

Tipo : Guía de Enunciado
Capítulo : ASP NET Core
Duración : 30 minutos

I. OBJETIVO

Agregando soporte a CORS.

II. REQUISITOS

Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Windows 10 (como mínimo Windows 8)
- Visual Studio 2017 como mínimo.

III. EJECUCIÓN DEL LABORATORIO

1. Abrir la solución “**Cibertec.NetCore.sln**” y ubicarse en el proyecto Cibertec.WebApi.

2. Instalar el siguiente paquete:

- Install-Package Microsoft.AspNetCore.Cors

3. Y configuramos en el “**Startup.cs**” lo siguiente:

(1) En el método “**ConfigureServices**” lo siguiente:

- services.AddCors();

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<IUnitOfWork>(option => new NorthwindUnitOfWork(Configuration.GetConnectionString("Northwind")));
    var tokenProvider = new RsaJwtTokenProvider("issuer", "audience", "token_cibertec_2017");
    services.AddSingleton<ITokenProvider>(tokenProvider);

    services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
        .AddJwtBearer(options => { });

    services.AddAuthorization(auth => { });
    services.AddCors();

    services.AddMvc();
}
```

(2) En el método “**Configure**” lo siguiente:

app.UseCors(builder =>

builder

.AllowAnyHeader()

.AllowAnyOrigin()

.AllowAnyMethod()

);

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseAuthentication();
    app.UseCors(builder =>
        builder
            .AllowAnyHeader()
            .AllowAnyOrigin()
            .AllowAnyMethod()
    );
    app.UseMvc();
}
```

4. Guardaos y compilamos.

IV. EVALUACIÓN

1. ¿Qué diferencia a .NET Core del .NET Framework?
2. ¿Por qué usar Asp Net Core?
3. ¿Cuáles son los beneficios de una Web Api con ASP NET Core?
4. ¿Dónde está el web.config?