
Bayes Editor: Desenvolvimento e validação de um editor de Redes Bayesianas para o uso no ensino na saúde

Bayes Editor: Development and validation of a Bayesian Network editor for use in health teaching

Marta Rosecler Bez
Universidade Feevale

Fernando Rafael Stahnke
Universidade Feevale

Paulo Ricardo Muniz Barros
Universidade Feevale

Fernando Alex Helwanger
Universidade Feevale

Resumo: Este artigo tem como objetivo apresentar o desenvolvimento e validação de um editor de redes bayesianas, denominado Bayes Editor, a ser utilizado no processo de ensino e aprendizado na área da saúde. Inicia com um referencial teórico que explica os principais conceitos da área de redes bayesianas, seguido de um comparativo de ferramentas com o mesmo propósito disponíveis no mercado. O processo de desenvolvimento do editor é descrito em detalhes. A validação do editor desenvolvido ocorreu com alunos dos cursos de enfermagem, biomedicina e informática. A partir dos resultados pode-se melhorar a ferramenta, bem como, destacar aspectos relevantes deste, como, a usabilidade e facilidade de uso, quando comparada com o uso de outro aplicativo similar.

Palavras-chave: Redes Bayesianas, Bayes Editor, Raciocínio clínico.

Abstract: *This article aims to present the development and validation of a Bayesian network editor, named Bayes Editor, to be used in the health area teaching and learning process. It begins with a theoretical reference that explains the main concepts of the Bayesian networks area, followed by a comparison of tools with the same purpose available in the market. The editor development process is described in detail. The validation of the developed editor occurred with students of the courses of nursing, biomedicine and informatics. From the results can improve the tool, as well, as highlight relevant aspects of the tool, such as usability and ease of use, compared to using another similar application.*

Keywords: Bayesian Networks, Bayes Editor, Clinical Reasoning.

1 Introdução

Existem diversos problemas que devem considerar um nível de incerteza em sua resolução, por possuírem amostras imprecisas ou teorias incompletas. Para este tipo de problema, podem ser usados cálculos probabilísticos, que trazem resultados com um determinado grau de confiança. Uma maneira de realizar estes cálculos probabilísticos é através da utilização de redes bayesianas (RUSSEL; NORVIG, 1995).

Redes bayesianas envolvem uma parte qualitativa e outra parte quantitativa. A parte qualitativa é representada por grafos acíclicos dirigidos, no quais os nodos representam variáveis aleatórias do problema em questão e as arestas representam as relações entre essas variáveis. A parte quantitativa é definida pelas probabilidades atribuídas a cada estado das variáveis aleatórias (WIEGERINCK; KAPPEN; BURGERS, 2010).

As redes bayesianas podem ser empregadas na solução de qualquer problema que envolva incerteza. Um exemplo deste tipo de problema é o processo de diagnóstico, no qual um especialista em determinada área precisa encontrar a causa de um problema a partir dos efeitos observados (RUSSEL; NORVIG, 1995).

No ensino na área da saúde, o aluno precisa praticar a construção de modelos hipotéticos que relacionem doenças com suas causas e sintomas. É preciso que ele possa avaliar a solução e tomar ações no decorrer deste processo (SEIXAS et al., 2002). Neste contexto, o uso de um software no qual o aluno possa simular situações reais e testar suas ações em tempo real mostra-se bastante útil. Para realizar estas simulações, o uso de redes bayesianas pode ser empregado.

Atualmente, existem softwares que permitem a construção e simulação de redes bayesianas. Porém, em projetos em que estas foram empregadas no ensino da saúde, há relatos de que alunos encontraram dificuldades na utilização do editor de redes (MARONI, 2013), o que é um indício de problemas de usabilidade.

Este artigo está dividido em 6 seções. A primeira apresenta a introdução. O referencial teórico que subsidia o trabalho é descrito na seção 2. A seção 3 explora trabalhos correlatos. A seção 4 descreve a metodologia empregada no desenvolvimento do editor e na sua validação. O editor é apresentado na seção 5, bem como os resultados de sua avaliação, seguidos das conclusões desta investigação.

2 Referencial Teórico

Russel e Norvig (1995) acreditam que o processo de diagnóstico (independente se for médico, de reparo automobilístico ou qualquer outra área) quase sempre envolve incerteza. Nesses casos, o especialista precisa identificar a causa do problema a partir de sinais e sintomas, ou seja, dos efeitos da causa observada. Este é um processo que exige treino e repetição, até que o profissional esteja apto a realizar sozinho e de forma direta o diagnóstico a partir de evidências.

Porém, a presença de um sintoma não necessariamente indica com 100% de certeza a causa do problema. Por exemplo, se um paciente possui dor de dente, há uma probabilidade de

ele possuir uma cavidade em algum dente, mas não há certeza disto. Nestes casos, uma representação utilizando lógica de primeira ordem não é o suficiente, é preciso usar teorias probabilísticas para modelar o problema (RUSSEL; NORVIG, 1995).

Portanto, "a principal vantagem de raciocínio probabilístico sobre raciocínio lógico é o fato de que agentes podem tomar decisões racionais mesmo quando não existe informação suficiente para se provar que uma ação funcionará" (CHARNIAK apud MAQUES; DUTRA, 2002, p. 2).

Redes bayesianas são estruturadas na forma de um grafo acíclico dirigido, no qual os nodos representam variáveis aleatórias e as arestas podem ser interpretadas como a indicação de uma relação de causa e efeito entre as variáveis. Utilizando esta rede, inferências podem ser feitas para verificar qual a probabilidade de determinado evento acontecer (NILSSON, 1998), utilizando-se para isso o teorema de Bayes.

2.1 Teorema de Bayes

Quando um problema que envolve incerteza é observado, são envolvidas variáveis aleatórias representadas por V_1, V_2, \dots, V_k . Os possíveis valores dessas variáveis são representados por v_1, v_2, \dots, v_k . Estas variáveis podem ser de diferentes tipos, de acordo com o problema em questão (NILSSON, 1998), como Booleanas (proposições de domínio "verdadeiro" ou "falso"; Numéricas (medidas físicas, como altura, largura, velocidade, entre outros; Categóricas (categorias, como cores, letras, etc).

