

## Programa CRUD hecho en C#

Esta aplicación hecha en C# es un CRUD (*create*, *read*, *update* y *delete*), es decir, tiene la función de conectarse a una base de datos y hacer varias operaciones en esa base, con una interfaz mediante una página web. Para este programa, la conexión es a una base de datos creada en MySQL (mediante el archivo `creacion-lab-mundo.sql`).

Este programa fue hecho en .NET 5.0, así que si se está usando otra versión de .NET, sólo hay que cambiar una línea de código en el archivo `LabApp.csproj` para que compile sin problemas. El contenido de este archivo es:

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net5.0</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference
      ↪ Include="Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore"
      ↪ Version="5.0.7" />
    <PackageReference Include="Microsoft.EntityFrameworkCore"
      ↪ Version="5.0.7" />
    <PackageReference
      ↪ Include="Microsoft.EntityFrameworkCore.Design"
      ↪ Version="5.0.7">
    <IncludeAssets>runtime; build; native; contentfiles;
      ↪ analyzers; buildtransitive</IncludeAssets>
    <PrivateAssets>all</PrivateAssets>
    </PackageReference>
    <PackageReference
      ↪ Include="Microsoft.EntityFrameworkCore.Tools"
      ↪ Version="5.0.7">
    <IncludeAssets>runtime; build; native; contentfiles;
      ↪ analyzers; buildtransitive</IncludeAssets>
    <PrivateAssets>all</PrivateAssets>
    </PackageReference>
    <PackageReference
      ↪ Include="Microsoft.VisualStudio.Web.CodeGeneration.Design"
      ↪ Version="5.0.2" />
    <PackageReference Include="Pomelo.EntityFrameworkCore.MySql"
      ↪ Version="5.0.1" />
    <PackageReference Include="X.PagedList" Version="8.0.7" />
    <PackageReference Include="X.PagedList.Mvc.Core"
      ↪ Version="8.0.7" />
  </ItemGroup>
```

```
</Project>
```

Nótese todos los paquetes usados en el programa. Para cambiar la versión de .NET, hay que cambiar una línea `<TargetFramework>net5.0</TargetFramework>` por la versión de .NET que quiera, por ejemplo, si es .NET 3.1 sería:

```
<TargetFramework>netcoreapp3.1</TargetFramework>
```

También se usan Entity Framework Core y Pomelo.EntityFrameworkCore.MySql para hacer la conexión con la base de datos. Además deberá agregar el servidor, el nombre de la base de datos, el usuario y la contraseña en `Models/lab_mundoContext.cs` para poder manipular la base de datos.

## Objetivo

El objetivo en este caso es crear una página web con una temática de laboratorio en el cual uno puede ver los visitantes e iniciar sesión para poder cambiar dichos visitantes. Cada visitante tiene un rol, así como también cada usuario que inicia sesión. Por parte de los visitantes, hay dos *roles*, uno es nacional y el otro extranjero. Por parte de los empleados, hay tres *roles*, uno es científico, otro es administrador y el último es asistente.

## Análisis

El primer paso es crear una página web sencilla con las páginas que uno crea que van a estar en el producto final, aunque puede ser que falten algunas. Luego se debe hacer la conexión con la base de datos usando un paquete adicional, Pomelo. A partir de ahí, se crea en .NET un proyecto con el estilo modelo vista controlador y se empieza a programar.

## Mockup de HTML

El boceto se puede ver en la carpeta LabMockup. Como se mencionó, esto sería el primer paso.

## Árbol de navegación

Para los controladores es así:

```
Controllers
  AuthController.cs
  HomeController.cs
  InicioController.cs
  NosotrosController.cs
  VisitantesController.cs
```

Para las vistas es así:

```

Views
  Auth
    Cuenta.cshtml
    Denegado.cshtml
    Login.cshtml
  Home
    Index.cshtml
    Privacy.cshtml
  Inicio
    Index.cshtml
  Nosotros
    Index.cshtml
  Shared
    Error.cshtml
    _Layout.cshtml
    _ValidationScriptsPartial.cshtml
  _ViewImports.cshtml
  _ViewStart.cshtml
  Visitantes
    Borrar.cshtml
    Crear.cshtml
    Detalles.cshtml
    Editar.cshtml
    Index.cshtml

```

## Acceso a datos

Para acceder a datos es necesario primero crear una base de datos, el archivo `creacion-lab-mundo.sql` permite hacer eso. Después hay que añadir un paquete que ayude a hacer la conexión a la base de datos, aquí entra Pomelo. Después es necesario instalar varios paquetes (como se ve en `LabApp.csproj`) para poder importar la base; otra forma de verlo es usar el comando `dotnet list package`, cuyo resultado es:

El proyecto "`LabApp`" tiene las referencias de paquete siguientes

```

[net5.0]:
Paquete de nivel superior
  ↪ Solicitado  Resuelto
> Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore
  ↪ 5.0.7        5.0.7
> Microsoft.EntityFrameworkCore
  ↪ 5.0.7        5.0.7
> Microsoft.EntityFrameworkCore.Design
  ↪ 5.0.7        5.0.7
> Microsoft.EntityFrameworkCore.Tools
  ↪ 5.0.7        5.0.7

```

```

> Microsoft.VisualStudio.Web.CodeGeneration.Design
  ↪ 5.0.2          5.0.2
> Pomelo.EntityFrameworkCore.MySql
  ↪ 5.0.1          5.0.1
> X.PagedList
  ↪ 8.0.7          8.0.7
> X.PagedList.Mvc.Core
  ↪ 8.0.7          8.0.7

```

Claramente no todos tiene esa finalidad, los dos últimos son para mostrar un lista entre varias páginas. Luego se usa el siguiente comando para importar todas las tablas:

```

dotnet ef dbcontext scaffold
  ↪ "Server=localhost;Database=lab_mundo;User=miusuario;Password=micontraseña"
  ↪ "Pomelo.EntityFrameworkCore.MySql" -o Models

```