

**MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
FARROUPILHA - *CAMPUS* SANTO ÂNGELO**

**INTEROPERABILIDADE DE INFORMAÇÕES DO PACIENTE USANDO UM
PADRÃO DE DADOS HL7 V2, API FHIR**

TRABALHO DE CONCLUSÃO DE CURSO

Eduardo Marques Ritter

**Santo Ângelo, RS, Brasil.
2021**

**INTEROPERABILIDADE DE INFORMAÇÕES DO PACIENTE USANDO UM
PADRÃO DE DADOS HL7 V2, API FHIR**

por

Eduardo Marques Ritter

**Monografia apresentada ao Instituto Federal de Educação, Ciência e
Tecnologia Farroupilha, como requisito parcial para obtenção do título de
Tecnólogo em Sistemas para Internet.**

Orientador: Prof. Me. Juliano Gomes Weber

**Santo Ângelo, RS, Brasil.
2021**

**Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia Farroupilha**

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

**INTEROPERABILIDADE DE INFORMAÇÕES DO PACIENTE USANDO UM
PADRÃO DE DADOS HL7 V2, API FHIR**

elaborada por
Eduardo Marques Ritter

como requisito parcial para obtenção do título de
Tecnólogo em Sistemas para Internet

COMISSÃO EXAMINADORA

Prof. Me. Juliano Gomes Weber (IFFar)
(Presidente/Orientador(a))

Prof. Me. Andrea Pereira (IFFar)

Prof. Me. Fabio Weber Albiero (IFFar)

Conceito Final: _____

Santo Ângelo, 18 de fevereiro de 2021

DEDICATÓRIA

*Dedico este trabalho ao meu filho Heitor e a minha esposa Cleide, pelo apoio nos
momentos de ausência.*

Aos meus pais, pelo apoio e pelos ensinamentos que me fizeram chegar até aqui.

A todos meus familiares, pelo apoio e palavras de incentivo.

*Aos professores, pela dedicação, carinho e empenho ao ensinar seus
conhecimentos.*

AGRADECIMENTOS

Agradeço a minha família, em especial ao meu filho Heitor que é a razão de todo e qualquer esforço.

À minha esposa Cleide, pela compreensão e pela paciência dedicada mesmo nos momentos em que tive que me ausentar, obrigado.

Aos meus pais pela educação e carinho, sempre dispostos a me ajudar. Sem o suporte de vocês jamais teria vindo até aqui.

Ao meu orientador professor Juliano, pela persistência em me fazer aprender, pelo apoio no desenvolvimento deste trabalho.

A todos os professores pelo empenho e dedicação, em especial aos professores (as) Lara, Marta e Marcus. Esse trabalho leva um pouco de cada um de vocês.

Aos meus colegas Fabrício e Daniel que não pouparam esforços para me ajudarem a entender o quanto o IFFar seria importante na minha trajetória de vida.

Ao meu colega de trabalho, Daniel Trois, que me incentivou na jornada de pesquisa do protocolo HL7.

RESUMO

INTEROPERABILIDADE DE INFORMAÇÕES DO PACIENTE USANDO UM PADRÃO DE DADOS HL7 V2, API FHIR

AUTOR: Eduardo Marques Ritter

Orientador: Prof. Me. Juliano Gomes Weber

Data e Local da Defesa: Santo Ângelo, 18 de fevereiro de 2021

Alcançar a Interoperabilidade entre sistemas de informação hospitalar é cada vez mais essencial. Ao longo do tempo, vários protocolos foram desenvolvidos com esse intuito. Este trabalho utiliza os padrões de dados desenvolvidos e mantidos pela Health Level Seven (HL7) para troca de mensagens. Sendo estudada a versão 2 da norma HL7 e o sistema construído teve por base a conversão de dados com informação hospitalar no formato *JSON (JavaScript Object Notation)* para obter uma mensagem HL7. As informações tratadas nesse estudo são oriundas do protocolo de vacinação nacional. Pode-se acessar o sistema por meio web bem como por um cliente Rest, tal como o Postman, o qual conecta na api e exporta a mensagem em JSON. Dentre as tecnologias estudadas, foi utilizado a REST pois é simples de entender e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS. Dentre as principais vantagens está a facilidade no desenvolvimento, o aproveitamento da infraestrutura web existente e um esforço de aprendizado pequeno.