Uma função chamada de *joint probability function*, que mapeia um conjunto de variáveis em um valor entre 0 e 1, é representada por $p(V_1 = v_1, V_2 = v_2, \dots, V_k = v_k)$ (NILSSON, 1998). Por exemplo, ao considerar três lançamentos de uma moeda, representados pelas variáveis M_1, M_2 e M_3 , que podem possuir os valores "cara" ou "coroa", $p(M_1 = cara) = 0,5$, representando 50% de chance de ser "cara" no primeiro lançamento. A probabilidade dos três lançamentos resultarem em "cara", representada por $p(M_1 = cara, M_2 = cara, M_3 = cara)$, é de 0,125 (HELWANGER, 2016).

A definição dessas probabilidades pode ser dada a partir do processamento de dados ou do julgamento de especialistas no domínio, desde que satisfaçam os seguintes axiomas básicos da probabilidade (NILSSON, 1998):

$$0 \leq p(V_1 = v_1, V_2 = v_2, \dots, V_k = v_k) \leq 1$$
$$\sum_{\text{todas as combinações}} p(V_1, V_2, \dots, V_k) = 1$$

onde o somatório é sobre todos os valores de todas as variáveis. A partir dessas definições, é possível montar uma tabela chamada de *full joint distribution*, que considera todas as combinações de valores para cada variável.

Caso o valor de alguma variável seja conhecido, ele pode ser utilizado para obter a probabilidade de que outras variáveis aconteçam. Para isso, utiliza-se a probabilidade condicional. A probabilidade de V_i dado V_j é representada por $p(V_i | V_j)$ e seu valor é definido pela seguinte equação (NILSSON, 1998):

$$p(V_i|V_j) = \frac{p(V_i, V_j)}{p(V_j)}$$

Este tipo de cálculo envolvendo probabilidade condicional é chamado de inferência probabilística (NILSSON, 1998). A equação da probabilidade condicional também pode ser escrita na forma da regra do produto (RUSSEL; NORVIG, 1995):

$$p(V_i|V_j) = p(V_i|V_j)p(V_j)$$

e aplicada de forma geral como (NILSSON, 1998), sendo:

$$p(v_1, v_2, \dots, v_k) = \prod_{i=1}^k p(V_i|V_{-1, \dots, V_{i-1}})$$

Utilizando a regra do produto e considerando que $p(A, B) = p(B, A)$, a seguinte equação pode ser escrita:

$$p(V_i, V_j) = p(V_i|V_j)p(V_j) = p(V_j|V_i)p(V_i) = p(V_j, V_i)$$

A partir dessa equação, é obtida a chamada de regra de Bayes (NILSSON, 1998):

$$p(V_i, V_j) = \frac{p(V_j|V_i)p(V_i)}{p(V_j)}$$

A regra de Bayes pode ser útil na prática para o processo de diagnóstico. Ela pode ser reescrita como:

$$p(causa|efeito) = \frac{p(efeito|causa)p(causa)}{p(efeito)}$$

Algumas variáveis podem não ter relação com as outras dependendo do problema modelado. Pode-se dizer que uma variável V é condicionalmente independente de um conjunto de variáveis V_i , dado um conjunto de variáveis V_j , se $p(V|V_i, V_j) = p(V|V_j)$. Para representar isto, é usada a notação $I(V, V_i|V_j)$. Isso significa que se o valor de V_j for conhecido, V_i não influencia em nada a probabilidade de V (NILSSON, 1998). A independência condicional é aproveitada para uma representação mais simples de problemas probabilísticos em estruturas chamadas de redes bayesianas.

2.2 Redes Bayesianas

Para resolver problemas probabilísticos, a tabela representando a *full joint distribution* pode ser utilizada. Porém, quanto mais variáveis são adicionadas ao problema, o número de entradas nessa tabela cresce exponencialmente. Problemas reais tendem a ter um número relativamente grande de variáveis, o que torna essa abordagem ineficiente e, muitas vezes, inviável. Para uma representação mais simples deste tipo de problema, aproveitando as propriedades de independência entre as variáveis, podem ser utilizadas redes bayesianas (RUSSEL; NORVIG, 1995). Estas são representadas por grafos acíclicos dirigidos, com a seguinte especificação (RUSSEL; NORVIG, 1995): Cada nodo representa uma variável aleatória; se houver uma aresta do nodo X ao nodo Y, X é considerado pai de Y; Cada nodo contém uma distribuição de probabilidade condicional que quantifica o efeito de seus pais em suas probabilidades.

A topologia da rede especifica quais são as relações de independência condicional entre as variáveis observadas. Cada variável representada por um nodo, dados os valores das variáveis representadas pelos seus nodos pais, é condicionalmente independente de todas as outras variáveis que não sejam descendentes dela.

As arestas que conectam os nodos da rede podem ser interpretadas como uma relação de causa e efeito, na qual o nodo pai é a causa do efeito representado pelo nodo filho. Este tipo de interpretação é muito próximo do raciocínio que seres humanos especialistas em alguma área utilizam, portanto, a representação é feita de forma natural (NILSSON, 1998).

A construção de uma rede bayesiana envolve duas partes, sendo uma qualitativa e a outra quantitativa. A parte qualitativa é a estrutura do grafo que representa a rede. A parte quantitativa é representada pelas tabelas de probabilidade (CTPs) associadas a cada variável. Essa especificação pode ser inferida de dados, construída manualmente por um especialista ou uma mistura das duas alternativas (WIEGERINCK; KAPPEN; BURGERS, 2010).

Além dessa representação das relações de independência entre as variáveis, é preciso associar uma tabela de probabilidade condicional a cada uma delas. A próxima seção apresenta ferramentas desenvolvidas para a geração de redes bayesianas.

3 Ferramentas Correlatas

Nesta seção são apresentadas três ferramentas utilizadas para o desenvolvimento de Redes Bayesianas. Ao final da seção é apresentada uma tabela comparativa entre estas ferramentas e, ao final da seção de desenvolvimento, com o Bayes Editor.

3.1 Hugin

O Hugin é considerado uma ferramenta de propósito geral para modelos probabilísticos gráficos (MADSEN et al., 2003). Ele é desenvolvido pela Hugin Expert, empresa dinamarquesa fundada em 1989. A empresa surgiu a partir de um projeto de pesquisa para ajudar a diagnosticar doenças musculares e nervosas (HUGIN, 2016).

