Development of an app in Blazor.

Get started for Access users

Microsoft

Blazor Explanation

What's that Blazor?

3 types (more or less)

Blazor vs Powers

Microsoft

Access and Sql Server on Azure sample

Microsoft

Steps

Create your app in 11 steps

Microsoft

1. New WebAssembly Application

View initial options

Microsoft

2. Nuggets (references)

Install-Package

Microsoft.EntityFrameworkCore.SqlServer

Install-Package

Microsoft.EntityFrameworkCore.Tools

Syncfusion???

3. Scaffold to data

- Scaffold-DbContext "Server=tcp:auge.database.windows.net,1433;Initial Catalog=appScripts;Persist Security Info=False;User ID=Juanjo;Password=#MadridAccess23;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" Microsoft.EntityFrameworkCore.SqlServer -ContextDir ApplicationDbContext -Context ApplicationDbContext -OutputDir Models

Microsoft

4. We move models

DbContext = ApplicationDbContext in Server

The models to Shared

5. Connection String in appSettings and program

- In appsettings.Development.json

```
 "ConnectionStrings": {
    "DefaultConnection": "Server=tcp:auge.database.windows.net,1433;Initial Catalog=appScripts;Persist Security Info=False;User ID=Juanjo;Password=#MadridAccess23;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"
  },
```

- In Program.cs

```
builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer("name=DefaultConnection"));
```

6. Create Rest/API → Controllers

- See Weather Forecast
- For each Entity:
    - Add Controller – API – API Controller with actions that use Entity Framework

Microsoft

7. Create Repositories (form functions)

HttpResponseWrapper.cs

Irepositorio.cs

Repositorio.cs

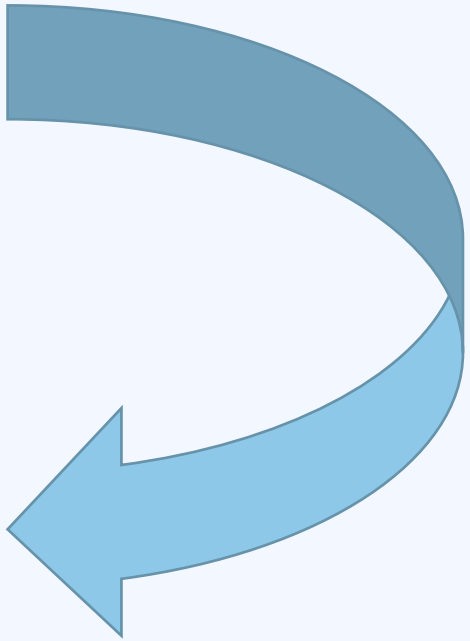
8. We declare the repositories in Program.cs

```
builder.Services.AddSingleton(sp => new HttpClient { BaseAddress = new Uri(builder.HostEnvironment.BaseAddress) });

ConfigureServices(builder.Services);

await builder.Build().RunAsync();

void ConfigureServices(IServiceCollection services)
{
    services.AddScoped<IRepositorio, Repositorio>();
}
```



Microsoft

9. WE TEST API

Ruta /api/Usuarios

Ruta /api/Usuarios/1

Use Postman or similar

10. Create Components→ Forms

- Pages – New component
- Reference to the desired entity---imports.razor
- Code
  - Dim

```
private List<Usuario> listaUsuarios { get; set; }
```

  - OnLoad

```
protected override async Task OnInitializedAsync()
{
    var respuestaHTTP = await repositorio.Get<List<Usuario>>($"api/Usuarios");
}
```

- Form Controls
  - Div and control name in html(use Bootstrap)

Microsoft

10.a. We create CRUD components

Create Usuario

Read Usuarios

Update Usuarios

Delete usuario

Microsoft

To allow includes in APIs

In Server.Program.cs

```
// To allow includes .Include(x => x.Usuario)

builder.Services.AddControllersWithViews()

        .AddJsonOptions(x => x.JsonSerializerOptions.ReferenceHandler = ReferenceHandler.IgnoreCycles);
```

Microsoft

Edit the Menu

```
<div class="nav-item px-3">
    <NavLink class="nav-link" href="fetchdata">
        <span class="oi oi-list-rich" aria-hidden="true"></span> Fetch data
    </NavLink>
</div>
```

Microsoft

11. Publish app

Azure may be free

Publish Server (non-Client)

Connect to the database

Microsoft

12. Use ready-made add-ons

Telerik

Syncfusion (free)

https://blazor.syncfusion.com/documentation/getting-started/blazor-webassembly-visual-studio

Microsoft

Thank you!!