**Clean Architecture Solution Template**

The purpose of this template is to provide a straightforward and efficient approach to developing enterprise applications by leveraging the power of Clean Architecture and ASP.NET Core. Using this template, you can easily create a single‑page application (SPA) with Blazor and ASP.NET Core while adhering to Clean Architecture principles. It’s easy to get started—just use this template or install the .NET template (see full details below).

**Technologies**

* ASP.NET Core 9
* Entity Framework Core 9
* ASP.NET Core Blazor (dotnet workload install wasm-tools-net6)
* MediatR
* AutoMapper
* FluentValidation
* NUnit, FluentAssertions, Moq & Respawn

**Getting Started**

The easiest way to begin is to install the .NET template and run:

bash

Copiar

dotnet new ca-sln

1. Install the latest .NET 9 SDK.
2. Run:

dotnet new install Clean.Architecture.Solution.Template

to install the template.

1. Run:

dotnet new ca-sln --output <NewSolutionName>

to scaffold a new solution.

1. Navigate into <NewSolutionName>/src/WebApplication and start the app with:

dotnet run

**Database**

By default, the template is configured to use an in‑memory database. This ensures everyone can run the solution without setting up additional infrastructure (e.g., SQL Server).

If you prefer SQL Server, update **WebApplication/appsettings.json** like this:

json

Copiar

"UseInMemoryDatabase": false

Then ensure the **DefaultConnection** string in appsettings.json points to a valid SQL Server instance.  
When you run the application, the database will be created automatically (if needed) and the latest migrations applied.

**Database Migrations**

To use dotnet-ef for your migrations:

1. Make sure "UseInMemoryDatabase" is set to false (see **Database** section).
2. From the repository root, add these flags to your EF command:

--project src/Infrastructure

(optional if you’re already in that folder)

--startup-project src/WebAplication

--output-dir Persistence/Migrations

For example, to add a new migration from the root:

dotnet ef migrations add "SampleMigration" \

--project src\Infrastructure \

--startup-project src\WebUI \

--output-dir Persistence\Migrations