No Hugin, cada rede bayesiana é aberta em uma janela para edição e simulação. Dentro dessa janela, há uma barra de ferramentas na parte superior. Nessa barra, ficam disponibilizadas uma série de ferramentas para inserir elementos na rede, sendo elas: nodos representando variáveis aleatórias discretas com um número finito de estados; nodos representando variáveis aleatórias com uma função de distribuições contínua; ligação causal entre um nodo pai e um nodo filho; nodos representando uma função de utilidade em diagramas de influência; nodos representado uma função; nodo representando uma distribuição discreta em função dos valores dos nodos pais; nodo que representa uma decisão a ser tomada pelo usuário com base nos valores de seus nodos pais em um diagrama de influência; nodos de instância.

Ao clicar com o botão direito em um dos nodos da rede, o usuário pode abrir a tabela de probabilidade condicional associada ao nodo. Desta forma é possível editar e definir a probabilidade de cada uma das combinações de estados possíveis.

Na barra de ferramentas superior o usuário pode clicar em um botão que altera a rede do modo de edição para o modo de execução, sendo possível, neste caso, visualizar as probabilidades de cada estado de cada nodo após a inferência. Também é possível informar os valores conhecidos de cada nodo que fica sinalizado em vermelho, desta forma, fazendo com que todos os nodos da rede sejam recalculadas.

Além do editor gráfico, o Hugin também disponibiliza um conjunto de APIs (C, C++, .NET, Java e ActiveX) para que programadores possam utilizar o seu motor de inferência para criar suas próprias aplicações. Esse fato dá liberdade para que programadores utilizem redes bayesianas criadas no Hugin em suas próprias aplicações. Outro aspecto interessante do Hugin é que ele possui um conjunto diversificado de tipos de nodos, o que torna a ferramenta útil, ou eficiente, para um número maior de problemas. Entretanto, o fato de o Hugin possuir o código fonte fechado, dependendo do cenário pretendido de aplicação das redes bayesianas, pode não ser o ideal, já que impede que modificações ou consultas sejam feitas no código do programa.

O fato de o modo de edição e execução serem separados pode diminuir a velocidade com que o usuário faça simulações. Por exemplo, se ele, durante o modo de execução, desejar alterar a probabilidade de algum estado de um nodo da rede, será necessário ir para o modo de edição, editar o estado desejado e então voltar para o modo de execução.

Um caso de uso que é confuso no Hugin é a edição das CPTs quando o nodo possui muitas combinações de estados. Não fica imediatamente claro quais são os estados dos nodos pais e os estados do nodo que está em edição no momento. Além disso, em aplicações reais, o número de estados normalmente é grande, o que dificulta ainda mais a edição.

3.2 Netica

De acordo com a definição presente em seu site, o Netica é “Um pacote de software completo para resolver problemas utilizando redes bayesianas e diagramas de influência” (NETICA, 2016). Ele possui um editor gráfico e um conjunto de APIs (Java, C, C#, Visual Basic/COM, C++, Matlab e CLisp) que programadores podem utilizar em suas aplicações.

Na ferramenta Netica existe uma tela principal de edição de redes bayesianas com uma barra de ferramentas na parte superior onde é possível selecionar recursos para serem incluídos na rede bayesiana, sendo eles: um nodo que representa uma variável aleatória na rede bayesiana (discreta ou contínua); um nodo que representa uma decisão a ser tomada por um usuário da rede; um nodo que representa um valor utilidade que deve ser maximizado em um diagrama de influência; ligações entre os nodos da rede.

Após montar a estrutura da rede, é preciso informar os estados para cada um dos nodos, bem como, os nomes dos estados. Feito isso, é preciso definir e probabilidade de cada um dos nodos e estados, sendo informadas as probabilidades para cada combinação possível.

No Netica, após definir as probabilidades de cada estado, elas não são propagadas automaticamente para todos os nodos da rede, é necessário compilar a rede. Após a rede estar compilada, é possível realizar inferências nos nodos criados. Para informar um estado já conhecido para um dos nodos, basta clicar em cima no estado desejado.

O Netica possui uma série de APIs para uso de programadores em suas aplicações, o que pode ser útil dependendo do contexto de aplicação. Ele também possui uma grande diversidade de tipos de nodos, o que faz com que a ferramenta seja útil em diferentes aplicações.

No Netica, a edição da tabela de probabilidades mostrou-se bastante natural, já que ficou evidente quais estados representam os estados dos nodos pai e quais são referentes ao nodo que é editado. Também é possível selecionar se as probabilidades serão informadas em valores percentuais, de 0 a 100, ou em valores numéricos representando a probabilidade, de 0 a 1. Além disso, como não existe a separação de um modo de edição ou execução, o Netica mostra-se como uma ferramenta com um fluxo ágil para simulações, já que o usuário fica a poucos cliques da edição das CPTs enquanto pode visualizar os resultados da inferência.

O Netica possui mais de uma maneira para informar os estados dos nodos, mas, apesar de elas tornarem a maneira de informar os valores mais flexível, ficou confuso para o usuário entender qual delas utilizar. Além disso, foram encontrados vários erros ao tentar realizar este processo, o que dificultou bastante a execução da tarefa. Outro aspecto observado é que a compilação da rede não é automática. Sendo assim, após o usuário informar as probabilidades da rede, ele precisa manualmente clicar no botão para recompilar. Caso o usuário não perceba isso, ele pode assumir valores incorretos na inferência da rede. Além destes pontos, também possui o código fonte fechado, o que pode não ser o ideal dependendo da aplicação pretendida.

3.3 UnBBayes

O UnBBayes é um software de código aberto para modelagem, aprendizado e raciocínio baseado em redes bayesianas (UNBBAYES, 2016). Ele é um *framework* para montagem e inferência de redes bayesianas desenvolvido pela Universidade de Brasília (SANTOS, 2009).

Na parte superior existe uma barra de ferramentas que pode ser utilizada para inserir nodos de probabilidade, decisão e utilidade, além de arcos para as ligações entre eles. No UnBBayes, não é possível adicionar nodos representando variáveis contínuas, apenas discretas. Após definir a estrutura da rede é possível adicionar os estados nos nodos presentes. Feito isso, um novo estado será adicionado na CPT na aba "Tabela", que pode ser editada imediatamente.

