Faculdade Boa Viagem

Prolog x Java

Paradigmas de Linguagem de Programação

Cleison Amorim e Rogério Peixoto
17/06/2013

Sumário

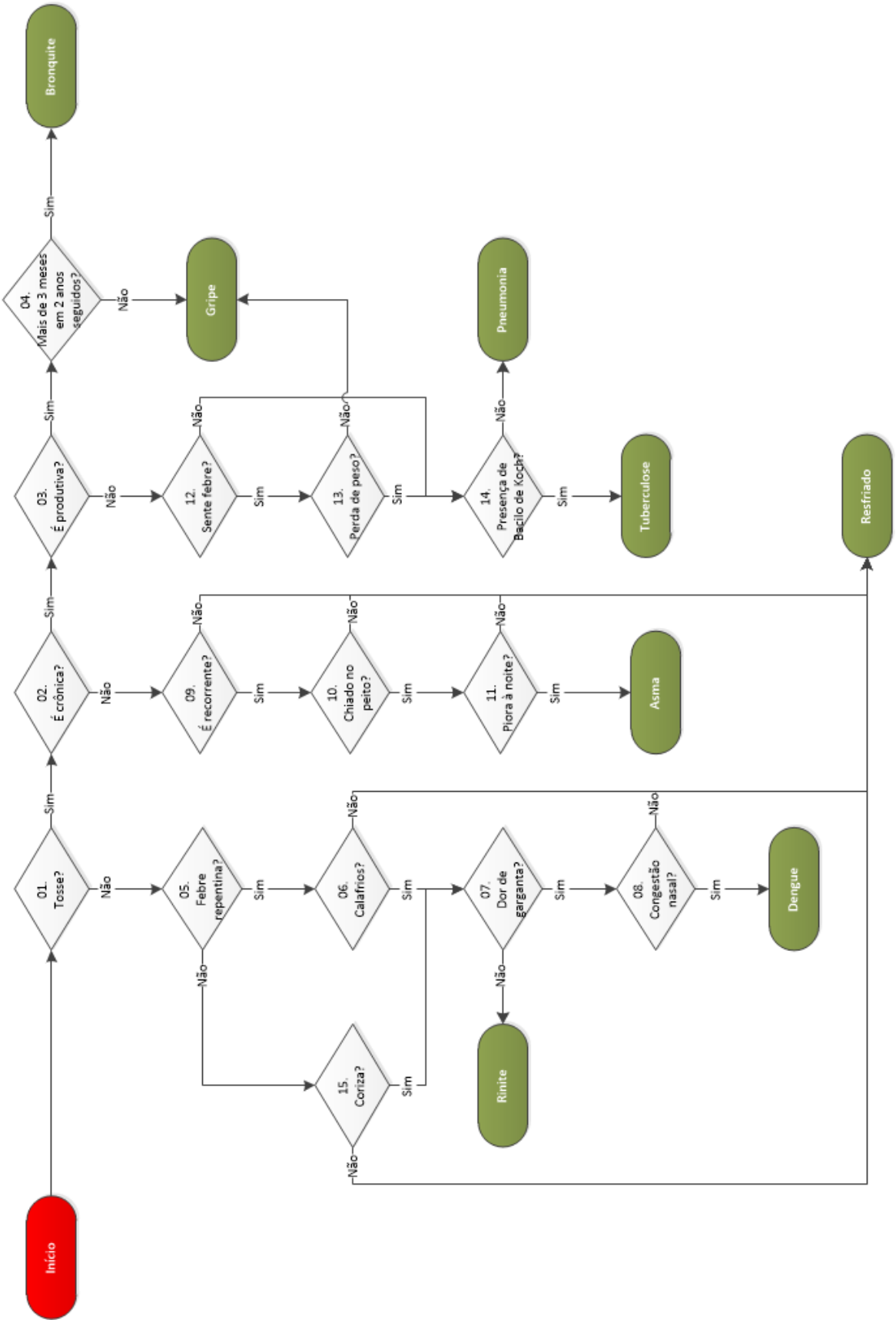
1. INTRODUÇÃO

O programa implementado refere-se a um simples sistema para diagnóstico de um pequeno número de doenças respiratórias, para o foi criado o fluxograma da seção seguinte, para representar seu comportamento e decisões até obtenção do diagnóstico.

Um diagnóstico médico é um processo analítico o qual o profissional médico se vê diante de diversos parâmetros que influenciam na determinação de um quadro clínico. A extração destes parâmetros é por muitas vezes feita e/ou mapeada através de perguntas e respostas e análise de exames durante a consulta.

