

USO DA PERSISTÊNCIA POLIGLOTA E DE ARQUÉTIPOS NO ARMAZENAMENTO DO REGISTRO ELETRÔNICO DE SAÚDE

USE OF POLYGLOT PERSISTENCE AND ARCHETYPES IN THE ELECTRONIC HEALTH RECORD STORAGE

André Magno Costa de ARAÚJO¹, Valéria Cesário TIMES²

1 Universidade Federal de Alagoas, Departamento de Sistemas de Informação, Penedo, Professor Adjunto (DE) da Universidade Federal de Alagoas, possui doutorado e mestrado em Ciência da Computação pela Universidade Federal de Pernambuco. Atuou como Analista de Sistemas e Coordenador de TI na área Informática em Saúde de 2003 a 2014. Desde 2002 é professor de ensino superior em cursos de Graduação e Pós-Graduação. Tem experiência na área de Ciência da Computação, com ênfase em Linguagens de Programação, Banco de Dados, Engenharia de Software, Modelagem de Dados e Desenvolvimento de Sistemas. E-mail: andre.araujo@penedo.ufal.br.

2 Universidade Federal de Pernambuco, Centro de Informática, Recife, Brasil. Possui graduação pela Universidade Católica de Pernambuco (1991), mestrado em Ciências da Computação pela Universidade Federal de Pernambuco (1994) e doutorado em Ciência da Computação (PhD on Computer Science) pela Universidade de Leeds (School of Computer Studies at University of Leeds) (1999). Participou de um programa de pós-doutorado na University of Ottawa, no Canadá, de 2013 a 2015. Atualmente é professor Associado II da Universidade Federal de Pernambuco. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Informação, atuando principalmente nos seguintes temas: data warehouse, banco de dados geográficos, sistemas de informação em saúde, sistemas de informações geográficas, segurança e privacidade de dados, sistemas NoSQL, banco de dados agregados e ferramentas OLAP. E-mail: yct@cin.ufpe.br.

RESUMO: Este artigo apresenta uma ferramenta computacional capaz de criar dinamicamente esquemas de dados relacionais e NoSQL para o armazenamento do Registro Eletrônico em saúde. Os esquemas de dados são gerados a partir dos atributos de dados, das terminologias e das restrições especificadas em um padrão em saúde chamado de arquétipo openEHR. Uma arquitetura de software foi projetada para que os esquemas de dados gerados pela ferramenta não tenham dependência de uma tecnologia específica de armazenamento. Para validar a solução proposta, utilizaram-se arquétipos disponíveis no repositório da openEHR para gerar esquemas de dados em diferentes sistemas de gerenciamento de banco de dados.

Palavras-Chave: Persistência poliglota. Sistemas de Informação em Saúde. Arquétipos. Registro Eletrônico de Saúde.

ABSTRACT. This paper presents a computational tool to dynamically create relational and NoSQL data schemas for the Electronic Health Record storage. Data schemas are generated from the data attributes, terminologies, and constraints specified in a health standard called openEHR archetype. A software architecture has been designed to allow data schemas generation without depending on a specific storage technology. To validate the proposed solution, we have used archetypes available in the openEHR repository to generate data schemas in different database management systems.

Keywords: Poliglot Persistence; Health Information Systems; Archetypes; Electronic Health Records.

1. INTRODUÇÃO

Pesquisas na área de Informática em Saúde mostram que a indústria de software e a academia têm debatido a importância do projeto de banco de dados no ciclo de vida de Sistemas de Informação em Saúde (SIS) (SAALFELD et al, 2017; PEDERSEN et al 2015). Nesse contexto, a criação e a evolução de esquemas de dados para o armazenamento do registro eletrônico em saúde (RES) vêm sendo objeto de estudo de diversas pesquisas (LIN et.al 2016, JUMAA et.al 2013). Segundo a norma ISO / TS 18308 (LEE et. al 2013), o esquema de dados do RES deve armazenar qualquer evento clínico relevante para os cuidados do paciente, incluindo, o registro de descrições textuais, valores numéricos, valores lógicos, expressões de data e hora e estruturas de dados hierárquicas. Além disso, o RES deve armazenar dados em tabelas de modo que as relações entre as linhas e colunas sejam preservadas, e também, permitir o armazenamento de dados na forma de pares de atributos.

O conceito de arquétipos e templates proposto pela openEHR representa um importante padrão em saúde para minimizar os problemas causados pela dificuldade de se modelar a heterogeneidade dos dados do RES e padronizar os atributos de dados, as terminologias e as restrições de um domínio da saúde (LEE et.al 2013). Um arquétipo pode ser definido como uma expressão computacional representada por restrições de domínio que modela e dá significado semântico ao RES, enquanto templates representam interfaces de interação do usuário criadas em tempo de execução a partir das especificações definidas nos arquétipos (BEALE, 2002).

Tradicionalmente, um único sistema de gerenciamento de banco de dados (SGBD) tem sido utilizado para representar os diferentes tipos de dados do RES de certa aplicação de saúde. Essa prática tem sido encontrada em esquemas de dados projetados com e sem o uso de arquétipos (KARAMJIT et. al2015, SELLAMI et. al 2012). Em virtude da variedade de tipos de dados (i.e., dados estruturados e não estruturados) encontra-

dos em um domínio da saúde, fazer uso de um único sistema de armazenamento pode aumentar a complexidade no desenvolvimento de SIS, criar rigidez no esquema de dados (i.e., a alteração no esquema de dados dependerá de uma equipe de desenvolvimento de software) e comprometer o desempenho da aplicação no processamento de dados.

Na área da saúde, o armazenamento do RES em diferentes sistemas de banco de dados torna-se uma alternativa para se representar a heterogeneidade das informações encontradas nas diversas aplicações que compõem um SIS (e.g., diagnóstico e terapia, clínica e assistencial, suprimentos, faturamento e gestão). Nesse sentido, o conceito de persistência poliglota propõe uma abordagem mista de armazenamento combinando as características de consistência da abordagem relacional com a flexibilidade dos modelos de dados NoSQL (i.e., *key-value*, *document* e *graph*).