Quando a rede estiver finalizada, é possível entrar no modo de execução e “Compilar árvore de junção”, sendo apresentada uma lista com os nodos e seus estados, onde são exibidos os resultados da inferência. Além disso, há uma exibição da rede com os resultados em cada nodo e, clicando em um dos estados de um nodo, tanto na listagem da esquerda quando diretamente no nodo, a crença de que aquela variável está naquele estado é inserida. Porém, a propagação das crenças não é automática, é preciso clicar no botão “Propagar evidências” para que os resultados da inferência sejam exibidos. O UnBBayes também disponibiliza uma API para utilização por programadores, mas apenas para a linguagem Java.

O UnBBayes é de código aberto, o que permite que modificações e consultas ao seu código sejam feitas. Isto o torna uma ferramenta bastante flexível, pois, dependendo da aplicação, isto pode ser necessário. Outro aspecto interessante é que a CPT é de fácil acesso para edição. Basta selecionar o nodo desejado e ela ficará disponível na parte superior da rede para edição.

O UnBBayes não possui nodos contínuos e, dependendo da aplicação da ferramenta, este fator pode ser importante. Apesar de a tabela para edição de estados ser de fácil acesso, não é óbvio para o usuário de que nesta tabela ele pode alterar o nome dos estados, o que pode fazê-lo pensar que não é possível editar. Como a tabela de edição das probabilidades é bem semelhante à do Hugin, ela possui os mesmos problemas em relação a complexidade, que cresce de forma proporcional ao número de estados.

No modo de edição, é possível informar os valores dos estados conhecidos. Porém, nas outras ferramentas, para remover uma crença informada basta clicar novamente no estado que foi informado. No UnBBayes, é preciso ir até o botão “Reiniciar crenças”, o que, além de não ser tão ágil, pode não ser óbvio ao usuário.

Além disso, apesar de possuir os modos de edição e execução separados, a propagação das probabilidades não é automática, é preciso realizar a ação manualmente. Desta forma, além de o usuário precisar ficar alternando entre os dois modos, ele precisa realizar a propagação manualmente, o que não facilita a tarefa de simulações, podendo visualizar probabilidades inconsistentes com os estados caso não perceba que a propagação precise ser feita.

3.4 Comparativo entre as ferramentas analisadas

A partir das características descritas nesta seção em relação a alguns dos editores de redes bayesianas existentes, é possível fazer uma comparação entre as funcionalidades presentes neles. Os critérios utilizados para a comparação levam em consideração as funções principais a serem atendidas nas ferramentas. O Quadro 1 exibe um comparativo para os editores analisados.

Como pode ser percebido no quadro, as três ferramentas atendem ao critério básico de nodos de estados discretos, porém o de estados contínuos não é atendido pelo UnBBayes, bem como a possibilidade de impressão da rede, sendo estas as principais deficiências da ferramenta. Em contrapartida, o UnBBayes possui código fonte aberto, o que permite que se estude e use características no desenvolvimento do Bayes Editor, proposto neste artigo. Nos três editores estudados, existem dificuldades na edição das CPTs, bem como a falta de agilidade na edição das probabilidades durante a inferência, sendo a principal contribuições do

editor proposto neste artigo, desenvolver um editor que tenha como foco a usabilidade, levando em consideração que o público que a usará é composto por professores e alunos da área da saúde, portanto, sem conhecimentos aprofundados de informática.

Quadro 1 – Comparativa de funcionalidades entre os editores analisados.

Características	Hugin	Netica	UnBBayes
Nodos de estados discretos	Sim	Sim	Sim
Nodos de estados contínuos	Sim	Sim	Não
Diagramas de influência	Sim	Sim	Sim
Impressão da estrutura da rede	Sim	Sim	Não
Impressão das CPTs da rede	Sim	Não	Não
Código fonte	Fechado	Fechado	Aberto
APIs	Diversas linguagens	Diversas linguagens	Linguagem Java
Facilidade de edição das CPTs	Dificuldade de visualizar quais estados são referentes ao nodo em edição e os são referentes aos estados dos nodos pai	Quais estados se referem aos nodos pais e quais se referem ao nodo atual de fácil visualização	Dificuldade de visualizar quais estados são referentes ao nodo em edição e quais são referentes aos estados dos nodos pai
Agilidade na edição das probabilidades durante a inferência	Modo de edição e execução separados, o que causa uma troca de contexto pelo usuário	Não existe separação entre a edição e execução, o que torna a ferramenta ágil para esta tarefa	Modo de edição e execução separados, o que causa uma troca de contexto pelo usuário

Fonte: elaborada pelos autores.

4 Metodologia de desenvolvimento e validação

Á partir das características dos softwares analisados foi desenvolvido um editor de redes bayesianas denominado Bayes Editor, com intuito de facilitar o uso deste tipo de ferramenta no ensino e aprendizagem de raciocínio diagnóstico. O software foi desenvolvido utilizando tecnologias da plataforma web, como JavaScript, HTML e CSS, permitindo que seja utilizado em múltiplas plataformas e sem a necessidade de instalação. Foi desenvolvido em dois projetos separados: uma biblioteca de inferência e a interface gráfica do editor. Desta forma, a biblioteca de inferência pode ser utilizada em diversos projetos no futuro, como *web services*, outras interfaces gráficas para criação de redes, simuladores aplicados ao ensino, entre outros.

Apesar da utilização da plataforma web para o desenvolvimento do editor, todo o processamento da edição e inferência da rede bayesiana é realizado no cliente via JavaScript. O único tráfego que ocorre na rede é o *download* inicial dos arquivos. Após isso, todo o processamento será feito no cliente para que se obtenha um tempo de resposta satisfatório.

O desenvolvimento do trabalho utiliza a especificação ECMAScript 2016 da linguagem JavaScript. Para que mesmo alguns navegadores não atualizados não tenham problemas em interpretar as funcionalidades da linguagem, foi utilizado o Babel, que compila o código ECMAScript 2016 para a versão ECMAScript 5, disponível em grande parte dos navegadores. Além disso, foi usado o Flow, uma ferramenta que pode ser utilizada para fazer verificação estática de tipos em um projeto JavaScript.

