

**FIAP – MBA<sup>+</sup>**  
**DECENTRALIZED AND DISTRIBUTED DEVELOPMENT**  
**Desenvolvimento de *Smart Contracts* e Tokenização**

Prof. João Kuntz

Instruções: crie um contrato para cada exercício no Remix. Ao final, faça o download do *workspace* (opção Download Backup na página inicial) e anexe o arquivo zipado.

**Lista de Exercícios 01**

1 – Você está trabalhando em um contrato que realiza o acompanhamento da vida dos usuários. Ele possui uma função que realiza a classificação do ser humano, conforme sua idade (recebida por parâmetro), em três fases da sua vida: criança (se tiver menos do que 18 anos), adulto (se tiver 18 anos ou mais) e idoso (se tiver 60 anos ou mais).

Implemente as funções necessárias para criar uma base de dados contendo os usuários (identificados pelo seu nome), sua idade e classificação, e para realizar a consulta, quando um nome for enviado como um parâmetro. Por simplificação, assumimos que nenhum nome de usuário se repetirá.

2 – Você está trabalhando em um contrato que se propõe a realizar operações matemáticas simples. Este contrato possui um contador de interações que, por decisão de negócio, deve ser inicializado com o valor de 50.

Este contrato deve possuir uma funcionalidade capaz de consultar o valor da variável de estado, e outra função que é invocada para incrementar o valor do contador em 1 – independentemente do endereço de origem da invocação, o incremento deve refletir para todos os usuários.

Faça as implementações necessárias.

3 – Modifique o contrato anterior para que ele armazene os endereços de origem de invocação da função de adição. Ele deve ser capaz de mostrar quem foi (o

endereço) e quantas vezes a pessoa invocou a funcionalidade de incremento. Deve possuir também uma funcionalidade de consulta por um endereço (por exemplo, quero descobrir quantas vezes o endereço A já invocou a execução).

4 – Seu contrato é especializado em manipular arrays. Ele deve ser capaz de receber como parâmetro um array de números inteiros (tamanho variável), sem repetição, em tempo de compilação, e deve possuir funcionalidades para remoção de determinado valor.

Por exemplo, em um array inicial [1,2,3], se eu invocar a funcionalidade de remoção passando o número 2 como parâmetro, o array inicial deve virar [1,3].

Da mesma maneira, quero ter uma funcionalidade que inclua um valor ao final do array, para que ele cresça. Contudo, nenhum valor repetido é aceitável; caso isto ocorra, não devo ser capaz de incluir o número.

Faça as implementações necessárias.

5 – Você está trabalhando em um contrato que cadastra pessoas (identificadas pelos seus endereços) aos serviços oferecidos pela sua solução. Em um dado momento, os status do contrato podem ser: Em análise, Liberado, Cancelado e Rejeitado.

Crie as funções necessárias para implementar funcionalidades que relacionem endereços com os status possíveis, e para fazer suas consultas (posso querer saber qual o status do endereço A, por exemplo).

Contratos Rejeitados não podem ter seu status alterado em qualquer hipótese.

Contratos Liberados podem ser Cancelados, porém, não podem voltar para o status Em Análise.

Contratos Cancelados, por sua vez, só podem passar para o status Em Análise.

Um contrato pode ser Rejeitado em qualquer ponto.

Por segurança, só posso modificar o status de um endereço após pelo menos 30 segundos da última alteração.

6 – Crie um contrato capaz de controlar a matrícula de alunos (identificados por seu nome e idade) à determinada disciplina (para facilitar, podemos considerar o contrato como sendo a própria disciplina). O aluno só pode se matricular uma única vez à disciplina.

Apenas alunos com idade igual a 18 anos podem se matricular na disciplina. O nome do aluno é um parâmetro obrigatório, e podemos ter alunos com o mesmo nome.

Na prática, cada aluno é um endereço de origem, então um endereço A pode se matricular uma única vez, por exemplo.

O contrato deve possuir funções para a inclusão deste aluno, para consulta (verificar se determinado aluno está matriculado), para verificar quantos alunos estão matriculados no total e posso querer remover um aluno da lista de matrícula – porém, apenas o endereço que fez a matrícula é o endereço que pode fazer a remoção.

7 – Você está trabalhando em um contrato que será utilizado por uma operadora de telefonia celular, como parte do seu programa de fidelidade. Segundo o regulamento, um usuário (identificado por seu endereço) possui pontos, e estes pontos podem ser trocados por produtos. Quando um usuário se inscreve para participar do programa, ele automaticamente ganha 2 pontos como boas-vindas.

A troca de pontos por produtos segue a lógica:

- Um relógio pode ser trocado por 10 pontos
- Um celular pode ser trocado por 50 pontos
- Um computador pode ser trocado por 100 pontos

O contrato deve ter funções capazes de fazer o cadastro do usuário, consulta dos pontos disponíveis, inclusão de pontos para o usuário (recebidos por parâmetro) e de realização de troca de pontos por produtos.

Deve ser possível verificar todos os produtos que um usuário já trocou em algum momento. Deve ser possível também verificar se um usuário já possui ou não determinado produto.

Por último, quero poder consultar, na lista de produtos, os usuários que escolheram determinado produto para trocar trocado (por exemplo, se João troca pontos por um Computador, e Pedro troca pontos por outro Computador, quando

eu consultar o produto “Computador”, quero saber que ele foi trocado por João e por Pedro).

Faça todas as implementações necessárias.