



# Certified Tech Developer

The Ultimate Degree



## Programação Imperativa

### Avaliação Final



### Objetivos

Vamos realizar uma série de exercícios conectados entre si. O objetivo é trabalhar um pouco com exercícios que dependem da etapa anterior para manter uma linha de trabalho.

Você pode consultar o material assíncrono e a internet, **mas** tenha atenção quanto a projetos iguais, cópias e plágio. A avaliação é individual e cada um deverá fazer o seu envio para seu github.

**Importante:** Ao final, lembre-se de:

- seu projeto deve estar completo no Github
- seu projeto deve ter um arquivo README explicando como executar seu projeto
- seu projeto não deve conter erros (de nenhum tipo)
- o código deve atender o mínimo do enunciado
- o código pode ser feito de várias formas, não necessariamente utilizando *arrow functions* - a escolha das estruturas que você irá utilizar fica a seu critério: MAS não pode conter falhas



## Micro desafio - Passo 1

Crie um objeto **aluno** que tenha como atributos: `nome` (string), `quantidade de faltas` (number) e `notas` (array de números). Crie um construtor para ele e importe-o como o módulo `aluno`.



## Micro desafio - Passo 2

Nosso objeto **aluno** terá o método `calcularMedia` que retorna a média de suas notas. Também terá um método chamado `faltas`, que simplesmente aumenta o número de faltas em 1.



## Micro desafio - Passo 3

Em um arquivo diferente, crie o objeto literal `curso` que tem como atributos: `nome do curso` (string), `nota de aprovação` (number), `faltas máximas` (number) e uma `lista de estudantes` (um array composto pelos alunos criados no passo 1).



## Micro desafio - Passo 4

Crie o método que permite adicionar alunos à lista do curso, ou seja, quando chamamos nosso método em nosso objeto `curso`, deverá adicionar um aluno a mais na propriedade `lista de estudantes` do objeto `curso`.



## Micro desafio - Passo 5

Crie um método para o objeto `curso` que receba um aluno (como parâmetro) e retorne `true` se ele aprovou no curso ou `false` em caso de reprovação. Para ser aprovado, o aluno tem que ter uma média igual ou acima da `nota de aprovação` e ter menos faltas que `faltas máximas`. Se tiver a mesma quantidade, tem que estar 10% acima da `nota de aprovação`.



## Micro desafio - Passo 6

Crie um método para o objeto `curso` que percorra a `lista de estudantes` e retorne um array de booleanos com os resultados se os alunos aprovaram ou não.



## Micro desafio - Passo 7

Importe o módulo `estudantes.js`, que contém uma lista de alunos, dentro do arquivo que contém o objeto `curso`. Substitua o conteúdo da propriedade `lista de estudantes`, pela `lista de estudantes` do arquivo `estudantes.js` e garanta que sigam funcionando todos os métodos corretamente. (revisar que o arquivo tenha todos os métodos corretamente).