Palavras-chave: Interoperabilidade; HL7 Fhir; Padrão de dados, REST.

ABSTRACT

Achieving interoperability between hospital information systems is increasingly essential. Over time, several protocols have been developed for this purpose. This work uses the data standards developed and maintained by Health Level Seven (HL7) to exchange messages. The version 2 of the HL7 standard was studied and the system built was based on the conversion of data with hospital information in JSON (JavaScript Object Notation) format to obtain an HL7 message. The information treated in this study comes from the national vaccination protocol. You can access the system through the web as well as a Rest client, such as Postman, which connects to the api and exports the message in JSON. Among the studied technologies, REST was used because it is simple to understand and can be adopted in almost any client or server with support for HTTP / HTTPS. Among the main advantages is the ease of development, the use of the existing web infrastructure and a small learning effort.

Keywords: Interoperability; HL7 Fhir; Data standard, REST.

LISTA DE FIGURAS

FIGURA 1 - Ficha do Paciente	15
FIGURA 2 - Gerenciador de Dispensação	16
FIGURA 3 - Tela de Atendimento	17
FIGURA 4 - Prontuário Eletrônico	18
FIGURA 5 - Sistema de Acolhimento	19
FIGURA 6 - Prontuário Eletrônico do Paciente	24
FIGURA 7 - Exemplo XML	25
FIGURA 8 - Exemplo JSON	27
FIGURA 9 - HAPI FHIR	29
FIGURA 10 - HAPI FHIR JSON	30
FIGURA 11 - Sistema Distribuído	31
FIGURA 12 - Sistema Distribuido 2	32
FIGURA 13 - SOAP	33
FIGURA 14 - REST API	34
FIGURA 15 - RESTful	35
Figura 22 - Modelo Lógico	37
Figura 23 - Modelo Físico	38
Figura 24 - Diagrama de Classe	39
Figura 25 - FrontEnd	40
Figura 26 - BackEnd	46
Figura 35 - Postman	48
Figura 27 - Tela Sistema 1	63
Figura 28 - Tela Sistema 2	64
Figura 29 - Tela Sistema 3	64
Figura 30 - Tela Sistema 4	65
Figura 31 - Tela Sistema 5	65
Figura 32 - Tela Sistema 6	66
Figura 33 - Tela Sistema 7	67
Figura 34 - Tela Sistema 8	67

LISTAS DE ABREVIATURAS E SIGLAS

API – Application Programming Interface.

DAO – Data Access Object.

GUI – Graphical User Interface.

IDE – Integrated Development Environment.

JDBC – Java Database Connectivity.

JPA – Java Persistence API.

JSF – JavaServer Faces.

JSON – JavaScript Object Notation

JVM – Java Virtual Machine.

MVC – Model-view-controller.

ORM – Object Relational Mapper (mapeador objeto relacional)

POO – Programação Orientada a Objetos.

SGBD – Sistema de Gerenciamento de Banco de Dados.

SQL – Structured Query Language.

XML – Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivos Específicos	13
1.2 Trabalhos correlatos	14
1.2.1 eSUS Hospitalar	15
1.2.2 Soul MV	17
1.2.3 Acesso móvel às informações de saúde do paciente utilizando computação ubíqua	20
1.3 Fundamentação teórica	20
1.3.1 Orientação a objetos	20
1.3.2 MySQL	21
1.3.3 Java	21
1.3.4 MVC (Model – View – Controller)	22
1.3.5 HI7 Fhir	22
1.3.6 Padrão Data Access Object – DAO	23
1.3.7 Prontuário eletrônico do paciente	23
1.3.8 Xml	24
1.3.9 Json	26
1.3.10 Maven	27
1.3.11 Hapi Fhir	28
1.3.12 Heterogeneidade	30
1.3.13 Sistemas distribuídos	31
1.3.14 Soap	32
1.3.15 Rest/Restfull	34
1.3.16 Spring Tool Suite	35
1.3.17 Postman	36
2 DESENVOLVIMENTO	36

