

Aplicação da mineração de dados para identificar padrões nos tratamentos clínicos

Aluno(a): Alan Michel Leicht

Orientador: Francisco Adell Péricas

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento
- Resultados e discussões
- Conclusões e sugestões

Introdução

- Auxiliar os profissionais da saúde;
- Indicar sugestões no processo de identificação do diagnóstico do paciente;
- Diminuir a incerteza na tomada de decisão clínica.

Objetivos

O objetivo deste trabalho é desenvolver uma ferramenta para detectar padrões em uma base de dados de tratamentos clínicos.

- Implementar o algoritmo C4.5 para realizar a mineração de dados;
- Demonstrar a consulta de predição sobre árvores de decisão;
- Disponibilizar uma interface gráfica com recursos visuais e interativos para análise clínica.

Fundamentação Teórica

- Informática na medicina
- Sistemas de apoio à decisão em medicina
 - Registros clínicos;
 - Decisão clínica.

Fundamentação Teórica

- Mineração de dados
 - Árvores de decisão: é uma estrutura que pode ser utilizada para, por meio de uma simples regra de decisão, dividir sucessivamente uma grande coleção de registros em conjuntos menores.
 - Indução na árvore de decisão: processo de classificação dos dados e construção da árvore de decisão.

Trabalhos Correlatos

- InteliMED
 - Dispositivos móveis;
 - Plataforma Andoid;
 - Diagnóstico de doenças respiratórias;
 - Árvore de decisão;
 - Questionário de evidências para determinar o diagnóstico.

Trabalhos Correlatos

- SADM-DST Desktop
 - Aplicação desktop;
 - Diagnóstico de doenças sexualmente transmissíveis;
 - Árvore de decisão;

Desenvolvimento

- Requisitos (funcionais e não funcionais)
- Especificação
- Implementação
- Operacionalidade da implementação

Requisitos

Requisitos funcionais	Caso de uso
RF01: permitir o cadastro de conexão da base de dados com informações clínicas	UC01
RF02: permitir o cadastro das configurações da árvore	UC02
RF03: permitir o cadastro das configurações dos atributos da árvore	UC03
RF04: permitir executar mineração de dados	UC04
RF05: permitir o cadastro das configurações de usuário	UC05

Requisitos

Requisitos funcionais	Caso de uso
RF06: permitir executar avaliação clínica	UC06
RF07: permitir visualizar árvore com diagnóstico geral	UC07
RF08: permitir visualizar árvore com o diagnóstico da avaliação clínica	UC08

