Clean architecture

## What is clean architecture

* Clean architecture is a set of design principles that divides software components/modules into ring layers. The main idea is that code dependencies are supposed to only go from the outer layers to the inner ones
* Clean architecture puts the business logic and application model at the center of the application. Instead of having business logic depend on data access or other infrastructure concerns
* Clean architecture layers:
  + **Domain**: contains enterprise logic and types that shared across many systems
  + **Application**: contains business logic and types that typically only be used within this system
  + **Infrastructure**: contains the implementation of interfaces defined in the Domain and Application layers
  + **Presentation**: contains the user interface and presentation logic
* Clean architecture rules:
  + Model all business rules and entities in the Core project
  + All dependencies flow toward the Core
  + Inner projects define interfaces, outer projects implement them
* Benefits:
  + **Independent of frameworks**: it does not require the existence of some tool or Framework
  + **Testable**: because **Core** has no dependencies on anything external
  + **Independent of UI**: logic is kept out of the UI so it is easy to change to another technology
  + **Independent of the database**: data-access concerns are cleanly separated
  + **Other benefits of layer architecture**

## Structure

* **Domain**: has no dependencies on anything external
  + entities
  + enums
  + domain exceptions
  + interfaces
  + specifications
* **Application**: dependent only on the Domain layer
  + services
  + dtos
* **Infrastructure**: represents the Infrastructure layer and contains classes for accessing external resources such as file systems, databases...
  + repositories
  + DbContext
  + Email implementations
  + Entities configuration
* **Web/API/Presentation**: depends on both the Application and Infrastructure layers (only to support dependency injection)
  + api endpoints / controllers
  + views
  + filters
  + middlewares