2.1 Bando de Dados	36
2.1.1 Modelo Lógico	36
2.1.2 Modelo Físico	38
2.1.3 Diagrama de Classe	39
2.2 APLICATIVO WEB	40
2.2.1 FrontEnd ReactJS.....	40
2.2.2 BackEnd Java JPA	41
3. RESULTADOS E DISCUSSÕES	47
4. CONSIDERAÇÕES FINAIS	49
REFERÊNCIAS.....	50
APÊNDICE A	52
APÊNDICE B	63

1 INTRODUÇÃO

As instituições médicas e hospitalares geram diariamente um grande volume de informações em formatos diversos e oriundos de várias fontes de informação. Além disso, é comum que uma instituição adote um sistema de informação (SI) para gerenciar seus dados e outra instituição adote outro SI completamente diferente. Uma das necessidades destas instituições está relacionada à integração destes SI's e garantir sua interoperabilidade.

Essa interoperabilidade se faz urgente tanto para produção de registros de conhecimento relativos quanto àqueles ao estado de saúde do paciente, voltadas a sua qualidade de vida. No campo da saúde já existem vários padrões de interoperabilidade, destacando-se o HL7.

Conforme Oliveira (2005, p.34):

Um dos aspectos importantes para a interoperabilidade de informações é a organização do conhecimento e a representação da informação. Para que duas pessoas, duas bases de dados ou até duas instituições possam trocar informações de forma eficaz é necessário o entendimento dos códigos utilizados por ambos e que eles tenham o mesmo entendimento quanto ao significado destes códigos. Para isso é necessária a implementação de padrões e normas que possibilitem o entendimento entre eles, rompendo barreiras trazidas pela hiperespecialização e pela fragmentação da informação, desenvolvendo e reforçando trabalhos cooperativos.

O HL7 tem por objetivo definir normas para a transmissão de dados como, por exemplo, dados sobre registros de pacientes, admissão, transferências de pacientes, seguros, taxas e contas a pagar, pedidos e testes de laboratório, exames de imagem, observações médicas e de enfermagem, prescrições de dieta, pedidos à farmácia, pedidos de suprimentos e arquivos; enfim, o padrão HL7 tem a capacidade de

comunicar sistemas considerados heterogêneos como o sistema administrativo de um hospital, o sistema financeiro e o sistema de informações clínicas do paciente (FERNANDES et al., 1999). O HL7 não restringe o sistema operacional, nem a linguagem de programação a ser utilizada para a troca de informação. Além disso, o HL7 é independente de meio físico e de protocolo de comunicação. O padrão HL7 permite que o implementador defina suas próprias mensagens, o que mais uma vez, torna-o flexível. (HENRIQUES, CARVALHO, 2005). É a possibilidade de interoperabilidade entre os diversos sistemas da área de saúde utilizando duas modelagens distintas, porém desenvolvidas conforme as especificações do padrão HL7 que serão abordadas como objeto de estudo.

Desta forma, hoje mais do que nunca, é fundamental haver compartilhamento de dados e informações.

O tema proposto (desenvolvimento de um paradigma de interoperabilidade utilizando HL7 API FHIR) para este trabalho é de grande relevância na área médica hospitalar, haja vista a necessidade de compartilhamento de informações e dados do paciente.

Este protótipo de software como ferramenta de interoperabilidade poderá ser utilizado em organizações públicas ou privadas. Suas funções se aplicarão no compartilhamento de dados entre diversos SIS.

1.1 OBJETIVOS

Neste tópico serão apresentados os objetivos gerais e específicos do trabalho proposto.

1.1.1 Objetivo Geral

Desenvolver um software de caráter médico hospitalar, que seja baseado em uma plataforma Java web, onde predomina o compartilhamento de informações do prontuário médico utilizando o protocolo HL7 v2, API FHIR.

