**INSTITUTO FEDERAL DE EDUCAÇÃO,  
CIÊNCIA E TECNOLOGIA DE SÃO PAULO** **CAMPUS – São Paulo** **Exercício**

**LP3A5 - Linguagem de Programação III**

**Data:** 04/10/2023  
**Nome:**Raphael Capistrano **Prontuário:** SP3063968  
**Professor : Wendel**

**1. De acordo com os autores do artigo “Um Estudo sobre Diferentes Linguagens de Programação para Introdução da Programação Funcional", quais são as principais dificuldades de um estudante ao se deparar com um novo paradigma?**

**R:** Aprender uma nova sintaxe (nova linguagem), eles precisam compreender diversos conceitos novos, como: as características do paradigma, o modelo de organização do código, a estratégia de modelagem dos problemas, o fluxo de execução e estado da aplicação, etc. São muitos aspectos que podem confundir um aprendiz quando focando em um paradigma que não está familiarizado.

**2. Como os autores definem “efeito colateral”? Qual sua relação com operações “puras”?**

**R:** As linguagens que levam o conceito de pureza muito forte, separando completamente operações que causam efeito colateral (como operações de entrada e saída) das operações que não alteram estado nem possuem efeito colateral.

**3. Quais são os dois grandes grupos de paradigmas descritos no artigo? Apresente a descrição destes e identifique a qual deles as linguagens funcionais pertencem.**

**R: Programação funcional:**

Busca emular funções matemáticas e enfatiza a aplicação de funções, que são tratadas como valores importantes e que podem ser utilizadas em qualquer parte da aplicação. Linguagem: Haskell e Clojure.

**Programação Orientada à Objetos:**

Eleva o nível de abstração ao permitir a criação de entidades que encapsulam lógicas específicas e podem ser reutilizados em diferentes contextos. Linguagem: JavaScript.

**4. A que se deve o crescimento do interesse por linguagens funcionais nos mecanismos de busca?**

**R:** O crescimento do uso de conceitos funcionais em linguagens populares como Java e o crescimento de Javascript, fizeram com que a comunidade tivesse outros olhos para a programação funcional.

**5. Por que os autores escolheram Haskell, Clojure e Javascript como objeto de estudo no artigo?**

**R: Haskell** foi escolhida por ser a linguagem atualmente utilizada na disciplina Paradigmas de Linguagem de Programação da UFCG para introdução do paradigma funcional.

**Clojure** é uma linguagem baseada no dialeto de Lisp, uma das mais antigas linguagens funcionais.

**Javascript** foi escolhida por ser uma linguagem multiparadigma, amplamente utilizada no desenvolvimento web e na criação de microsserviços dada a sua sintaxe simples e poderosa.

**6. Quais foram os pontos positivos e negativos elencados pelos autores para a adoção das linguagens Haskell, Clojure e Javascript em um curso de programação?**

**R: Haskell** é uma das linguagens mais referenciadas no ensino de programação funcional.

**Clojure** tem crescido bastante nos últimos anos e o mercado vem adotando a linguagem, especialmente em instituições financeiras. O fato de ser executada sob a

JVM (Java Virtual Machine) garante interoperabilidade com classes feitas em Java e familiaridade do programador para com o ambiente.

**Javascript** possibilita o uso de função como valores de primeira classe. Javascript foi uma das grandes responsáveis pelo crescimento do interesse no paradigma funcional nos últimos anos, devido à implementação de funções anônimas, funções como valores de primeira ordem e sua facilidade por ser flexível o suficiente para permitir que programadores acostumados com linguagens procedurais pudessem se aventurar, quando confortáveis, em um pensamento mais declarativo.

**7. No Caso de Uso 3 os autores falam sobre as estruturas de repetição (“laços”). Quais são as diferenças entre as linguagens funcionais e as imperativas nesse  
sentido?**

**R:** Linguagens funcionais não costumam oferecer laços como as imperativas (for e while), sendo utilizada a recursão para esses casos. De maneira muito simples, apenas com o uso de argumentos. Casamento de padrões ou sobrecarga paramétrica não é uma opção nessa linguagem, dificultando a legibilidade. Além disso, o programador precisa ficar o tempo inteiro fixado em não cometer vícios de linguagem imperativa, como uso de laços, variáveis, estado.