

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**HEPATIC: UM PROTÓTIPO DE APOIO À DESCOBERTA DE**  
**FATORES QUE AUMENTAM O OBITUÁRIO DE PACIENTES**  
**HEPÁTICOS TRANSPLANTADOS**

**CAUÃ SCHERER**

**BLUMENAU**  
**2018**

**CAUÃ SCHERER**

**HEPATIC: UM PROTÓTIPO DE APOIO À DESCOBERTA DE  
FATORES QUE AUMENTAM O OBITUÁRIO DE PACIENTES  
HEPÁTICOS TRANSPLANTADOS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU  
2018**

**HEPATIC: UM PROTÓTIPO DE APOIO À DESCOBERTA DE  
FATORES QUE AUMENTAM O OBITUÁRIO DE PACIENTES  
HEPÁTICOS TRANSPLANTADOS**

Por

**CAUÃ SCHERER**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, Mestre – FURB

Blumenau, 17 de julho de 2018

Dedico este trabalho primeiramente a minha família, que sempre com muito amor, ensinaram-me a ser um humano de princípios e valores. Dedico também aos futuros pacientes transplantados, e espero com toda esperança, que trabalhos deste tipo lhes permitam possuir uma sobrevida feliz ao lado da família, amigos ou companheiros.

## **AGRADECIMENTOS**

Aos meus pais Iara Maria Scherer e Almir Rogério Scherer, e minhas irmãs Larissa Scherer e Lisara Scherer, por todos os ensinamentos e direcionamentos da minha vida.

Ao meu orientador Aurélio, pela seriedade e comprometimento na orientação deste trabalho.

A especialista Maíra Silva de Godoy e sua assistente Jane, pela disponibilidade, auxílio e atenção para a construção deste trabalho.

Ao meu amigo Lucas Gomes, pelo auxílio no desenvolvimento.

Aos meus amigos Diego Garcia e Marcio Ropelato, pelos conhecimentos técnicos e forte incentivo à conclusão deste trabalho.

Aos meus gestores Ricardo Lepinski e Fabio Campigotto, pela compreensão nos meus momentos de ausência para conclusão neste trabalho.

Aos meus amigos que compreenderam minha ausência durante esse período, mas sempre estiveram comigo.

E a todos que participaram de alguma forma direta ou indiretamente no desenvolvimento deste trabalho.

O primeiro pecado da humanidade foi a fé, a primeira virtude foi a dúvida.

Carl Sagan

## RESUMO

Este trabalho apresenta a construção de um protótipo para extração de informação de dados dos pacientes transplantados de fígado, e com isso, analisar padrões que influenciam no obituário desses pacientes. Com o auxílio de profissionais da saúde, analisou-se os processos do centro de transplantes do Hospital Santa Isabel e uma amostra de dados de 158 pacientes transplantados. A partir da amostra realizou-se uma correlação manual de fatores que contribuem para o obituário de pacientes. Para isso, foram agrupados dados relevantes dos pacientes e divididos em dois grupos para correlação: grupo do óbito e da sobrevivência. Diante dessa correlação, analisou-se possíveis motivos para o obituário dos pacientes. Validou-se esta análise com a médica Maíra Godoy, responsável pelo centro de transplantes do Hospital Santa Isabel. Neste processo, descartou-se algumas destas análises e adotou-se as restantes. Definiu-se a criação de um protótipo que permite a moldagem de correlações a partir de uma base de dados importada, resultando no desenvolvimento do Hepatic, um protótipo Web que permite a correlação direcionada aos obituários de pacientes e aos demais dados técnicos e relevantes do centro de transplantes. Confrontaram-se os resultados gerados a partir do protótipo em relação ao levantamento manual de causas do obituário, anteriormente validado pelos profissionais da saúde. Obteve-se o resultado de 97% de acerto das informações do Hepatic sobre o levantamento manual. Conclui-se que o trabalho atingiu aos objetivos propostos considerando que o protótipo permite a análise de causas que contribuem para a taxa de mortalidade de pacientes transplantados.

Palavras-chave: Transplante hepático. Correlação. Óbito.

## **ABSTRACT**

This work presents the construction of a prototype for extraction of data from patients with liver transplants, and with that, analyze patterns that influence the obituary of these patients. With the assistance of health professionals, we analyzed the procedures of the Santa Isabel Hospital transplant center and a data sample of 158 transplant patients. From the sample a manual correlation of factors contributing to the obituary of patients was performed. For this, relevant data were grouped into patients and divided into two groups for correlation: death and survival groups. Faced with this correlation, possible reasons for the obituary of the patients were analyzed. This analysis was validated with Maíra Godoy, a physician responsible for the Santa Isabel Hospital transplant center. In this process, some of these analyzes were discarded and the rest were accepted. We defined the creation of a prototype that allows the molding of correlations from an imported database, resulting in the development of Hepatic, a Web prototype that allows the correlation directed to patient obituaries and other relevant technical data of the center of transplants. We compared the results generated from the prototype in relation to the manual obituary causes, previously validated by health professionals. The result of 97% correctness of the Hepatic information on the manual survey was obtained. It is concluded that the work reached the proposed objectives considering that the prototype allows the analysis of causes that contribute to the mortality rate of transplanted patients.

**Keywords:** Liver transplantation. Correlation. Death.



## LISTA DE FIGURAS

Figura 1 – Degradação anatômica do fígado .....	17
Figura 2 – Fórmula do coeficiente de r-de-Pearson .....	18
Figura 3 – Correlação linear perfeita entre as variáveis X e Y.....	19
Figura 4 – Correlação nula entre as variáveis X e Y .....	19
Figura 5 – Cálculo do índice MELD .....	21
Figura 6 – Atuação dos agentes na aplicação .....	21
Figura 7 – Fases mineração de dados .....	23
Figura 8 – Arquitetura básica dos sistemas especialistas .....	24
Figura 9 – Arquitetura geral da aplicação .....	25
Figura 10 – Controle atual dos pacientes.....	26
Figura 11 – Dados clínicos pré-cirúrgico .....	27
Figura 12 – Informações incompletas .....	27
Figura 13 – Informações sobre os doadores e os fígados transplantados.....	28
Figura 14 – Diagrama de casos de uso .....	30
Figura 15 – Diagrama de classes .....	32
Figura 16 – Tela inicial do Hepatic .....	34
Figura 17 – Digrama de objetos importação de dados .....	36
Figura 18 – Tela inicial após a importação.....	41
Figura 19 – Atributos de correlação para manipulação dos dados .....	41
Figura 20 – Opção de acesso para as propriedades .....	42
Figura 21 – Opções de filtros .....	42
Figura 22 – Filtros por distinção.....	43
Figura 23 – Filtros por informação.....	43
Figura 24 – Filtros por intervalo de data .....	44
Figura 25 – Filtros por intervalo numérico.....	44
Figura 26 – Definição atributo divisor de dados .....	47
Figura 27 – Grupos por distinção .....	48
Figura 28 – Divisão por informação.....	48
Figura 29 – Adicionando atributo confrontador .....	49
Figura 30 – Seleção de dados por diagnóstico do paciente .....	50
Figura 31 – Layout dos filtros na correlação.....	50

Figura 32 – Tabela de correlação .....	51
Figura 33 – Adição de atributos para correlação .....	52
Figura 34 – Resultado da execução da consulta ao BD.....	55
Figura 35 – Diagrama de objetos árvore de correlação .....	56
Figura 36 – Montagem estrutura árvore de correlação.....	56
Figura 37 – Layout gráfico de barras.....	58
Figura 38 – Forma de visualização a partir da correlação .....	58
Figura 39 – Valores apresentados no gráfico .....	59
Figura 40 – Gráfico comparativo de acerto do Hepatic.....	60

## LISTA DE QUADROS

Quadro 1 – Requisitos funcionais do protótipo .....	29
Quadro 2 – Requisitos não funcionais do protótipo .....	29
Quadro 3 – Estrutura da planilha para importação .....	34
Quadro 4 – Transmissão dos dados de importação .....	34
Quadro 5 – Estrutura das colunas no servidor .....	35
Quadro 6 – Processo de transformação das informações .....	36
Quadro 7 – Criação de objetos Column e definição de estrutura .....	37
Quadro 8 – Obtendo o tipo da coluna.....	38
Quadro 9 – Conversão dos títulos em rótulos para criação da tabela no BD .....	39
Quadro 10 – Demonstração de conversão .....	39
Quadro 11 – Persistência no banco de dados .....	40
Quadro 12 – Estrutura dos atributos de correlações .....	45
Quadro 13 – Obtenção dos atributos de correlação.....	46
Quadro 14 – Estrutura dos atributos e propriedades no servidor .....	52
Quadro 15 – Montagem dinâmica do código SQL.....	53
Quadro 16 – Definição restrição SQL por grupo .....	54
Quadro 17 – Código de consulta ao BD .....	54
Quadro 18 – Estrutura da classe OutputColumn .....	55
Quadro 19 – Código-fonte árvore de correlação .....	57

## LISTA DE TABELAS

Tabela 1 – Resultados levantamento manual <i>versus</i> Hepatic.....	60
Tabela 2 – Correlação estatístico dos pacientes .....	62
Tabela 3 – Correlação dos dados dos doadores e pacientes .....	62

## **LISTA DE ABREVIATURAS E SIGLAS**

BD - Banco de Dados

CRUD - Create, Retrive, Update and Delete

EUA - Estados Unidos da América

IDE - Integrated Development Environment

KDD - Knowledge Discovery in Databases

MELD - Model for End-Stage Liver Disease

RBT - Registro Brasileiro de Transplantes

RC - Rejeição Celular

RF - Requisitos Funcionais

RNF - Requisitos Não Funcionais

SQL - Structured Query Language

TX - Transplante

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA DO TRABALHO .....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>16</b>
2.1 TRANSPLANTE HEPÁTICO .....	16
2.2 CORRELAÇÃO DE DADOS .....	18
2.3 TRABALHOS CORRELATOS .....	20
2.3.1 Modelo de simulação de fila de transplantes de fígado baseado em sistemas multiagentes .....	20
2.3.2 Aplicação de técnicas de mineração de dados ao desenvolvimento de um sistema de apoio à decisão clínica para acompanhamento do pós-transplante renal .....	22
2.3.3 Protótipo de um sistema especialista educativo para diagnóstico médico .....	23
<b>3 DESENVOLVIMENTO .....</b>	<b>26</b>
3.1 ANÁLISE DO PROCESSO ATUAL .....	26
3.2 REQUISITOS .....	29
3.3 ESPECIFICAÇÃO .....	30
3.3.1 Diagrama de casos de uso .....	30
3.3.2 Diagrama de classes .....	31
3.4 IMPLEMENTAÇÃO .....	33
3.4.1 Importação dos dados.....	33
3.4.2 Atributos de correlação .....	41
3.4.3 Tabela comparativa .....	47
3.5 ANÁLISE DOS RESULTADOS .....	59
3.5.1 Resultados da correlação dos dados .....	59
3.5.2 Resultados e discussões gerais .....	61
<b>4 CONCLUSÕES .....</b>	<b>65</b>
4.1 EXTENSÕES .....	66
<b>REFERÊNCIAS .....</b>	<b>67</b>

## 1 INTRODUÇÃO

Cita-se o transplante de órgãos como uns dos passos primordiais para o avanço medicinal (MARINHO, 2006). Os responsáveis por iniciarem as técnicas de transplante de órgãos foram Carrel e Ullmann, no ano de 1922. Porém, somente na década de 1960 direcionou-se por Thomas Starzl as linhas de pesquisas para o transplante hepático (PACHECO et al., 2002). Primeiramente, Thomas aperfeiçoou as técnicas em cães, porém em 1963 realizou-se o primeiro transplante hepático em um ser humano. O paciente era uma criança de 3 anos, que morreu no período pós-operatório devido a uma hemorragia (FERREIRA; VIEIRA; SILVEIRA, 2000).

Segundo Fonseca e Tavares (2015, p. 2), no Brasil, “o número de transplantes [...] vem crescendo de forma contínua. Onde, em 2007 eram 6,2 doadores por milhão de habitantes e em 2014 atingiu 12,8 doadores efetivos”. No Registro Brasileiro de Transplantes (RBT) tem-se que no ano de 2017, 2109 transplantes do fígado foram realizados no Brasil, sendo que no estado de Santa Catarina, realizou-se no total de 127 transplantes hepáticos (GARCIA, 2017). Segundo a RBT, só no Hospital Santa Isabel, localizado no município de Blumenau, realizaram-se no total de 1175 transplantes, sendo que em 108 destes transplantes, os pacientes vieram à óbito, representando um índice de 9,19% de mortalidade.

Nunes e Moreira (2007) consideram que os transplantes de fígado são necessários, pois o órgão realiza importantes funções vitais essenciais para manutenção da homeostasia corporal<sup>1</sup>. Para Pinto (2016), as causas mais frequentes de doença hepática estão relacionadas ao consumo de álcool (cirrose de Laennec), distúrbios colestáticos (congenitos ou adquiridos) e hepatite viral<sup>35</sup>.

Entretanto, Fonseca e Tavares (2015) também ressaltam que complicações técnicas (na ferida operatória, hemorragias, vasculares, biliares, nutricionais e gastrintestinais), clínicas (imunológicas, infecciosas, pulmonares, cardiovasculares, renais, neurológicas e metabólicas) ou até mesmo uma prescrição médica errada podem influenciar negativamente na sobrevivência dos pacientes. Ainda segundo o autor, pode-se encontrar a partir do confronto ou correlação dessas informações, associações que influenciam para a taxa de quase 10% de mortalidade em pacientes transplantados de fígado.

Ferreira, Vieira e Silveira (2000) destacam que, após a alta hospitalar, os pacientes devem realizar uma checagem frequente. Mesmo assim, no Brasil os índices de sobrevivência

---

<sup>1</sup> Homeostase corporal é o processo pelo qual o organismo mantém seu equilíbrio, e conseqüentemente sua saúde (SENS, 1998).

têm sido baixos sendo que em média, 65% dos pacientes transplantados estarão vivos após um ano de transplante e, apenas 61,30% após 5 anos (MIGUEL et al., 2007).

Diante deste cenário, este trabalho apresenta o desenvolvimento de um protótipo de software que realize o processo de organização e análise dos dados a partir de históricos clínicos, tendo como objetivo correlacionar fatores que podem influenciar no óbito de pacientes hepáticos transplantados.

## 1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um protótipo de software que organize e analise os dados a partir de históricos clínicos para descobrir fatores que podem influenciar ou contribuir com o obituário de pacientes hepáticos transplantados.

Os objetivos específicos são:

- a) importar os dados históricos dos transplantados;
- b) modelar e validar regras de correlação a partir de uma base de dados exemplo, tendo a ajuda de um especialista da área médica;
- c) disponibilizar uma aplicação web para visualização das principais causas mortis e suas relações técnicas ou clínicas.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta os objetivos e a motivação para desenvolvimento do trabalho. O segundo capítulo trata da fundamentação teórica do trabalho, explicando os principais conceitos e técnicas utilizadas no desenvolvimento do protótipo. No terceiro capítulo são descritos a arquitetura do trabalho através de diagramas, o detalhamento da implementação do protótipo e os resultados obtidos nos testes realizados. Por fim, são apresentadas as conclusões e limitações do trabalho, assim como sugestões para trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo explorar os principais assuntos necessários para a realização e entendimento do trabalho. Desta forma, o capítulo foi subdividido em três partes, onde a seção 2.1 conceitua transplante hepático. Já a seção 2.2 aborda correlação de dados. Por fim, na seção 2.3 são descritos três trabalhos correlatos.

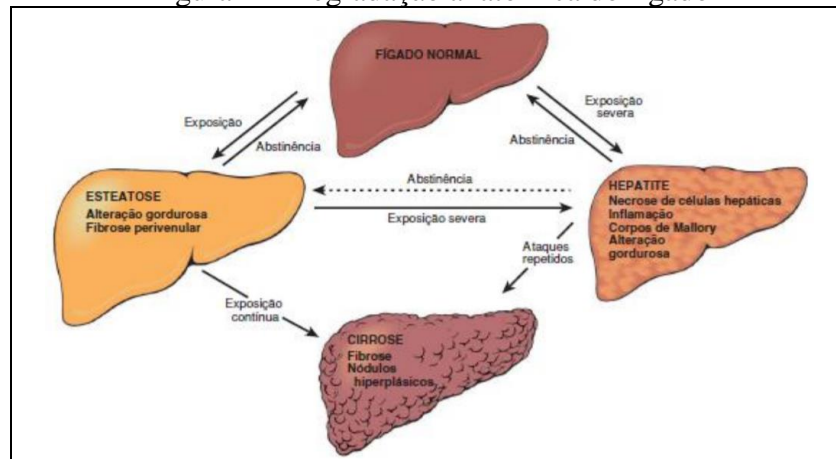
### 2.1 TRANSPLANTE HEPÁTICO

O fígado é o maior órgão do corpo humano, possuindo um peso médio de 1500g, representando 2,5 a 4,5% da massa corporal (NUNES; MOREIRA, 2007). O fígado situa-se no sistema circulatório (quadrante superior direito da cavidade abdominal, no domo do diafragma) e possui fornecimento sanguíneo duplo: cerca de 20% de sua fluência é rica em O<sub>2</sub> e provém da artéria hepática, já os demais 80% são ricos em nutrientes e provém da veia porta.

Segundo Nunes e Moreira (2007), o funcionamento incorreto do fígado pode gerar fadiga, náuseas, vômitos, mal-estar, anorexia, dor, hemorragia intestinal, entre outros. Esses sintomas não são específicos, como a fadiga, por exemplo, que pode ser atribuída a outras situações como ansiedade ou distúrbio do sono, portanto, uma junção desses sintomas é necessária para a detecção da doença hepática.

Insuficiências hepáticas causadas por lesões agudas, maciças ou crônicas podem ser consideradas como as mais severas. Para um fígado ser considerado insuficiente, basta 80% a 90% de seu parênquima (célula) estar destruído. Uma doença comum é a cirrose hepática, que é considerada 12<sup>a</sup> causa de morte mais comum nos Estados Unidos da América (EUA). A cirrose pode ser causada por abuso de álcool e hepatites virais (VIEIRA et al, [s.d.]). A Figura 1 apresenta a evolução da degradação de um fígado saudável a partir das doenças esteatose e hepatite, finalizando seu estágio na cirrose.

Figura 1 – Degradação anatômica do fígado



Fonte: Vieira et al. ([s.d.]).

Para avaliação de uma doença hepática, é necessário analisar o histórico clínico e exames físicos. No histórico clínico é observado os fatores que podem ter levado à doença hepática, como por exemplo, o consumo de álcool, drogas ou medicamentos, contatos sexuais de riscos, histórico familiar de doença hepática, diabetes, consumo exagerado de bebidas entre outros (NUNES; MOREIRA, 2007). Caso seja constatado insuficiência hepática crônica terminal cuja expectativa de vida seja inferior a 20% ao final de 12 meses, um transplante é necessário (PACHECO et al., 2002).

O transplante somente é realizado quando os demais tratamentos clínicos já foram aplicados, as exceções são: casos de infecções, doenças cardiovasculares ou pulmonares pré-existentes, metástases de tumores malignos e hipotensão arterial (MIGUEL et al., 2007). Apesar disso, um transplante não é algo que depende somente do paciente com doença hepática. A falta de doadores é algo crítico para realização do procedimento (MIES, 1998).

