Exercícios

Todos os exercícios devem ser feitos usando a mesma plataforma, e preferencialmente no mesmo projeto.

Não há uma restrição geral da tecnologia ou linguagem esperada para a resolução desses exercícios, contudo todos devem ter testes automatizados. Ou seja, qualquer tecnologia escolhida para o treinamento, ela deverá contar com um ambiente e framework de testes instalado.

A exceção ficará por conta da equipe - se uma equipe trabalha com uma determinada tecnologia, sugere-se que a tecnologia usada para o treinamento seja a mesma da equipe.

Exercício 1 - Múltiplos de 3 ou 5

Dado todos os números naturais abaixo de 10;

Se selecionarmos aqueles que sejam múltiplos de 3 ou 5 temos: 3, 5, 6 e 9.

Se somarmos todos esses valores, teremos o resultado = 23

Desenvolva um sistema que responda às seguintes dúvidas:

Qual é o valor da soma de todos os números múltiplos de 3 ou 5 de números naturais abaixo de 1000?

Qual é o valor da soma de todos os números múltiplos de 3 e 5 de números naturais abaixo de 1000?

Qual é o valor da soma de todos os números múltiplos de (3 ou 5) e 7 de números naturais abaixo de 1000?

Definition of Done

Uma solução completamente testada, de preferência com testes para cada passo essencial para a checagem do resultado.

O mínimo esperado é que tenham 3 testes, uma para cada pergunta, **além** de outros testes validando os algoritmos usados para se chegar na solução final.

Exercício 2 - Números felizes

Os números felizes são definidos pelo seguinte procedimento:

Começando com qualquer número inteiro positivo, o número é substituído pela soma dos quadrados dos seus dígitos.

Repete-se esse processo até que o número seja igual a 1.

Tomamos o 7, que é um número feliz:

- $7^2 = 49$
- $4^2 + 9^2 = 97$
- $9^2 + 7^2 = 130$
- $1^2 + 3^2 + 0^2 = 10$
- $1^2 + 0^2 = 1$

Um número não é feliz quando, em seu processo de cálculo, em algum momento ele entra em loop, ou seja, ele passe por um número que ele já passou anteriormente (não é possível determinar um número específico que ele sempre irá passar).

Faça um programa que, dado um número natural qualquer, determine se é um número feliz.

Definition of Done

Um trecho de código onde é possível invocar um método checando se o número é feliz ou não, e o sistema consiga responder.

A solução final deve ser quebrada em diversas etapas, passos que precisam ser executados para se chegar na solução final, e cada uma dessas etapas devem estar cobertas com testes automatizados.

Sugestão: Fazer seguindo TDD.

Exercício 3 - Palavras em números

Neste problema, dado uma palavra composta somente por letras [a-zA-Z], cada letra possui um valor específico, 'a' vale 1, 'b' vale 2 e assim por diante, até a letra 'z' que vale 26. Do mesmo modo 'A' vale 27, 'B' vale 28, até a letra 'Z' que vale 52.

O valor da palavra será a soma total dos valores de todas as letras da palavra.

Você precisa definir se cada palavra em um conjunto de palavras é:

Prima ou não;

Feliz ou não;

Múltipla de 3 ou 5;

Qualquer caracter na palavra que não seja uma letra deve ser desconsiderado.

Definition of Done

Um sistema que, quando executado, transforme uma palavra em um número, seguindo a lógica acima, e responda às três questões: se é prima, feliz e múltipla de 3 ou 5.

Não há a necessidade de ter interação com o usuário para requisitar a palavra.

É esperado que as soluções anteriores sejam reusadas, e cada novo componente criado seja coberto com testes automatizados.

Exercício 4 - Cálculo de Frete

Em todos os e-commerces, o usuário pode criar um carrinho de compras, adicionar um produto e calcular o valor do frete para a entrega.

O valor do frete é calculado a partir do CEP do usuário (destinatário), e geralmente é provido pelos serviços de fretamento (correios ou particular), muitas vezes sendo invocado uma API que, dado o CEP (dentre outros dados), traz o valor do frete.

Desenvolva um sistema simplificado do carrinho, com os seguintes requisitos:

- Um produto possui um nome e um valor
- Um carrinho recebe um conjunto de produtos e a quantidade de itens de cada produto
- Um carrinho pertence à um usuário, que tem nome e seu endereço de entrega representado por um CEP
- Um serviço que recebe o carrinho, e retorna o valor final para o usuário

Esse serviço

- Faz a soma total de valores de todos os produtos do carrinho
- Caso o valor seja < $100,00, o sistema requisita para um serviço externo o valor do frete de acordo com o CEP do dono do carrinho
- Retorna o valor final do carrinho (com ou sem frete)

Sugestão de solução:

1. Crie a estrutura necessária para o Carrinho
    a. Usuário (nome e CEP)
    b. Produto (nome e Valor)
    c. Carrinho (Usuário e lista de Produtos)
2. Faça o carrinho responder o valor total das compras - com testes
    a. Pergunte-se:
        i. Qual o valor se ele estiver vazio?

        ii. E se eu adicionar novos produtos?

        iii. E se eu adicionar produtos que já tinham sido adicionados?

        iv. E como eu removo o produto do carrinho?

        v. E se eu adicionar dois produtos ao mesmo tempo?

        vi. E se eu adicionar ou remover a quantidade de produtos no carrinho?

        vii. E se eu zerar a quantidade de produtos do carrinho?

    b. Não há requisitos formais de como um carrinho deve funcionar, ou suas interfaces

    c. O carrinho **deve** ter um método final, que retorne o valor total do carrinho

3. Criem uma interface que representará o Serviço do Correios

    a. Ele terá apenas um método registrado - recebe o CEP e retorna um valor de Frete

    b. Não precisamos da implementação real (no momento, e para esse exercício)

4. Crie um classe que representará o serviço de cálculo

    a. Ela receberá em sua construção uma instância da interface do serviço de Correios (injeção de dependência)

    b. Ela terá um método que recebe um carrinho como parâmetro, e retorna o valor total

5. Crie os testes antes da implementação

    a. Qual o valor total, caso a soma total dos produtos do carrinho (feito anteriormente e com seus testes já funcionando) seja< $100?

        i. Eu preciso invocar o método real do Carrinho?

ii. Como eu simulo o retorno do serviço do Correios?

iii. Como eu garanto que a lógica decisória está correta?

iv. Como eu garanto que eu chamei apenas uma única vez o serviço do correio?

b. E se o valor for >= $100?

i. Qual será o valor final do cálculo?

ii. Como eu garanto que eu não precisei chamar os serviços do correio?

6. Crie os testes usando Mocks, mockando tanto o Carrinho quanto o Serviço de Correios

    a. A implementação do serviço que calcula valor total com ou sem frete deve ser concreta

Definition of Done

Todos os requisitos devem estar cobertos por testes automatizados.

Deve existir pelo menos uma classe de testes para o serviço, e esse deverá cobrir todas as variações das regras do serviço. Além do mais, a comunicação com o serviço do correio deverá ser através de mocks.