityFrameworkCore.Relational
- dotnet add package System.Configuration.ConfigurationManager
- dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design
- dotnet tool install --global dotnet-ef
- dotnet tool install --global dotnet-aspnet-codegenerator
- Execute:
  - export PATH=$HOME/.dotnet/tools:$PATH
  - dotnet restore
  - dotnet build

# Instalar os pacotes do projeto

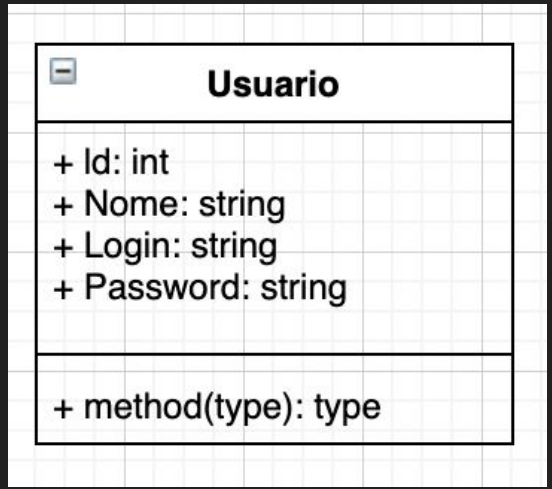
appcsharp.csproj

```
1   <Project Sdk="Microsoft.NET.Sdk.Web">
2
3     <PropertyGroup>
4       <TargetFramework>netcoreapp3.1</TargetFramework>
5     </PropertyGroup>
6
7     <ItemGroup>
8       <PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="3.1.4">
9         <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
10        <PrivateAssets>all</PrivateAssets>
11      </PackageReference>
12      <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="3.1.4" />
13      <PackageReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Design" Version="3.1.4" />
14      <PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="3.2.0" />
15    </ItemGroup>
16
17  </Project>
18
```

# Configurando acesso ao banco de dados

Adicionar o modelo: a classe Usuario dentro da pasta Models

```
namespace democsharp.Models
{
    public class Usuario
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Login { get; set; }
        public string Password { get; set; }
    }
}
```



# Configurando acesso ao banco de dados

Adicionar o modelo: a classe Usuario dentro da pasta Models

```
namespace appcsharpmvc.Models
{
    public class Pessoa
    {
        public int Id {get; set;}
        public string? Nome {get; set;}
    }
}
```

# Configurando acesso ao banco de dados

Adicionar o modelo: a classe Usuario dentro da pasta Models

```
namespace appcsharpmvc.Models
{
  public class Professor : Pessoa
    {
        public string Formacao {get; set;}
        public double Salario {get; set;}
        public virtual ICollection<Turma> ProfessorsTurmas {get;set;}
    }
}
```

# Configurando acesso ao banco de dados

Adicionar o modelo: a classe Usuario dentro da pasta Models

```
namespace appcsharpmvc.Models
{
    public class Turma
    {
        public int Id {get;set;}
        public string Sigla {get; set;}
        public int Ano{get; set;}
        public virtual ICollection<Aluno> Alunos {get;set;}
        public virtual ICollection<Professor> Professores {get;set;}
    }
}
```

# Configurando acesso ao banco de dados

Adicionar a classe DdContext dentro da pasta Models

```
using Microsoft.EntityFrameworkCore;

namespace democsharp.Models
{
    public class MyDbContext : DbContext
    {
        public MyDbContext(DbContextOptions<MyDbContext> options) :
base(options)
        {
        }

        public DbSet<Usuario> Usuario {get; set;}
    }
}
```

# Configurando acesso ao banco de dados

- Toda vez que Modificar o projeto faça
  - <dotnet restore> para restaurar o projeto
  - <dotnet build> para compilar o projeto
  - <dotnet run> para executar o projeto

