| CENTRO DE CIÊNCIAS E TECNOLOGIA  DEPARTAMENTO DE COMPUTAÇÃO  **PARADIGMAS DE PROGRAMAÇÃO**  ATIVIDADE 3 |
| --- |

**1)** Utilizando **Prolog**...

**a)** Escreva um código com os seguintes fatos/regras:

∙ Alexsandro, Antônio, Arton, Britney, Bruno, Caio, Camila, Fernando, Gabriel, Ian, Joanderson, Kelvyn, Luana, Lucas, Luiz, Maiara, Marcelo, MateusViana, MateusSilva, Mikaelle, Natália, Thiago, Victor e Wilton cursam PP; ∙ Alexsandro, Antônio, Arton, Bruno, Caio, Fernando, Gabriel, Ian e Joanderson cursam SO; *[dados fictícios]* ∙ Britney, Camila, Luana, Maiara, Mikaelle e Natália cursam TEES; *[dados fictícios]*

∙ Kelvyn, Lucas, Luiz, Marcelo, MateusViana, MateusSilva, Thiago, Victor e Wilton cursam TEBD; *[dados fictícios]* ∙ PP e SO são disciplinas obrigatórias;

∙ TEES e TEBD são disciplinas eletivas;

∙ Janderson leciona PP (Paradigmas de Programação);

∙ Janderson leciona SO (Sistemas Operacionais);

∙ Sabrina leciona TEES (Tópicos Especiais em Engenharia de Software);

∙ Vladimir leciona TEBD (Tópicos Especiais em Banco de Dados);

∙ Fulano é professor de Sicrano uma vez que Sicrano cursa uma disciplina que Fulano leciona.

Com base nisso:

i. Liste todas as disciplinas obrigatórias. ii. Liste todos os alunos de Janderson.

iii. Liste todos os teus professores.

iv. Liste todos os professores de disciplinas eletivas.

DICA: procure sobre o predicado findall

**b)** Baseando-se nos slides 89 a 97, utilize Prolog, via Java, considerando que o arquivo teste.pl deve consistir dos fatos e regras da ***letra a***. O programa em Java deve permitir ao usuário digitar o nome de um aluno, listando, por conseguinte, seus professores (***letra a - item iii***). SUGESTÃO: Para pedir o nome, use JOptionPane.showInputDialog("Nome?").

OBS. 1: Trocar ancestral(X, jose) pelo que foi solicitado!

OBS. 2: Não esqueça de adicionar o *jpl.jar* ao *Java Project*.

OBS. 3: Em vez de import jpl.\*, utilize: import org.jpl7.\*.

OBS. 4: Em vez de q1.query(), utilize: q1.hasSolution().

OBS. 5: Em vez de Hashtable[], utilize: Map<String, Term>[].

OBS. 6: Ao se deparar com esta possível exceção java.lang.UnsatisfiedLinkError: no jpl in java.library.path, consulte o link a seguir:

https://stackoverflow.com/questions/12283471/jpl-swi-prolog-configuration-failure

(Veja a resposta: “Try adding your path to java.library.path via Run > Run Configuration > [...]”.)

**2)** Utilizando **Haskell**... 1

**a)** Baseando-se no slide 17, elabore um algoritmo em Haskell para indicar, a partir da frequência (porcentagem de faltas) e das notas das duas unidades de um determinado aluno, se este está ‘Reprovado por falta’, ‘Reprovado por nota’, ‘Aprovado por média’ ou ‘Na final’.

**b)** Procure como seria a implementação do algoritmo *Quicksort* em Haskell, entenda o código e modifique-o para ordenar uma lista, de maneira decrescente, considerando apenas os números pares. Teste-o “ordenando” a seguinte lista: [9,1,8,2,5,7,3,6,4]. Resultado esperado: [8,6,4,2]

1 Sugestão: usar um compilador online, como, por exemplo, tutorialspoint.com/compile\_haskell\_online.php ou rextester.com/l/haskell\_online\_compiler.

Paradigmas de Programação – 2022.1 – Prof. Janderson Aguiar Curso de Ciência da Computação – CCT/UEPB

**3)** Considere Python uma linguagem de programação multiparadigma (incluindo características da Programação Funcional). Mostre duas formas de se implementar o algoritmo *Quicksort* em Python: de maneira imperativa (como tradicionalmente se aborda em uma disciplina de Estrutura de Dados) e de maneira funcional (similarmente ao que você procurou para a questão ***2b***, podendo ser codificado em uma linha!). Use a mesma sequência de elementos da questão ***2b*** para ilustrar a execução de ambas

as formas. Comente sobre os códigos (em relação àqueles critérios comentados no início da disciplina: *readability* e *writability*).

**4)** Comente brevemente sobre...

**a)** Programação Orientada a Aspectos (POA) e a linguagem AspectJ.

**b)** Programação Concorrente (baseando-se no capítulo 17 do livro do Tucker (2010), e/ou em outra referência relevante).

**c)** Teoria de domínios (relacionado ao conteúdo sobre Semântica Denotacional).

Paradigmas de Programação – 2022.1 – Prof. Janderson Aguiar Curso de Ciência da Computação – CCT/UEPB