

# PROGRAMA STARTER FULL STACK WEB DEVELOPER

POO

Atividade Prática – Typescript

## Termo de uso

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal.

É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

## Objetivo do documento

Este material tem como objetivo descrever a atividade prática que realizaremos durante as aulas para fixação do conteúdo.

## Vamos praticar!

Chegou a hora de aplicar o conhecimento adquirido em nosso encontro. Os exercícios e desafios serão nossos principais indicadores sobre o conhecimento de vocês, tanto para ajudá-los como na hora do direcionamento para as vagas.

### Configuração de um projeto

Crie um projeto Node.js do zero e o configure para utilizar o Typescript, conforme foi feito em aula com o mentor. Faça a publicação deste projeto no seu GitHub e utilize para a resolução dos exercícios de fixação.

### Exercícios de fixação

Usando o projeto Node.js + Typescript configurado acima, realize os seguintes exercícios:

1. Crie uma função que receba 2 números e retorne um objeto contendo a média e também um indicador booleano de aprovado/reprovado. Considere aprovado com média  $\geq 6$ .
2. Crie uma função que receba uma lista de objetos contendo **nota** e **peso**, realize a média das notas considerando o peso. Exemplos:

Lista com 2 notas:  $(N1*P1) + (N2*P2) / 2 = Resultado$

Lista com 3 notas:  $(N1*P1) + (N2*P2) + (N3*P3) / 3 = Resultado$

3. Crie um programa que simule uma carteira de dinheiro. Este programa vai precisar ter uma carteira contendo saldo, transações de entrada e saídas. Ou seja, será um objeto com estas propriedades. Depois crie uma função para lançar uma entrada e uma saída. Caso ao lançar uma saída e não tiver saldo, precisa dar um erro ou avisar.
4. Crie um programa para cadastrar, listar e excluir produtos de uma lista com tipagem de Produto.
5. Crie um programa para mostrar informações de usuários (User) de uma empresa. Crie o tipo User com as seguintes propriedades: nome, idade, ocupação e salário (opcional). Caso o salário do usuário não seja informado, mostre o valor "N/A". Exemplo:

a. *"Daphne, 23 anos, analista de TI, salário R\$ 1000"*

b. *"Daphne, 23 anos, analista de TI, salário N/A"*

6. Usando o contexto do exercício 6, crie um tipo de usuário que representa funcionários da diretoria da empresa. O tipo Diretor deve conter as propriedades: nome, idade, salário (opcional) e nível de comissionamento (numérico). Crie uma função que receba um Diretor e mostre suas informações. Exemplos:

a. *"Diretor(a) Daphne, 23 anos, comissão nível 5, salário R\$ 1000"*

b. *"Diretor(a) Daphne, 23 anos, comissão nível 5, salário N/A"*

7. Crie um tipo que seja composto por um User OU por um Diretor usando interseção de tipos. Desenvolva uma função que receba uma lista desse novo tipo e, para cada item da lista, imprima:
- O mesmo que o exercício 5, em caso de objeto User.
  - O mesmo que o exercício 6, em caso de objeto Diretor.

## Regras

- Os exercícios acima são apenas TS sem front. Rodar no terminal.
- Deve ser utilizado type e interface para tipagem dos dados.
- Separar exercícios e tipagens em arquivos específicos.

---

**ATENÇÃO!** Este exercício será corrigido na próxima aula com o Tech Helper e deverá ser postado na Class até o horário estipulado da tarefa na plataforma.

Crie um arquivo compactado contendo os arquivos com a resolução da atividade e realize o upload no post da atividade no Class, ou suba no Github e poste o link para o respectivo repositório.

---

Para que possamos construir uma base sólida de aprendizado é preciso praticar. **Bora implementar tudo isso!**



# PROGRAMA STARTER FULL STACK WEB DEVELOPER

BORA GROWDEVERS!