Após a realização do transplante de fígado, as células brancas (leucócitos) identificam o novo órgão. A partir disso, ocorre a multiplicação dos leucócitos e ataques ao fígado devido ao corpo estranho detectado no organismo. O processo de resposta imunológica é natural do corpo humano e nos protege contra vírus, bactérias e outros microrganismos (MIGUEL et al., 2007).

Depois de três semanas do transplante, as complicações que podem surgir são: efeitos colaterais da imunossupressão; infecções virais; estenoses biliares e trombozes tardias; entre outros. Ao longo prazo, a idade (crianças menores de 1 ano); estado nutricional e a severidade da doença no pré-operatório são fatores que afetam diretamente na sobrevivência dos pacientes (FERREIRA; VIEIRA; SILVEIRA, 2000).

## 2.2 CORRELAÇÃO DE DADOS

Utiliza-se a correlação quando há necessidade de verificar o grau de associação entre duas ou mais variáveis (FEIJOO, 2010, p. 70). Define-se o termo associação como a medição do relacionamento entre as variáveis quantitativas e/ou qualitativas. Variáveis qualitativas originam-se de um adjetivo, como por exemplo, cor, raça, sexo. Já as variáveis quantitativas definem-se a partir de valores mensuráveis, como por exemplo, valores numéricos, monetários, notas de uma avaliação, entre outros (MIRANDA, 2008).

Na correlação, utiliza-se diversas formas para medir a associação entre múltiplas variáveis, considera-se como o caso trivial a medição entre duas variáveis, X e Y (LIRA, 2004). Pode-se medir a dependência entre as variáveis X e Y utilizando o coeficiente amostral, conhecido também pelo coeficiente de r-de-Pearson. A Figura 2 exhibe a designação do coeficiente r, onde apresenta-se associação entre X e Y quando o resultado de  $r = 1$ . Também há correlação quando o resultado de  $r = -1$  (este chamado de correlação inversa ou negativa). Caso o valor de r vier a ser 0, não há correlação linear entre as duas variáveis (REGRA, [s.d.]).

Figura 2 – Fórmula do coeficiente de r-de-Pearson

$$r = \frac{\text{Cov}(X, Y)}{s_X \cdot s_Y}$$

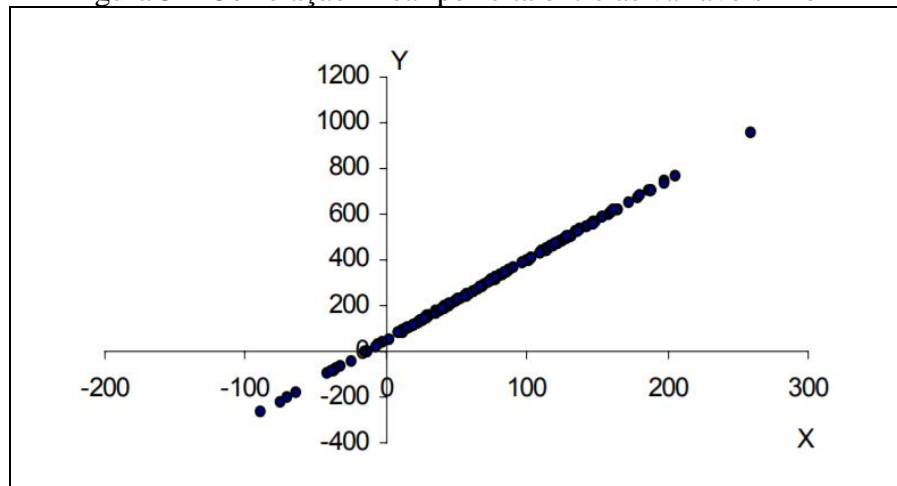
$$\text{Onde } \text{Cov}(X, Y) = \frac{\sum_{i=1}^n XY}{n} - \bar{X} \bar{Y} ,$$

$$s_X = \sqrt{\frac{\sum_{i=1}^k f_i (X_i - \bar{X})^2}{n}} \text{ e } s_Y = \sqrt{\frac{\sum_{i=1}^k f_i (Y_i - \bar{Y})^2}{n}}$$

Fonte: Regra ([s.d.]).

Simplifica-se a visualização de correlação linear entre as variáveis X e Y a partir do diagrama de dispersão (MIRANDA, 2008). Para montá-lo, necessita-se inserir como pontos as variáveis X e Y num plano cartesiano. A partir disso, observa-se visualmente a relação linear entre as duas variáveis. Caso haja correlação linear entre as variáveis, os pontos dispõem-se no plano cartesiano em forma de uma reta (REGRA, [s.d.]). A Figura 3 apresenta um exemplo do diagrama de dispersão, nota-se que há correlação linear perfeita entre as variáveis, devido aos pontos do plano cartesiano formarem uma reta.

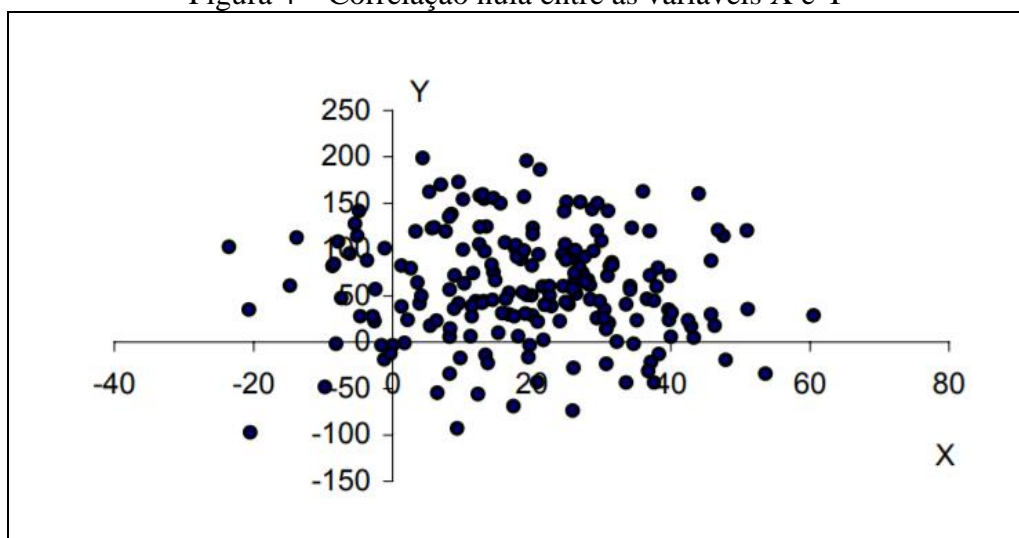
Figura 3 – Correlação linear perfeita entre as variáveis X e Y



Fonte: Lira (2004).

Para situações contrárias, apresenta-se a fraca correlação linear entre as variáveis devido aos pontos se apresentarem de forma dispersa no gráfico, conforme pode ser visto na Figura 4.

Figura 4 – Correlação nula entre as variáveis X e Y



Fonte: Lira (2004).

Para estudos de relações ou associações entre as variáveis, utiliza-se o método denominado de Análise de Correlação. Esta análise proporciona a visualização do grau de relacionamento entre as variáveis e como elas variam de forma conjunta. Dentre as técnicas deste estudo estão: Análise de Confiabilidade, a Análise da Estrutura de Variância-Covariância e o Teste de Normalidade ou Gaussianidade (LIRA, 2004). Entretanto, segundo Lira (2004) neste método não há necessidade de verificação das dependências e independências entre essas variáveis, somente o comportamento associativo entre elas.

## 2.3 TRABALHOS CORRELATOS

Neste capítulo são descritos trabalhos que se assemelham aos objetivos propostos. Foram encontrados 3 trabalhos que servem como base de conhecimento e fundamentação para criação do protótipo de rastreabilidade de transplantes hepáticos. A seção 2.3.1 descreve o modelo de simulação de fila de transplantes de fígado baseado em sistemas multiagentes desenvolvido por Oliveira (2010). Na seção 2.3.2 Hummel (2011) criou uma aplicação utilizando as técnicas de mineração de dados para auxiliar na decisão clínica de pacientes pós-transplantados do rim. Por fim, Santos (2010) utilizou como técnica um sistema especialista, onde o autor criou uma ferramenta educativa que auxilia estudantes da área de medicina em diagnósticos médicos.

### 2.3.1 Modelo de simulação de fila de transplantes de fígado baseado em sistemas multiagentes

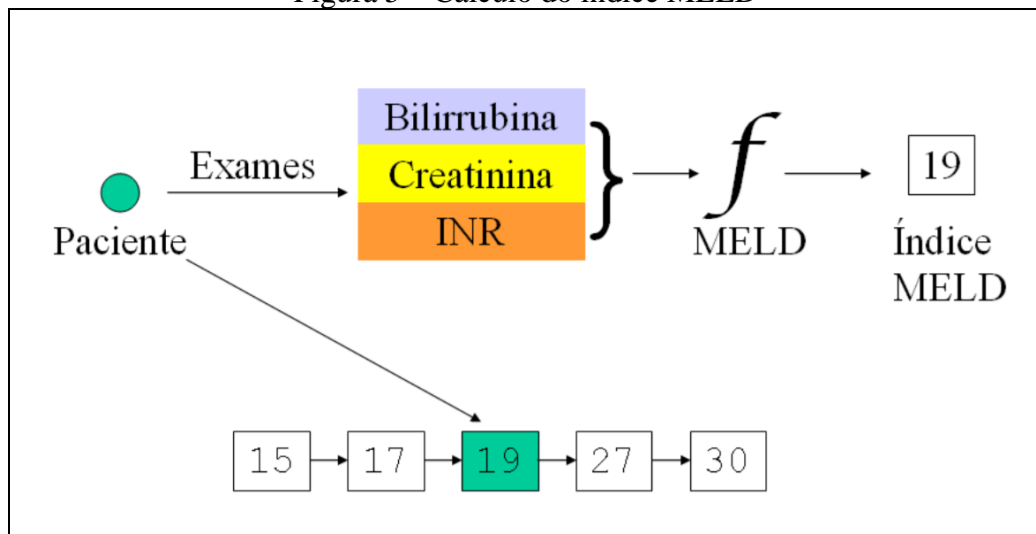
Oliveira (2010) afirma que os sistemas de simulação são importantes para auxiliar nas tomadas de decisões, pois seus eventos são reproduzidos com baixo custo, alta segurança e flexibilidade. Para Oliveira (2010), esses fatores também influenciaram a área médica, onde tem-se a necessidade de sistemas de simulação para fila de espera de pacientes. Diante disso, Oliveira (2016) propôs um modelo de simulação para fila de espera de pacientes utilizando o paradigma baseado em agentes. O autor ressalta que no Brasil, o Modelo para Doença Hepática Terminal (MELD) é utilizado para organizar e definir a prioridade na fila de transplantes de fígado.

Oliveira (2010) também destaca que o MELD é um modelo matemático que utiliza os resultados dos exames realizados pelos pacientes (bilirrubina, creatinina, INR<sup>2</sup>) para definir a prioridade/ordem do paciente na fila de transplante. A Figura 5 demonstra o processo de atribuição do índice MELD ao paciente para alocação na fila de espera.

---

<sup>2</sup> INR – Conhecido também como protrombina, é uma proteína que auxilia no processo da coagulação. Sua síntese ocorre no fígado (OLIVEIRA, 2010).

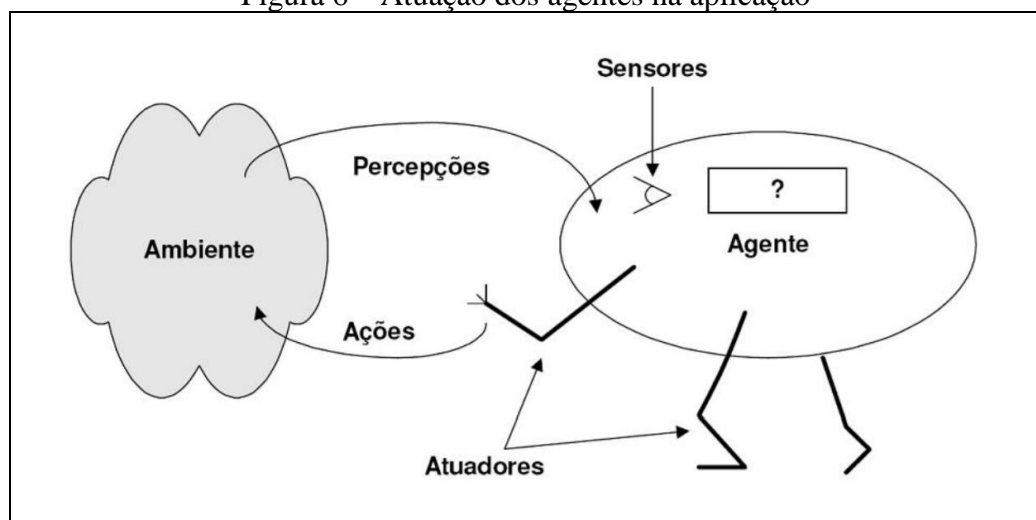
Figura 5 – Cálculo do índice MELD



Fonte: Oliveira (2010).

Oliveira (2010) explica que o sistema baseado em agentes modela cada agente como um indivíduo com racionalidade, tendo capacidade para a realização de tarefas. Já os sistemas multiagentes são caracterizados por interações contínuas desses agentes, podendo emergir um padrão como resultado dessa atuação. Segundo Oliveira (2010), os agentes na aplicação “representam individualmente, cada paciente receptor e a doença que o acomete, cada fígado doado, o sistema de filas e as interações entre eles”. O autor aplicou a ferramenta Repast para realizar a simulação baseada em agentes. A Figura 6 demonstra a arquitetura dos agentes e a tomada de decisões através de suas percepções.

Figura 6 – Atuação dos agentes na aplicação



Fonte: Oliveira (2010).

Segundo Oliveira (2010), os resultados da simulação ocorreram conforme esperado. O autor também afirma que a aplicação pode auxiliar os gestores na aplicação de recursos disponíveis e na tomada de decisões para diferentes cenários. Ele também destaca que para a área médica, os resultados alcançados podem significar fatores que não contemplam apenas

resultados financeiros, mas que influenciem na qualidade de vida e na quantidade de óbitos. Como extensão, Oliveira (2010) sugere a implantação de agentes capazes de analisar os dados de entrada para definição do melhor procedimento a ser utilizado, possibilitando a reavaliação de sua decisão através da análise dos dados de saída da simulação.

### 2.3.2 Aplicação de técnicas de mineração de dados ao desenvolvimento de um sistema de apoio à decisão clínica para acompanhamento do pós-transplante renal

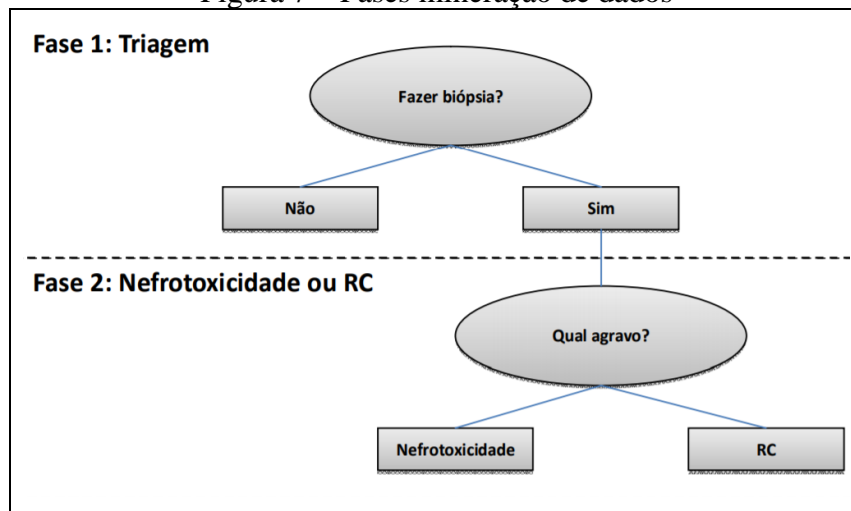
Segundo Hummel (2011), o transplante de órgãos está associado à melhora da qualidade de vida ou sobrevivência do paciente. Porém, o autor pondera que a baixa taxa de doação de órgãos dificulta tais fatores e que a tecnologia da informação pode auxiliar neste processo.

Hummel (2011) destaca que dentro do universo dos transplantes renais, há alguns cuidados que devem-se ter no período pós-transplante tais como a Nefrotoxicidade e a Rejeição Celular (RC). Hummel (2011) também aponta que se esses dois fatores forem diagnosticados de maneira precoce, pode-se evitar a causa mortis do paciente. Diante desta constatação, ele desenvolveu uma aplicação de apoio à decisão para auxiliar os médicos a identificar a presença desses fatores em pacientes pós-transplantados.

Para Hummel (2011), a mineração de dados é uma técnica de extração de conhecimento ou obtenção de informações relevantes a partir de uma base de dados. No desenvolvimento do seu trabalho, ele utilizou-se as ferramentas RapidMiner e Weka. Segundo Hummel (2016), o RapidMiner foi utilizado em testes e pesquisas exploratórias dos métodos de mineração de dados, já para a realização do middleware, foi utilizado o software Weka.

A aplicação de Hummel (2011) foi desenvolvida para as plataformas web e dispositivos móveis. Para realizar a mineração dos dados, Hummel (2011) dividiu o processo em duas etapas: triagem e identificação da nefrotoxicidade e RC, conforme pode ser visto na Figura 7.

Figura 7 – Fases mineração de dados



Fonte: Hummel (2011).

Hummel (2011) utilizou como base de dados 158 prontuários de pacientes com suspeita clínica de RC e/ou nefrotoxicidade do Centro de Transplante da Casa de Saúde Santa Efigênia, em Caruaru-PE. Desses pacientes, o autor incluiu os dados de 102 eventos de acompanhamento (tempo da consulta pós-transplante, aumento de temperatura, tremores, dosagem da ureia, dosagem de creatinina, glicemia, indução, tipo de doador) do período pós-transplante.

A aplicação obteve uma taxa de acerto de 82% na identificação da nefrotoxicidade e 93% na rejeição celular (RC). Hummel (2011) ressalta que com o conhecimento adquirido ao decorrer do trabalho auxiliará na construção de um sistema de monitoramento de pacientes no estágio pós-transplante renal. Como trabalhos futuros, Hummel (2011) sugere a construção de um sistema web que englobe todo o processo de transplante renal, acoplando como módulo este trabalho.