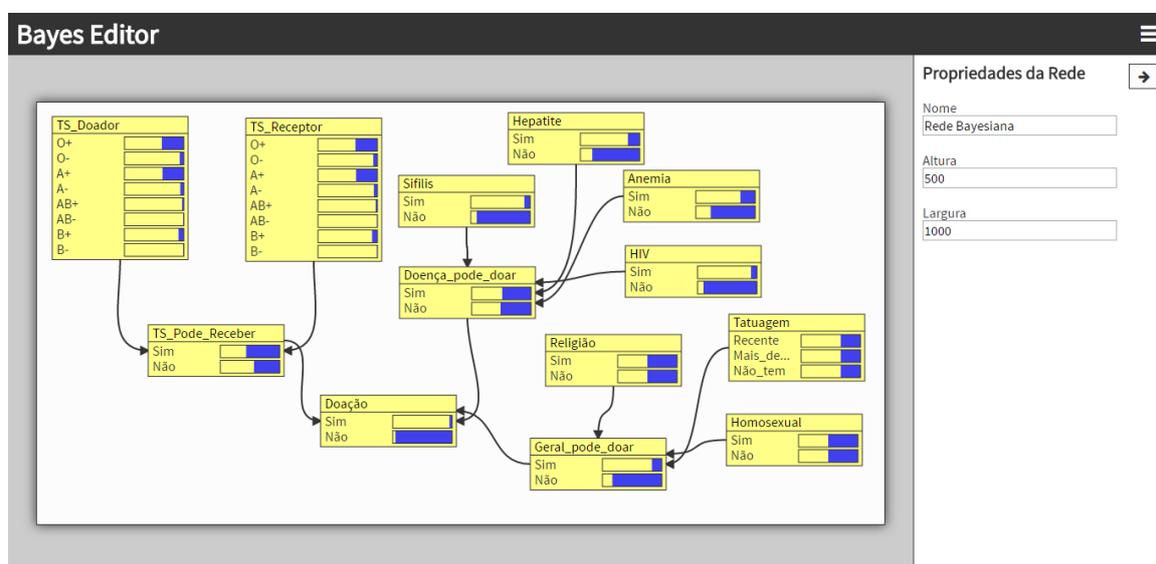
Para a validação do trabalho, foram realizados testes com nove usuários (2 estudantes de biomedicina, 2 estudantes de enfermagem e 5 estudantes de áreas relacionadas à

computação). Além disso, foram realizados testes também em um dos editores estudados, o Hugin, possibilitando uma comparação relativa entre os resultados.

Antes de os usuários utilizarem os editores, foi dado a eles um treinamento a respeito de redes bayesianas. O objetivo foi esclarecer os seguintes itens: Aplicações de redes bayesianas; Estrutura de uma rede bayesiana; Tipos de inferência; Passos para construir uma rede bayesiana (selecionar as variáveis relevantes; identificar o relacionamento entre as variáveis; definir as probabilidades de cada uma das variáveis e avaliar o resultado da rede).

Após a apresentação destes conceitos, foi criada por todos uma rede em ambos os editores. Após isso, foi solicitado que os usuários criassem nos editores, individualmente, a rede da Figura 1. Essa rede foi elaborada por alunos do curso de biomedicina, que representa o problema de verificar se uma pessoa é apta para realizar uma doação de sangue.

Figura 1 – Rede Bayesiana de Doação de Sangue.



Fonte: elaborada pelos autores.

Nos testes, foi solicitado que os usuários falassem quais dificuldades encontraram durante a utilização dos editores, que foram observadas pelos autores. Além disso, ao final, foi respondido a um questionário contendo nove questões optativas sobre usabilidade, seguindo a escala Likert e uma questão descritiva para críticas e sugestões gerais. Nas questões optativas, haviam opções de 1 (discordo totalmente) a 5 (concordo totalmente).

5 O Bayes Editor

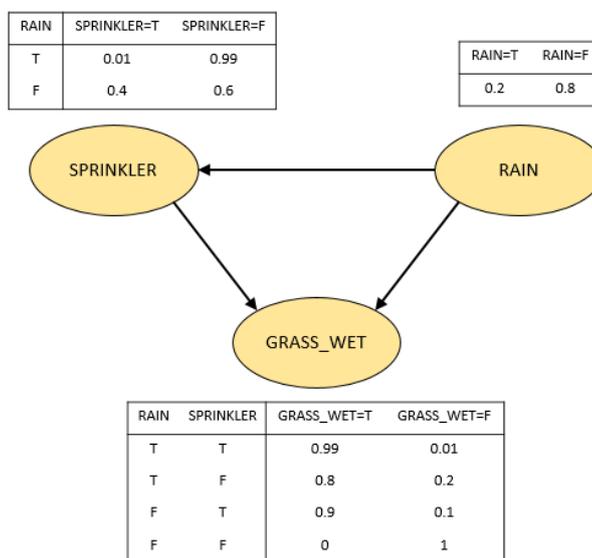
A Figura 1 apresenta uma captura de tela do editor de redes bayesianas desenvolvido. Para a construção da interface gráfica do software, foi utilizada a biblioteca *React*. Esta biblioteca serve para que a interface gráfica possa ser desenvolvida de forma declarativa, baseada em componentes. Ao final do desenvolvimento, foram criados os seguintes componentes:

- AddNodeModal: tela para adicionar nova variável;
- App: componente raiz que engloba a aplicação inteira;

- Button: componente para todos os botões da aplicação;
- Canvas: área branca em que as variáveis da rede são exibidas;
- ContextMenu: menu de contexto que é exibido quando o usuário clica com o botão direito no canvas de edição;
- EditCptModal: tela para a edição da tabela de probabilidades de uma variável;
- EditStatesList: lista de estados, usada tanto na tela de edição de estados quanto na tela de adicionar variável;
- EditStatesModal: tela para a edição de estados de uma variável;
- Header: cabeçalho da aplicação, com título e acesso ao menu;
- Node: representa uma variável da rede;
- PropertiesPanel: painel de propriedade para editar a rede e variáveis.

Na biblioteca de inferência, a rede bayesiana é representada por um objeto que pode ser descrito em formato JSON (*JavaScript Object Notation*). Uma rede de exemplo pode ser observada na Figura 2 e sua representação em formato JSON na Figura 3. O objeto da rede possui uma propriedade para cada variável, e o nome da propriedade é o identificador da respectiva variável. Cada variável possui as seguintes propriedades: id (identificador da variável); states (lista de estados possíveis para a variável); parents (lista de identificadores das variáveis que são pais da variável em questão); cpt (tabela de probabilidades condicionais associada à variável).