A persistência poliglota vem sendo aplicada nas mais diversas áreas de conhecimento, como por exemplo, na especificação de uma arquitetura *PaaS* de autoscaling da IBM (SEELAM et. al 2015), na construção de ferramentas para identificação automática de esquemas de dados a partir do código fonte de uma aplicação (KARAJIT et. al 2015) e na reengenharia de bases de dados legadas para modelos de dados heterogêneos (SELLAMI et. al 2015). Embora o conceito de persistência poliglota venha sendo debatido e aplicado na área da saúde, percebe-se a falta de arquiteturas de ferramentas computacionais que persistam dados de aplicações de saúde, construídas a partir de arquétipos, em diferentes sistemas de banco de dados. A principal motivação para o uso de persistência poliglota e de arquétipos é

que: i) esquemas de dados podem ser construídos a partir de um padrão em saúde que uniformiza os requisitos de dados do RES; ii) a representação da heterogeneidade dos requisitos do RES é mantida com o uso de diferentes sistemas de BD e iii) a geração dinâmica de esquemas de dados melhora a manutenibilidade de aplicações de saúde.

Este trabalho apresenta uma ferramenta computacional para geração de esquemas de dados relacionais e NoSQL a partir dos atributos de dados, das terminologias e das restrições especificadas em arquétipos. Para garantir que os esquemas de dados gerados sejam independentes de tecnologia de armazenamento, especificou-se uma arquitetura de software que permite que as regras de negócios da ferramenta acessem a camada de persistência de dados sem que a mesma tenha conhecimento da tecnologia do SGBD utilizada. Um Meta esquema foi proposto para gerar as instâncias de esquemas de dados relacionais, enquanto a geração de esquemas de dados NoSQL é exemplificada por meio de um algoritmo. Por fim, para validar a solução proposta, utilizaram-se arquétipos disponíveis no repositório da openEHR para gerar esquemas de dados em diferentes sistemas de BD.

As demais seções deste artigo estão organizadas da seguinte forma. A Seção 2 descreve os conceitos básicos utilizados no desenvolvimento desta pesquisa e traz uma análise dos principais trabalhos correlatos identificados no estado da arte. A Seção 3 apresenta a ferramenta para geração de esquemas de dados utilizando arquétipos, enquanto a Seção 4 discute os resultados alcançados. Por fim, a Seção 5 traz as considerações finais e sugestões de trabalhos futuros.

2. CONCEITOS BÁSICOS E TRABALHOS

CORRELATOS

2.1. Arquétipos

A modelagem dual proposta pela fundação openEHR objetiva a criação de um RES único, preservando o histórico e a evolução dos dados clínicos do paciente, podendo ser compartilhado e reutilizado por outros modelos de domínios da saúde (JEFFREY et. al 2012). A arquitetura dual possibilita a separação entre as propriedades clínicas e demográficas do RES, dos padrões e das terminologias utilizadas para dar significado semântico aos dados. O primeiro nível da modelagem dual contempla os componentes de linguagem de programação, linguagem de troca de informações e todos os outros componentes relacionados ao desenvolvimento de softwares, enquanto o segundo nível é representado por arquétipos e templates (JEFFREY et. al 2012).

Em um arquétipo, a especificação dos atributos é realizada por meio de construtores de entrada de dados chamados de estruturas de dados genéricas. As estruturas de dados genéricas permitem representar a heterogeneidade dos dados do RES por meio dos tipos, ITEM_SINGLE, ITEM_LIST, ITEM_TREE e ITEM_TABLE.

O tipo ITEM_SINGLE modela um único atributo de dado. Por exemplo, ele pode ser usado para especificar o sexo do paciente. O tipo ITEM_LIST agrupa um conjunto de atributos e é fisicamente representado na forma de uma lista. O endereço de um paciente com os itens, rua, número e CEP, exemplifica a modelagem por meio de um ITEM_LIST. O tipo ITEM_TREE especifica uma estrutura hierárquica de dados que é logicamente representada na forma de uma árvore. Ela pode ser usada, por exemplo, para modelar a avaliação

física e neurológica de um paciente. Por fim, o tipo ITEM_TABLE modela os elementos de dados por meio de linhas e colunas, onde as linhas representam a definição do elemento, e as colunas, o valor da informação. O resultado de um exame de análises clínicas contendo os campos dispostos em linhas, e os valores de referências em colunas, pode ser indicado como exemplo do ITEM_TABLE.

Cada atributo de uma estrutura de dados é caracterizado por um tipo de dado e pode ter ainda um conjunto de restrições de domínio e terminologias associadas. As terminologias dão significado semântico aos dados clínicos e podem ser representadas por meio de uma padronização de termos em saúde ou uma informação textual definida por um especialista do domínio.

A Figura 1 mostra o exemplo de um arquétipo (i.e., family history) que modela os antecedentes familiares de um paciente utilizando um ITEM_TREE. Neste arquétipo, os atributos de dados estão organizados em uma estrutura hierárquica de vários níveis. Além disso, é possível observar na Figura 1, que o atributo de dado *Problem/diagnosis* possui uma terminologia em saúde (i.e., ICD10CM) que padroniza o diagnóstico do paciente e uma restrição (i.e., occurrences) que indica que pelo menos um diagnóstico deve ser informado.