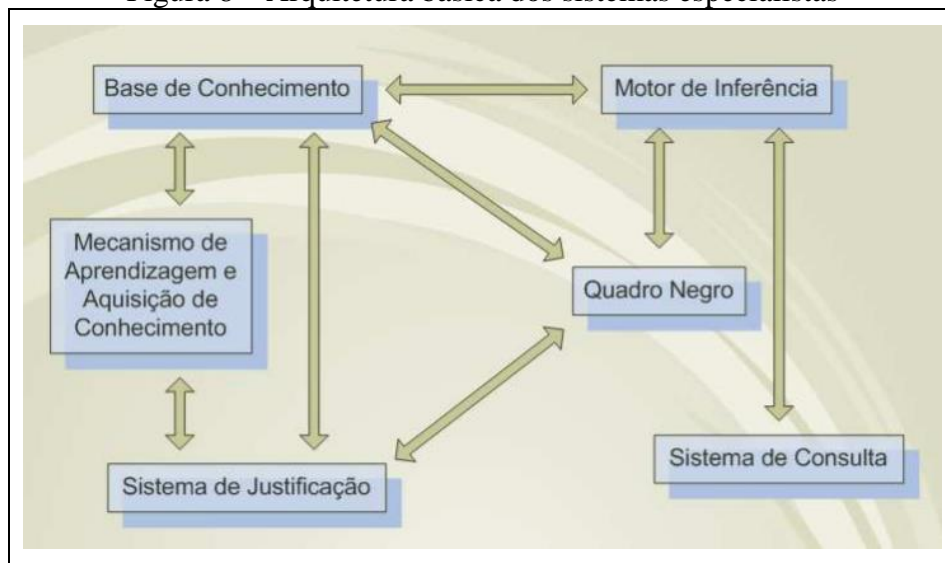
### 2.3.3 Protótipo de um sistema especialista educativo para diagnóstico médico

Segundo Santos (2010), os sistemas especialistas ajudam a solucionar problemas de áreas com conhecimentos específicos e que possuem restrições de mão de obra especializada. Devido a essas restrições, Santos (2010) afirma que os estudantes da área médica podem se deparar com situações delicadas e que necessitam de certo grau de conhecimento para sua execução. Para Santos (2010), o aprimoramento do conhecimento na área médica é delicado pois um diagnóstico falso pode resultar na perda de vida de um paciente. Diante disso, Santos (2010) desenvolveu um protótipo para auxílio de profissionais de educação para capacitação de estudantes na área médica, que auxiliará na ascensão de seus conhecimentos.



Santos (2010) descreve que os sistemas especialistas possuem como característica evidente a centralização do conhecimento especialista e na capacidade autônoma da progressão, ascensão e adaptação do conhecimento. Santos (2010) também afirma que o grande objetivo dos sistemas especialistas é a busca pela autossuficiência. A Figura 8 demonstra de forma simples a arquitetura dos sistemas especialistas, onde o mesmo possui o armazenamento de conclusões (quadro negro) e interpretador e escalonador de regras (motor de inferência).

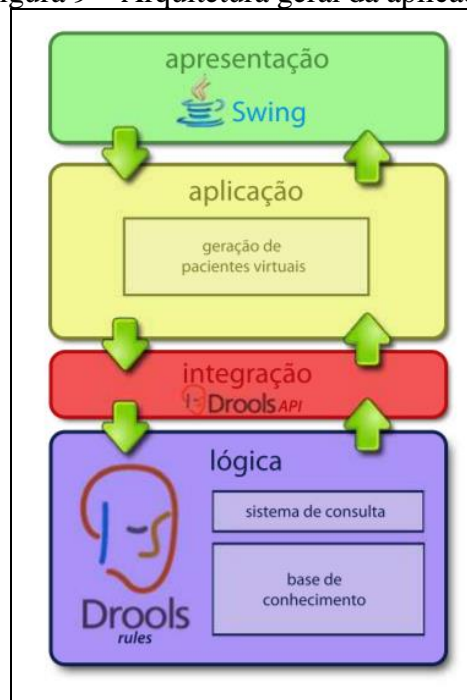
Figura 8 – Arquitetura básica dos sistemas especialistas



Fonte: Santos (2010).

Santos (2010) tinha como objetivo a construção de uma ferramenta educativa que conciliasse dois paradigmas de programação, o lógico e o orientado a objetos. A aplicação lógica de Santos (2010) foi desenvolvida em cima de uma derivação da Inteligência Artificial, onde o funcionamento do sistema especialista funcionaria com o auxílio da ferramenta Drools. Segundo Santos (2010), Drools é um sistema que gerencia regras de negócios e permite a criação de regras para emular conhecimento. Junto com o Java Swing para interface gráfica e a camada de geração de pacientes virtuais para base de conhecimento, na Figura 9, pode-se visualizar a localização das regras da ferramenta Drools.

Figura 9 – Arquitetura geral da aplicação



Fonte: Santos (2010).

Santos (2010) não obteve resultados numéricos, mas concluiu que o desenvolvimento da aplicação foi válido e trouxe grande conhecimento na área de desenvolvimento de sistemas. Dentro os desafios descritos por Santos (2010), estão no desenvolvimento da integração de dois paradigmas de programação (lógico e orientado a objetos). Para projetos posteriores, Santos (2010) recomendou a incrementação do protótipo com o desenvolvimento de um módulo tutor, onde é possível a inserção de novas regras e doenças na base de conhecimento. Também foi recomendado por Santos (2010) o acompanhamento de rendimentos de estudantes utilizando o sistema.

### 3 DESENVOLVIMENTO

Neste capítulo são demonstradas as etapas do desenvolvimento do protótipo, chamado de Hepatic. Na seção 3.1 estão descritas todas as etapas do processo atual realizado manualmente. A seção 3.2 apresenta os principais requisitos do problema. A seção 3.3 apresenta a especificação. Por fim, na seção 3.4 são demonstrados os resultados dos testes, sugestões e melhorias do protótipo.

#### 3.1 ANÁLISE DO PROCESSO ATUAL

Este trabalho foi realizado com o auxílio dos profissionais do centro de transplante de órgãos do hospital Santa Isabel, localizado no município de Blumenau. Para a concretização dos requisitos do protótipo, acompanhou-se os processos e o controle dos pacientes com objetivo de levantar problemas que barram a extração de conhecimentos estatísticos a partir dos registros de eventos dos pacientes do centro de transplantes.

A coleta de dados e eventos dos pacientes do centro de transplantes ocorrem em três etapas: pré-operatório, transplante, pós-operatório. No decorrer do processo do transplante, são registradas informações importantes dos pacientes pelos profissionais do centro de transplantes, sendo que os dados gerais e técnicos diferem a cada etapa. Conforme é apresentado na Figura 10, pode-se notar que uma planilha do tipo .csv é a fonte do armazenamento das informações dos eventos do processo de enxerto, nela é realizado o controle dos pacientes no centro de transplantes.

Figura 10 – Controle atual dos pacientes

PACIENTE	DT INCLUSAO	DT TX	DO ENXERTO DIAS	IDADE	PESO	ALTURA	ABO	RELIGIAO	ESCOLARIDADE	PROFISSAO	SEXO	DIAGNOSTICO
XXX	31/08/17	02/09/17	7	61	72	162	A+	Católica	Ensino Médio	Do Lar	F	Cirrose Hepática VCH + HCC
XXX	16/08/17	01/09/17	2	26	64	160	AB	Católica	Superior	Desempregado	F	Retransplante, Recidiva Hepatite Auto Imune
XXX	22/06/17	19/08/17	290	71	57	165	O+			Funcionário Público	M	Hepatocarcinoma, VHC
XXX	16/08/17	18/08/17	291	38	85	162	O-	Católica	Superior	Médico Veterinário	M	Cirrose Alcoolica
XXX	05/07/17	18/08/17	293	53	91	168	A	Evangelica	Ensino Médio	Operador Empilhadeira	M	Hepatocarcinoma, VHC
XXX	09/08/17	16/08/17	293	67	64	169	A+	Espírita	Ensino Médio	Funcionária Pública	F	Hepatocarcinoma, VHC
XXX	13/06/17	12/08/17		56	136	163	O	Católica	Ensino Fundamental	Não informado	M	Esteato hepatite não alcoolica
XXX	08/08/17	11/08/17	298	26	67	167	O	Católica	Ensino Médio	Aposentada	F	Hepatite Auto Imune

Fonte: elaborado pelo autor.

A primeira etapa ocorre no pré-operatório onde são coletados dados pessoais como: idade, sexo, peso, altura, religião, escolaridade e profissão. Dados clínicos como diagnóstico, índice MELD e tipo sanguíneo também são documentados nessa etapa. Também é visto como importante pelo centro de transplante o armazenamento de eventos clínicos anteriores ao transplante, como doenças, fraturas, infecções, entre outros. Estes são agrupados e documentados como intercorrências antes do transplante (TX), como pode ser visto alguns exemplos na Figura 11.

Figura 11 – Dados clínicos pré-cirúrgico

DIAGNOSTICO	MELD	INTERCORRENCIAS ANTES DO TX
Cirrose Hepática VCH + HCC	8	PO HISTERECTOMIA, HAS
Retransplante, Recidiva Hepatite Auto Imune	37	TX, DIALISE, PSORIASE
Cirrose Alcoólica	31	APENDICECTOMIA, ASCITE, ENCEFALOPATIA
Hepatocarcinoma, VHC	10	QUIMIOEMBOLIZACAO TTA TUMOR, TTO SOFO, DACLATASVIR, RIBAVIRINA
Esteato hepátite não alcoólica	15	TROMBOSE TRONCO DA PORTA, CIRCULACAO COLATERAL ABUNDANTE OBESO
Hepatite Auto Imune	26	DIALISE
Hepatocarcinoma, alcool	12	QUIMIOEMBOLIZACAO TTA TUMOR

Fonte: elaborado pelo autor.

As etapas posteriores são a realização do enxerto e a recuperação pós-cirúrgico. A documentação dos dados referentes aos eventos dentro do centro cirúrgico é realizada após a cirurgia. As principais informações armazenadas são: início cirúrgico, tempo de duração da cirurgia, quantidade de plaquetas e hemácias utilizadas, procedimento de Cell Saver e quantidade de sangramento no procedimento.

O processo de documentação dos eventos cirúrgicos ocorre no sistema do hospital Santa Isabel. No encerramento do transplante, os cirurgiões responsáveis relatam no prontuário do paciente em formato de relatório, os eventos que ocorreram no procedimento. Para controle destes eventos no centro de transplantes, há um processo de retirada de informações do sistema e adequação para o layout da planilha. Como consequência, pode-se visualizar na Figura 12 que muitas informações são perdidas nesse processo, isto pois a documentação em formato de relatório após o procedimento cirúrgico pode dar margens ao esquecimento, que posteriormente afetam levantamentos analíticos e controle dos dados dos pacientes.

Figura 12 – Informações incompletas

CELL SAVER	DURAÇÃO CIRÚRGICA	QUALIDADE PERFUSÃO	SANGRAMENTO NA CIRURGIA	TECNICA
0		3	MINIMO	
2460		3		PIGGBACK
0		3	POUCO	
		3	MODERADO	
0		3	MODERADO	
0		3	MODERADO	
0		3	INTENSO	
736	04:40	3	INTENSO	
0	03:50		NAO	
0			NAO	
0	05:05		INTENSO	
0			NÃO	
0	05:10		MODERADO	

Fonte: elaborado pelo autor.

Durante o período do pós-cirúrgico, são coletados dados de acompanhamento e evolução da saúde do paciente. Caso venha a necessidade de um novo transplante, é documentado a nova data do procedimento, o mesmo procedimento ocorre caso o paciente vir a óbito. Os principais dados desta etapa são: intercorrências pós-transplante, alta hospitalar, infecção, data do retransplante (caso houver), data e motivo do óbito (caso houver).

Estende-se o armazenamento de informações para os doadores do fígado. Dados pessoais como: origem, sexo, idade, altura e peso do doador são documentados. Referente aos dados clínicos, a ênfase das informações é direcionada para o fígado do doador, conforme é apresentado na Figura 13. Como exemplo de informações armazenadas sobre o fígado do doador, temos: TGO, TGP, GGT, BTF, sódio e procedimentos como tempo de isquemia.

Figura 13 – Informações sobre os doadores e os fígados transplantados

CAUSA MORTE DOADOR	SEXO D	IDADE D	ABO D	ALTURA D	PESO D	SODIO D	BTF D	TGO D	TGP D	GGT D
AVC	M	42	A	164	84	141	0,41	51	24	68
TCE	M	20	A	176	71	140	0,23	143	45	10
AVC	F	35	O	168	82	167	0,06	38	49	210
AVC	F	43	O	145	60	150	0,77	61	60	199
HSA	M	42	A	180	75	155	0,6	28	27	43
TCE	M	21	A	176	90	144	1,1	80	59	24
AVC	M	38	O	165	120	151	0,98	71	36	79

Fonte: elaborado pelo autor.

Uma parte dos profissionais que atuam no centro de transplantes do Santa Isabel possuem acesso de visualização e edição dos dados de controle dos pacientes. Referente a isto, pode-se citar como consequência a falta de normalização dos dados de controle dos pacientes. Isso ocorre, pois há redundância nas informações cadastradas na planilha devido à múltiplas edições oriundas de profissionais distintos. Esse processo inviabiliza a extração de informações gerais e controle sobre os pacientes transplantados no centro.

É obrigatório para a continuidade das atividades do centro de transplantes do Santa Isabel o envio de informações estatísticas sobre os pacientes à órgãos do governo. No processo atual, a extração destas informações é manual e demandam tempo. Porém mesmo com os dados completos e normalizados, o tempo de obtenção dessas informações seriam inferiores ao comparados de um processo de correlação e controle de consideráveis quantidades de dados de forma automatizada.

Dados do Registro Brasileiro de Transplantes apontam que o centro de transplantes do Santa Isabel possui um histórico de 1175 de transplantes de fígado realizados, sendo que a partir do total, 108 pacientes vieram a óbito (GARCIA, 2017). No cenário atual de controle

dos pacientes, devido as informações estarem dispersas, ainda um processo com barreiras o levantamento de evidências que fundamentam a taxa de óbito de aproximadamente 10%. Situações como: liberação de orçamentos para compra de equipamentos, aplicações de mudanças nos processos cirúrgicos e possíveis mudanças no transporte dos órgãos, necessitam de comprovação a partir de extração de informações nos históricos dos pacientes. Somente evidências vindas do atual controle de transplantes, podem auxiliar em aplicações de medidas para uma possível diminuição do obituário.

### 3.2 REQUISITOS

Nesta seção são apresentados os Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF) concretizados, conforme Quadro 1 e Quadro 2 respectivamente.

Quadro 1 – Requisitos funcionais do protótipo

Requisitos Funcionais	Casos de Uso
RF01: permitir a importação dos dados e o reconhecimento dos títulos das colunas a partir de arquivos .csv. Caso haja posterior adição ou exclusão de colunas na planilha, ao efetuar uma nova importação, o protótipo deverá reconhecer essas alterações.	UC01; UC02
RF02: apresentar a partir do arquivo importado, os atributos disponíveis para correlação e todos os seus valores de forma distinta.	UC03
RF03: possibilitar a seleção, divisão e filtragem dos dados ao usuário de três maneiras: por intervalo numérico ou de datas, distinção das informações ou por verificação dos dados preenchidos ou vazios.	UC05
RF04: permitir a partir dos atributos apresentados, a construção e especificação da correlação dos dados.	UC04; UC06
RF05: permitir a visualização numérica das correlações em forma de uma tabela.	UC09
RF06: permitir a visualização dos dados correlacionados na forma de gráfico de barras.	UC10

Fonte: elaborado pelo autor.

Quadro 2 – Requisitos não funcionais do protótipo

Requisitos Não Funcionais
RNF1: garantir o balanceamento da árvore de correlação, para que a cada nível da árvore, os dados agrupados referenciem a mesma informação.
RNF2: utilizar frameworks Java Script para a camada de interface e usabilidade do usuário.
RNF3: utilizar BD relacional, com armazenamento em memória RAM.
RNF4: deverá ser utilizado o protocolo Rest para comunicação entre o Front-end e o Back-end.
RNF5: construir o protótipo para permitir posterior execução em servidores na nuvem.

Fonte: elaborado pelo autor.

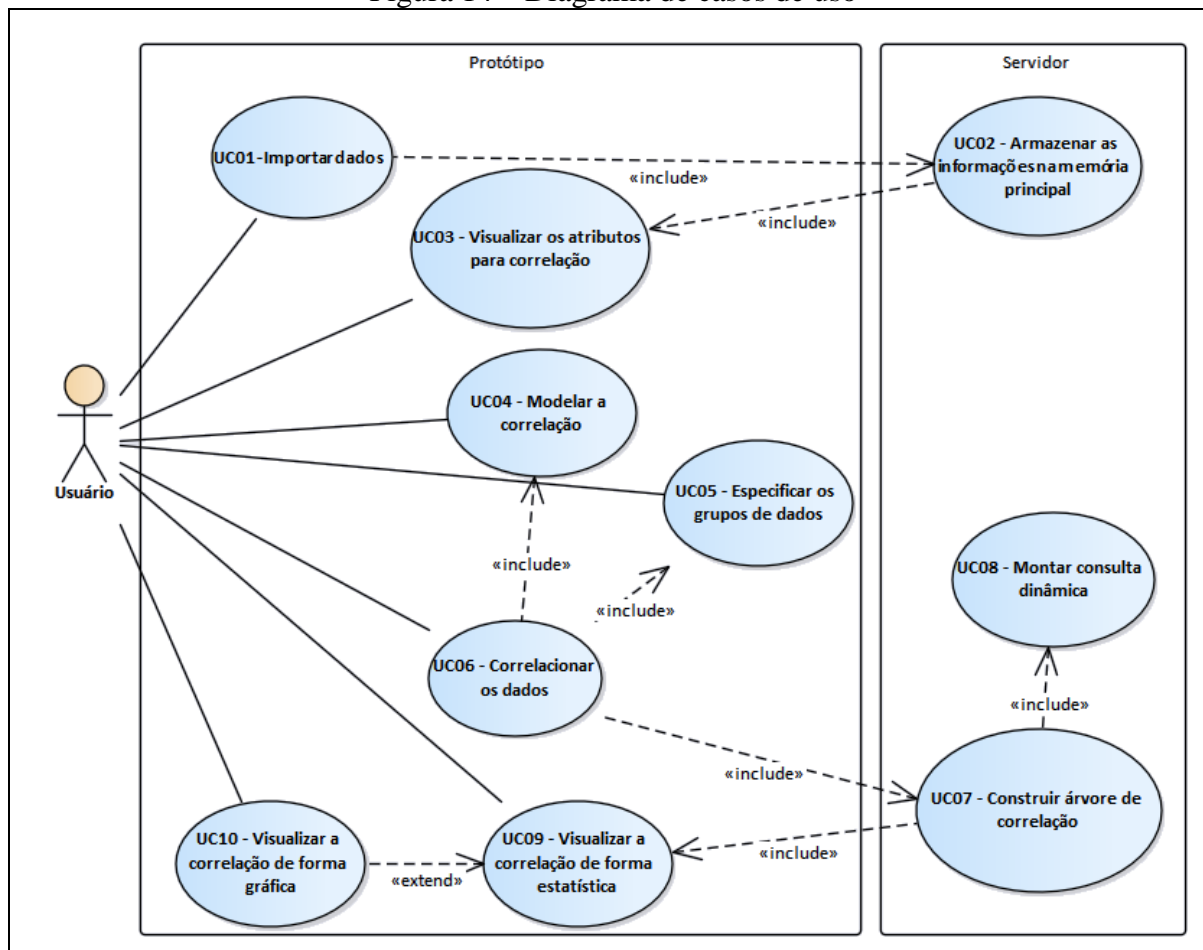
### 3.3 ESPECIFICAÇÃO

Com objetivo de melhor apresentar visualmente a especificação do protótipo, utilizou-se a ferramenta Enterprise Architect. Na seção 3.3.1 encontra-se o diagrama de casos de uso e um breve explicativo de suas relações. Posteriormente, na seção 3.3.2, é possível visualizar a arquitetura física do protótipo a partir do diagrama de classes.