1.1.2 Objetivos Específicos

- Pesquisar e adequar uma metodologia que permita apoiar o desenvolvimento da solução proposta;
- Analisar esta padronização dos dados, HL7 v2, API FHIR.;
- Permitir que as organizações de saúde compartilhem facilmente informações clínicas e administrativas;
- Teoricamente, essa capacidade de troca de informações deve ajudar a minimizar a tendência de os cuidados médicos serem geograficamente isolados e altamente desfragmentados;
- Pesquisar sobre o Paradigma de Orientação Objeto;
- Investigar a Linguagem de Programação Java;
- Analisar e ampliar conhecimento sobre o SGBD Mysql;
- Desenvolver um software baseado no estudo desse trabalho.

1.2 Trabalhos correlatos

A proposta deste trabalho é criar um sistema para gerenciar e implementar o padrão de interoperabilidade HL7 API FHIR nas instituições de saúde, para tanto é necessário entender como funciona esse processo e quais suas etapas.

Para atender essa demanda foi realizada uma pesquisa sobre softwares já existentes no mercado que utilizam esse padrão. Um deles com licença de uso do Ministério da Saúde e outro sob licença comercial. O objetivo desse estudo é encontrar semelhanças e diferenças entre eles para posterior implementação.

1.2.1 eSUS Hospitalar

O e-SUS Hospitalar tem seu foco no HIS (Hospital Information System) e PEP (Prontuário Eletrônico do Paciente), oferece ferramentas para geração de relatórios, painel de indicadores, módulos de agendamento, internação, laboratório, etc. Ele pode atender tanto hospitais quanto UPAs e emergências.

FIGURA 1 - Ficha do Paciente.

The screenshot shows a web-based form for entering patient data. The form is organized into several sections:

- Dados do Paciente:** Includes a checkbox for 'Paciente não identificado neste momento'. Fields for 'Núm. do Cartão SUS', 'Código', 'Nome', 'Nascimento', 'Recém nascido', and 'Declaração'.
- Identificação:** Fields for 'Sexo', 'CEP', 'Tipo de Logradouro', 'Endereço', 'Número', 'Complemento', 'Estado', 'Cidade', and 'País'.
- Contato:** Fields for 'Telefone Res.', 'Telefone Com.', 'Telefone Cel.', 'Fax', 'E-mail', and 'Correspondência'.
- Identificação Pessoal:** Fields for 'Registro Geral', 'CPF', 'Certidão de Nascimento', 'PIS/PASEP', 'Estado Civil', 'Tipo Sanguíneo', and 'Indicado por'.
- Família:** Fields for 'Nome do Cônjuge', 'Nome da mãe', and 'Nome do pai'.
- Outros Dados:** Fields for 'Religião', 'Cor', 'Natural de', 'Estado', and 'Nacionalidade'.
- Educação:** Field for 'Grau de Instrução'.

The form also includes several interactive elements like 'Clique aqui para selecionar a cidade' and 'Clique aqui para buscar a ocupação'.

Fonte: do Autor.

Nesta ficha (Figura 1) inserimos os dados particulares do paciente, como nome, endereço, etc.

FIGURA 2 - Gerenciador de Dispensação.

Dispensação > Gerenciador de dispensação

Data Inicial: 05/07/2012 Data Final: 05/07/2012 Exibir dispensações parciais

Prescrições imediatas

<input type="checkbox"/>	Nº DOC	Unidade	Paciente	Status	Data / Hora	Médico	Tipo
<input type="checkbox"/>							

Clique sobre o item para selecioná-lo.