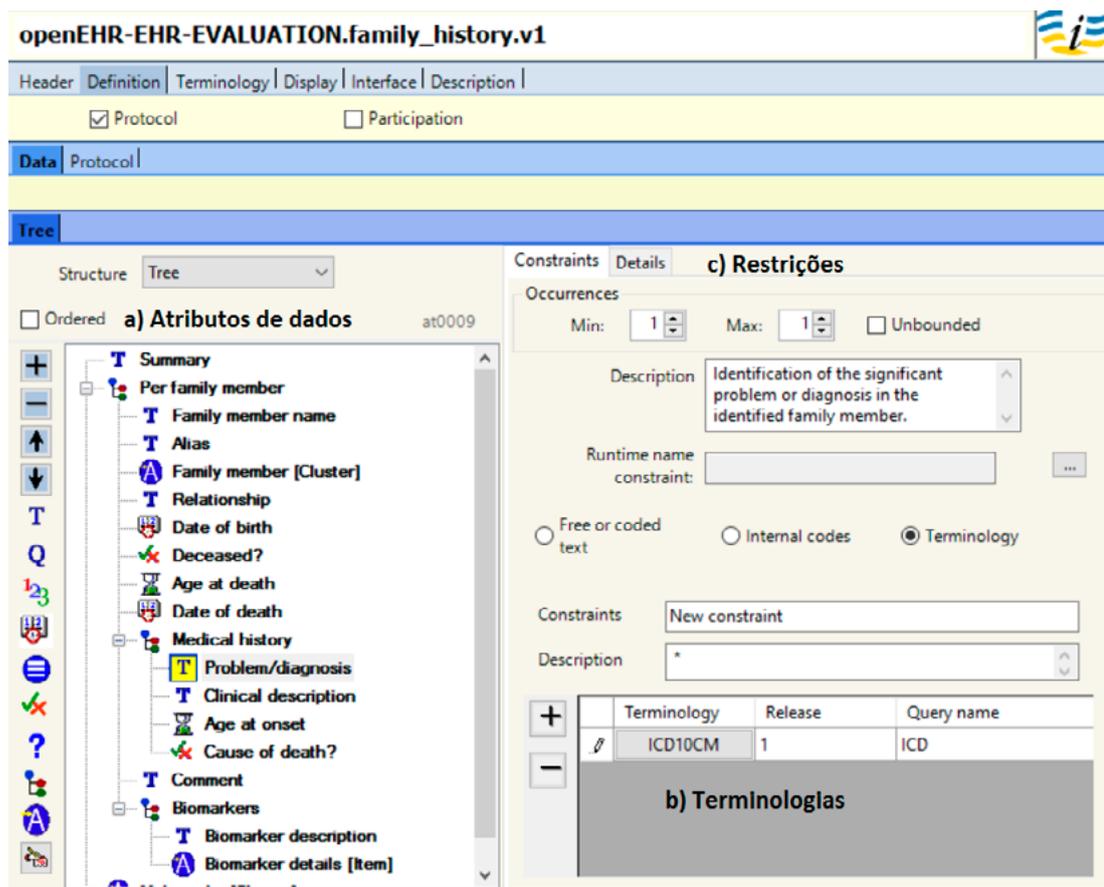


Figura 1. Exemplo da estrutura e organização de um arquétipo.

Arquétipos podem ser descritos e especificados por meio de uma linguagem formal denominada Archetype Definition Language (ADL) (BEALE, 2002). A linguagem ADL descreve restrições baseadas em modelos de conteúdo de domínio e possui três elementos básicos na sua sintaxe: (1) dADL (data ADL) usada para definição de dados, (2) cADL (constraint ADL) possibilita a definição de restrições e (3) FOPL (First-Order Predicate Logic) que se destina à construção de expressões lógicas baseadas em predicados de primeira ordem.

2.2. Trabalhos Correlatos

Esta seção discorre sobre o desenvolvimento de aplicações em saúde baseadas em ar-

quétipos, e sobre mapeamento e persistência de arquétipos em banco de dados.

Späth e Grimson (2010) utilizaram a especificação fornecida pela openEHR para modelar e reestruturar o RES em um sistema legado. Inicialmente, o esquema do banco de dados proprietário foi reestruturado para que cada campo das tabelas do banco de dados estivesse vinculado a um elemento do arquétipo. Os autores indicam que durante o processo de mapeamento dos arquétipos para o banco de dados, diversas dificuldades foram encontradas como, a falta de ferramentas e metodologias que apoiem o processo de modelagem de arquétipos em banco de dados.

De forma similar, Georg, Judith e Christoph (2013), propõem uma abordagem para ma-

pear arquétipos em uma base de dados legada e gerar templates para uma aplicação de saúde chamada de ArchiMed. Um conjunto de regras de mapeamento foi especificado em um algoritmo, de modo a vincular cada elemento de dado do arquétipo com um campo de uma tabela no banco de dados. Os resultados apresentados indicam que a solução proposta não conseguiu mapear os arquétipos que continham estruturas de dados hierárquicas.

Em Wang et al. (2015), uma ferramenta faz o mapeamento de arquétipos em BD relacional. A solução consiste em um conjunto de regras de mapeamento que extraem os atributos de dados dos arquétipos e os armazenam em tabelas e colunas de dados. Além disso, estende-se a linguagem Archetype Query Language (AQL), adicionando os comandos de *insert*, *update* e *delete* para prover a manipulação de dados. Observando o esquema de dados gerado pela ferramenta, percebe-se que as restrições e terminologias especificadas nos arquétipos não são consideradas e não há definição de restrições de integridade referencial entre as tabelas.

Freire et al. (2016) avaliaram o desempenho de sistemas de bancos de dados NoSQL com relação à recuperação de dados clínicos gerados a partir das especificações da openEHR. Dois aspectos principais foram avaliados. Primeiro, o tamanho do espaço alocado para o armazenamento dos dados. Segundo, o tempo de processamento de um conjunto de consultas. Para isso, quatro bancos de dados relacionais legados foram utilizados para gerar dados clínicos nos formatos XML e JSON. Os resultados apresentados mostraram que o espaço de armazenamento de JSON é menor que o espaço correspondente em XML. Além disso, foi verificado o baixo desempenho de docu-

mentos XML em relação a JSON, quando usados no processamento de um conjunto de consultas para recuperação dos dados.

Embora os trabalhos aqui apresentados representem um avanço no desenvolvimento de aplicações em saúde baseadas na arquitetura da openEHR, entende-se que a forma como os arquétipos são mapeados e persistidos em banco de dados necessita evoluir. Além disso, em virtude da heterogeneidade dos requisitos de dados encontrados em um domínio da saúde, identificou-se que pouca atenção tem sido dada na investigação de uma abordagem que crie dinamicamente esquemas de dados para o RES utilizando o conceito da persistência poliglota e de arquétipos.