Figura 2 – Rede Bayesiana de exemplo.



Fonte: elaborada pelos autores.

Figura 3 – Rede Bayesiana representada em JSON

```
{
  "RAIN": {
    "id": "RAIN",
    "states": [ "T", "F" ],
    "parents": [],
    "cpt": { "T": 0.2, "F": 0.8 }
  },
  "SPRINKLER": {
    "id": "SPRINKLER",
    "states": [ "T", "F" ],
    "parents": [ "RAIN" ],
    "cpt": [
      { "when": { "RAIN": "T" }, "then": { "T": 0.01, "F": 0.99 } },
      { "when": { "RAIN": "F" }, "then": { "T": 0.4, "F": 0.6 } }
    ]
  },
  "GRASS_WET": {
    "id": "GRASS_WET",
    "states": [ "T", "F" ],
    "parents": [ "RAIN", "SPRINKLER" ],
    "cpt": [
      { "when": { "RAIN": "T", "SPRINKLER": "T" }, "then": { "T": 0.99, "F": 0.01 } },
      { "when": { "RAIN": "T", "SPRINKLER": "F" }, "then": { "T": 0.8, "F": 0.2 } },
      { "when": { "RAIN": "F", "SPRINKLER": "T" }, "then": { "T": 0.9, "F": 0.1 } },
      { "when": { "RAIN": "F", "SPRINKLER": "F" }, "then": { "T": 0, "F": 1 } }
    ]
  }
}
```

Fonte: elaborada pelos autores

Para as variáveis que não possuem nenhum nodo pai, a CPT é representada por um objeto no qual cada propriedade define a probabilidade de um de seus estados, como pode ser observado na variável RAIN. Quando a variável possuir pais, a CPT é representada por uma lista de objetos com as propriedades *when* e *then*. A propriedade *when* é representada por um objeto descrevendo uma das combinações possíveis dos estados de cada um dos nodos pais da variável. A propriedade *then* representa as probabilidades para cada um dos estados da variável, no mesmo formato da CPT quando a variável não possui pais. Desta forma, essa CPT pode ser entendida com uma lista de objetos que descrevem as probabilidades (objeto *then*) para uma condição (objeto *when*). Essa lista de objetos deve possuir todas as combinações possíveis de estados dos pais da variável.

Durante o trabalho, foram realizadas duas implementações de algoritmos diferentes para fazer a inferência na rede. Ambos utilizam a mesma estrutura para a representação da rede bayesiana apresentada acima, porém, uma com inferência por enumeração e a outra por eliminação de variáveis (HELWANGER, 2016).

O Quadro 2 mostra um comparativo de funcionalidades entre os editores estudados e o editor desenvolvido. É possível visualizar que o Bayes Editor tem um número de funcionalidades mais limitado em relação aos editores existentes. Porém, a decisão durante o desenvolvimento foi justamente criar um conjunto mínimo de funcionalidades que permitam aos usuários criar redes bayesianas, desde que as funcionalidades disponíveis possuam uma boa usabilidade.

Quadro 2 – Comparativo de funcionalidades entre os editores analisados e o desenvolvido.

	Hugin	Netica	UnBBayes	Bayes Editor
Nodos de estados discretos	Sim	Sim	Sim	Sim
Nodos de estados contínuos	Sim	Sim	Não	Não
Diagramas de influência	Sim	Sim	Sim	Não
Impressão da estrutura da RB	Sim	Sim	Não	Não
Impressão das CPTs da RB	Sim	Não	Não	Não
Código fonte	Fechado	Fechado	Aberto	Aberto
APIs	Disponíveis em diversas linguagens	Disponíveis em diversas linguagens	Disponível apenas na linguagem Java	Disponível na linguagem JavaScript
Facilidade de edição das CPTs	Dificuldade de visualizar quais estados são referentes ao nodo em edição e referentes aos estados dos nodos pai	Quais estados se referem aos nodos pais e quais se referem ao nodo atual de fácil visualização	Dificuldade de visualizar quais estados são referentes ao nodo em edição e referentes aos estados dos nodos pai	Quais estados se referem aos nodos pais e quais se referem ao nodo atual de fácil visualização
Agilidade na edição das probabilidades durante a inferência	Modo de edição e execução separados, o que causa uma troca de contexto pelo usuário	Não existe separação entre o modo de edição e execução, o que torna a ferramenta bastante ágil para esta tarefa	Modo de edição e execução separados, o que causa uma troca de contexto pelo usuário	Não existe separação entre o modo de edição e execução, o que torna a ferramenta bastante ágil para esta tarefa

Fonte: elaborada pelos autores.

Além disso, apesar de o editor desenvolvido não ter uma funcionalidade específica para impressão da estrutura da rede, como ele é uma aplicação web, pode ser utilizada a funcionalidade de impressão do navegador.

Tanto a biblioteca de inferência como o editor gráfico de redes bayesianas foram desenvolvidos com código aberto e podem ser encontrados, respectivamente, em: <https://github.com/fhelwanger/bayesjs> e <https://github.com/fhelwanger/bayesjs-editor>.

5 Resultados e Discussão

A Tabela 1 apresenta as médias de respostas para cada pergunta após os testes de usabilidade para os dois editores.

Tabela 1 – Respostas do questionário dos testes de usabilidade

Número	Questão	Hugin	Bayes Editor
1	O editor fornece as funcionalidades necessárias para a criação/inferência em uma RB	4,33	4,68
2	O editor permite que a criação e inferência em uma RB seja realizada com rapidez	2,78	4,44
3	O editor possui um fluxo de trabalho que é claro e não gera confusões	2,33	4,67
4	Os textos e mensagens utilizados no editor são claros	3,22	4,44
5	O editor não apresenta erros durante a execução	4	4,89
6	O editor informa ao usuário quando dados inválidos são informados	3,78	4,78
7	A interface gráfica utilizada pelo editor é intuitiva	2,67	4,44
8	É fácil lembrar de como executar as operações no editor	3,33	4,89
9	O editor fornece os resultados com tempo de resposta satisfatório	4	4,44

Fonte: elaborada pelos autores

Com base nas respostas da primeira questão, é possível observar que ambos os editores se mostraram satisfatórios em relação ao fornecimento de funcionalidades para a criação de redes bayesianas. As respostas da segunda questão já mostram que o fluxo de trabalho adotado no Bayes Editor se mostrou mais ágil, provavelmente por se basear na ideia de não possuir telas separadas para a montagem e a inferência na rede.