#### 3.3.1 Diagrama de casos de uso

Nesta seção é apresentado o diagrama de casos de uso, conforme ilustrado na Figura 14. Nele, estão representadas as funcionalidades na perspectiva do ator *Usuário*, sendo responsável pela montagem da estrutura de correlação da base dos dados.

Figura 14 – Diagrama de casos de uso



Fonte: elaborado pelo autor.

Na primeira utilização do protótipo, o *Usuário* deverá importar uma planilha do tipo *.csv* contendo os dados e suas colunas tituladas, conforme caso de uso UC01 - Importar dados. O caso de uso UC02 - Armazenar as informações na memória principal demonstra o processo de importação dos dados para tabelas em memória RAM do BD relacional H2. Ao término dessa importação, são apresentados ao *Usuário* os atributos que

permitirão a montagem da estrutura de seleção e correlação dos dados, conforme caso de uso UC02 - Visualizar os atributos para correlação.

O protótipo permite que o usuário realize a divisão dos dados em dois grupos distintos (caso de uso UC05 - Especificar o grupo de dados). A divisão dos grupos permitirá a comparação de informações tais como: dados pessoais, técnicas, eventos, dados do fígado, dados dos doadores, intercorrências, entre outros. Para correlacionar, é necessário que o Usuário modele de sua preferência a estrutura para posterior apresentação da correlação (caso de uso UC04 - Modelar a correlação). Somente ao finalizar a especificação da estrutura dos dados, o sistema habilitará a opção para a ação de correlação.

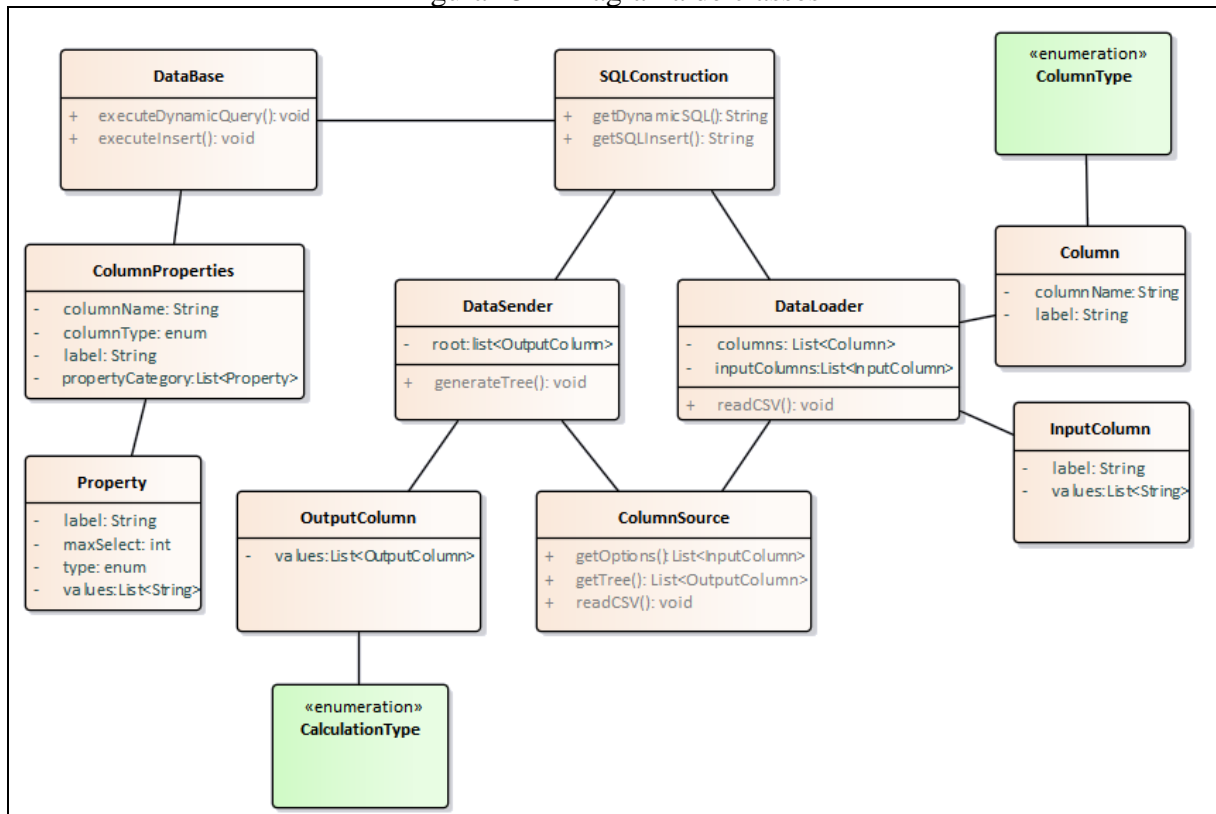
Ao submeter a montagem da correlação (caso de uso UC06 - Correlacionar os dados), o protótipo fará uma requisição ao servidor que construirá a árvore de correlação dos dados, conforme UC07 - Construir árvore de correlação. A criação da árvore de correlação possui como exemplo os resultados de uma consulta dinâmica do tipo Structured Query Language (SQL) ao BD H2 (caso de uso UC08 - Montar consulta dinâmica). Após a conclusão da montagem das informações, é apresentado ao usuário a correlação de duas maneiras, a primeira é uma tabela com dados numéricos (média ou quantidade) entre os grupos, e a segunda através de gráfico de barras. A forma de apresentação corresponde aos casos de usos UC11 - Visualizar a correlação de forma estatística e UC10 - Visualizar a correlação de forma gráfica respectivamente.

### 3.3.2 Diagrama de classes

Nesta seção é apresentado o diagrama de classes, conforme ilustrado na Figura 15. No diagrama, está representada a arquitetura das classes responsáveis pela comunicação ao BD, estruturação das informações e construção da árvore de correlação. Por não fundamentarem o modelo de negócio, optou-se pela retirada do diagrama as classes irrelevantes, estas com objetivo de realizar os tratamentos de serviços (responsáveis pela comunicação cliente servidor e pela realização das conversões dos objetos para JSON) e o tratamento de exceções.



Figura 15 – Diagrama de classes



Fonte: elaborado pelo autor.

A classe `ColumnSource` contém os métodos responsáveis por desencadear todo o processamento necessário para o funcionamento da aplicação. Na classe, requisita-se o método `readCSV()` no momento da importação do arquivo, onde por meio dele realiza-se o processo de leitura dos dados do arquivo para gravação no banco de dados relacional H2. Monta-se os atributos de correlação dos dados a partir do método `getOptions()`, sua requisição ocorre ao abrir a aplicação ou através da função importar arquivo. Para construção da árvore requisita-se o método `getTree()`, retornando as correlações para posterior apresentação ao usuário.

Responsabilizam-se as classes `Column`, `ColumnProperties`, `Property`, `InputColumn` e `OutputColumn` por definir as estruturas dos dados no servidor. Para cada uma destas classes, há diferentes atributos e métodos que auxiliam a manipulação estrutural e de valores. Já nas classes `DataLoader` e `DataSender`, implementou-se a geração e o gerenciamento destas estruturas. Nessas classes, localiza-se toda a lógica para permitir o funcionamento do protótipo.

Somente a classe `DataBase` é responsável pela realização da comunicação com o banco de dados. Esta classe possui configurações de acesso ao banco, como também métodos para realização das ações de criação, recuperação, atualização e exclusão (do inglês `Create`,

Retrieve, Update and Delete - CRUD) dos dados na tabela. Ressalta-se que nenhuma outra classe do protótipo possui acesso ao BD. Como parte da camada do banco de dados, a classe `SQLConstruction` é responsável pelo dinamismo da aplicação. Seus métodos são requisitados pelas classes `DataLoader` e `DataSender`, classes responsáveis por controlar e gerar a correlação do protótipo.

### 3.4 IMPLEMENTAÇÃO

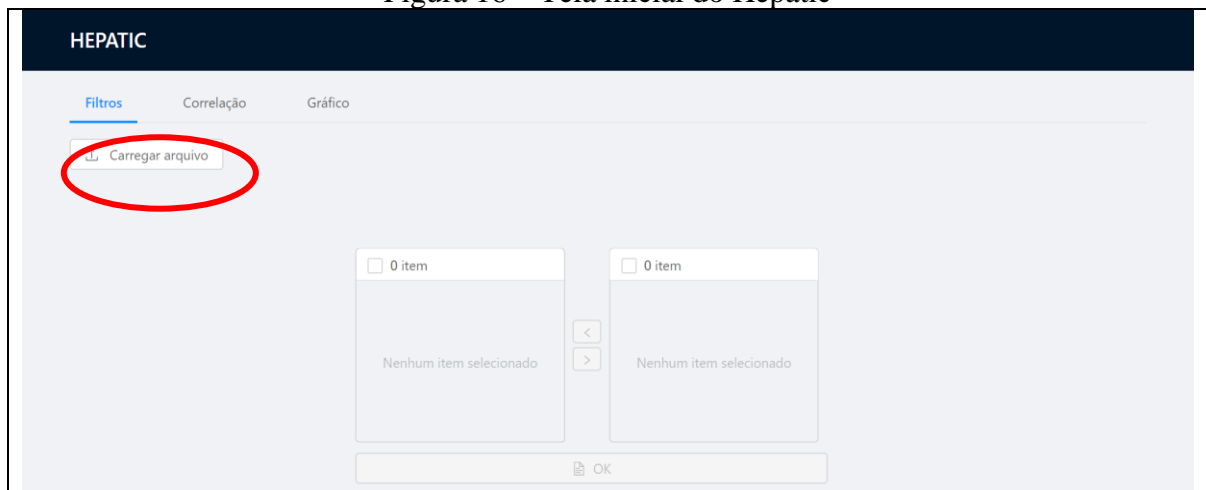
Nesta seção encontram-se os trechos de códigos juntamente com breves explicações mais relevantes para a execução do protótipo. Objetivou-se para cada item desta seção em apresentar primeiramente à funcionalidade do protótipo, e na sequência a operação via código-fonte.

Na seção 3.4.1, apresenta-se o processo de importação de dados. Nele descreve-se o processo para importar o arquivo de dados, define-se a estrutura da planilha, transmissão dos dados para o servidor e funcionamento via código-fonte. Na seção 3.4.2, mostram-se as regras para divisão e seleção dos dados a partir dos atributos originados da importação da planilha. Na seção 3.4.3, apresenta-se a operacionalidade de correlação dos dados a partir dos atributos de correlação, a forma de apresentação por tabela comparativa e gráfico de barras, e também a execução para geração da árvore de correlação em nível de código-fonte.

#### 3.4.1 Importação dos dados

Apresenta-se na Figura 16 a tela inicial do protótipo. Nela, visualiza-se dois quadros alinhados em paralelo no centro da tela que são responsáveis por apresentar os atributos para correlações e seleção de grupo de dados. Para iniciar o processo de utilização do protótipo, necessita-se importar um arquivo do tipo `.csv` contendo as informações desejadas para correlação. Utiliza-se o botão “Carregar arquivo”, localizado no canto superior esquerdo da tela para dar início a importação.

Figura 16 – Tela inicial do Hepatic



Fonte: elaborado pelo autor.

Para o correto reconhecimento das informações no processo de importação, necessita-se de uma planilha corretamente estruturada. Todas as colunas da planilha devem estar rotuladas por um título. Objetiva-se com isso uma apresentação clara das opções para correlação ao usuário após a importação. Já para os valores das colunas, adotou-se alguns padrões na sua formação, conforme apresenta-se no Quadro 3. Quando os valores se referem à data, definiu-se como formato padrão dia/mês/ano (2 dígitos/ 2 dígitos /4 dígitos respectivamente). Já para os valores numéricos, o protótipo reconhece a divisão de decimais com “.” (ponto) ou “,” (vírgula). Transmite-se somente o conteúdo do arquivo para o servidor.

Quadro 3 – Estrutura da planilha para importação

Profissão	Sexo	MELD	Data de óbito	Motivo óbito	Tempo isquemia
texto	texto	10	Dia / mês /ano	texto	10
Texto	Texto	10,5	Dia / mês /ano	Texto	10,5
TEXT0	TEXT0	10.5	Dia / mês /ano	TEXT0	10.5

Fonte: elaborado pelo autor.

Ao obter os dados, inicia-se um processo de conversão e extração das informações para posterior processamento. Mostra-se no Quadro 4, o processo de obtenção dos dados em forma binária e sua conversão para um buffer de leitura.

Quadro 4 – Transmissão dos dados de importação

```

1 | @POST
2 | @Consumes(MediaType.MULTIPART_FORM_DATA)
3 | @Path("/upload")
4 | public void recordFile(MultipartFormDataInput input) throws
5 | IOException, DatabaseException, BuilderException {
6 |     InputStream inputStream =
7 | input.getFormDatumap().get("file").get(0).getBody(InputStream.class,
8 | null);
9 |     new ColumnSource().recordColumns(new BufferedReader(new
10 | InputStreamReader(inputStream)));
11 | }

```

Fonte: elaborado pelo autor.

Envia-se o conteúdo do arquivo para o servidor em um `array` de bytes. Para obtê-los no lado do servidor, necessita-se especificar o tipo de recebimento como `MULTIPART_FORM_DATA`, conforme definido na linha 2. A partir da especificação do cabeçalho do método, realiza-se a conversão dos bytes recebidos em um buffer de leitura (linha 7). Utilizou-se as classes da biblioteca IO (Java 7) para realização da conversão. Na linha 9, ocorre a passagem do buffer de leitura para o método `recordColumns()` da classe `ColumnSouce`, inicia-se por esta instrução o processo para persistência das informações no BD relacional H2.

Posteriormente, molda-se a estrutura das informações para importação das informações no servidor a partir da classe `Column`. Conforme apresenta-se no Quadro 5, definiu-se atributos como: `label`, nome da coluna no BD, o tipo da coluna e os valores.

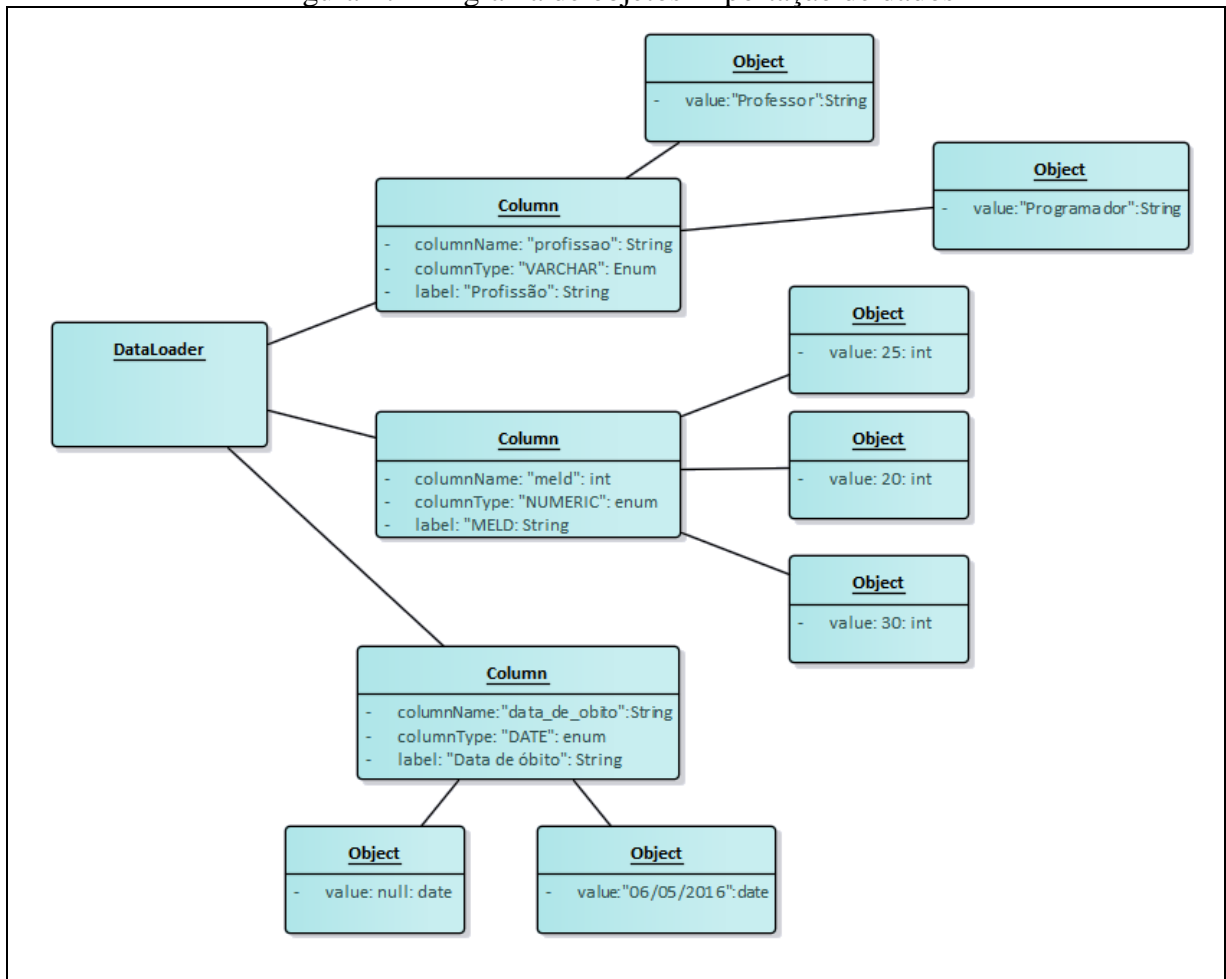
Quadro 5 – Estrutura das colunas no servidor

1	<code>public class Column {</code>
2	
3	<code>    private String label;</code>
4	<code>    private String columnName;</code>
5	<code>    private ColumnType columnType;</code>
6	<code>    private List&lt;Object&gt; values;</code>

Fonte: elaborado pelo autor.

Armazena-se o título original da coluna no atributo `label` na linha 3, título que posteriormente será utilizado para o usuário identificar as opções na tela para correlação. O atributo `columnType` (linha 5) define qual é o tipo do dado que será trabalhado no BD, são três: textual (`VARCHAR`), datas (`DATE`) e numéricos (`NUMERIC`). A partir disso, define-se como `columnName` (linha 4) o rótulo da coluna no BD, objetiva-se com esta tratativa facilitar a criação dinâmica de comandos SQL pela aplicação. Já na linha 6, armazena-se em uma lista todos os valores que continham na coluna do arquivo importado, estes valores posteriormente serão inseridos no BD com o seu tipo já definido no atributo `columnType`. Para melhor compreensão, a Figura 17 apresenta um diagrama de objetos com uma exemplificação da estrutura da classe `Column` montada.

Figura 17 – Digrama de objetos importação de dados



Fonte: elaborado pelo autor.