3. FERRAMENTA PARA GERAÇÃO DE ESQUEMAS DE DADOS

A ferramenta proposta neste trabalho consiste em um ambiente computacional voltado para a construção dinâmica de esquemas de dados a partir da especificação do RES existente nos arquétipos. O ponto principal da abordagem aqui proposta consiste na extração dos atributos de dados, das terminologias e das restrições dos arquétipos para armazenamento do RES em BD relacionais e NoSQL.

Para garantir que os esquemas de dados gerados sejam independentes de tecnologia, utilizou-se um padrão arquitetural que permite que as regras de negócios da aplicação acessem a camada de persistência de dados sem que a mesma tenha conhecimento da tecnologia de SGBD utilizada. Chamado de repositório, este padrão encapsula o conjunto de objetos persistidos na camada de dados e as operações realizadas sobre eles. Além disso, a mediação entre os componen-

tes das camadas de apresentação, regras de negócio, repositório e acesso a dados é realizada por meio de interfaces.

A Figura 2 mostra a arquitetura de software criada para a ferramenta proposta utilizando a notação do diagrama de pacotes da UML e do formalismo do padrão repositório. Nela, é possível observar que os componentes se comunicam por meio de interfaces. A principal vantagem do uso de interfaces é que elas estabelecem a forma de comunicação entre os componentes e permitem que modificações sejam feitas nas regras de negócio sem afetar o funcionamento das entidades externas que interagem com a ferramenta. Por exemplo, é possível que uma aplicação externa esteja consumindo os métodos de geração de esquemas de dados relacionais do componente *DataSchema* (Figura 2), enquanto que novos métodos estejam sendo implementados por meio do

componente *Generator* para criar esquemas de dados NoSQL.

Além disso, a Figura 2 ilustra o funcionamento das interfaces especificadas na arquitetura de software. A interface *IRepository* contém as assinaturas dos métodos de persistência a serem implementadas nos diferentes sistemas de bancos de dados (e.g., Oracle, SQL Server, Cassandra, Neo4J). A interface *IMapper* fornece os contratos de assinatura para que as classes da camada de negócio realizem o mapeamento dos elementos dos arquétipos em esquemas de dados. Por fim, a interface *IDataSource* disponibiliza os parâmetros de conexão com diversos tipos de SGBD.

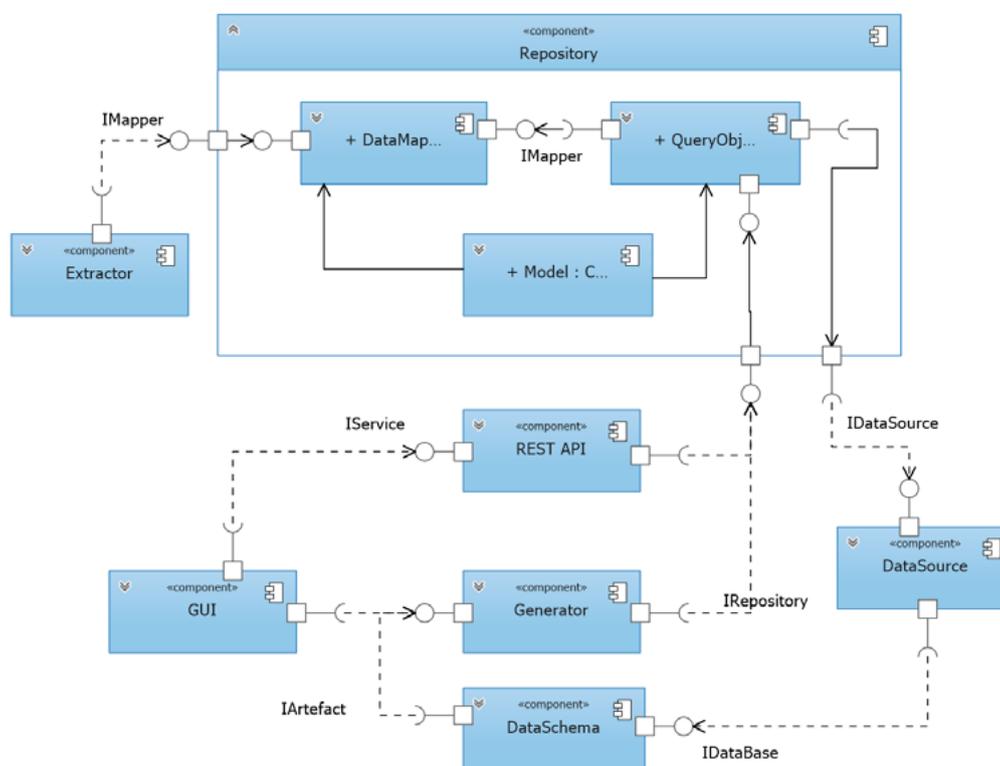


Figura 2. Arquitetura de software da ferramenta proposta.