Requisitos

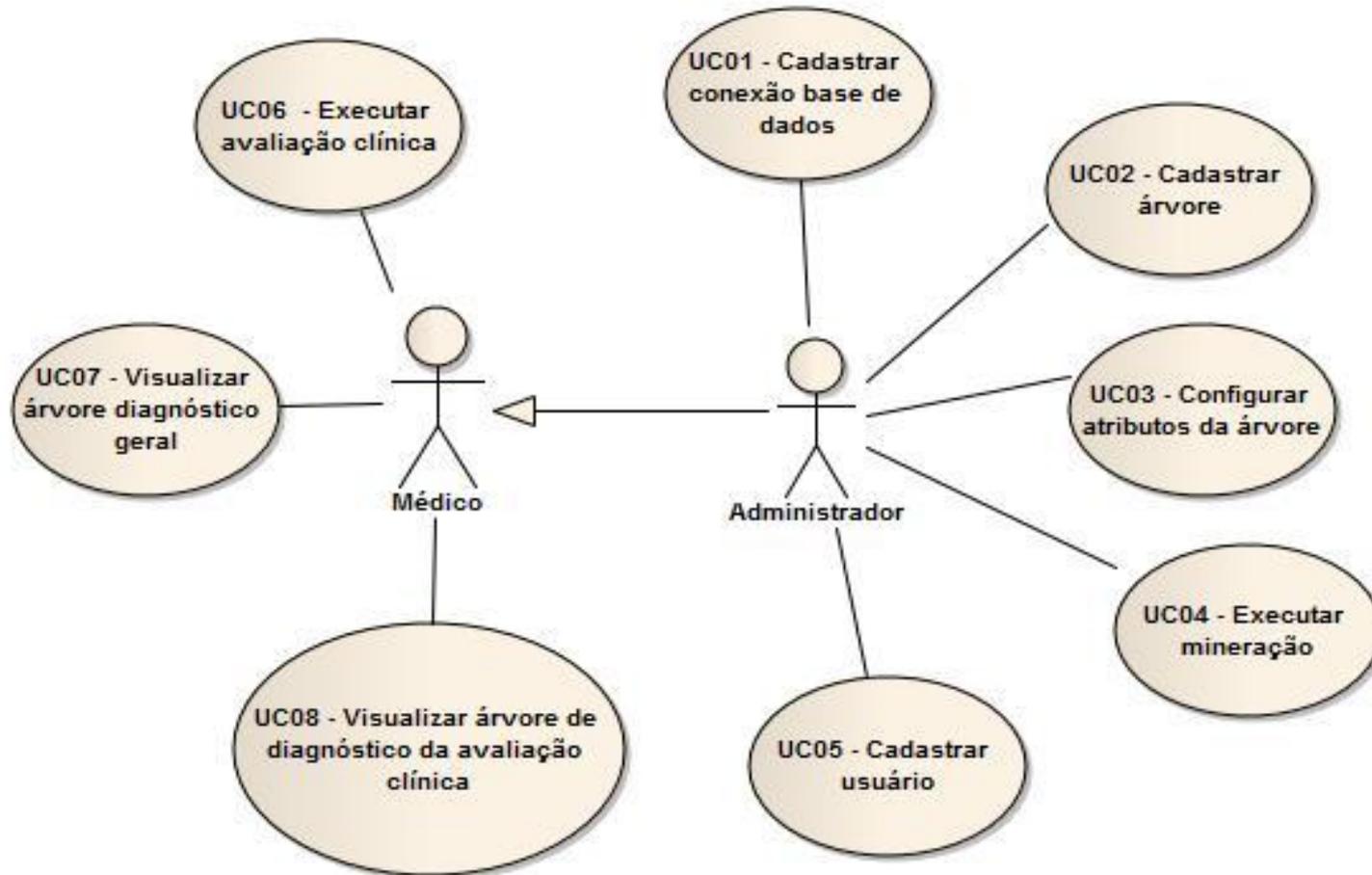
Requisitos não funcionais

RNF01: utilizar o ambiente de desenvolvimento Netbeans e a linguagem Java

RNF02: utilizar o banco de dados MySQL

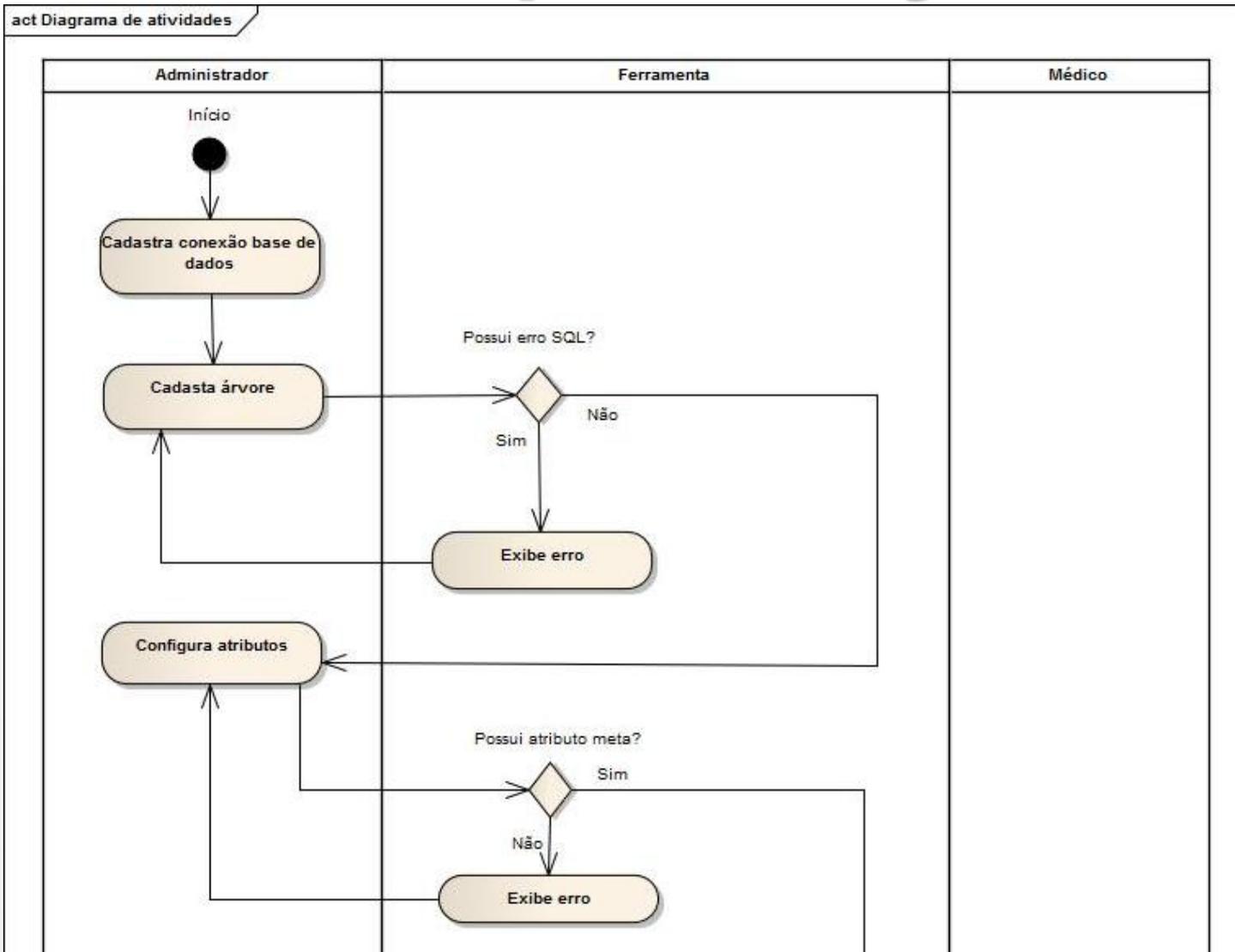
Especificação

uc Modelo de casos de uso

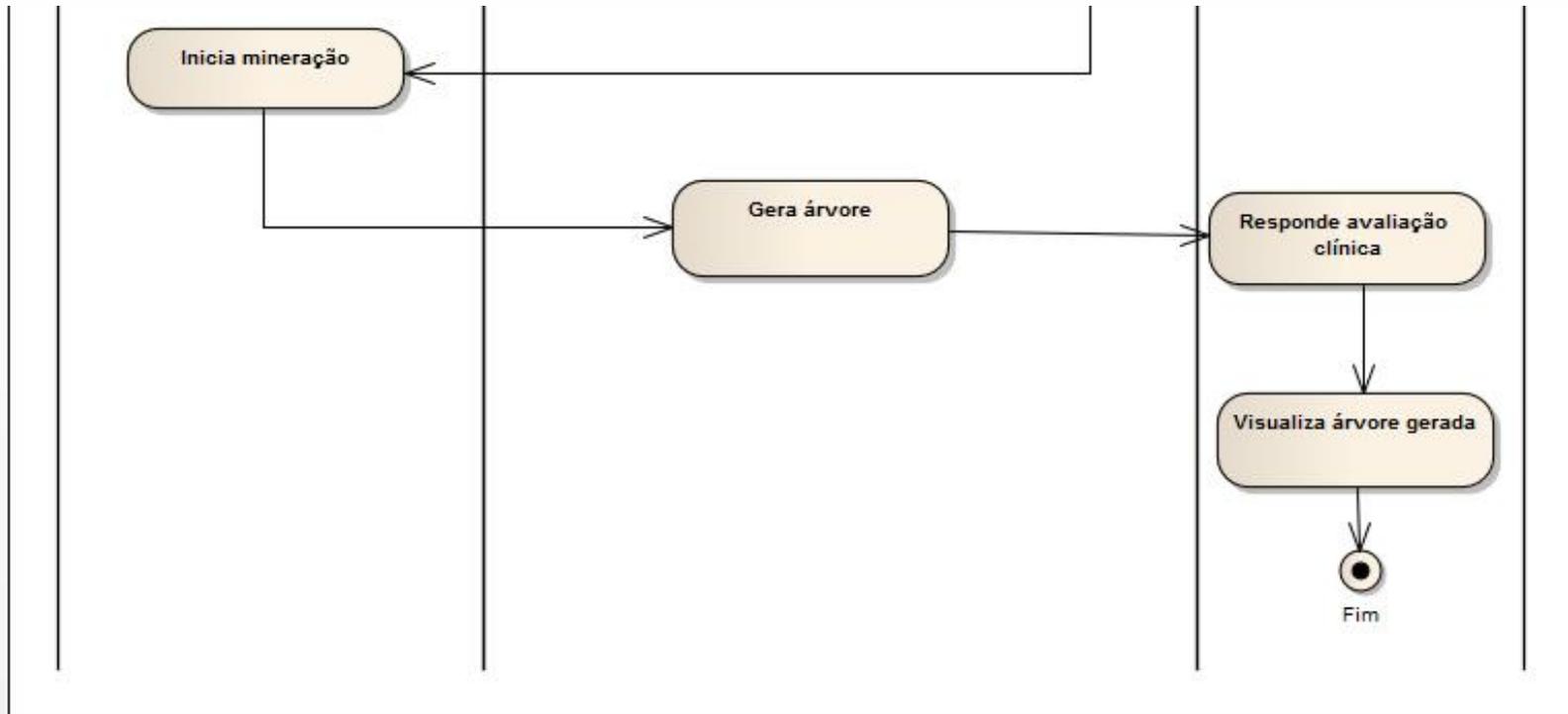


Especificação

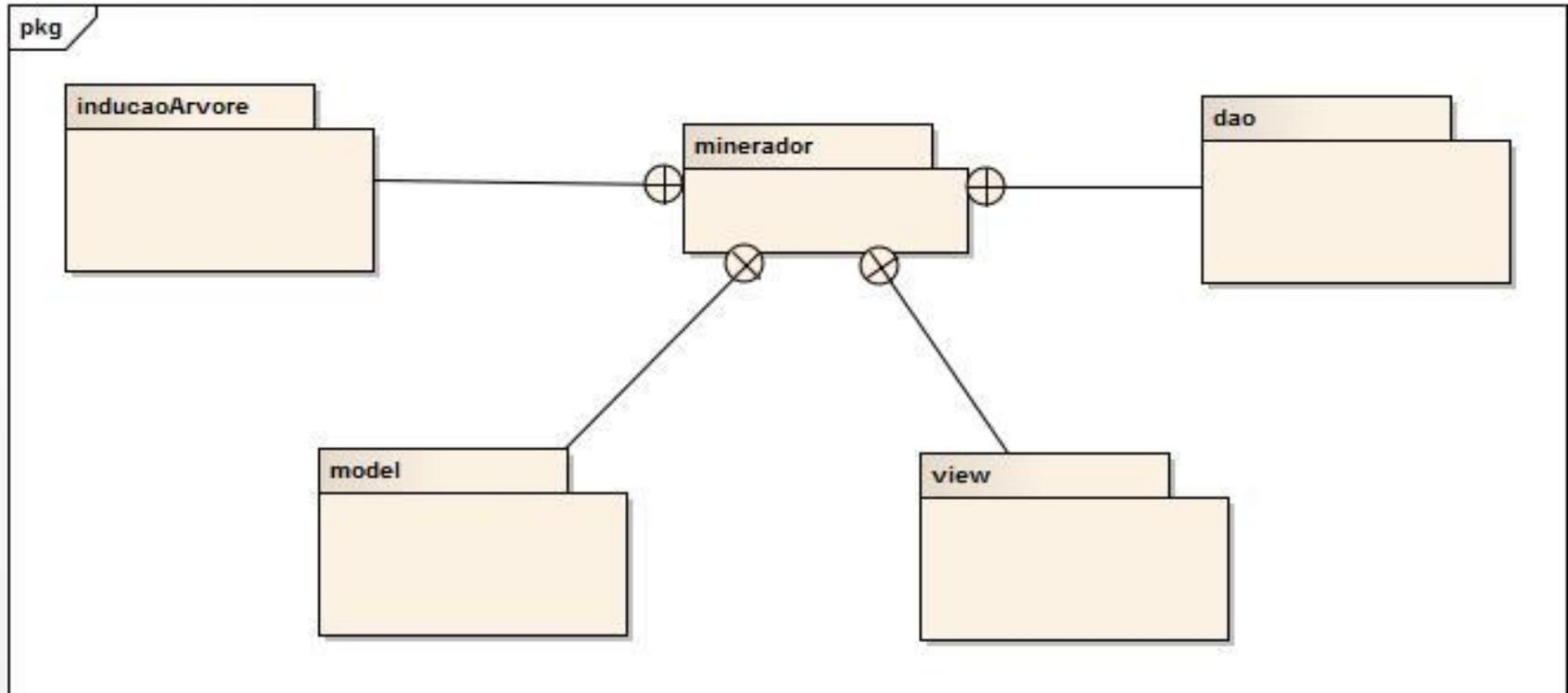
act Diagrama de atividades



Especificação



Especificação



Implementação

```
public void executaMineracaoTodasArvores() {