As respostas para as questões de número 3, 7 e 8 mostram que o fluxo de trabalho do Bayes Editor se mostrou mais intuitivo. Conforme relatado na seção de trabalhos correlatos, a edição da tabela de probabilidades no Hugin é confusa. Nos testes com usuários esse problema mostrou-se bastante aparente e, em geral, as pessoas conseguiram compreender melhor a forma adotada no Bayes Editor para a disposição das tabelas de probabilidades. Além disso, a redução no número de telas para tornar o editor mais simples também contribuiu para o melhor entendimento.

Na questão 4, os usuários acreditam que os textos e mensagens do Bayes Editor se mostraram mais claros do que os do Hugin. Por conta de o Bayes Editor possuir um conjunto mais limitado de funcionalidades, foi possível utilizar termos bastante simples para descrever os elementos da interface gráfica, já o Hugin, apresenta termos bastante técnicos, como *Discrete change tool* e *Function tool*. Isto pode explicar a escolha dos usuários.

Em geral os usuários concordam que os editores não apresentaram erros durante a execução, conforme mostra a questão 5. Alguns usuários reclamaram que o Hugin não aceitava acentos ou espaços nos nomes das variáveis, isso pode ter sido percebido por alguns como erro e explicaria a média um pouco inferior em relação ao Bayes Editor.

Conforme pode ser observado nas respostas da questão 9, os editores forneceram resultados em tempo satisfatório para as redes montadas. Porém, um dos usuários gerou algumas ligações mais complexas durante a montagem da rede no Bayes Editor e foi possível observar lentidões durante a execução do editor. Para que os resultados sejam mostrados em tempo real no editor, é preciso que seja feita uma inferência para cada estado de cada variável da rede a cada modificação. Devido a isto, mesmo o algoritmo de eliminação de variáveis levando alguns milissegundos para inferências isoladas, ao somar todas elas, em redes complexas fica perceptível a lentidão.

Além das respostas das questões optativas, foram obtidas diversas sugestões de melhorias, tanto por meio do campo para sugestões e críticas gerais do questionário, quanto por meio de observações feitas durante os testes. A maioria dessas sugestões foram itens bastante simples, mas que aumentaram ainda mais a usabilidade do software.

Um dos itens sugerido por boa parte dos usuários foi a questão de que o Bayes Editor poderia usar teclas de atalho para algumas operações ao invés do mouse. Ao adicionar uma variável, editar os estados ou editar as tabelas de probabilidades das redes, o usuário é obrigado a clicar no botão de confirmação da janela para concluir o processo. Foi sugerido que, ao pressionar a tecla ENTER, o processo também seja concluído. Também foi sugerido que, além de possuir um menu de contexto para apagar uma variável da rede, quando a variável estiver selecionada, seja possível excluí-la pressionando a tecla DELETE.

Outra sugestão foi que seja possível ligar as variáveis da rede sem a necessidade de acessar um menu de contexto. Alguns softwares existentes mostram pontos nas bordas dos objetos que podem ser ligados, e basta ligar estes pontos com o mouse para estabelecer a relação entre eles.

Também foram dadas sugestões para que a estrutura lógica da rede possa ter uma representação visual mais organizada. Uma delas é que, ao selecionar uma das variáveis, todas as setas que estão ligadas a ela fiquem realçadas com uma cor diferente. Desta forma, quando houverem ligações complexas, o usuário pode ter uma melhor visualização da estrutura da rede. Além disso, foi sugerido que a posição das setas possa ser ajustada manualmente. Na versão do software utilizada na validação, o usuário só pode ajustar as posições das variáveis e as setas são ajustadas automaticamente. Porém, o ajuste automático pode nem sempre trazer a disposição das setas para o entendimento do usuário.

As telas de edição de estados e edição de probabilidades possuem como título, respectivamente, os textos "Editar Estados" e "Editar Tabelas de Probabilidade". Porém, em nenhum lugar dessas telas é exibido o nome da variável em edição. Isso pode gerar confusão caso não lembrem qual variável estava sendo editada. Foi notado que os usuários sentiram falta da exibição desta informação, que poderia ser incluída no título dessas telas.

Alguns usuários informaram que prefeririam que, para a exibição dos resultados das inferências, fossem utilizados números ao invés das barras gráficas. A versão utilizada na validação exibe um texto com o valor numérico quando o usuário passa o mouse sobre a barra, porém, o acesso à essa informação é mais difícil. Isso foi resolvido com a inclusão de uma opção no software para que seja possível escolher se a exibição dos resultados será na forma de números ou barras.

Outro item levando foi que a versão do software utilizada na validação não permite a seleção de múltiplas variáveis simultaneamente. Dessa forma, mover várias variáveis de lugar se torna bastante trabalhoso. Com a implementação dessa funcionalidade, as operações de mover e apagar múltiplas variáveis se tornaria mais rápida.

Foi observado que, quando a tabela de probabilidades possui uma lista muito grande de combinações, é exibida uma barra de rolagem para que o usuário possa navegar por toda a tabela. Porém, ao mover a barra de rolagem para baixo, o cabeçalho da tabela é escondido, o que pode gerar confusão durante a edição da tabela. O ideal seria que o cabeçalho permanecesse fixo e apenas o corpo da tabela rolasse.

6 Considerações finais

A utilização de redes bayesianas mostra ser uma maneira eficaz para a solução de problemas que envolvem a incerteza. Por representarem relações de causa e efeito entre variáveis, são semelhantes ao raciocínio de especialistas em determinada área. Por este motivo, podem ser aplicadas no processo de ensino e aprendizagem na área da saúde.

A usabilidade pode ser um fator muito importante no sucesso de um software pois, caso ele possua uma usabilidade baixa, o usuário ficará frustrado e simplesmente poderá abandonar a ferramenta. Questionários de avaliação de usabilidade mostram-se bastante úteis para que um

índice satisfatório seja atingido ao desenvolver um software, principalmente como este, voltado ao ensino.