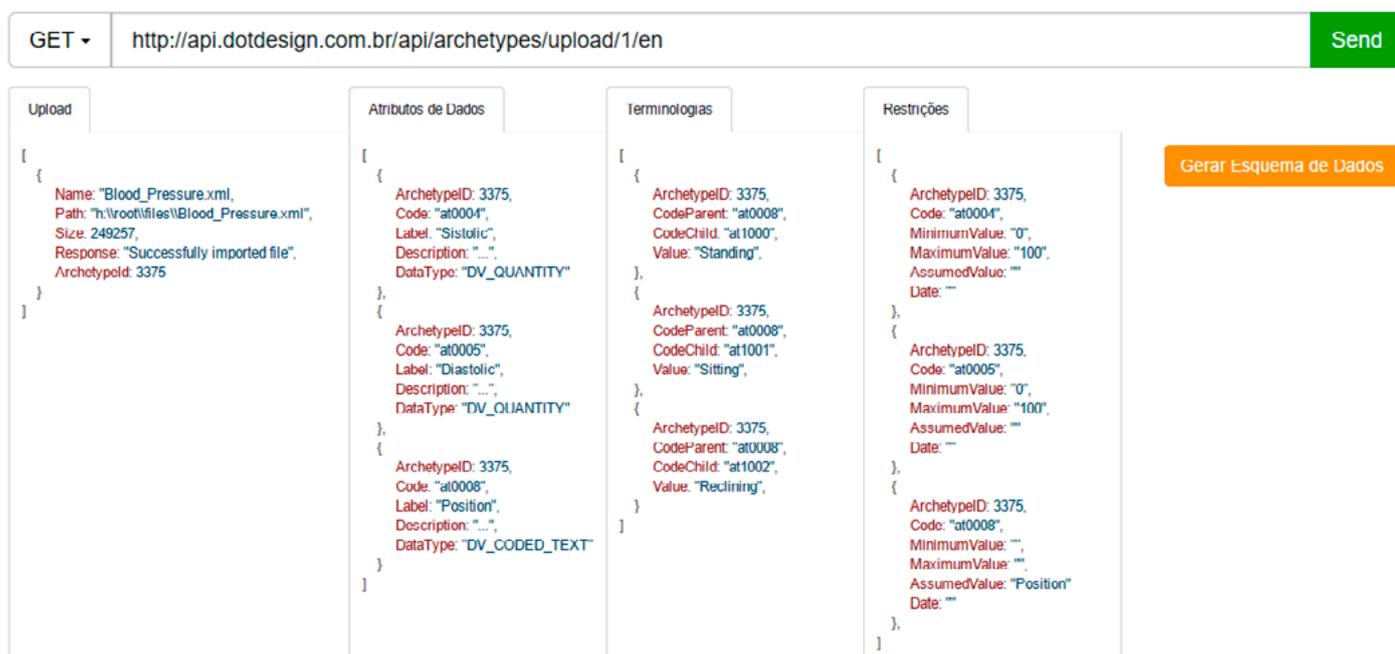


Figura 3. Interface gráfica da ferramenta proposta.

A Figura 3 mostra a interface gráfica e as principais funcionalidades da ferramenta desenvolvida para geração de esquemas de dados. Por meio de uma requisição POST no método upload (), envia-se os arquétipos para a extração dos seus elementos e, após a confirmação do método, o serviço obtém e mostra em formato JSON, os atributos de dados, as terminologias e as restrições dos arquétipos. A seguir, detalha-se como é feita a geração de esquemas de dados relacionais e NoSQL.

3.1. Geração de Esquemas de Dados Relacionais

A geração de esquemas de dados relacionais consiste na criação de tabelas, de restrições de integridade referencial e de campos, a partir da extração dos elementos de arquétipos (i.e.,

atributos de dados, terminologias e restrições). Assim, cria-se o esquema de dados em tempo de execução de acordo com o meta esquema da Figura 4.

O meta esquema proposto é composto por seis tabelas, cinco restrições de integridades referenciais e um conjunto de campos que armazenam os elementos dos arquétipos. A tabela *Archetype* armazena os metadados de configuração do arquétipo informado, como por exemplo, o tipo, o autor e a versão. A tabela *Archetype_Details* armazena o tipo da estrutura de dados e as restrições encontradas no arquétipo, enquanto a tabela *Terminology*, armazena a referência dos atributos de dados com suas respectivas terminologias, caso elas existam.

A especificação dos atributos de dados de um arquétipo pode ser realizada na forma de um único atributo, uma lista vertical de atributos, uma

estrutura de dados hierárquica ou uma tabela com linhas e colunas. Para armazenar as diferentes formas que os atributos de dados são organizados em um arquétipo, utiliza-se as tabelas *Item_Tree* e *Item_Table*. Como a organização de uma estrutura hierárquica já contempla a definição de um único atributo ou um conjunto de atributos mapeia-se os atributos dos tipos ITEM_SINGLE, ITEM_LIST e ITEM_TREE para a tabela *Item_Tree*. Como o nome sugere, a tabela ITEM_TABLE ma-

peia os atributos dos arquétipos que são organizados em linhas e colunas. Uma vez mapeados os atributos de dados em suas respectivas tabelas, o usuário pode escolher quais atributos de dados serão utilizados na geração das interfaces gráficas e, conseqüentemente, quais atributos armazenarão os dados manipulados pelos usuários finais. Nesse caso, os atributos escolhidos são adicionados em tempo de execução, como colunas na tabela *Data_Item*.

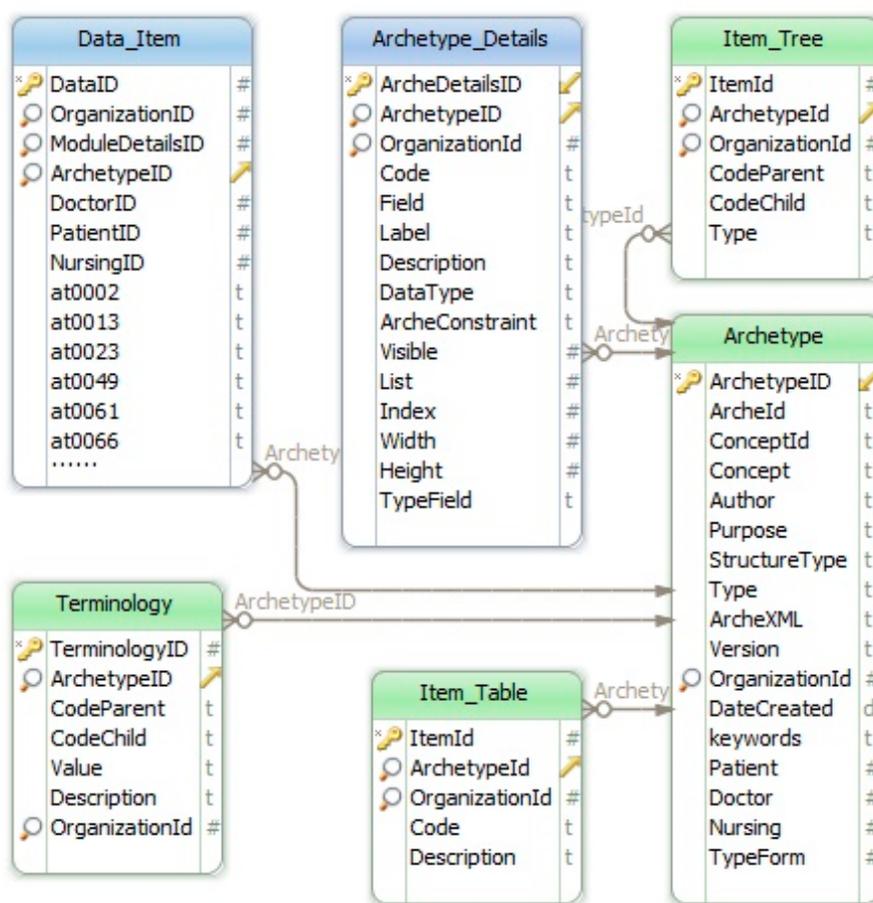


Figura 4. Meta esquema de dados relacional.