O processo em que uma conclusão é inferida através de múltiplas observações é chamado processo dedutivo ou indutivo. Para computar tal análise e criar um autômato que leve de seu estado inicial ao diagnóstico válido, é necessária a implementação da inferência lógica, isto é, o ato de derivar conclusões através de premissas conhecidas. Esta que será usada para constituir uma árvore de decisão, mapeando todas as combinações de parâmetros e seus possíveis resultados. Para tal implementação é possível a utilização de diversos paradigmas de linguagem de programação, apesar de tal problemática estar mais propensa a ser implementada utilizando o paradigma lógico, neste documento será analisada e descrita as diferenças, vantagens e desvantagens entre a implementação da solução utilizando o paradigma lógico e orientado a objeto.

2. FLUXOGRAMA



3. PARADIGMA LÓGICO

3.1. INTRODUÇÃO

A linguagem Prolog foi escolhida para o desenvolvimento da solução no paradigma lógico, pois é capaz de representar de maneira satisfatória o uso do paradigma.

3.2. PROLOG

O Prolog (ou ***Pro****gramation em* ***Log****ique*) é uma linguagem de programação que se enquadra no paradigma de Programação em Lógica Matemática, e utiliza como gramática a Lógica de Predicados de Primeira Ordem. É uma linguagem declarativa, ou seja, limita-se a fornecer uma descrição do problema que se pretende computar e faz o uso de uma coleção base de dados de fatos e de relações lógicas (ou regras), que exprimem o domínio relacional do problema a resolver. Comumente é utilizado de um modo interativo, a partir de consultas (*queries*) formuladas pelo usuário sob os fatos e regras relacionais declarados.

Prolog nasceu a partir de um projeto na Universidade de Marselha que não tinha por foco a implementação de uma linguagem de programação, mas o processamento de linguagens naturais. Em 1971, desenvolveu-se uma versão preliminar de linguagem, mas apenas em 1972 é que surge sua versão definitiva por Alain Colmerauer e Philippe Roussel. Hoje, seu uso é especialmente associado às áreas de inteligência artificial e linguística computacional.

3.3. CARACTERÍSTICAS

Prolog consiste numa linguagem puramente lógica (ou Prolog puro) adicionado de componentes "extra-lógicos" que possibilitam funcionalidades como leitura (*read()*) e escrita (*write()*), entre outras.

Não há tipos de dados como nas linguagens de programação mais convencionais o fazem. Todos os dados são tratados como sendo um único tipo – **Termo**, cuja natureza depende da forma como este foi declarado. Ou seja, os elementos léxicos utilizados na sua declaração é que definem se o termo será um número, texto, variável, termo composto, etc.

A seguir, alguns conceitos importantes para a compreensão da linguagem:

- Átomos – São palavras, de um modo geral, e não possuem um significado específico. Constantes são declaradas como átomos. É uma sequencia constituída de letras, números ou "_" iniciados por letra minúscula ou conjunto de sequencias limitadas por aspas simples. Ex.: *blue, blue_atom, x, 'Taco', 'i am an atom'*.

- Números – Sequencia de dígitos, que permite números reais, negativos e notação negativa.

- Variáveis – São declaradas de maneira semelhante aos átomos, porém iniciados com letra maiúscula ou "_". Uma variável é como uma incógnita, cujo valor é desconhecido a princípio, mas após descoberto não sofre mais mudanças. Prolog também permite o uso de variáveis anônimas (uma "variável qualquer"), representadas como um único subtraço "_".

- Termos compostos – É a composição de um átomo (functor) e uma lista de argumentos (que define a aridade do termo). São utilizados para expressar estruturas de dados complexas em Prolog. Ex.: *atom('arg1', 234), 'My_atom'(2, 3, 4)*.

- Aridade – Número de argumentos de um termo composto.

- Strings – Sequencia de caracteres indicados entre aspas.

- Regras – São as cláusulas, ou um subset da lógica de predicados de primeira ordem. Ex.: *luz(acesa) :- interruptor(ligado).* (é uma representação para o predicado de que se uma luz está acesa, logo o interruptor está ligado *(P -> Q)*).

- Fatos – São a representação dos predicados (ou fatos do mundo real que o programa deve conhecer), informados à base de dados do programa para que as inferências possam ser realizadas. Ex.: *cat(tom) :- true.* (tom é um gato).

Uma busca em Prolog segue o padrão de busca em profundidade (*depth-first search*), ou seja, a árvore é percorrida sistematicamente de cima para baixo e da esquerda para a direita. Quando esta pesquisa falha, ou é encontrado um nó terminal da árvore, entra em funcionamento o mecanismo de *backtracking*, que faz com que o sistema retorne pelo mesmo caminho percorrido com a finalizada de encontrar soluções alternativas.