# Configurando acesso ao banco de dados

Conection String para MySql - arquivo appsettings.json

```
"ConnectionStrings": {
   "MyDbContext":
"server=localhost;port=3306;database=appcsharp;uid=root;pwd=1
2345678;"
 }
```

# Configurando acesso ao banco de dados SDK 3.1

Edite o arquivo Startup.cs e modifique as linhas em verde. Lembre-se que `MyDbContext` pode ter qualquer nome e representa o contexto do seu banco de dados.

```
public class Startup
    {
        public Startup(IConfiguration configuration, IWebHostEnvironment env)
        {
            Environment = env;
            Configuration = configuration;
        }
        public IConfiguration Configuration { get; }
        public IWebHostEnvironment Environment { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddDbContext<MyDbContext>(options =>
            {
                var connectionString = Configuration.GetConnectionString("MyDbContext");

                if (Environment.IsDevelopment())
                {
                    options.UseMySql(connectionString);
                }
                else
                {
                    //options.UseSqlServer(connectionString);
                }
            });
        }
```

# Configurando acesso ao banco de dados SDK 6.0

Edite o arquivo Program.cs e modifique as linhas em verde. Lembre-se que `MyDbContext` pode ter qualquer nome e representa o contexto do seu banco de dados.

```
var builder = WebApplication.CreateBuilder(args);
// Add services to the container.
builder.Services.AddRazorPages();
string mysqlconnection =
builder.Configuration.GetConnectionString("MyDbContext");
builder.Services.AddDbContext<AnimalContext>(options =>
options.UseMySql(mysqlconnection,
ServerVersion.AutoDetect(mysqlconnection)));
var app = builder.Build();
```

# Configurando acesso ao banco de dados

Gerar os Migrations do model para o banco

```
$ dotnet ef migrations add InitialCreate
```

```
$ dotnet ef database update
```

# Gerando os controllers e viewes do modelo

```
$ dotnet-aspnet-codegenerator controller -name UsuariosController
-m Usuario -dc MyDbContext --relativeFolderPath Controllers
--useDefaultLayout --referenceScriptLibraries
```

```
Depois execute:
$ dotnet restore
$ dotnet build
$ dotnet run
```

# Gerando os controllers somente de for para API

```
$ dotnet-aspnet-codegenerator controller -name UsuarioController
-async -api -m Usuario -dc ApiDbContext -outDir Controllers
```

Depois execute:

