Test Driven Development

BOOTCAMP

HELLO!

I am Eduardo Flores.

You can find me at eduardo-flores846 on Linkedin

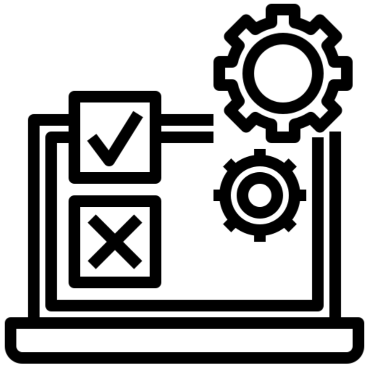
1.
TDD

Test Driven Development

You write the test first, for the code you what to write, then you write the code to make the test pass.

Your unit test cases drive the design.

keeps you mind focused on the purpose of the code.

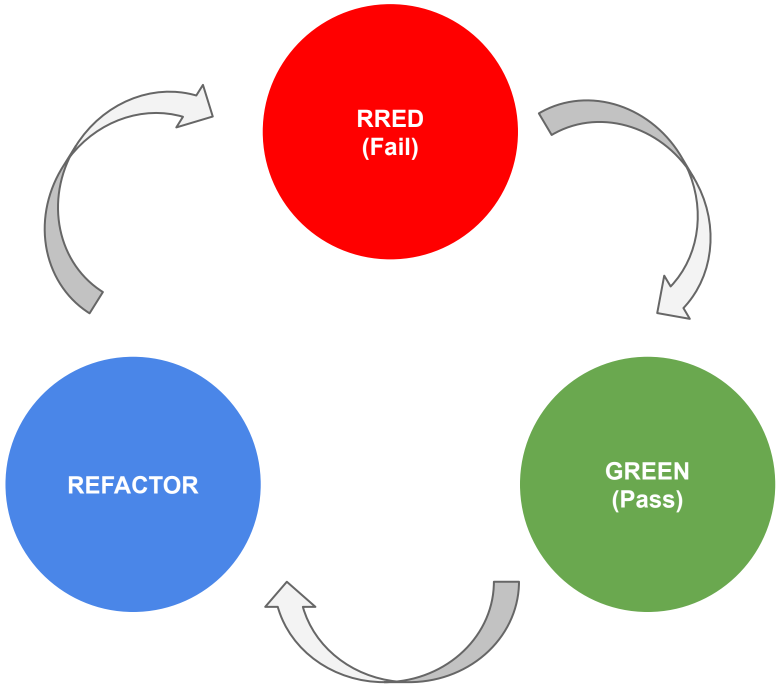


TDD workflow

Write the test case that fails

Write the code to make the test pass

Improve code quality



The importance of TDD

- It saves time when developing
- It allows you to code with more confidence
- It that your code works as expected
- It ensures you that future changes don;t break you code
- Creates a habit of refactoring
- Support clean and maintainable code
- Lays the foundation for automation

.

Automated testing

The practice of running test automatically, with data management, and improving the code quality.

Free developers from manual testing
Changes how we as developers write our tests

Testing frameworks
xUnit series
- JUnit for **Java**
- PyUnit for **Python**
- NUit for **.net**
- Embunit for C/C++

Jasmine for **JavaScript**
Jest for **node.js**
Simpletest for **PHP**

Happy paths and Sad paths

Happy paths verify that a functions returns positive outcomes when expected

Sad paths verify that a function responds to exceptions appropriately and without breaking

Test fixtures

Establish the initial state for our tests
Run test in isolation
Ensure repeatable results

Commons steps

- Preparation of the test data
- Setup/creation of fake or mock objects
- Loading the database with a predefined set of data
- Copying a specific set of files

Mocking

Mocking is the process of creating object that mimic the behaviour of real objects, functions etc.

When should you mock

- External services that are not under tests
- Need to change the behavior of a dependency
- Don't have a remote connection to a dependency
- Anytime you want to isolate your tests

Considerations when writing tests

- One assert per test
- Test should be fast
- Test has to run independently
- It can be repeated
- It should be self-validating

2.
Let's start testing!

Create the save order endpoint

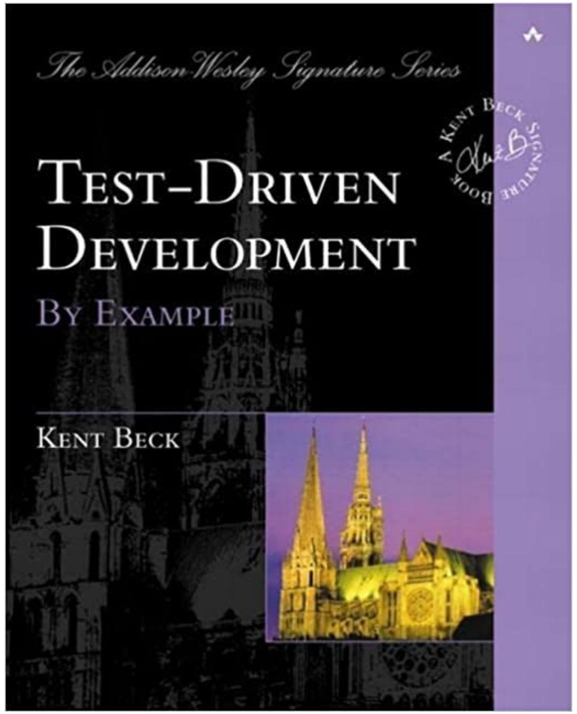
The endpoint should be a **POST** called **/orders**

It should calculate the **total** of the order
It should return the created **order**
It should return a status code **201**

It should return status code **500** when an error is catched

Resources

TDD,(Ian Cooper)
https://youtu.be/EZ05e7EMOLM


The Addison-Wesley Signature Series
A Kent Beck Signature Book
Test-Driven Development
By Example
Kent Beck