**Clean architecture** is a software design philosophy that separates the elements of a design into ring levels.

An important goal of clean architecture is to provide developers with a way to organize code in such a way that it encapsulates the business logic but keeps it separate from the delivery mechanism.

The main rule of clean architecture is that code dependencies can only move from the outer levels inward.

Code on the inner layers can have no knowledge of functions on the outer layers. The variables, functions and classes (any entities) that exist in the outer layers cannot be mentioned in the more inward levels. It is recommended that data formats also stay separate between levels.

**Layers of Clean Architecture**

- Entities/domain: Entity layers consist of business logic which remains intact the entire time.

- Use Cases/ Application: This layer contains application-specific business rules and implements use cases for business logic

- Controller/Presentation/Adapters: In this layer, The software is a set of adapters that convert data from the format most convenient for the use cases and entities to the format most convenient for some external agency such as the Database or the Web

- Framework & Driver/Details: This layer is generally composed of frameworks tools such as the Database, the Web Framework, etc.

**Examples**

A project needs to be Modular, so that the project is divided into Modules for ease of work on and reuse in other projects and the separation between as a Module and another Examples are: CustomerModules, UserModules, and so on. It is necessary to work, that each Module contains its own DbContext, only the tables that belong to it, and it is necessary to work with the Bounded Context, and it will be as follows:

- App.Users.Core  
 Entities , Interfaces , DTOs  
- App.Users.Infrastructure  
 UserContext , UserService , AutoMapperProfiles, Startup  
- App.Customers.Core  
 Entities , Interfaces , DTOs  
- App.Customers.Infrastructure  
 CustContext , CustService,AutoMapperProfiles, Startup

Clean Code architecture is used when dealing with large project, In order to keep business logic intact and separating the files or classes into components that can change independently of other components