    listaArvores = configuracaoArvore.getRegistros();
    for (ConfiguracaoArvoreVO arvore : listaArvores) {
        listaRetorno =
prepararArvore.getValores(arvore.getNrSequencia());
        inserirLog("Preparando atributos da árvore -> " +
arvore.getNmArvore());
        criarAtributosMineracao();
        inserirLog(" Executando o algoritmo C4.5");
        ArvoreGerada.getInstace().addArvore(executarC45(arvore));
    }
}
```

Implementação

```
private void criarAtributosMineracao() {
    listaAtributosMineracao.clear();
    String valoresExibir = "";
    for (AtributoRetorno retorno : listaRetorno) {
        valoresExibir = "  Atributo=" + retorno.getNmAtributo()
            + " Tipo=" + retorno.getIeTipoAtributo()
            + " Meta=" + retorno.getIeMeta();
        inserirLog(valoresExibir);
        inserirLog("          Valores=" +
retorno.getValores().toString());

        Atributo novo = new Atributo();
        novo.setNome(retorno.getNmAtributo());
        novo.setTipoAtributo(retorno.getIeTipoAtributo().equals("CO") ?
Tipo.Contínuo : Tipo.Categorico);

        if (retorno.getIeTipoAtributo().equals("CA")) {
            ArrayList<String> lista = new ArrayList<String>();

            lista.addAll(retorno.getValores());
            novo.setValores(lista);
        } else if (retorno.getIeTipoAtributo().equals("CO")) {
            ArrayList<Integer> lista = new ArrayList<Integer>();
            lista.addAll(retorno.getValores());
            novo.setValores(lista);
        }
        listaAtributosMineracao.add(novo);
    }
}
```