Utiliza-se a classe `DataLoader` para processar as informações do arquivo e convertê-los em uma lista com objetos do tipo `Column`. Para realizar a criação dos objetos `Column`, a classe `DataLoader` realiza os tratamentos para manipular, formatar e atribuir os tipos aos valores retirados da planilha. No Quadro 6, demonstra-se em nível código-fonte de maneira abstraída a conversão do buffer de leitura para a lista de objetos de `Column`.

Quadro 6 – Processo de transformação das informações

```

1 public List<Column> readCSV(BufferedReader br) throws BuilderException
2 {
3     try {
4         List<String[]> l = new ArrayList<>();
5         String line;
6         while ((line = br.readLine()) != null) {
7             l.add(line.split(";"));
8         }
9         return transformValues(l);
10    } catch (ParseException | IOException ex) {
11        throw new BuilderException("Erro ao importar o arquivo.", ex);
12    }
13 }
  
```

Fonte: elaborado pelo autor.

Trata-se no método `transformValues()` na linha 9, a transformação dos valores brutos da planilha para objetos de `Column`. Conforme apresentado no Quadro 7, recebe-se como parâmetro uma lista de `string` contendo os valores brutos da importação. Para cada coluna da planilha, será criado um objeto `Column`, atribuindo o resultado da verificação do tipo da coluna a partir do método `getFieldType()` e a atribuição do nome da coluna no banco de dados através do método `getFieldName()`.

**Quadro 7 – Criação de objetos `Column` e definição de estrutura**

```

1 private List<Column> transformValues(List<String[]> data) throws
2 ParseException {
3     List<Column> columns = new ArrayList<>(data.get(0).length);
4
5     for (int i = 0; i < data.get(0).length; i++) {
6         Column c = new Column();
7         c.setColumnName(getFieldName(data.get(0)[i]));
8         c.setColumnType(getFieldType(data, i));
9         c.setLabel(data.get(0)[i]);
10        columns.add(c);
11    }
12
13    for (int i = 1; i < data.size(); i++) {
14        for (int j = 0; j < columns.size(); j++) {
15            Column c = columns.get(j);
16            try {
17                c.addValue(transformValue(data.get(i)[j],
18 c.getColumnType()));
19            } catch (IndexOutOfBoundsException ex) {
20                c.addValue(null);
21            }
22        }
23    }
24
25    return columns;
26 }

```

Fonte: elaborado pelo autor.

Para armazenar os valores com o tipo correto das informações é essencial para posteriores manipulações ao BD. Para atribuição do tipo, chama-se conforme linha 8 o método `getFieldType()`. No método, realiza-se verificações de forma total ou parcial dos valores das colunas. No Quadro 8, pode-se verificar em nível de código-fonte o processo de obtenção do tipo da coluna.

Quadro 8 – Obtendo o tipo da coluna

```

1 private Column.ColumnType getFieldTypes(List<String[]> s, int column)
2 {
3     Column.ColumnType typeValue = null;
4     for (int i = 1; i < s.size(); i++) {
5         if (column >= s.get(i).length) {
6             continue;
7         }
8         String value = s.get(i)[column];
9         if (value == null || "".equals(value)) {
10            continue;
11        }
12        try {
13            Double.parseDouble(value);
14            if (typeValue == null) {
15                typeValue = Column.ColumnType.NUMERIC;
16            } else if (!Column.ColumnType.NUMERIC.equals(typeValue)) {
17                return Column.ColumnType.VARCHAR;
18            }
19        } catch (NumberFormatException e) {
20            //is not a number
21            try {
22                dateFormat.parse(value);
23                if (typeValue == null) {
24                    typeValue = Column.ColumnType.DATE;
25                } else if (!Column.ColumnType.DATE.equals(typeValue)) {
26                    return Column.ColumnType.VARCHAR;
27                }
28            } catch (ParseException ex) {
29                //is not a date nor a number
30                return Column.ColumnType.VARCHAR;
31            }
32        }
33    }
34    return typeValue == null ? Column.ColumnType.VARCHAR : typeValue;
35 }

```

Fonte: elaborado pelo autor.

Para evitar determinados erros, utiliza-se tratamentos de exceções como parte do processo de verificação dos tipos de cada coluna importada. Verifica-se cada valor até identificar se o tipo da coluna é textual, numérica ou de data. Com a conversão do valor para o tipo `Double` (linha 13), define-se primeiramente a coluna como numérica. Caso a conversão do valor resulte em uma exceção (linha 19), descobre-se que o mesmo não é numérico, então ocorre a conversão do valor para data (linha 22). Caso a conversão também resulte numa exceção, interrompe-se o processo e o algoritmo assume a coluna como textual. Assume-se também como texto nos casos de uma coluna possuir número e datas (validação das linhas 16 e 25, respectivamente). Somente, valida-se uma coluna como numérica ou de datas, caso o algoritmo percorra todos os valores sem exceções, ou seja, para identificar que uma coluna é composta de datas, nenhuma exceção do tipo `ParseException` deverá ser lançada. A partir disso, obtém-se o rótulo da coluna do banco a partir do método `getFieldName()`, conforme apresentado no Quadro 9.

Quadro 9 – Conversão dos títulos em rótulos para criação da tabela no BD

```

1 private String getFieldName(String s) {
2     if (s == null) {
3         return null;
4     }
5
6     String specialCharacters = "áâãäåèéëìíîïóôõöùüçñýÿ -";
7     String replaceCharacters = "aaaaeeeeiiiioooouucnyy__";
8
9     StringBuilder sb = new StringBuilder(s.length());
10
11    for (char c : s.trim().toLowerCase().toCharArray()) {
12        int index = specialCharacters.indexOf(c);
13        if (index < 0) {
14            sb.append(c);
15        } else {
16            sb.append(replaceCharacters.charAt(index));
17        }
18    }
19
20    return sb.toString();
21 }

```

Fonte: elaborado pelo autor.

O algoritmo para conversão do nome para criação da tabela no BD é trivial em comparação com o algoritmo que define o tipo da coluna. Adotou-se um padrão para conversão, que se constitui na eliminação de caracteres especiais, acentuação, traços, entre outros. Ao encontrar algum caractere especial (linha 6), substitui-se pelo seu caractere correspondente (linha 7), já os espaços são substituídos por underline (\_). Mostra-se no Quadro 10 uma representação das transformações dos títulos das colunas em nomes apropriados para criação da tabela no banco de dados.

Quadro 10 – Demonstração de conversão

Título da coluna	Nome no banco
Profissão	profissao
Sexo	sexo
MELD	meld
Data de óbito	data_de_obito
Motivo óbito	motivo_obito
Tempo isquemia	tempo_isquemia

Fonte: elaborado pelo autor.

A partir da lista de objetos do Column, pode-se persistir no BD as informações para conclusão do processo de importação do arquivo, conforme mostra o Quadro 11.



Quadro 11 – Persistência no banco de dados

```

1 public void recordColumns(BufferedReader br) throws BuilderException,
2 DatabaseException {
3     List<Column> columns = new DataLoader().readCSV(br);
4     try (Database d = new Database(columns, null)) {
5
6         d.createTable();
7
8         Object[] param;
9         Column.ColumnType[] types;
10        String[] metaData = new String[3];
11
12        for (int i = 0; i < columns.get(0).getValues().size(); i++) {
13            param = new Object[columns.size()];
14            types = new Column.ColumnType[columns.size()];
15
16            for (int j = 0; j < columns.size(); j++) {
17                param[j] = columns.get(j).getValues().get(i);
18                types[j] = columns.get(j).getColumnType();
19
20                if (i == 0) {
21                    metaData[0] = columns.get(j).getColumnName();
22                    metaData[1] = columns.get(j).getLabel();
23                    metaData[2] =
24 columns.get(j).getColumnType().getSqlType();
25                    d.insertMetaData(metaData);
26                }
27            }
28
29            d.executeInsert(param, types);
30        }
31    }
32 }

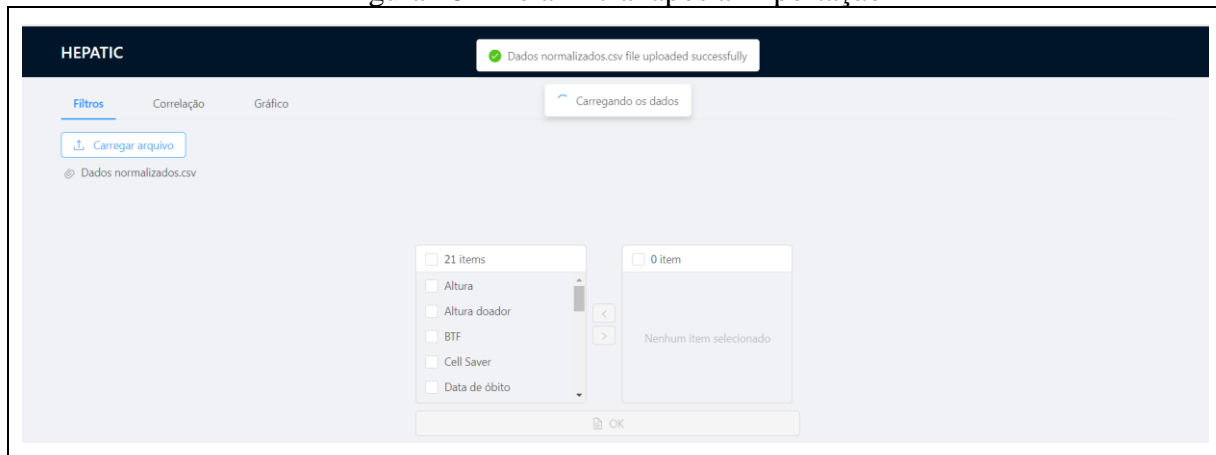
```

Fonte: elaborado pelo autor.

Apresenta-se na linha 4 a comunicação com o BD, que é realizada pela classe `DataBase`. Por meio desta, são construídos dinamicamente comandos SQL para criação da estrutura da tabela no BD (linha 6). Cria-se também na linha 6 uma tabela para persistir informações estruturais da classe `Column`, como: nome original da coluna, tipo da coluna e o nome correspondente no BD. Com a persistência da estrutura na linha 25, objetiva-se que o usuário não precise sempre importar uma planilha para realizar a correlação. Persiste-se os valores para correlação no método `executeInsert()`, na linha 29. Finaliza-se o processo de importação após a execução deste algoritmo.

Após o término da importação, apresenta-se ao usuário os atributos de correlação com as descrições das colunas do arquivo recém importado, conforme apresentado na Figura 18. Como a apresentação dos atributos de correlação é um processo que não depende só da importação, sua execução será exemplificada na seção 3.5.2.

Figura 18 – Tela inicial após a importação

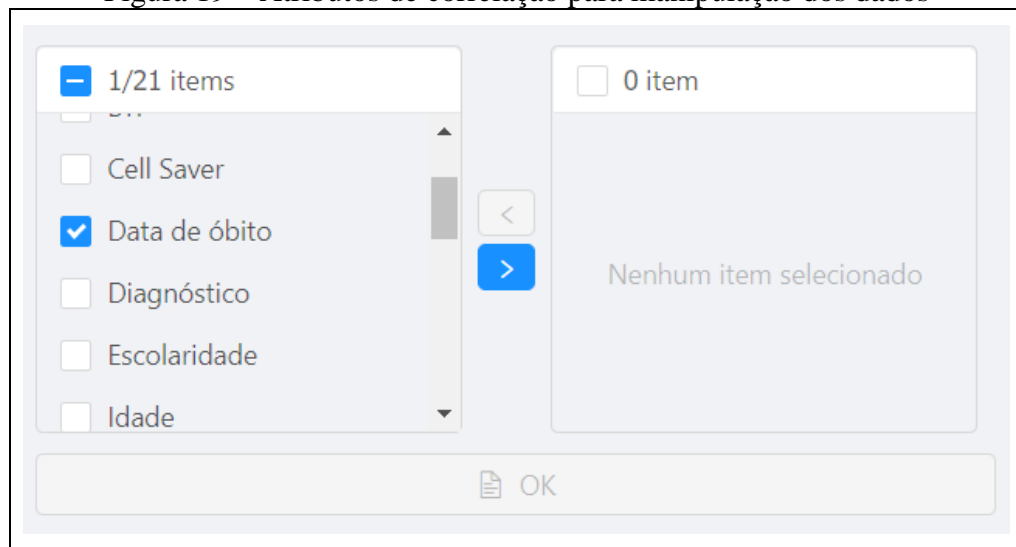


Fonte: elaborado pelo autor.

### 3.4.2 Atributos de correlação

Com os atributos de correlação, define-se a divisão do grupo de dados e especifica-se a seleção de grupos de informações. Nesta seção apresenta-se somente as opções dos filtros a partir das propriedades dos atributos de correlação, a modelagem para correlação encontra-se na seção 3.5.3. Para iniciar a exploração dos filtros, seleciona-se a um atributo do quadrante esquerdo, conforme ilustra a Figura 19.

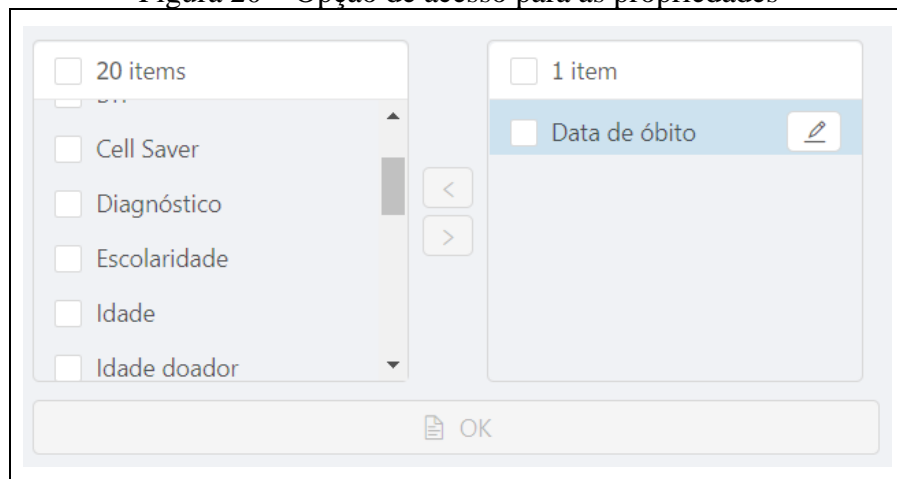
Figura 19 – Atributos de correlação para manipulação dos dados



Fonte: elaborado pelo autor.

Após a escolha da opção, clica-se na seta indicativa para adicionar o atributo a lista de correlação. Para cada opção adicionada à lista, habilita-se um botão para acesso às propriedades dos filtros, conforme é apresentado na Figura 20.

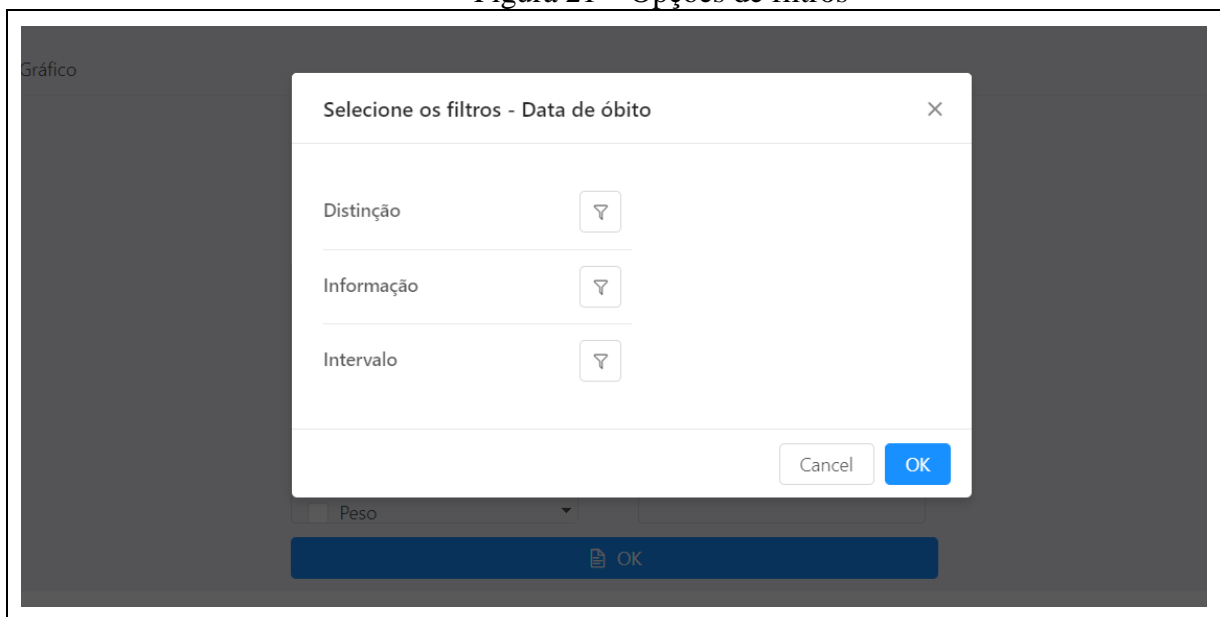
Figura 20 – Opção de acesso para as propriedades



Fonte: elaborado pelo autor.

Ao clicá-lo, apresenta-se uma modal com diferentes tipos de filtros. As opções de filtros para atributos numéricos e de datas são: distinção, informação e intervalo, conforme Figura 21. Já para atributos do tipo textual, a opção de filtro do tipo “intervalo” não é apresentada.

Figura 21 – Opções de filtros



Fonte: elaborado pelo autor.

Apresentam-se todos os valores de forma distinta ao selecionar a opção *distinção*. Objetiva-se pela filtragem por distinção a seleção de grupos específicos ou unitários de dados na base de dados. Nota-se na Figura 22, que o protótipo trabalhará somente com registros que possuam data de óbito correspondentes aos valores selecionados.

Figura 22 – Filtros por distinção

Selecione os filtros - Data de óbito

Distinção  Selecionar todos

Informação

Intervalo

09/06/2016 22/06/2016

03/08/2016 10/08/2016

13/09/2016 16/09/2016

17/09/2016 08/10/2016

12/10/2016 29/10/2016

01/11/2016 08/11/2016

13/11/2016  22/12/2016

23/12/2016 05/01/2017

26/01/2017 31/01/2017

17/02/2017 18/02/2017

28/03/2017 06/04/2017

13/04/2017  16/05/2017

19/05/2017 23/05/2017

31/05/2017 25/06/2017

28/06/2017 04/07/2017

12/08/2017 03/09/2017

09/09/2017

Cancel OK

Fonte: elaborado pelo autor.

Utiliza-se a opção *Informação* quando há objetivo de restringir os registros por valores vazios ou preenchidos. Esta opção sempre apresentará apenas dois valores, *Sim* e *Não*. Marca-se a opção *Sim* para agrupar registros que possuam valor no atributo selecionado. Já a opção *Não* atua de forma contrária, restringindo a seleção de dados para os casos de quem não possuam valor. Seleciona-se as duas opções quando se objetiva em selecionar todos os registros da base de dados, conforme Figura 23.

Figura 23 – Filtros por informação

Selecione os filtros - Data de óbito

Distinção  Sim  Não

Informação

Intervalo

Cancel OK

Peso

OK

Fonte: elaborado pelo autor.