Cada elemento de um arquétipo é referenciado por um identificador. Todo identificador é iniciado pelas letras *at* seguido por um valor sequencial. Por exemplo, o identificador *at000* armazena o nome dado a um arquétipo. Utiliza-se a nomenclatura definida por esse identificador para criar o nome dos campos no esquema de dados. Toda vez que um novo campo for inserido na tabela *Data_Item*, a ferramenta verifica se o mesmo já existe, e caso não exista, o campo é adicionado.

3.2. Geração de Esquemas de Dados NoSQL

A partir da identificação do tipo de estrutura de dados utilizada para especificar os atributos de dados de um arquétipo (i.e., ITEM_SINGLE,

ITEM_LIST, ITEM_TREE e ITEM_TABLE), cria-se os esquemas de dados da seguinte forma. O ITEM_SINGLE é representado em um arquétipo, por um único atributo, enquanto o tipo ITEM_LIST agrupa um conjunto de atributos representados fisicamente como uma lista vertical. Nesse caso, em virtude da organização dos atributos, utiliza-se o sistema de banco de dados chave-valor para armazenar os tipos ITEM_SINGLE e ITEM_LIST. O ITEM_TREE permite que os atributos de dados sejam representados por meio de uma estrutura hierárquica de vários níveis em um arquétipo. Para garantir que a mesma representação seja mantida, utiliza-se o formato de armazenamento baseado em documento.

```
1 Algorithm - Document Generator
2 Input: MetaData, Elements, DataStructures, Terminologies
3 Output: Void
4 BEGIN
5 INITIALIZE DataBase to DocumentsDataBase
6 CREATE Terminology as Collection
7 CREATE Constraints as Collection
8 CREATE Attributes as Collection
9 SET i to 1
10 FOR each node on the DataStructures
11 SET Index GET index on the node
12 SET Description GET description on the node
13 SET DataType GET datatype on the node
14 SET Constraint GET constraint on the node
15 STORE Index in Attributes[i]
16 STORE Description in Attributes[i]
17 STORE DataType in Attributes[i]
18 STORE Constraint in Constraints[i]
19 SET Terms GET Terminologies on the node
20 SET j to 1
21 FOR each node2 on the Terms
22 SET Child GET child on the node2
23 SET Parent GET parent on the node2
24 SET Value GET value on the node2
25 STORE Child in Terminologies[j]
26 STORE Parent in Terminologies[j]
27 STORE Value in Terminologies[j]
28 INCREMENT j
29 END FOR
30 INCREMENT i
31 END FOR
32 END
```

Figura 5. Algoritmo para geração de esquema de dados do tipo documento.

O algoritmo *DocumentGenerator* (Figura 5) recebe como parâmetros de entrada os elementos extraídos do algoritmo e, de posse destes elementos, cria um banco de dados (linha 5) e um conjunto de coleções para armazenar separadamente, os atributos de dados, as terminologias e as restrições dos arquétipos, como mostra o código descrito entre as linhas 6 e 9 da Figura 5. Depois disso, passa-se então à inserção dos elementos dos arquétipos em suas respectivas coleções de documentos. O laço iniciado na linha 10 e finalizado na linha 31 percorre a lista de elementos dos arquétipos e adiciona na coleção de atributos (linha 13), o código, a descrição e o tipo de dado. Já para a coleção de restrições (linha 18), armazena-se o código de referência do atributo de dado e a descrição da restrição. Por fim, caso o atributo de dado tenha uma ou mais terminologias, armazena-se na sua respectiva coleção (linha 21), o código de referência do atributo de dado, juntamente com a lista de terminologias encontradas.

A estrutura de dados ITEM_TABLE organiza os atributos por meio de linhas e colunas. Para esse tipo, utiliza-se a geração de esquemas de dados relacionais descrita na Seção 3.1.

3.3. Avaliando a Geração de Esquemas de Dados

Um dos aspectos principais da ferramenta proposta é a geração de esquemas de dados, em diferentes sistemas de bancos de dados. Para isso, propõe-se uma arquitetura de software que possibilita a partir da leitura de arquétipos, a geração de esquemas de dados independente da tecnologia de um SGBD. Para avaliar tal característica, foi feita a geração de esquemas de dados nas seguintes tecnologias: SQL Server 2015 e ArangoDB

3.1.12.

Uma das principais vantagens nesse modelo de arquitetura é que, caso seja necessário, novas interfaces para outras tecnologias de SGBD podem ser desenvolvidas sem afetar o funcionamento das demais camadas da arquitetura de software. Para isso, basta mapear (*IMapper*) os elementos dos arquétipos em esquemas de dados relacionais ou NoSQL e utilizar as assinaturas dos métodos de persistência a serem implementadas em diferentes sistemas de bancos de dados (*IRepository*), conforme descrito na Seção 3.

A partir da arquitetura da Seção 3, criou-se interfaces específicas para geração de esquemas de dados relacionais (i.e., MS SQL Server) e NoSQL (i.e., ArangoDB). Para os esquemas de dados relacionais, investigou-se a geração das tabelas, dos campos e das restrições de integridades referenciais, enquanto para o esquema de dados orientado a documento, investigou-se a representação hierárquica dos elementos extraídos do arquétipo.

Nesta avaliação, utilizou-se o arquétipo (i.e., Admin_Entry_Admission) que contém os requisitos de dados da admissão de um paciente em uma unidade de saúde. O arquétipo está disponível no site da openEHR e foi escolhido por ter mais de 50 atributos de dados, 70 terminologias e 25 restrições. Além disso, os atributos de dados estão organizados por meio de uma estrutura hierárquica de diversos níveis.