Operacionalidade da Implementação

The screenshot displays the 'Minerador' application interface. At the top, there is a navigation bar with five icons: a doctor for 'Avaliação clínica', a tree for 'Gerar árvore', gears for 'Configurar árvore', a database for 'Configurar base', and two people for 'Usuários'. Below this, the 'Avaliação clínica' section is active, showing a dropdown menu for 'Tipo diagnóstico' set to 'Doença cardíaca' and an 'Avaliar' button. The main area contains a form with several sections, each with radio buttons for 'Sim' and 'Nao': 'Usuario de drogas' (selected 'Nao'), 'Fumante' (selected 'Sim'), 'Possui diabetes' (selected 'Nao'), 'Sedentario' (selected 'Sim'), and 'Acima do peso' (selected 'Sim'). To the right of the form is a decision tree diagram. The root node is 'Diagnóstico', which branches into 'Arterioesclerose: 6.0%' and 'Infarto: 12.0%'. The 'Arterioesclerose' node has a child node 'Usuario de drogas=Nao', and the 'Infarto' node has a child node 'Fumante=Sim'.

Minerador

Avaliação clínica

Tipo diagnóstico:
Doença cardíaca

Avaliar

Avaliação clínica Doença cardíaca

Usuario de drogas
 Nao Sim

Fumante
 Sim Nao

Possui diabetes
 Nao Sim

Sedentario
 Sim Nao

Acima do peso
 Sim Nao

Diagnóstico
Arterioesclerose: 6.0%
... Usuario de drogas=Nao
Infarto: 12.0%
... Fumante=Sim

Usuário: Alan Michel Leicht Administrador

Operacionalidade da Implementação

The screenshot displays the 'Minerador' application interface. At the top, there is a navigation bar with five icons: a doctor (Avaliação clínica), a tree diagram (Gerar árvore), gears (Configurar árvore), a database (Configurar base), and two people (Usuários). Below this, the 'Avaliação clínica' section is active, showing a dropdown menu for 'Tipo diagnóstico' set to 'Doença cardíaca' and an 'Avaliar' button. A tree view shows the following results:

- Avaliação clínica: Doença cardíaca
 - Diagnóstico
 - Arteriosclerose: 6.0%
 - Usuario de drogas=Nao
 - Infarto: 12.0%
 - Fumante=Sim
 - Arteriosclerose: 12.0%
 - Fumante=Nao
 - Angina: 8.0%
 - Usuario de drogas=Sim

At the bottom left, the user information is displayed: 'Usuário: Alan Michel Leicht Administrador'.

Resultados e Discussões

- Principais dificuldades
 - Definição dos atributos para construir uma árvore de qualquer diagnóstico de doença;
 - Obtenção de uma base de dados com registros clínicos reais.
- Validação
 - Simulando as atividades do administrador;
 - Simulando as atividades do médico.

Resultados e Discussões

	Ferramenta	Morais et al. (2012)	Zoby (2009)
Linguagem	Java	Java	Java
Plataforma	Desktop	Móvel	Desktop
Utiliza o algoritmo C4.5	Sim	Sim	Não
Permite o diagnóstico de várias doenças	Sim	Não	Não
Técnica de classificação utilizada	Arvore de decisão	Arvore de decisão	Arvore de decisão

Conclusões e Sugestões

- Objetivos alcançados
 - Identificar padrões em uma massa de dados;
 - Criar uma ferramenta de suporte no processo de definição do diagnóstico da doença de um paciente;
 - Indicar quais doenças são mais caracterizadas pelos sintomas informados.

Conclusões e Sugestões

- Pontos positivos
 - Permitir criação de várias árvores de diagnóstico;
 - Exibir todas as doenças do diagnóstico pretendido, juntamente com a combinação de fatores que podem caracterizar cada doença.
- Pontos negativos
 - Utilização de uma base de dados com registros não reais;
 - Falta de conhecimento em medicina para criar as árvores de diagnóstico.

Conclusões e Sugestões

- Trabalhos futuros
 - Migrar para a plataforma Web;
 - Criar modo de visualização da árvore gerada em formato de gráfico;
 - Adicionar funcionalidade para agendar a execução da mineração de dados;
 - Permitir acesso a uma base que esteja na nuvem.