Disponibiliza-se também a opção de seleção por intervalo. Esta opção só é disponível para atributos numéricos ou de datas, pois não há como definir intervalos textuais. Nesta opção, permite-se apenas a seleção de dois valores, pois o protótipo retornará todos os dados no intervalo entre os dois valores selecionados. A Figura 24 demonstra a seleção de dois

valores: datas 17/09/2016 e 31/01/2017. No exemplo, todos os registros que serão selecionados possuem data de óbito dentro de um intervalo de 17/09/2016 até 31/01/2017.

Figura 24 – Filtros por intervalo de data

Selecione os filtros - Data de óbito

Você pode selecionar até 2 itens

Distinção  09/06/2016  22/06/2016  
 03/08/2016  10/08/2016  
 13/09/2016  16/09/2016

Informação  17/09/2016  08/10/2016  
 12/10/2016  29/10/2016

Intervalo  01/11/2016  08/11/2016  
 13/11/2016  22/12/2016  
 23/12/2016  05/01/2017  
 26/01/2017  31/01/2017  
 17/02/2017  18/02/2017  
 28/03/2017  06/04/2017  
 13/04/2017  16/05/2017  
 19/05/2017  23/05/2017  
 31/05/2017  25/06/2017  
 28/06/2017  04/07/2017  
 12/08/2017  03/09/2017  
 09/09/2017

Cancel OK

Fonte: elaborado pelo autor.

Disponibiliza-se também a opção de intervalo para dados numéricos. Sua utilização é ilustrada no exemplo da Figura 25, onde selecionou-se um intervalo de idade de 32 até 44 anos.

Figura 25 – Filtros por intervalo numérico

Selecione os filtros - Idade

Você pode selecionar até 2 itens

Distinção  21.00  22.00  23.00  
 26.00  27.00  28.00

Informação  32.00  33.00  35.00  
 36.00  38.00  40.00  
 41.00  43.00  44.00

Intervalo  45.00  46.00  47.00  
 48.00  49.00  50.00  
 51.00  52.00  53.00  
 54.00  55.00  56.00  
 57.00  58.00  59.00  
 60.00  61.00  62.00  
 63.00  64.00  65.00  
 66.00  67.00  68.00  
 69.00  70.00  71.00  
 72.00  74.00  76.00

Cancel OK

Fonte: elaborado pelo autor.

Para exibição das propriedades e atributos de correlação para o usuário, estruturou-se estas informações nas classes `ColumnProperties` e `Property`. Ao usuário entrar no protótipo, requisita-se o servidor para montagem dos atributos de correlação. Adentrando-se em nível de código-fonte, o Quadro 12 apresenta as definições estruturais das classes `ColumnProperties`, e `Property`.

Quadro 12 – Estrutura dos atributos de correlações

1	<code>public class ColumnProperties implements Serializable {</code>
2	
3	<code>private String label;</code>
4	<code>private String columnName;</code>
5	<code>private Column.ColumnType columnType;</code>
6	<code>private List&lt;Property&gt; propertyCategory;</code>
7	
8	
9	<code>public class Property implements Serializable {</code>
10	
11	<code>private String label;</code>
12	<code>private InputColumn.CategoryProperty type;</code>
13	<code>private List&lt;String&gt; values;</code>
14	<code>private int maxSelect;</code>

Fonte: elaborado pelo autor.

A classe `ColumnProperties` (linha 1) define os atributos apresentadas nos quadrantes da tela principal. Já a classe `Property` (linha 9) define as propriedades para filtros dos atributos do quadrante de correlação. Para as propriedades em nível de código-fonte, define-se seu tipo a partir do atributo `type`, onde somente três valores lhe são aplicáveis: `Distinção`, `Informação` e `Intervalo`. Monta-se a estrutura de atributos e propriedades sempre quando o usuário entra na tela principal ou realiza uma importação de planilha. O Quadro 13 apresenta em nível código-fonte a montagem da estrutura de atributos e propriedades.

Quadro 13 – Obtenção dos atributos de correlação

```

1 public List<ColumnProperties> getColumnProperties() throws
2 DatabaseException {
3     try (Statement statement = this.getConnection().createStatement())
4     {
5         List<ColumnProperties> columnProperties = new ArrayList<>();
6
7         ResultSet resultSet =
8 statement.executeQuery(sqlConstruction.getColumnsSQLMetaData());
9
10        while (resultSet.next()) {
11            columnProperties.add(new
12 ColumnProperties(resultSet.getString("LABEL"),
13                    resultSet.getString("COLUMN_NAME"),
14                    resultSet.getString("COLUMN_TYPE")));
15        }
16        List<Property> properties;
17
18        for (ColumnProperties c : columnProperties) {
19            List<String> distinctValues =
20 getDistinctValues(c.getColumnName(), c.getColumnType());
21
22            properties = new ArrayList<>();
23            properties.addAll(Arrays.asList(
24                new Property(InputColumn.CategoryProperty.DISTINCT,
25                    distinctValues),
26                new
27 Property(InputColumn.CategoryProperty.INFORMATION,
28                    Arrays.asList("Sim", "Não"))));
29
30            if
31 (!"VARCHAR(255)".equalsIgnoreCase(c.getColumnType().getSqlType())) {
32                properties.add(new
33 Property(InputColumn.CategoryProperty.INTERVAL,
34                    distinctValues));
35            }
36
37            c.setPropertyCategory(properties);
38        }
39
40        return columnProperties;
41    } catch (SQLException ex) {
42        throw new DatabaseException("Erro ao buscar as colunas", ex);
43    }
44 }
45

```

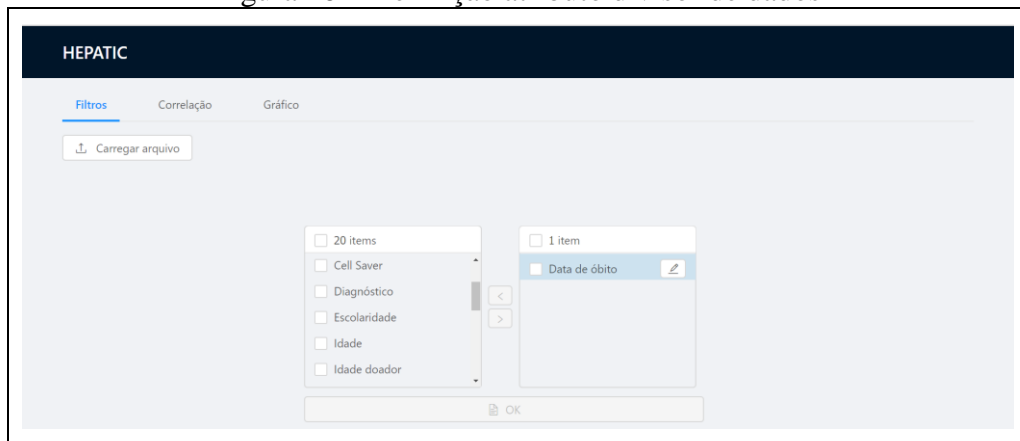
Fonte: elaborado pelo autor.

Divide-se em duas etapas a montagem dos atributos e suas propriedades. Na primeira, realiza-se a busca das informações das colunas previamente inseridas no BD (linha 7), com o retorno, monta-se uma lista de objetos do tipo `ColumnProperties` (linha 12). Já na segunda, a partir dos tipos e nomes das colunas no BD, buscam-se os valores de cada coluna de forma distinta (linha 20). Atribui-se os valores à objetos do tipo `Property` (linha 23), que posteriormente são agregados aos objetos `ColumnProperties` (linha 37). Para apresentação dos atributos, os objetos são serializados em JSON e enviados para o lado do cliente.

### 3.4.3 Tabela comparativa

O processo de correlação dos dados no protótipo foi subdividido em três etapas, que são: divisão de grupos, seleção dos dados e modelagem da correlação. Definiu-se como divisor dos grupos sempre o primeiro atributo na lista da correlação, esta regra é vigente para todas as correlações realizadas no Hepatic. Para exemplificar, apresenta-se na Figura 26 que o atributo divisor dos dados é a data do óbito, pois está na primeira posição da lista de correlação.

Figura 26 – Definição atributo divisor de dados



Fonte: elaborado pelo autor.

No protótipo desenvolvido, permite-se somente dividir os dados em dois grupos. Para estes dois grupos, atribuiu-se a nomenclatura de grupo dos dados selecionados e grupo dos dados restantes. Divide-se os grupos a partir das propriedades do primeiro atributo, onde alocam-se para o grupo dos selecionados todos os registros correspondentes às propriedades selecionadas pelo usuário. Já para o grupo restante, alocam-se todos os registros que as propriedades não foram selecionadas pelo usuário. Conforme apresentado na Figura 27, exemplifica-se a seleção dos pacientes que vieram a óbito nos dias 28/03/2017 e 06/04/2017, logo movimentam-se os registros destes pacientes para o grupo selecionado. Para o grupo restante, alocam-se todos os pacientes que não morreram na data 28/03/2017 e 06/04/2017.



Figura 27 – Grupos por distinção

Fonte: elaborado pelo autor.

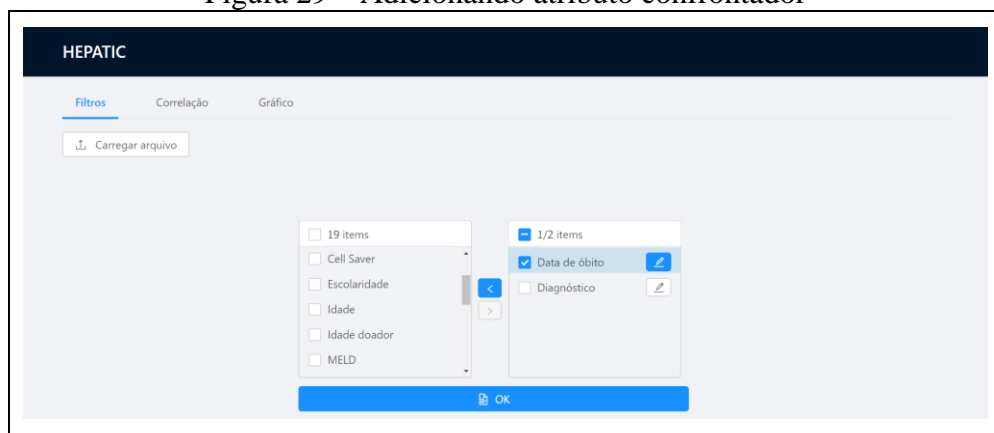
Estende-se a divisão de grupos também por *Intervalo* e *Informação*. Para divisão por *Intervalo*, define-se o intervalo de valores, consequentemente registros dentro do intervalo estipulado alocam-se para o grupo selecionado. Já para os registros que possuem os valores fora do intervalo selecionado, alocam-se para o grupo restante. O processo para o tipo *Informação* assemelha-se com os demais, onde atribui-se ao grupo selecionado somente registros com os valores preenchidos ou registros com os valores não preenchidos. Exemplifica-se na Figura 28 a divisão de dados com o filtro *Informação*, no exemplo, marca-se para o grupo selecionado todos os pacientes que possuam algum valor na data de óbito (entende-se como pacientes que vieram à óbito). Para o grupo restante, sobrarão os que não possuem valor no campo data de óbito (entende-se os pacientes que sobreviveram).

Figura 28 – Divisão por informação

Fonte: elaborado pelo autor.

Inicia-se a correlação após o processo de divisão dos grupos. Para comparar algum atributo entre o grupo selecionado e o grupo restante, adiciona-se um segundo atributo ao quadrante da esquerda. Para isso, seleciona-se o atributo desejado do quadrante direito e move-se ao esquerdo utilizando o botão com direcionamento indicativo. Previamente definiu-se a divisão dos grupos por data de óbito (grupo selecionado, filtro *Informação*, valor *Sim*). Conforme Figura 29, adiciona-se o atributo *Diagnóstico* para comparar entre os grupos de pacientes que vieram à óbito (grupo selecionado) e os pacientes que tiveram sobrevida (grupo restante).

Figura 29 – Adicionando atributo confrontador



Fonte: elaborado pelo autor.

Para cada atributo, necessita-se sempre realizar a filtragem do universo de dados para correlação. Exemplifica-se o processo na Figura 30, onde utilizou-se os valores *Cirrose* e *Hepatite* da propriedade filtro por *Distinção*. Com estas propriedades marcadas, objetiva-se em comparar a quantidade de pacientes que morreram de cirrose e hepatite, dos que obtiveram os mesmos diagnósticos, mas sobreviveram.

Figura 30 – Seleção de dados por diagnóstico do paciente

Fonte: elaborado pelo autor.

Após definir o atributo divisor de dados e escolher o atributo comparador, realiza-se a correlação ao clicar no botão OK abaixo dos quadrantes. Verifica-se o resultado da comparação na aba Correlação, localizado no canto esquerdo superior.

Na aba Correlação apresentam-se os filtros utilizados (Data de óbito e Diagnóstico). Conforme Figura 31, possibilita-se a verificação de valores e propriedades utilizados para correlação ao mover o cursor do mouse para cima do filtro.

Figura 31 – Layout dos filtros na correlação



Fonte: elaborado pelo autor.

Apresenta-se em quatro colunas o resultado da comparação na tabela de correlação, são elas: Informação, Grupo selecionado, Grupo restante e Total geral. Mostra-se na coluna Informação os valores agrupados em forma textual que dão sentido às quantidades numéricas (valores presentes nas demais colunas). Já nas colunas grupo selecionado e grupo restante, apresenta-se a quantidade de pacientes que se encaixam no grupo selecionado (previamente definido como que vieram a óbito) e grupo restante (previamente definido como pacientes com sobrevida) respectivamente. Na coluna Total geral, apresentam-se o total de pacientes entre o grupo selecionado e o grupo restante. Mostra-se na Figura 32 o resultado da correlação de pacientes com os diagnósticos Hepatite e Cirrose que morreram em comparação aos pacientes que não morreram e possuíram os mesmos diagnósticos.

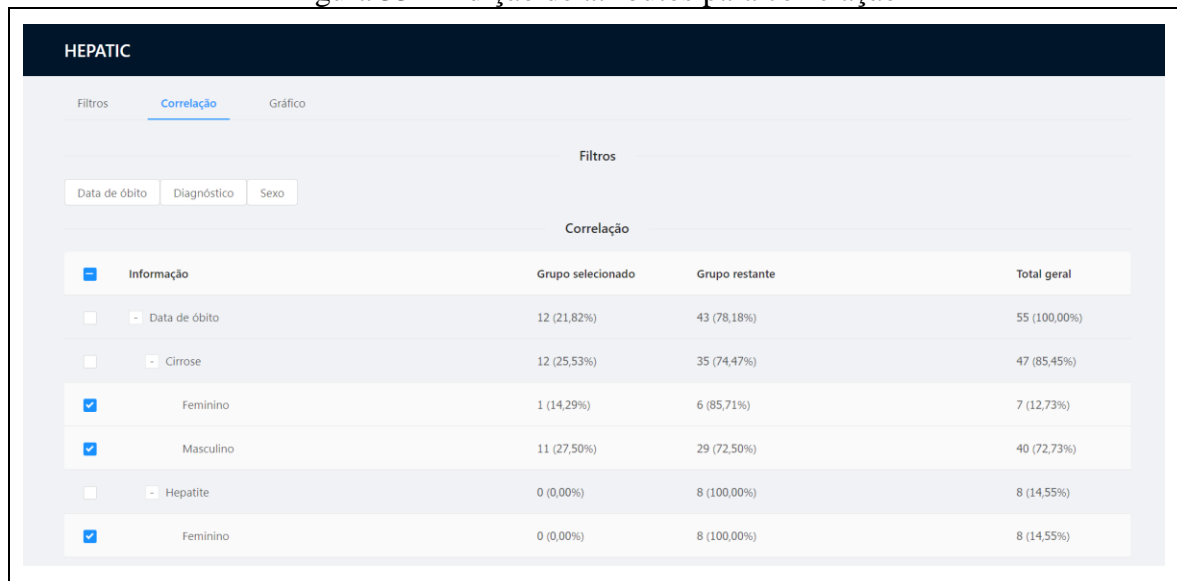
Figura 32 – Tabela de correlação

Informação	Grupo selecionado	Grupo restante	Total geral
Data de óbito	12 (21,82%)	43 (78,18%)	55 (100,00%)
Cirrose	12 (25,53%)	35 (74,47%)	47 (85,45%)
Hepatite	0 (0,00%)	8 (100,00%)	8 (14,55%)

Fonte: elaborado pelo autor.

Apresenta-se a correlação em uma tabela com linhas expansíveis e retráteis. Particionam-se cada vez mais linhas da tabela ao adicionar novos atributos para correlação. Limita-se a correlação à medida que as quantidades chegam próximo a valores unitários, isto ocorre quando não há dados suficientes para suportar a quantidade de atributos adicionados para correlação. Exemplifica-se a partição/abertura do atributo Diagnóstico com a adição do atributo Sexo, objetiva-se comparar entre o grupo de óbito (grupo selecionado) e os grupos de sobrevida (grupo restante) a as doenças Hepatite e Cirrose particionadas por sexo Masculino e Feminino. A Figura 33 apresenta o resultado da correlação.

Figura 33 – Adição de atributos para correlação



Fonte: elaborado pelo autor

Para a construção da correlação no servidor, cria-se uma árvore em cinco etapas, que são:

- montagem dinâmica da consulta SQL;
- estruturação da árvore;
- rastreamento de valores;
- limpeza da árvore;
- cálculo da porcentagem.

Na primeira etapa, monta-se o código SQL para consulta ao BD a partir dos atributos e propriedades escolhidas pelo usuário. Para isso, converte-se estas propriedades e atributos para objetos da classe `InputColumn`. Apresenta-se no Quadro 14 a estruturação dos atributos e propriedades escolhidas pelo usuário.

Quadro 14 – Estrutura dos atributos e propriedades no servidor

```

1 public class InputColumn implements Serializable,
2   Comparable<InputColumn> {
3
4     private int order;
5     private String label;
6     private String columnName;
7     private CategoryProperty type;
8     private Column.ColumnType columnType;
9     private List<String> values;
10    private CalculationType calculationType;

```

Fonte: elaborado pelo autor.

Armazena-se todos os valores dos filtros escolhidos pelos usuários no atributo de classe `values` (linha 9). Posteriormente, utiliza-se estes valores para montar as restrições na cláusula `where` da consulta SQL. Para a cláusula `select`, monta-se os campos a partir dos valores do atributo de classe `columnName` (linha 6). Com uma lista de objetos `InputColumn`, torna-se possível a montagem dinâmica do SQL, conforme apresenta-se no Quadro 15.

Quadro 15 – Montagem dinâmica do código SQL

```

1  for (InputColumn inputColumn : this.inputColumns) {
2      if (inputColumn.getValues() != null &&
3          !inputColumn.getValues().isEmpty()) {
4          switch (inputColumn.getType()) {
5              case DISTINCT:
6                  where.append(" and ")
7                      .append(inputColumn.getColumnName())
8                      .append(" in ")
9                      .append(getInClause(inputColumn.getValues().size()));
10                 break;
11             case INTERVAL:
12                 where.append(" and
13 ").append(inputColumn.getColumnName()).append(" between ? and ?");
14                 break;
15             case INFORMATION:
16             default:
17                 if (inputColumn.getValues().size() == 1) {
18                     where.append(" and
19 ").append(inputColumn.getColumnName())
20
21 .append("Sim".equalsIgnoreCase(inputColumn.getProperties(0))
22                                     ? " is not null " : " is null ");
23                 }
24                 break;
25             }
26     }

```

Fonte: elaborado pelo autor.

