

**3º Trabalho de Paradigmas de Linguagem de Programação**  
**Universidade Federal do Amazonas**  
**Instituto de Computação - ICOMP**

**Alunos(as):** Brenda Aguiar Oliveira, Elizamara Karina Almeida.

**1) Entre as características associadas com linguagens lógicas, podemos citar a programação baseada em motores de inferência, a unificação, a representação via cláusulas lógicas e a falsidade como falha. O que você entende por cada uma delas?**

**R:**

<b><i>Motores de Inferência:</i></b>	O prolog tem base de dados e os predicados. Quando chama esses predicados, ele busca na base de dados que o usuário forneceu e faz inferências que é extraída das informações implícitas.
<b><i>Unificação:</i></b>	A unificação é justamente o casamento de padrão, pois ele busca aquela condição dentro do banco de dados fornecida pelo usuário.
<b><i>Cláusulas Lógicas:</i></b>	São os fatos e regras definidas por Lógicas de Primeiro Ordem, “dois pontos e hífen” ( :- ) = se, a “vírgula “(,) é e, barra e igual (\=) significa diferença, com as cláusulas podemos formular os predicados para o pensamento lógico dentro do algoritmo. Por exemplo: %maria gosta de joao se joao for rico, isso em prolog ficaria: condicoes(joao,rico). gostar(maria, X) :- condicoes(X, rico).
<b><i>Falsidade como falha:</i></b>	Usa-se o falso para avaliar regras naquele mundo, se está dentro do nosso banco de dados, verifica se ele não é um mundo exterior e retorna true.

**2) Por que o uso de predicados assert e retract é associado à programação com efeitos colaterais?**

**R:** Porque na definição de efeitos colaterais, uma função ou variável durante a execução do programa pode modificar alguma variável de estado, que é o caso do assert e do retract que alteram o programa em execução por adicionar e remover os dados no banco de dados.

**5) Baseado em sua experiência com Prolog, a compare com Python OO, considerando os seguintes critérios:**

**a. Simplicidade, Ortogonalidade, Tipos de Dados, Projeto de Sintaxe, Suporte à Abstração, Expressividade, Checagem de Tipos, Manipulação de Exceções e Restrição de Aliases.**

<p><b>Simplicidade:</b> “Linguagens com menos operadores e verbosidade tendem a ser mais fácil de ler e entender.”</p>	<p><b>Python OO:</b> No começo tem um estranhamento por parte do programador iniciante, porém se torna mais simples e bem intuitivo de escrever, pois ele foi justamente projetado para ser simples.</p> <p><b>Prolog:</b> É estranho no começo, pois seu objetivo é a lógica de primeira ordem, por isso seus operadores são baseados em predicados. Dessa forma Prolog é fácil de aprender pois não possui muitos componentes básicos e também possui pouca multiplicidade de recursos.</p>
<p><b>Ortogonalidade :</b> “A falta de ortogonalidade em uma linguagem leva a exceções às regras da linguagem.”</p>	<p><b>Python OO:</b> É ortogonal que por usa classes e heranças, onde classes herdaram os atributos de outras classes.</p> <p><b>Prolog:</b> Vendo a pequena quantidade de componentes básico da linguagem e que os operadores se aplicam a eles a linguagem é ortogonal.</p>
<p><b>Tipos de Dados :</b> “A presença de mecanismos adequados para definir tipos e estruturas de dados, isso é significativo a legibilidade.”</p>	<p><b>Python OO:</b> É mais criterioso em relação a declaração de atributos, como no caso do “self” dentro das classes.</p> <p><b>Prolog:</b> Ela não tem suporte a tipos de dados porque é uma linguagem declarativa onde considera os argumentos e variáveis como símbolos e não como “ int float “ e estruturas de dados presente em POO .</p>
<p><b>Projeto de Sintaxe:</b> “Conceito bom para uma boa escrita são indentação, blocos, como se escreve certas coisas tipo os ifs.”</p>	<p><b>Python OO:</b> Força o programador a indentar o código para uma boa legibilidade.</p> <p><b>Prolog:</b> Não tem escrita em blocos em seus códigos, por ser uma Linguagem Lógica tenta se aproximar mais da linguagem humana, tendo com fechamento dos predicados o ponto(.).</p>