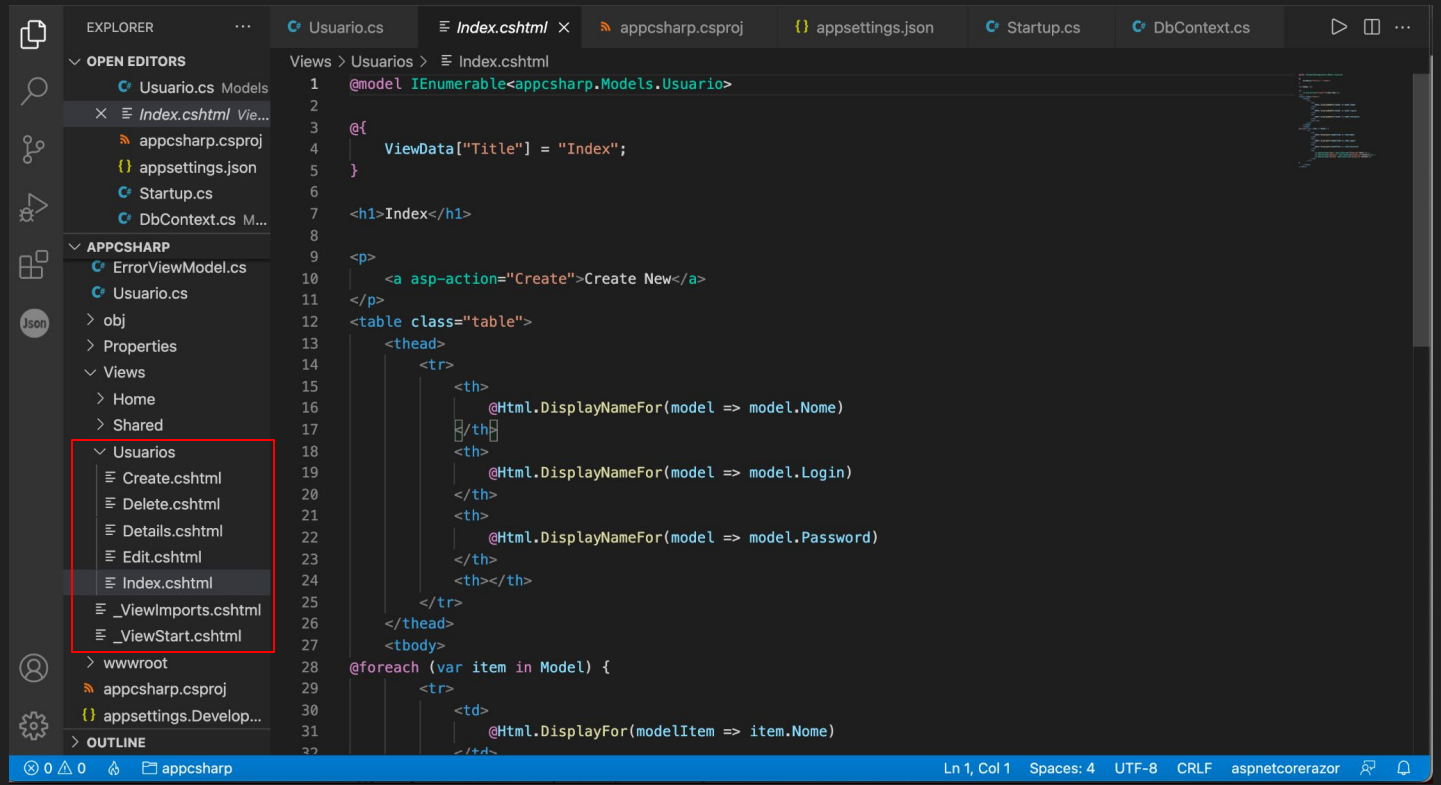
```
$ dotnet restore
```

```
$ dotnet build
```

```
$ dotnet run
```

# Configurando Views e Controllers do Modelo

Criar as  páginas Viewes e Controllers do modelo



# Data Annotations

```
$ using System.ComponentModel.DataAnnotations;
$ using System.ComponentModel.DataAnnotations.schema;
```

# Data Annotations

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace appcsharp.Models
{
        public class Usuario
        {
                [Display(Name = "Código")]
                public int Id { get; set; }
                [Display(Name = "Nome")]
                public string Nome { get; set; }
                [RegularExpression(@"^([\w\.\-]+)@([\w\-]+)((\.(\w){2,3})+)$")]
                public string Login { get; set; }
                [Display(Name = "Senha")]
                public string Password { get; set; }

        }
}
```

# Linkar o modelo na home

Na pasta Views -> shared modifique o _Layout.cshtml conforme as linhas verdes abaixo. Alterando asp-controller="Usuarios" asp-action="Index">Usuarios

```
<div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse" >
                    <ul class="navbar-nav flex-grow-1" >
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area=""
asp-controller="Home" asp-action="Index">Home</a>
                        </li>
                         <li class="nav-item">
                            <a class="nav-link text-dark" asp-area=""
asp-controller="Usuarios" asp-action="Index">Usuarios</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area=""
asp-controller="Home" asp-action="Privacy">Privacy</a>
                        </li>
                    </ul>
                </div>
```

# Projeto final

- Construir um projeto .NET Core MVC
- Usar o Entity FrameWork
- Criar os Models conforme diagrama de classe
- Criar o DdContext dos Models
- Usar conexão de banco de dados com MySql
- Gerar os controllers e as Views
- Entrega 07/12/2022

# Projeto final