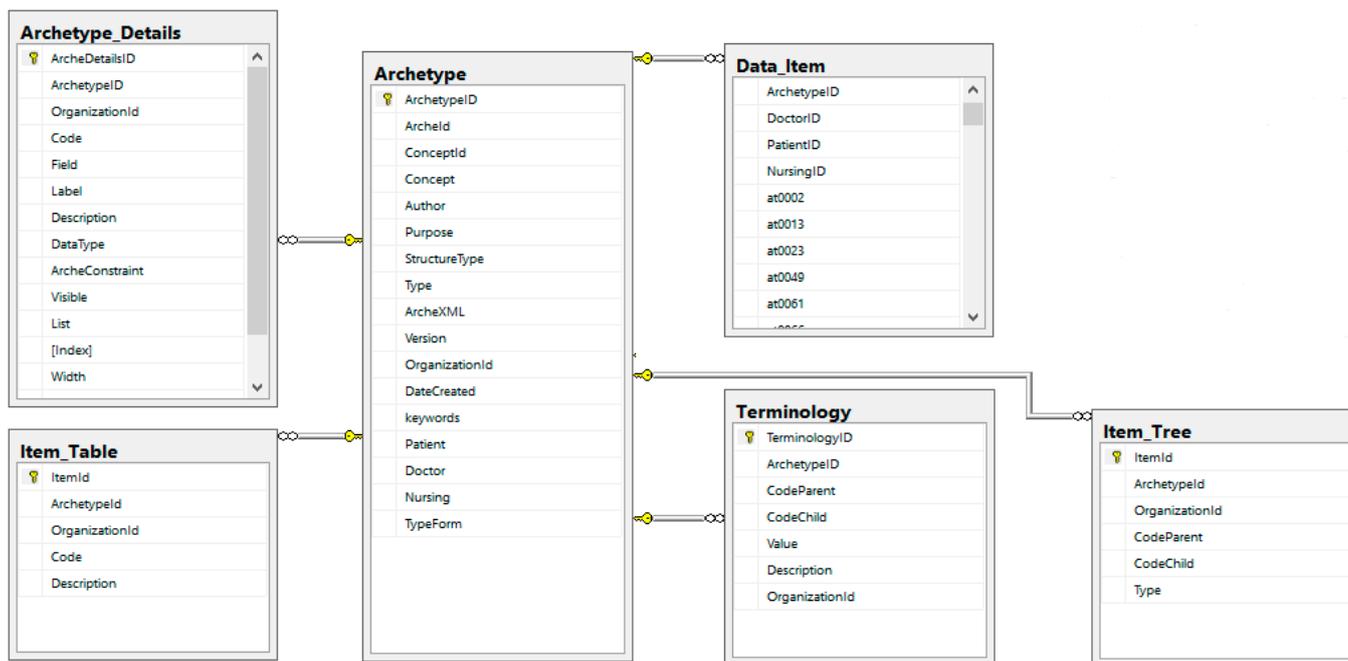


Figura 6. Esquema de dados gerado para o SGBD SQL Server .

A Figura 6 mostra o esquema de dados relacional gerado para o SGBD MS SQL Server 2015. Observa-se que as tabelas, os campos e as restrições de integridade referenciais foram geradas corretamente.

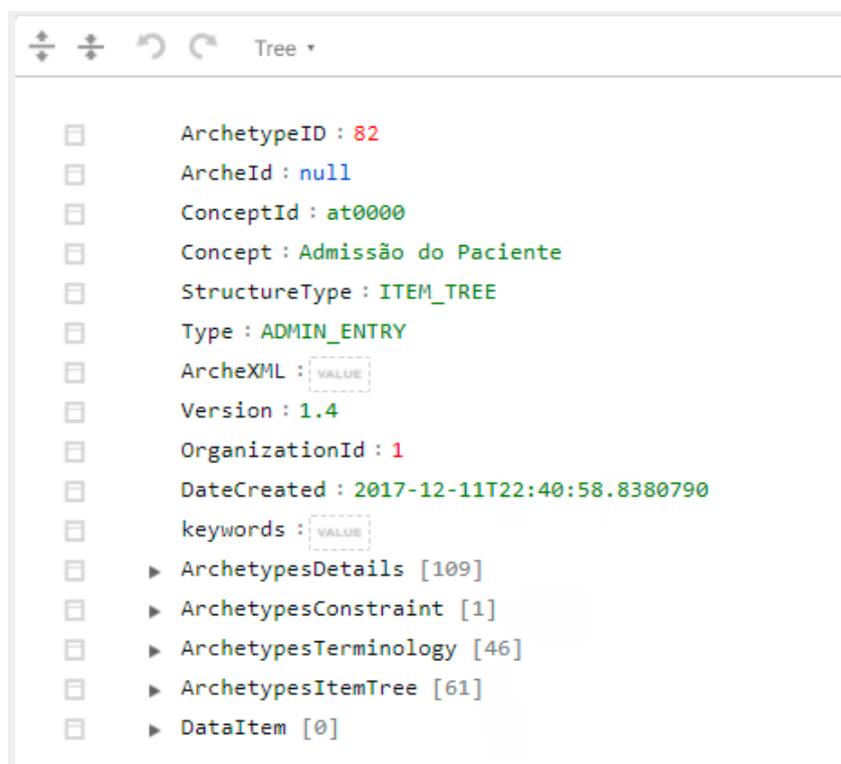


Figura 7. Esquema de Dados gerado para o SGBD ArangoDB

Esquemas de dados orientados a documento utilizam coleções para armazenar a estrutura hierárquica dos elementos de um arquétipo. Como mostra a Figura 7, armazena-se em uma coleção, os metadados do arquétipo, os atributos de dados com a mesma representação hierárquica, as terminologias e as restrições. O esquema de dados da Figura 7 foi gerado a partir da interface que gera documentos NoSQL para o SGBD ArangoDB. Outros SGBD podem utilizar a mesma interface para a geração de esquemas de dados. Nesse caso, deve-se ajustar os parâmetros de conexão com o SGBD e alterar a sintaxe de criação das coleções de documentos.