<p><b>Expressividade:</b> “Formas convenientes de especificar computações, onde uma expressão representa muitas computações.”</p>	<p><b>Python OO:</b> É expressivo por conta da sua simplicidade, não robusto como a linguagem C, por exemplo.</p> <p><b>Prolog:</b> É bastante expressivo por conta da sua lógica e escrita, tendo um conceito de mundo fechado dentro do código.</p>
<p><b>Suporte a Abstração:</b> “Abstração: capacidade de definir e, depois usar estruturas ou operações complicadas de uma maneira que seja possível ignorar muito do detalhes exemplo disso é o uso de funções de bibliotecas.”</p>	<p><b>Python OO:</b> Sua declaração de classes é intuitiva e fácil de escrever e entender, com isso a linguagem tem um suporte de abstração.</p> <p><b>Prolog:</b> Com o conceito de mundo fechado, o prolog tem capacidade de abstração quando declarado um predicado que usa uma cláusula por exemplo, porém por ter casamento de padrão fica complicado do programa entender as vezes o que o programador quer se ele não for especificado ao máximo.</p>
<p><b>Checagem de tipos:</b> “ É a fase da compilação responsável por verificar se o tipo de uma expressão “casa” com o esperado em seu contexto.”</p>	<p><b>Python OO:</b> Python tem declaração de variável implícita portanto se uma variável for criada ela pode ser mudada durante o programa. Por exemplo, se for criada uma variável x que recebe um inteiro, e se, no meio do programa ela receber um caracter, o python permite isso.</p> <p><b>Prolog:</b> Tem casamento de padrão, portanto se um predicado tem os seguintes argumentos durante todo o código ele precisa ter os mesmos argumentos quando passados no predicado.</p>
<p><b>Manipulação de Exceções:</b> “A capacidade de identificar os erros durante a execução de um programa. Também proporcionando a possibilidade de corrigir o problema e continuar a sua execução”</p>	<p><b>Python OO:</b> Como python varre o código todo antes de executar e consegue identificar erros antes da execução, na maioria das vezes.</p> <p><b>Prolog:</b> Na maioria das vezes não tem capacidade de resolver um problema quando encontrado, outra dificuldade que o prolog tem é justamente o programador tentar identificar o erro que o prolog está indicando.</p>

<p><b>Restrição de Aliases:</b> “Existência de métodos diferentes de acesso a uma mesma célula de memória. Mais de um apontador para a mesma variável.”</p>	<p><b>Python OO:</b> O python OO tem a capacidade de fazer referências usando implicitamente ponteiros.</p> <p><b>Prolog:</b> Prolog é uma linguagem pode se usar várias cláusulas com objetos diferentes. Por exemplo, podemos declarar uma família(pai), família(mãe), família(filho).</p>
---	--

**b. Legibilidade, escrita e confiabilidade.**

<p><b>Legibilidade:</b> “Facilidade de ler e entender a linguagem.”</p>	<p><b>Python OO:</b> É fácil de ler e entender.</p> <p><b>Prolog:</b> É fácil de ler e entender por se aproximar mais da linguagem humana.</p>
<p><b>Escrita:</b> “A capacidade de escrita é a medida da facilidade em que uma linguagem pode ser usada para criar programas para um domínio de problema escolhida”</p>	<p><b>Python OO:</b> O python tem a necessidade de ser um programa bem escrito.</p> <p><b>Prolog:</b> Precisa ser bem escrito para o entendendimento do código.</p>
<p><b>Confiabilidade:</b> “Um programa é confiável se ele se comportar de acordo com suas especificações sob todas as condições.”</p>	<p><b>Python OO:</b> O python sabe trabalhar bem com problemas, por isso ele tem alta taxa de confiabilidade.</p> <p><b>Prolog:</b> O prolog tem alguns problemas em relação a confiabilidade por exemplo os efeitos colaterais que ele tem durante a execução do código.</p>