O objetivo deste artigo foi apresentar o desenvolvimento e validação de um editor de redes bayesianas com foco em usabilidade que possa ser empregado no processo de ensino e treinamento do raciocínio clínico. Conforme relatado neste artigo, o desenvolvimento do editor foi realizado e nele podem ser construídas diversas redes bayesianas. Pela própria definição destas redes, elas podem ser aplicadas a qualquer problema que envolva incerteza, portanto, o editor também se aplica a diferentes domínios.

Além disso, foram realizadas pesquisas sobre os editores de redes bayesianas existentes no mercado, o que foi bastante útil para encontrar pontos fortes e fracos nas soluções existentes, buscando manter as características positivas e eliminar as negativas na nova solução.

Tendo como base os resultados dessas pesquisas, foi feita uma especificação do novo editor utilizando como principal ferramenta os *wireframes*, além da escolha de tecnologias que estejam alinhadas com os objetivos do novo editor. Após essa especificação, foi feito o desenvolvimento deste novo editor.

Por fim, foi realizada uma validação parcial pois, conforme relatado no artigo, existem diversas outras possibilidades e rodadas de validação que devem ser realizadas, com mais usuários, buscando novos ajustes necessários antes de coloca-lo efetivamente em salas de aula. Esta validação deve ocorrer ainda este ano, pois o objetivo é que a ferramenta fique disponível ao uso de alunos e professores no início do próximo semestre.

Como um dos resultados da validação realizada, obteve-se uma quantidade valiosa de melhorias que estão sendo realizadas no editor.

Conforme relatado pelos participantes da pesquisa, para redes complexas o editor apresentou lentidão, principalmente pelo fato de que, para mostrar todos os resultados em tempo real, diversas inferências precisam ser feitas na mesma rede. O algoritmo *junction tree* é mais indicado para essa situação, pois é dividido em duas partes: a compilação e a inferência. A inferência na *junction tree* é bastante eficiente, e a mesma *junction tree* construída para uma rede pode ser reutilizada para todas as inferências nela (BEAL, 2003). Desta forma, o estudo e implementação deste algoritmo na biblioteca de inferência está sendo realizado.

Uma funcionalidade que facilitaria a migração de usuários de outros editores de redes bayesianas para o editor desenvolvido neste trabalho seria a importação e exportação de redes entre os editores. Desta forma, caso o usuário possua redes já salvas, não seria preciso recriá-las. Para isso, um trabalho futuro para estudo dos formatos utilizados por estes editores e implementação dessas funcionalidades fica definido.

Conforme relatado no Capítulo 1, redes bayesianas podem ser construídas manualmente ou automaticamente a partir de dados. Este trabalho focou na construção manualmente por meio de um software editor. No futuro, poderia ser incluída a funcionalidade de gerar uma rede automaticamente a partir de alguma fonte de dados externa, com a possibilidade de o especialista fazer ajustes manuais após essa geração utilizando o editor desenvolvido.

Por fim, foi desenvolvido um editor de redes bayesianas que pode ser utilizado para criar e observar o resultado de redes para o processo de ensino e aprendizagem do raciocínio clínico

para qualquer área da saúde. Também foi desenvolvida uma biblioteca para realizar a inferência em redes bayesianas que pode ser utilizada nos mais diversos softwares. Ambos os projetos foram desenvolvidos com código aberto e podem servir como referência para outros trabalhos.

Referências

- BEAL, M. J. *Variational algorithms for approximate Bayesian inference*. London: University of London, 2003.
- HELWANGER, F. A. Um editor de redes bayesianas com foco em usabilidade. Novo Hamburgo: Universidade Feevale. Trabalho de Conclusão do Curso de Ciência da Computação. 2016.
- HUGIN. *Company Information*. Disponível em: <<http://www.hugin.com/company-information>>. Acesso em: agosto de 2017.
- MADSEN, Anders L. et al. The Hugin tool for learning Bayesian networks. In: *Symbolic and quantitative approaches to reasoning with uncertainty*. Springer Berlin Heidelberg, 2003. p. 594-605.
- MARONI, V. Construção de um motor de inferência para análise de desempenho em ambientes virtuais de aprendizagem aplicados ao ensino da medicina de família e comunidade. 2013. 103 f. Dissertação (Mestrado em Ciências da Saúde) – Universidade Federal de Ciências da Saúde de Porto Alegre, Porto Alegre, RS, 2013.
- MARQUES, R. L.; DUTRA, I. Redes Bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. Coppe Sistemas – Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2002.
- NETICA. Netica Application. Disponível em: <<https://www.norsys.com/netica.html>>. Acesso em: julho de 2017.
- NILSSON, N. J. *Artificial intelligence: a new synthesis*. San Francisco: Morgan Kaufmann, 1998. 513 p.
- RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A modern approach*. 3. ed. New Jersey: Prentice-Hall. 1995. 1132 p.
- SANTOS, F. S. Desenvolvimento de módulo de redes bayesianas para o amplia. 2009. 70 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2009.
- SEIXAS, L. J.; FLORES, C. D.; VICARI, R. M.; LADEIRA, M. *An architecture for an intelligent learning environment with a constructivist approach*. ITS2002. San Sebastian, Spain. 2002.
- UNBBAYES. *UnBBayes Overview*. Disponível em: <<http://unbbayes.sourceforge.net/index.html>>. Acesso em: julho de 2017.
- WIEGERINCK, W.; KAPPEN, B.; BURGERS, W. *Bayesian networks for expert systems: Theory and practical applications*. In: *Interactive collaborative information systems*. Springer Berlin Heidelberg, 2010. p. 547-578.

Recebido em agosto de 2017
Aprovado para publicação em novembro 2017.

Marta Rosecler Bez

Professora na Universidade Feevale – Novo Hamburgo. Doutora pelo Programa de Pós-Graduação em Informática da Educação – Universidade Federal do Rio Grande do Sul – UFRGS, Brasil, martabez@gmail.com.

Fernando Rafael Stahnke

Bacharel em Ciência da Computação e Mestre em Indústria Criativa pela Universidade Feevale, Brasil, fsthank@gmail.com.

Paulo Ricardo Muniz Barros

Professor na Universidade Feevale – Novo Hamburgo. Mestre pela Universidade Federal de Ciências da Saúde de Porto Alegre e doutorando em Computação Aplicada pela Unisinos. pbarros1979@gmail.com.

Fernando Alex Healwanger

Bacharel em Ciência da Computação pela Universidade Feevale, Brasil, fhelwanger@gmail.com.