

## Laboratório 4 – TypeScript com Módulos, Arquivos e Funções Assíncronas

### 1 Exercícios

Utilize o ambiente Node.js para executar no console o código TypeScript destes exercícios. Você pode utilizar como base o projeto configurado no Laboratório 1.

1. Retome o exemplo dos objetos *Cofrinho* e *Moeda*. Realize as seguintes tarefas:
  - a) Crie um módulo em um arquivo chamado “entidades” para conter as definições das classes *Cofrinho* e *Moeda* e exporte suas definições.
  - b) Crie um módulo em um arquivo chamado “persistência” para conter o código de funções para ler e escrever o conteúdo do *Cofrinho* em arquivos texto serializados no formato JSON. Nomeie as funções como *salvarCofrinho(Cofrinho, nomeArquivo)* e *lerCofrinho(nomeArquivo)*. Utilize o mecanismo de tratamento de exceções e caso encontre uma falha na leitura ou escrita do arquivo, lance um novo tipo de exceção chamada *PersistenciaErro*.
  - c) Crie um arquivo chamado “index.ts” que cria um novo cofre, adiciona diversas moedas nele, salva um arquivo e depois lê o arquivo e informa no console o conteúdo do arquivo.
2. Se você utilizou métodos síncronos de acesso aos arquivos na questão 1, refaça o programa utilizando métodos assíncronos com *callback*.
3. Refaça a questão 2 fazendo uso do modelo de programação com *promises* via as palavras-chave *async/await*.
4. Acesse o serviço REST <https://reqres.in/> e escreva um pequeno programa que consuma o serviço. Mostre o acesso a URIs diferentes para as operações de GET/POST/PUT/PATCH/DELETE. Pelo menos uma das operações deve testar tanto o retorno com sucesso quanto com falha.