3.4. ESTUDO DE CASO

Um sistema de diagnósticos é de forma geral um problema adequado à utilização do *paradigma lógico*, uma vez que seu objetivo principal é simplesmente obter uma conclusão a partir de decisões (ou respostas) fornecidas pelo usuário. Para isto, adota-se um conjunto de predicados conhecidos como verdadeiros ao início da execução do programa, e que contêm os relacionamentos lógicos entre outros possíveis sintomas (predicados) e conduzem a um diagnóstico final. Para representar este paradigma neste projeto, foi utilizada a linguagem *Prolog (SWI)*.

No código Prolog anexo, os fatos foram modelados e adicionados à base de dados segundo o algoritmo sugerido pelo fluxograma. Durante a execução, cada resposta indicada pelo usuário adiciona proposições verdadeiras ou falsas, que por sua vez são pré-condições para outras novas proposições (ou fatos) possam também ser consideradas como verdadeiras. Esta cadeia de dependência lógica entre as proposições é o que permite que ao Prolog caminhar entre os fatos declarados e chegar a um diagnóstico.

3.4.1.VANTAGENS

- A natureza do problema é adequada ao paradigma da linguagem, não havendo necessidade de implementação de estruturas de repetição e controle;
- A sintaxe simples possibilita o uso da linguagem por pessoas com pouco conhecimento em programação;
- O código final é pouco complexo em comparação com outras linguagens de programação;
- O mecanismo de execução e inferência é implícito, expondo apenas uma interface lógica simples para construção do programa;

3.4.2.DESVANTAGENS

- A linguagem é voltada a problemas específicos e sua utilização ainda é limitada aos meios acadêmicos e de comprovação teórica;
- O código é monolítico, e não pode ser modularizado para acomodar de maneira mais legível algoritmos mais extensos;
- As entradas do usuário no sistema devem ser feitas por meio de regras de inferência lógica, o que não é muito intuitivo;

4. PARADIGMA ORIENTADO A OBJETOS

4.1. INTRODUÇÃO

Java foi escolhida por ser uma linguagem orientada a objetos comumente utilizada no mercado de trabalho atual.

4.2. JAVA

Java é uma linguagem de programação desenvolvida na década de 90. Diferentemente de outras linguagens que são compiladas em código nativo, Java é compilada para um *bytecode* que é executado em uma máquina-virtual. A linguagem de programação Java é a linguagem convencional da plataforma, porém não é sua única linguagem.

4.3. CARACTERÍSTICAS

Java implementa o paradigma orientado a objetos. Java é uma linguagem multiplataforma, foi projetada sob o lema *"write once, run anywhere"* isto é, escreva uma vez execute em qualquer lugar (o que na prática nem sempre é assim). As versões mais recentes de Java possuem extensa API nativa com bibliotecas de segurança, rede, interface gráfica e suporte a programação distribuída assim como sistemas multitarefas. Outro diferencial de outras linguagens é que Java possui alocação e desalocação dinâmica de memória, através do processo de *garbage collector* (coletor de lixo).

4.4. ESTUDO DE CASO

O mesmo sistema foi também implementado sob o paradigma orientado a objetos, utilizando a linguagem Java. Neste caso, como o paradigma é mais genérico com relação aos tipos de problemas em que pode ser aplicado, pode-se observar de antemão um programa mais extenso em comparação ao mesmo em Prolog, visto que agora mais unidades são necessárias para representar a modelagem das entidades em objetos. Optou-se por adotar um algoritmo de árvore binária (devido a sua semelhança com o paradigma lógico em relacionar sintomas) em que cada nó representa uma pergunta (ou possível sintoma) e está ligado a dois outros nós, que representam respectivamente o caminho a ser percorrido no caso de uma resposta afirmativa e outro no caso de uma resposta negativa.

4.4.1.VANTAGENS

- Modularização do código.

- Melhor leitura do código, Java promove o uso de identificadores de variáveis e métodos inteligíveis.
- Alta manutenibilidade de código. Como o código é bem modularizado e legível, é mais fácil dar manutenção de *bugs*.

4.4.2.DESVANTAGENS

- Maior esforço de implementação. É necessária a implementação de uma estrutura de dados para representar a árvore de decisão que define a problemática da inferência no contexto.
- Maior esforço de configuração de ambiente. Java requer a instalação de algumas ferramentas para criação de uma aplicação, a configuração de um ambiente de trabalho se feita a mão não é um processo muito produtivo, porém existem IDE's que eliminam este problema.