As interfaces desenvolvidas para a geração dos esquemas de dados desta avaliação estão disponíveis em <https://doi.org/10.5281/zenodo.167623>.

4. CONSIDERAÇÕES FINAIS

Neste artigo, apresentou-se uma ferramenta computacional capaz de extrair e mapear dinamicamente os elementos de arquétipos openEHR em esquemas de dados relacionais e NoSQL. Para garantir que os esquemas de dados gerados sejam independentes de tecnologia, especificou-

-se uma arquitetura de software que permite que as regras de negócios da ferramenta acessem a camada de persistência de dados sem que a mesma tenha conhecimento da tecnologia do SGBD utilizada.

Um Meta esquema foi proposto para gerar instâncias de esquemas de dados relacionais, enquanto a geração de esquemas de dados NoSQL foi exemplificada por meio de um algoritmo. Por fim, para validar a solução proposta, utilizou-se arquétipos disponíveis no repositório da openEHR para gerar esquemas de dados para diferentes SGBD. Existem duas vantagens principais na criação de esquemas de dados utilizando arquétipos. Primeiro, utiliza-se um padrão em saúde que uniformiza os atributos de dados e terminologias em aplicações de saúde. Segundo, a especificação do RES feita por um especialista do domínio em um arquétipo é utilizada na geração de esquemas de dados de aplicações de saúde.

Como sugestão de trabalhos futuros, indica-se: i) a criação dinâmica de interfaces gráficas de usuário a partir de arquétipos e ii) a especificação de um serviço que persista os dados manipulados na interface gráfica nos esquemas de dados criados.

REFERÊNCIAS

BEALE, T. Archetypes: Constraint-based domain models for future-proof information systems. **Eleventh OOPSLA Workshop on Behavioral Semantics: Serving the Customer 2002**, pp. 16-32. Disponível:

<https://www.academia.edu>., Último acesso em: 05-mai-2020

FREIRE, M.S, et al. (2016). Comparing the Performance of NoSQL Approaches for Managing Archetype-Based Electronic Health Record Data. **PLOS ONE** 2016, 11(3):

e0150069, <https://doi.org/10.1371/journal.pone.0150069>. Disponível: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0150069>. Último acesso em: 05-mai-2020.

GEORG, D., JUDITH, C., CHRISTOPH, R. (2013). Towards plug-and-play integration of archetypes into legacy electronic health record systems: the ArchiMed experience. **BMC Medical Informatics and Decision Making**. 2013; 1-12. Disponível: <https://bmcmmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-13-11>. Último acesso em: 05-mai-2020.

JEFFREY, A. L., JEFFREY, L. S., BLACKFORD, M. (2012). Method of Electronic Health Record Documentation and quality of primary care. **J Am Med Inform Assoc**, pp.1019-1024, 2012. Disponível: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3534457/>. Último acesso em: 05-mai-2020.

JUMAA, H., RUBEL, P., FAYN, J. (2013). An XML-based framework for automating data exchange in healthcare. **IEEE International Conference on e-Health Networking Applications and Services - Healthcom**, pp.264 – 269, 2010. Disponível: <https://cyberleninka.org/article/n/514784>. Último acesso em: 05-mai-2020.

KARAMJIT, K, RINKLE, R. Managing Data in Healthcare Information Systems: Many Models, **One Solution Computer**. 2015, 52-59, 10.1109/MC.2015.77. Dispo-

nível: <https://books.google.com.br/books>. Último acesso em: 05-mai-2020.

LEE, K. K., TANGB, W., CHOIA, K. (2013). Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage. **Computer Methods and Programs in Biomedicine**, pp. 99-109, 2013. Disponível: <https://www.ncbi.nlm.nih.gov/pubmed>. Último acesso em: 05-mai-2020.

LIN, C. H, FANN YC, LIOU, D.M. (2016). An exploratory study using an openEHR 2-level modeling approach to represent common data elements. **J Am Med Inform Assoc**. 2016 Sep;23(5):956-67. doi: 10.1093/jamia/ocv137. Disponível: <https://www.ncbi.nlm.nih.gov/pubmed>. Último acesso em: 05-mai-2020.

PEDERSEN, R., WYNN, R., ELLINGSEN, G. (2015). Semantic Interoperable Electronic Patient Records: The Unfolding of Consensus based Archetypes. **Stud Health Technol Inform**. 2015; 210:170-4. Disponível: <https://www.ncbi.nlm.nih.gov/pubmed>. Último acesso em: 05-mai-2020.

SAALFELD, B, TUTE, E, WOLF, K. H, MARSCHOLLEK, M. Introducing a Method for Transformation of Paper-Based Research Data into Concept-Based Representation with openEHR. **Stud Health Technol Inform**. 2017;235:151-155. Disponível: <https://www.ncbi.nlm.nih.gov/pubmed>. Último acesso em: 05-mai-2020.

SEELAM, S. R., DETTORI, P., WESTERINK, P., BO YANG, B. (2015). Polyglot Application Auto Scaling Service for Platform As A Service Cloud, IEEE International Conference on Cloud Engineering. Disponível: <https://www.semanticscholar.org/paper/Polyglot-Application-Auto-Scaling-Service-for-as-a-Seelam>. Último acesso em: 05-mai-2020.

SELLAMI, R., BHIRI, SS., DEFUDE, B. (2012). Supporting multi data stores applications in cloud environments, IEEE Transactions on Services Computing, 2015. Disponível: <https://scholar.google.com/citations>. Último acesso em: 05-mai-2020.

SPÄTH, M. B, GRIMSON, J. (2010). Applying the archetype approach to the database of a biobank information management system. International Journal of Medical Informatics. 2010; 1-22. Disponível: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3534457/>. Último acesso em: 05-mai-2020.

WANG, L, MIN, L, LU, X, DUAN, H. (2015). Archetype relational mapping - a practical openEHR persistence solution"; BMC Medical Informatics and Decision Making, pp. 1-18, 2015. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3534457/>. Último acesso em: 05-mai-2020.