Para montagem do SQL dinâmico, necessita-se do tipo e valores das propriedades de filtros (Distinção, Informação e Intervalo) definidos pelo usuário na construção da correlação. A partir disso, define-se qual cláusula de restrição aplica-se para a restrição `where` (linha 4). Converte-se no servidor a modelagem da correlação realizada pelo usuário para uma lista de objetos do tipo `InputColumn`. Para cada objeto desta lista, a verificação do tipo da propriedade é primordial, pois define qual cláusula de restrição aplicar ao SQL. Para o tipo de propriedade `Distinção`, utiliza-se a cláusula `in` (linhas 6, 7, 8 e 10), já para o tipo `Informação`, utiliza-se na linha 23 a cláusula `is null` nos casos em que o valor da informação selecionado for `Não`, ou `is not null` nos casos em que o valor da informação selecionado for `Sim`. Por último, utiliza-se a cláusula `between` (linha 14) quando a propriedade é `Intervalo`. Para cada atributo escolhido pelo usuário, concatena-se a partir do método `append()` as restrições `where`. Finaliza-se a execução da montagem da cláusula `where` ao término da lista de objetos `InputColumn`.

Divide-se os grupos a partir do primeiro atributo da lista de objetos de `InputColumn`. Conforme já mencionado, denomina-se a divisão por grupo dos dados selecionados e grupo dos dados restantes. Para estes dois grupos, adiciona-se sempre duas restrições opostas, que são responsáveis por explicitamente, dividir os grupos na consulta SQL. Apresenta-se no Quadro 16 a correlação das sintaxes das restrições utilizada na cláusula `where` distribuídas para cada tipo de propriedade.

Quadro 16 – Definição restrição SQL por grupo

Propriedade	Grupo selecionado	Grupo oposto
Distinção	and campos in [valores])	and campo not in ([valores])
Informação	and campo is not null	and campo is null
Intervalo	and campo between valor and valor	and campo not between valor and valor

Fonte: elaborado pelo autor.

Utiliza-se as palavras chaves `union all` para separar os resultados dos grupos na consulta SQL. Com a definição da divisão de grupos, finaliza-se a construção dinâmica do código SQL. Apresenta-se no

Quadro 17 a consulta SQL construída a partir da modelagem da correlação dos atributos: Data de óbito (Propriedade: Informação, Valores: “Sim”), Diagnóstico (Propriedade: Distinção, Valores: Hepatite e Cirrose) e por fim Sexo (Propriedade: Informação, Valores: Sim).

Quadro 17 – Código de consulta ao BD

1	SELECT 1 group_id,
2	diagnostico,
3	sexo,
4	Count(*) counter
5	FROM data_values
6	WHERE 1 = 1
7	AND data_de_obito IS NOT NULL
8	AND diagnostico IN ( 'Cirrose', 'Hepatite' )
9	AND sexo IN ( 'Feminino', 'Masculino' )
10	GROUP BY group_id,
11	diagnostico,
12	sexo
13	UNION ALL
14	SELECT 2 group_id,
15	diagnostico,
16	sexo,
17	Count(*) counter
18	FROM data_values
19	WHERE 1 = 1
20	AND data_de_obito IS NULL
21	AND diagnostico IN ( 'Cirrose', 'Hepatite' )
22	AND sexo IN ( 'Feminino', 'Masculino' )
23	GROUP BY group_id,
24	diagnostico,
25	sexo
26	ORDER BY group_id,
27	diagnostico,
28	sexo

Fonte: elaborado pelo autor.

Para divisão dos grupos ao executar a consulta SQL, adicionou-se de forma estática os campos `group_id`, presentes nas linhas 1 e 14. Responsabiliza-se os valores fixos e distintos dos campos `group_id` pela diferenciação dos grupos no resultado da consulta. Para divisão total dos grupos de dados, adiciona-se nas linhas 7 e 20 as restrições opostas a partir do atributo divisor. Já para os atributos restantes utilizados para correlação, adiciona-se à cláusula `select` os nomes das colunas correspondentes no BD. Restringe nas linhas 8, 9, 21 e 22 os valores das propriedades dos atributos para correlação definidos. Após a finalização da construção, executa-se a consulta SQL e a partir de seu resultado, monta-se a estrutura da árvore de correlação. Na Figura 34, apresenta-se o resultado do comando SQL executado a partir do ambiente de desenvolvimento integrado (do inglês Integrated Development Environment – IDE) disponibilizada pelo banco de dados H2.

Figura 34 – Resultado da execução da consulta ao BD

GROUP_ID	DIAGNOSTICO	SEXTO	COUNTER
1	Cirroze	Feminino	1
1	Cirroze	Masculino	11
2	Cirroze	Feminino	6
2	Cirroze	Masculino	29
2	Hepatite	Feminino	8

Fonte: elaborado pelo autor.

Verifica-se que no retorno da consulta define-se o grupo de cada linha a partir do rótulo `group_id`. Alocam-se para o grupo selecionado as linhas que possuem `group_id` com valor 1, já para o grupo restante, utiliza-se as linhas com o valor 2. Retorna-se na coluna `counter` a quantidade de pacientes agrupados pelos campos `diagnostico` e `sexo`. Estrutura-se a árvore de correlação após o retorno da consulta, para isso, definiu-se uma estrutura de árvore a partir da classe `OutputColumn`. O Quadro 18 representa a estruturação em código-fonte da classe `OutputColumn`.

Quadro 18 – Estrutura da classe `OutputColumn`

1	<code>public class OutputColumn implements Serializable {</code>
2	
3	<code>    private String label;</code>
4	<code>    private String columnName;</code>
5	<code>    private double value;</code>
6	<code>    private double totalValue;</code>
7	<code>    private List&lt;OutputColumn&gt; values;</code>
8	<code>    private double support;</code>
9	<code>    private double confidence;</code>
10	<code>    private InputColumn.CalculationType calculationType;</code>

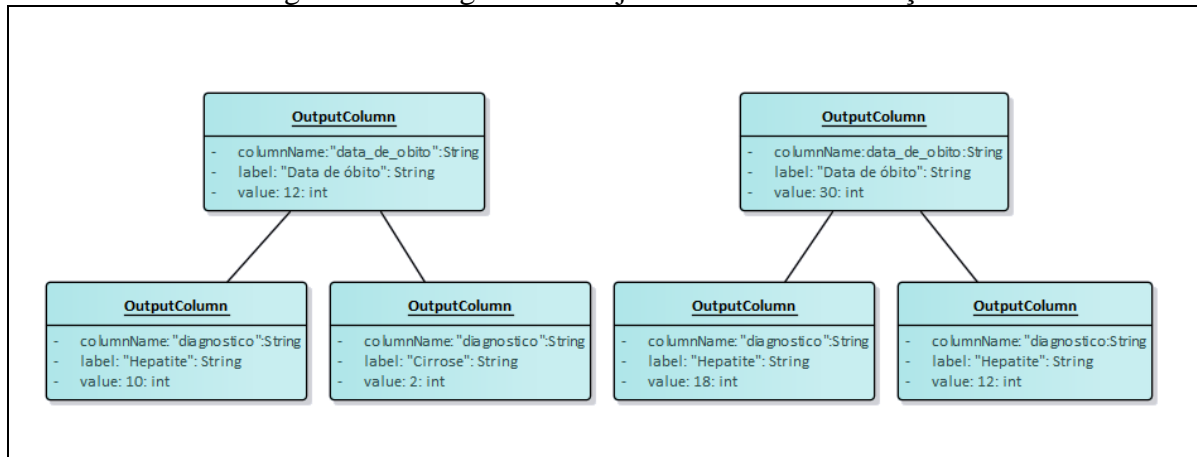
Fonte: elaborado pelo autor.

Corresponde-se a um nó da árvore de correlação cada objeto `OutputColumn` instanciado. Em cada objeto, possui-se uma lista de nós filhos, ou seja, uma lista de objetos de `OutputColumn` (linha 7), resulta-se a partir dessa estrutura, uma árvore. Para correlação,



duplica-se a estrutura da árvore gerada, onde cada nível, pode-se confrontar os valores dos nós. Apresenta-se na Figura 35 um diagrama de objetos para facilitar no entendimento da estrutura da árvore.

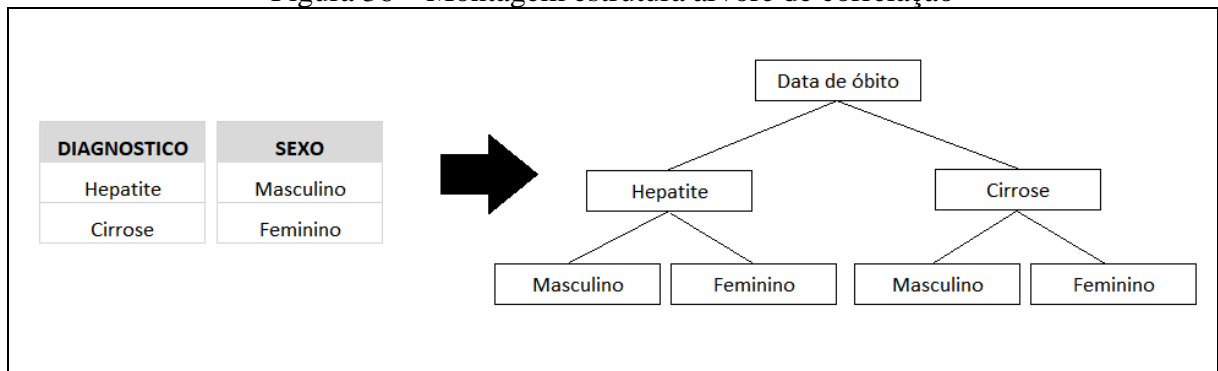
Figura 35 – Diagrama de objetos árvore de correlação



Fonte: elaborado pelo autor.

Para construção da árvore com objetos de `OutputColumn`, adiciona-se uma coluna de valores distintos a uma lista a partir do retorno da consulta, criando-se assim, uma matriz onde que o número de linhas sofre variação de coluna para coluna. Posteriormente, transforma-se em um nó cada um dos valores da matriz. Para cada valor, transforma-se a coluna à direita como filhos, gerando assim uma estrutura de árvore que atende a correlação para os grupos distintos de dados. A Figura 36 demonstra a transformação dos valores distintos do retorno da consulta em uma árvore considerando a coluna à esquerda como filhas da direita.

Figura 36 – Montagem estrutura árvore de correlação



Fonte: elaborado pelo autor.

No Quadro 19, apresenta-se o código-fonte que gera a estrutura da árvore a partir da lista dos valores distintos.

Quadro 19 – Código-fonte árvore de correlação

```

1 private void createTreeColumnsRecursive(final int index, final
2   OutputColumn outputColumn) {
3     if (index >= this.distinctValues.size()) {
4       return;
5     }
6
7     String[] values = this.distinctValues.get(index).toArray(new
8   String[0]);
9     String columnName = this.orderColumns.get(index).getColumnName();
10    String label;
11
12    for (int i = 0; i < values.length; i++) {
13      label = values[i];
14      OutputColumn o = new OutputColumn();
15      o.setLabel(label);
16      o.setColumnName(columnName);
17      outputColumn.addSon(o);
18    }
19
20    for (OutputColumn out : outputColumn.getValues()) {
21      createTreeColumnsRecursive(index + 1, out);
22    }
23  }

```

Fonte: elaborado pelo autor.

Ao chamar pela primeira vez o método `createTreeColumnsRecursive()` (linha 2), deve-se passar o objeto raiz da árvore de `OutputColumn` e o índice inicial da coluna com os valores distintos (linha 1). Verifica-se sempre se o índice de colunas ultrapassou a quantidade de colunas distintas (linha 3), caso sim, não há mais filhos para criar e adicionar na árvore (linha 4). Cria-se os filhos previamente adicionando-os na lista de filhos do nó atual (linha 17). Após toda a inserção dos filhos, chama-se recursivamente o método `createTreeColumnsRecursive()`, passa-se cada filho como parâmetro e incrementa-se o índice das colunas de valores distintos (linha 21).

Com a estrutura da árvore pronta, ocorre a rastreabilidade dos valores a partir do resultado a consulta dinâmica executada anteriormente. Para isso, atribui-se o valor da coluna `counter` ao seu respectivo nó. Com a árvore estruturada e com valores, limpa-se a árvore removendo níveis com valores totalmente zerados. Objetiva-se em poluir a tabela de correlação com linhas sem relevância, devido a seus valores zerados.

Além da correlação por tabela, disponibiliza-se também a visualização das correlações por gráfico. Ao acessar a aba “Gráficos”, verifica-se que o campo de seleção, com os atributos utilizados para a correlação, e abaixo, apresenta-se o gráfico, conforme Figura 37.

Figura 37 – Layout gráfico de barras



Fonte: elaborado pelo autor.

Apresenta-se em cada barra no gráfico, a quantidade de pacientes. As barras agrupam-se subdividindo entre o grupos selecionados e grupos restantes. Os valores que são apresentados no gráfico, são de forma totalitária, ou seja, percorre-se a árvore de correlação agrupando as informações e somando seus valores conforme atributo selecionado, após essa tratativa, atualiza-se os valores do gráfico. No exemplo da Figura 38, selecionou-se a visualização por “Diagnóstico”, na atualização do gráfico, soma-se no protótipo as quantidades de cada diagnóstico em cada grupo e agrupa-se as informações.

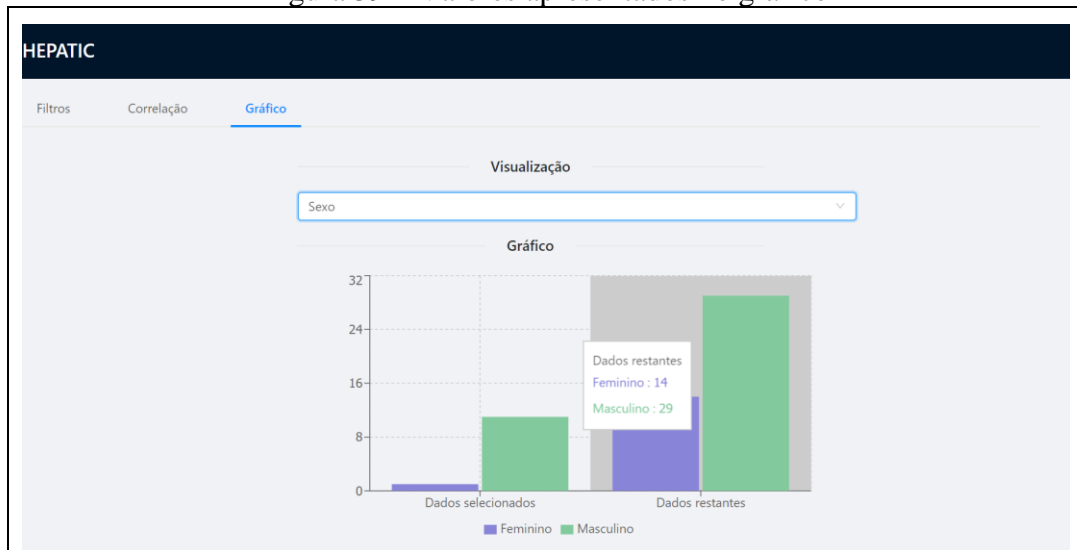
Figura 38 – Forma de visualização a partir da correlação



Fonte: elaborado pelo autor.

Demonstra-se também na Figura 39, a visualização pela opção *Sexo*. Nota-se que ao apresentar a tabela de correlação no item 3.5.3, o atributo *Sexo* encontrava-se um nível abaixo de *Diagnóstico*, no gráfico, ignora-se o diagnóstico e apresenta-se a soma dos valores dos *SEXOS* Masculino e Feminino.

Figura 39 – Valores apresentados no gráfico



Fonte: elaborado pelo autor.

Para montagem do gráfico, utiliza-se a biblioteca Recharts disponível para o framework React, em JavaScript. Monta-se os valores dos dados a partir da tabela de correlação, ou seja, não há requisição para o servidor, devido todas as informações já estarem em formato JSON no Front-end.

### 3.5 ANÁLISE DOS RESULTADOS

Esse capítulo apresenta os resultados obtidos na execução do protótipo. A seção 3.5.1 demonstra os resultados alcançados pelo Hepatic em relação a correlação dos dados. Na seção 3.5.2 são apresentados os resultados e discussões gerais do protótipo.

#### 3.5.1 Resultados da correlação dos dados

Para a análise dos resultados, confrontou-se a correlação gerada pelo Hepatic com o levantamento manual. Para isso, extraiu-se do MySQL para uma planilha normalizada os dados dos 158 pacientes. Na sequência, importou-se a planilha no protótipo, dando início aos testes de correlação do Hepatic.

No cálculo do percentual de acerto, utilizou-se a fórmula  $((\text{óbito manual} - \text{óbito Hepatic}) / \text{óbito manual}) + ((\text{sobrevida manual} - \text{sobrevida Hepatic}) / \text{sobrevida manual})$ . A partir desta fórmula, obteve-se em percentual as diferenças de valores entre o levantamento manual e o Hepatic. A Tabela 1 demonstra o confronto entre os dados gerados pelo Hepatic e os levantados de forma manual.

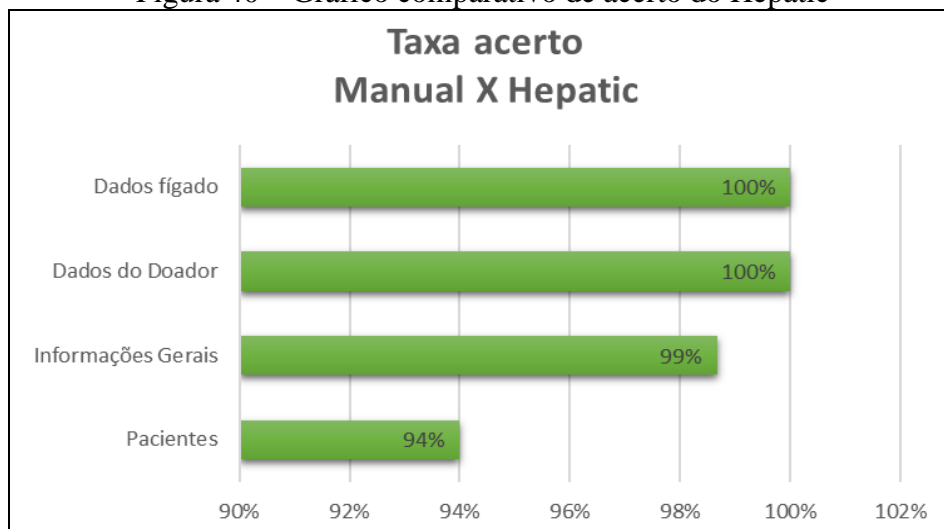
Tabela 1 – Resultados levantamento manual *versus* Hepatic

Informações Gerais	Levantamento Manual		Gerados pelo Hepatic		% Acerto
	Óbito	Sobrevida	Óbito	Sobrevida	
Masculino	27	85	27	85	100%
Feminino	9	37	9	37	100%
Tiveram Sangramento	28	79	29	79	96%
<b>Dados do Doador</b>					
Idade	40,16	42,18	40,17	42,18	100%
<b>Dados do Fígado</b>					
Sódio	153,97	151,36	153,97	151,37	100%
BTF	0,78	0,64	0,78	0,64	100%
TGO	97,49	97,81	97,49	97,81	100%
TGP	85,58	102,34	85,58	102,34	100%
GGT	79,81	85,99	79,81	85,99	100%
<b>Pacientes</b>					
MELD	16,41	14,95	16,42	14,96	100%
Tempo de isquemia	435	492	435,78	492,88	100%
Duração da cirurgia	313,00	303,00	261,25	273,25	74%
Plaquetas	1,65	0,72	1,61	0,73	99%
Cell Saver	445,68	153,75	433,31	152,49	96%
Espera transplante	37,86	68,36	36,67	57,97	82%
Idade	58	55	58	55	100%
Peso	82,44	76,9	82,44	76,91	100%

Fonte: elaborado pelo autor.

