Conceptos clave I

- ports and adapters
- hexagonal architecture
- outside-in thinking
- primary port
- secondary port
- adapter
- test double (mocks arent stubs)
- three layers architecture

Conceptos clave II

- interfaz, polimorfismo
- modules: horizontals and verticals
- vertical slicing- TDD, Test first, testing
- inversión de control- branch by abstraction
- technical debt
- parallel change
- feature toggle
- walking skeleton / service template

Ports and adapters

Ports and adapters

The main idea of Ports & Adapters is to define the structure of an application so that it could be run by different kinds of clients (humans, tests cases, other applications,…), and it could be tested in isolation from external devices of the real world that the application depends on (databases, servers, other applications, …).

Hexagonal architecture

Hexagonal architecture

Es lo mismo que Ports and adapters. En hexagonal architecture no hay seis de nada.

Outside-in thinking

Outside-in thinking

Hacer lo contrario que DDD, que sería: Many developers focus on implementing the Domain Model before defining how it is going to be used by the external world.

Primary port

Primary port

Primary ports are the main API of the application. They are called by the primary adapters that form the user side of the application. Examples of primary ports are functions that allow you to change objects, attributes, and relations in the core logic.

Secondary port

Secondary port

Secondary ports are the interfaces for the secondary adapters. They are called by the core logic. An example of a secondary port is an interface to store single objects. This interface simply specifies that an object be created, retrieved, updated, and deleted. It tells you nothing about the way the object is stored.

Adapter

Adapter

An adapter is a bridge between the application and the service that is needed by the application. It fits a specific port.

Test double (mocks arent stubs)

Test double (mocks arent stubs)

Test Double is a generic term for any case where you replace a production object for testing purposes. There are various kinds of double that Gerard lists:

Three layers architecture

Three layers architecture

A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture.

Interfaz, polimorfismo

Interfaz, polimorfismo

explicación

Modules: horizontals and verticals

Modules: horizontals and verticals

explicación

Vertical slicing

Vertical slicing

explicación

TDD, Test first, testing

TDD, Test first, testing

explicación

Inversión de control- branch by abstraction

Inversión de control- branch by abstraction

explicación

Technical debt

Technical debt

explicación

Parallel change

Parallel change

explicación

Feature toggle

Feature toggle

explicación

Walking skeleton / service template

Walking skeleton / service template

explicación