No total, obteve-se uma taxa de correlação de 97% em relação ao Hepatic e a forma manual. Para uma análise detalhada, apresenta-se graficamente na Figura 40 a taxa de acertos agrupado pelo tipo de informações.

Figura 40 – Gráfico comparativo de acerto do Hepatic



Fonte: elaborado pelo autor.

Verifica-se a partir da Figura 40 que a taxa de acerto dos pacientes foi a menor em relação aos outros, com 94%. Os itens pertencentes aos pacientes que mais influenciaram para a taxa de acerto são: duração da cirurgia (com uma taxa de acerto de 74%) e espera do transplante (82% de taxa de acerto). Verificou-se que a causa da taxa destes itens decorreu-se a partir de erros de migração da massa de dados para o banco MySQL, gerando incompatibilidade em algumas informações do banco de dados em relação com a planilha de controle de pacientes. Outro fator que influenciou foi a desconsideração dos valores “0” para o cálculo da média no Hepatic. De maneira geral, os resultados obtidos foram satisfatórios e o Hepatic se mostrou uma ferramenta prestativa para auxílio na extração de informações da base de pacientes.

### 3.5.2 Resultados e discussões gerais

A partir dos problemas apontados na seção 3.1, levantou-se as necessidades do centro de transplantes do Santa Isabel. Para concretizar os requisitos, obteve-se o acesso amostral de dados de 158 pacientes da planilha de controle do centro de transplante. O objetivo foi conhecer e integrar-se nos dados técnicos, procedimentos e suas correlações, a fim de realizar os levantamentos estatísticos dos pacientes e aproximar os requisitos do protótipo para a resolução dos atuais problemas.

Com os dados da planilha, iniciou-se um trabalho de normalização e migração de suas informações para o banco de dados relacional MySQL. No processo migratório, foram inseridos na base do BD todos os dados e eventos relacionados aos 158 pacientes amostrais repassados inicialmente pelo centro de transplantes. Deu-se como objetivo desta migração, permitir a extração de informações de forma assertiva para correlação posterior, permitindo a realização de um levantamento de diferenças de informações entre grupos dos pacientes que morreram e os que tiveram sobrevida.

Realizou-se um trabalho de extração manual das informações da base de dados migrada juntamente com as informações originalmente contidas na planilha de controle de pacientes. Para realização da correlação, foram divididos dois grupos em uma tabela, que são: pacientes que vieram a óbito e os que tiveram sobrevida. Para correlação, adicionou-se uma linha para cada evento, procedimento, técnicas ou dados pessoais, com o objetivo de confrontar de forma geral os números entre óbito e sobrevida. O levantamento realizado é exibido na Tabela 2.

Tabela 2 – Correlação estatístico dos pacientes

	<b>Óbito (36)</b>	<b>Sobrevida (122)</b>
<b>Informações Gerais</b>	Quantidade	Quantidade
Masculino	27	85
Feminino	9	37
Infecção	20	54
Tiveram Sangramento	28	79
<b>Diagnóstico</b>	-	-
Cirrose	18	56
Hepatocarcinoma	6	37
Retransplante	6	3
Hepatite	1	9
Outros	3	12

Fonte: elaborado pelo autor.

Os levantamentos dos dados estatísticos prolongaram-se para a correlação dos dados técnicos, cirúrgicos, informações sobre os fígados transplantado e dados dos doadores. A Tabela 3 apresenta a correlação de médias dos doadores como: IMC, Sódio, BTF, TGO, TGP, GGT e médias clínicas como: MELD, tempo de isquemia (sem distinção de isquemia fria e quente), tempo de cirurgia, número de plaquetas, procedimento Cell Saver, tempo de espera na fila de transplante, idade e peso.

Tabela 3 – Correlação dos dados dos doadores e pacientes

	<b>Óbito (36)</b>	<b>Sobrevida (122)</b>
<b>Dados do Doador</b>	<b>Média</b>	<b>Média</b>
Idade	40,16	42,18
IMC	26,68	26,38
<b>Dados fígado</b>	-	-
Sódio	153,97	151,36
BTF	0,78	0,64
TGO	97,49	97,81
TGP	85,58	102,34
GGT	79,81	85,99
<b>Pacientes</b>	-	-
IMC	28,96	26,73
MELD	16,41	14,95
Tempo de isquemia	07:15	08:12
Duração da cirurgia	05:13	05:03
Plaquetas	1,65	0,72
Cell Saver	445,68	153,75
Espera transplante	37,86	68,36
Idade	58	55
Peso	82,44	76,9

Fonte: elaborado pelo autor.

Com a correlação, possibilitou-se a extração de possíveis causas que podem contribuir para a taxa de óbito no centro de transplantes do hospital Santa Isabel. O objetivo foi analisar o confronto entre os dois grupos (óbito e sobrevida), para posterior enumeração dos resultados da análise. Para validação, levou-se os resultados da análise da correlação para a médica responsável pelo centro de transplantes. O objetivo foi verificar se os itens levantados estavam coerentes dentro da perspectiva medicinal. Segue abaixo a enumeração dos 12 itens da análise do confronto entre o grupo de óbitos e sobrevidas, levantados a partir da correlação dos dados dos transplantes:

- a) todos os retransplantados vieram à óbito;
- b) a taxa de infecção é 33% maior no grupo de óbito;
- c) dos 122 pacientes com sobrevida, somente 7 realizaram Laparotomia, já nos 36 obituários, o número de pacientes que realizaram o procedimento chega a 11;
- d) taxa de utilização do Cell Saver é 190% maior no grupo que veio à óbito;
- e) índice MELD está em média dois pontos acima no grupo de óbito;
- f) taxa de sangramento é 13% maior no grupo que veio à óbito;
- g) média de TGP 16,25 mais baixa no grupo que veio à óbito;
- h) 44% dos óbitos ocorreram por choque;
- i) 1h à menos em média de isquemia no grupo de óbito;
- j) BTF do fígado está numericamente 0,14 a maior no grupo de óbito;
- k) TGP do fígado está numericamente 16,76 mais baixa no grupo de óbito.

Todos os 12 itens foram validados com o auxílio da médica responsável pelo centro de transplantes. O item sobre o tempo de isquemia (i) foi invalidado, isto porque pelo aspecto clínico, não é lógico o tempo ser inferior no grupo de óbito. Explicou-se que não há sentido em deixar mais tempo o fígado em solução fria para que se aumente a probabilidade de sobrevida. Retirou-se também conclusões referentes ao BTF (j) e TGP (k) devido sua variação não apresentar diferenças significativas. Conclusões referentes a infecção (b), retransplante (a), utilização do Cell Saver (d) e sangramento na cirurgia (f) validou-se como possíveis contribuições para a taxa de óbito do ponto de vista medicinal.

O conhecimento das informações do controle dos pacientes nos apresentou de forma clara que a necessidade prioritária do centro de transplante era a automatização da correlação de dados, que atualmente são armazenados na planilha. Descartou-se a aplicação dos métodos de descoberta de conhecimento em banco de dados (do inglês *Knowledge Discovery in Databases* – KDD) na base de dados, devido ao centro de transplantes já possuir um projeto



paralelo que visa automatizar o cadastro e consulta dos pacientes transplantados (elimina etapas como seleção e limpeza dos dados no processo KDD). Outra premissa que invalida a utilização do KDD é o conhecimento prévio sobre os dados.

Pensou-se então na construção de um protótipo para correlação dos dados de forma interativa, que permite ao usuário (especialista da área médica) a montagem de toda a estrutura de correlação. No protótipo, o usuário poderá dividir os dados em dois grupos, e possuirá controle total sobre os atributos a serem aplicados nesses dois grupos para correlação. Verificou-se que o protótipo deve permitir a adequação a diferentes situações do presente e futuro, isto porque no futuro, novos dados técnicos e eventos podem virar relevantes e serem adicionados ao escopo de controle do centro de transplante, o levantamento de requisitos do protótipo deverá abordar essa premissa.

## 4 CONCLUSÕES

O objetivo deste trabalho era desenvolver um protótipo de software para correlacionar os dados a partir de históricos clínicos para descobrir fatores que podem influenciar ou contribuir com o obituário de pacientes hepáticos transplantados. Para chegar ao objetivo, obteve-se o histórico de 158 pacientes do centro de transplantes do Hospital Santa Isabel.

Na construção do protótipo foi fundamental o auxílio dos profissionais do centro de transplantes do Hospital Santa Isabel. A partir da parceria, modelaram-se os requisitos para atender aos problemas relacionados a análise da massa de dados dos pacientes. Nesta etapa, uns dos principais desafios decorrente ao tempo e escopo do trabalho, foi a normalização e a limpeza da base destes 158 pacientes.

Para o armazenamento dos históricos dos pacientes pelo Hepatic, utilizou-se o banco de dados H2. Sua principal vantagem é a persistência de dados em memória principal, que proporciona acesso rápido aos registros e alto desempenho nas consultas. Outro benefício deu-se ao fato que o H2 não utiliza espaço em disco durante a execução do protótipo. Para a construção das telas, utilizou-se as bibliotecas Ant Design e Recharts, do framework de JavaScript React. Sua utilização proporcionou a criação de um ambiente intuitivo e de fácil usabilidade para realização dos mais variados tipos de correlações. Não se utilizou ferramentas para geração da árvore de correlação, pois objetivou-se em direcionar a correlação do protótipo para permitir a implementação futura de eventuais necessidades do centro de transplantes do Santa Isabel.

Uma das principais vantagens em relação ao protótipo, está na facilidade de correlação e nas fortes possibilidades de correlação que o Hepatic proporciona. A partir do protótipo, pode-se correlacionar fatos não só direcionados aos óbitos dos pacientes, mas confrontar dados para extração de outras informações ou eventos de pacientes requeridos pelo centro de transplantes. Também se vê como uma vantagem, a capacidade do protótipo de se adequar a mudanças de processos e controle de dados que podem futuramente vir a ocorrer no centro de transplantes do Santa Isabel.

Contudo, o protótipo ainda se limita em atender todas as necessidades que o centro de transplantes do Santa Isabel possui, dos quais pode-se citar: análise estatística da correlação a partir de testes T-Student e Qui-quadrado, permitir a definição de múltiplos grupos de correlação, gráficos, entre outros. A partir disso, concluiu-se que o Hepatic atingiu seu objetivo, ou seja, ele pode auxiliar na identificar causas que podem influenciar no obituário dos pacientes, porém ainda falta amadurecimento.

#### 4.1 EXTENSÕES

O protótipo não atende atualmente todas as necessidades do centro de transplantes do Hospital Santa Isabel. Para atendê-las, segue abaixo algumas sugestões de extensões para o protótipo:

- a) permitir a divisão de múltiplos grupos de dados para correlação;
- b) implementar os testes estatísticos de T-Student e Qui-quadrado;
- c) implementar a visualização de correlações para vários tipos de gráfico;
- d) implementar a escolha de visualização de média, quantidade ou soma na correlação e simplificar a escolha das propriedades e seus valores.

## REFERÊNCIAS

- FEIJOO, Ana Maria Lopez Calvo de. Correlação. In: FEIJOO, Ana Maria Lopez Calvo de. **A pesquisa e a estatística na psicologia e na educação**. Rio de Janeiro: Centro Edelstein, 2010. p. 70-77. Disponível em: <<http://books.scielo.org/id/yvnmwq/pdf/fejoo-9788579820489-11.pdf>>. Acesso em: 19 jul. 2018.
- FERREIRA, Cristina Targa; VIEIRA, Sandra Maria Gonçalves; SILVEIRA, Themis Reverbel da. Transplante hepático. **Jornal de Pediatria**, Porto Alegre, v. 76, n. 2, p.1-11, jan. 2000. Disponível em: <<https://lume.ufrgs.br/bitstream/handle/10183/54621/000335420.pdf?sequence=1>>. Acesso em: 19 jul. 2018.
- FONSECA, Paula Isabella Marujo Nunes da; TAVARES, Claudia Mara de Melo. O manejo das emoções dos coordenadores em transplantes na realização da entrevista familiar para doação de órgãos. **Revista Portuguesa de Enfermagem de Saúde Mental**, Porto, n. spe2, p. 39-44, fev. 2015 . Disponível em: <[http://www.scielo.mec.pt/scielo.php?script=sci\\_arttext&pid=S1647-21602015000100007&lng=pt&nrm=iso](http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S1647-21602015000100007&lng=pt&nrm=iso)>. acessos em 19 jul. 2018.
- GARCIA, Valter Duro. **Dimensionamento dos Transplantes no Brasil e em Cada Estado**. São Paulo: Abto - Associação Brasileira de Transplante de Órgãos, n. 4, 2017. Disponível em: <<http://www.abto.org.br/abtov03/Upload/file/RBT/2017/rbt-imprensa-leitura-compressed.pdf>>. Acesso em: 11 jul. 2018.
- HUMMEL, Anderson Diniz. **Aplicação de técnicas de mineração de dados ao desenvolvimento de um sistema de apoio à decisão clínica para acompanhamento do pós-transplante renal**. 2011. 90 f. Tese (Doutorado) - Curso de Ciências, Departamento de Informática em Saúde, Universidade Federal de São Paulo, São Paulo, 2012. Disponível em: <<http://telemedicina.unifesp.br/pub/unifesp/teses/2012-AndersonDinizHummel-mestrado.pdf>>. Acesso em: 26 mar. 2018.
- LIRA, Sachiko Araki. **Análise de correlação: Abordagem Teórica e de Construção dos Coeficientes com Aplicações**. 2004. 209 f. Dissertação (Mestrado) - Curso de Engenharia dos Setores de Ciências Exatas e de Tecnologia, Universidade Federal do Paraná, Curitiba, 2004. Disponível em: <[http://www.ipardes.gov.br/biblioteca/docs/dissertacao\\_sachiko.pdf](http://www.ipardes.gov.br/biblioteca/docs/dissertacao_sachiko.pdf)>. Acesso em: 11 jul. 2018.
- MARINHO, Alexandre. Um estudo sobre as filas para transplantes no Sistema Único de Saúde brasileiro. **Cad. Saúde Pública**, Rio de Janeiro , v. 22, n. 10, p. 2229-2239, Outubro. 2006 . Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-311X2006001000029&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2006001000029&lng=en&nrm=iso)>. Acesso em: 19 jul. 2018.
- MIES, S. Transplante de fígado. **Revista da associação médica brasileira**, São Paulo, v. 44, n. 2, p. 127-134, Junho 1998. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-42301998000200011&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-42301998000200011&lng=en&nrm=iso)>. Acesso em: 15 fev. 2018.
- MIGUEL, Gustavo Peixoto Soares et al. **Manual do paciente candidato ao transplante de fígado**. 2007. Equipe de Transplante Hepático do Hospital Meridional. Disponível em: <[http://www.doencasdofigado.com.br/Manual do candidato a transplante de figado- ES.pdf](http://www.doencasdofigado.com.br/Manual%20do%20candidato%20a%20transplante%20de%20figado-ES.pdf)>. Acesso em: 01 mar. 2018.

- MIRANDA, Mara Lúcia de. **Correlação e Regressão em Curso de Engenharia: Uma Abordagem com Foco na Leitura e Interpretação de Dados.** 2008. 160 f. Dissertação (Mestrado) - Curso de Ensino de Ciências e Matemática, Programa de Pós-graduação em Ensino de Ciências e Matemática, Universidade Católica de Minas Gerais, Belo Horizonte, 2008. Disponível em: <[http://www1.pucminas.br/imagedb/documento/DOC\\_DSC\\_NOME\\_ARQUI20150313114827.pdf](http://www1.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20150313114827.pdf)>. Acesso em: 11 jul. 2018.
- NUNES, Pedro Pimentel; MOREIRA, Doutor Adelino Leite. **Fisiologia hepática: Texto de Apoio.** 2007. Faculdade de Medicina da Universidade do Porto, Serviço de Fisiologia. Disponível em: <<http://www.doencasdofigado.com.br/fisiologia-hepatica.pdf>>. Acesso em: 01 mar. 2018.
- OLIVEIRA, Alexander Flávio de. **Modelo de simulação de fila de transplantes de fígado baseado em sistemas multiagentes.** 2010. 101 f. Dissertação (Mestrado) - Curso de Informática, Programa de Pós-graduação em Informática, Universidade Católica de Minas Gerais, Uberlândia, 2010. Disponível em: <[http://www.biblioteca.pucminas.br/teses/Informatica\\_OliveiraAF\\_1.pdf](http://www.biblioteca.pucminas.br/teses/Informatica_OliveiraAF_1.pdf)>. Acesso em: 26 mar. 2018.
- PACHECO, Eduardo et al. Transplante de fígado: indicação e sobrevida. **Acta Cir. Bras.**, São Paulo, v. 17, supl. 3, p. 83-91, 2002. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-86502002000900018&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-86502002000900018&lng=en&nrm=iso)>. Acesso em: 19 jul. 2018.
- PINTO, Andressa dos Santos. **Efeito da dieta individualizada e do aconselhamento dietético na dislipidemia em pacientes pós-transplante hepático.** 2016. 87 f. Dissertação (Mestrado) - Curso de Medicina, Medicina, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/148852>>. Acesso em: 03 mar. 2018.
- SANTOS, Luciana Rocha dos. **Protótipo de um sistema especialista educativo para diagnóstico médico.** 2010. 85 f. TCC (Graduação) - Curso de Ciência da Computação, Ciência da Computação, Centro Universitário Vila Velha, Vila Velha, 2010.
- SENS, Monica Justino. **A criatividade sob a ótica do processo comportamental.** 1998. 92 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1998. Disponível em: <<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/77409/138770.pdf?sequence=1&isAllowed=y>>. Acesso em: 08 mar. 2018.
- VIEIRA, Sander Glauco Melo et al. Patologias do fígado: patologia sistêmica, [s.d.]. 9f. Notas de Aula. Mimeografado. Disponível em: <<http://disciplinas.famerp.br/Patologia/Documentos%20%20Fgado/Resumo%20-%20F%C3%ADgado.pdf>>. Acesso em: 19 out. 2018.