Prescrições diárias

<input type="checkbox"/>	Nº DOC	Unidade	Paciente	Status	Data / Hora	Médico
<input type="checkbox"/>	17221	HFL 5ª ANDAR POSTO 2 - 534.2		Pendente	05/07/2012 18:00:00	FERNANDO SPADA
<input type="checkbox"/>	17224	HFL 3ª ANDAR CTI PEDIATRICO - 350.04-CTI-PEDIATRIC	XXXXXXXXXX	Pendente	05/07/2012 18:00:00	MARIA ISABEL BRANDAO PIRES E ALBUQUERQUE
<input type="checkbox"/>	17226	HFL 5ª ANDAR POSTO 2 - 530.2		Pendente	05/07/2012 18:00:00	RAFAEL CARRARA SANGLARD AMARAL
<input type="checkbox"/>	17227	HFL 3ª ANDAR CTI PEDIATRICO - 350.03-CTI-PEDIATRIC	XXXXXXXXXX	Pendente	05/07/2012 18:00:00	MARIA ISABEL BRANDAO PIRES E ALBUQUERQUE
<input type="checkbox"/>	17228	HFL 7ª ANDAR UNIDADE CORONARIANA - 748.1-U.CORONARIANA	XXXXXXXXXX	Pendente	05/07/2012 18:00:00	BEATRIZ BUSSADE PILLAR
<input type="checkbox"/>	17233	HFL 5ª ANDAR POSTO 2 - 538.1		Pendente	05/07/2012 18:00:00	TATIANA MARTINS BENVENUTO LOURO BERBARA
<input type="checkbox"/>	17232	HFL 7ª ANDAR CTI ADULTO - 700.10 - CTI ADULTO	XXXXXXXXXX	Pendente	05/07/2012 18:00:00	NÚBIA DE MATTOS SILVA
<input type="checkbox"/>	17236	HFL 7ª ANDAR UNIDADE CORONARIANA - 748.2-U.CORONARIANA	XXXXXXXXXX	Pendente	05/07/2012 18:00:00	FABIOLA LÚCIO CARDÃO
<input type="checkbox"/>	17237	HFL 3ª ANDAR CTI PEDIATRICO - 350.08-CTI-PEDIATRIC	XXXXXXXXXX	Pendente	05/07/2012	MARIA ISABEL BRANDAO

Fonte: do Autor.

Controle de medicamentos e materiais fracionados (Figura 2), de empréstimos, além de relatórios como: escrituração de controlados, estoque por unidade, movimentação de estoque, itens a vencer e outros.

1.2.2 Soul MV

Segundo o site do desenvolvedor (MV Informática Nordeste Ltda, 2019), “eleito o melhor Prontuário Eletrônico por quatro vezes consecutivas, o Prontuário Eletrônico do SOUL MV é considerado o melhor da América Latina segundo o instituto norte-americano de pesquisa e insights, KLAS.”

Este sistema reúne um conjunto de soluções que facilitam o fluxo de dados entre os setores e integram todos os processos hospitalares. O sistema gerencia informações clínicas, assistenciais, administrativas, financeiras e estratégicas, proporcionando uma gestão mais eficiente e melhor atendimento para os pacientes.

As figuras 3, 4 e 5, demonstram algumas funcionalidades da ferramenta no que diz respeito à gestão clínica e administrativa de instituições de saúde.

FIGURA 3 - Tela de Atendimento.

Fonte: do Autor.

FIGURA 4 - Prontuário Eletrônico

The screenshot displays the 'soulmv' electronic medical record (EMR) interface. At the top, the header includes the logo 'soulmv', the title 'NOVO PRONTUÁRIO ELETRÔNICO DO PACIENTE WEB', and the user's role 'Prescrição'. The patient's name 'MARIA JOSE DA SILVA' and demographic information 'Feminina, 59 Anos 8 Meses 9 Dias, PSEMO, GERIATRIA, 10º ANDAR, 1012-1' are visible. The main content area is divided into several sections:

- Prescrição Médica:** A search bar and a 'Data de referência: 10/11/2015' field are present.
- Prescrição Paciente:** A list of medical categories with expandable details:
 - Dieta:** Nenhum item adicionado. Clique ao lado, no nome "Dieta", para adicionar.
 - Enfermagem:** Nenhum item adicionado. Clique ao lado, no nome "Enfermagem", para adicionar.
 - Soro/terapêut:** Nenhum item adicionado. Clique ao lado, no nome "Soro/terapêut", para adicionar.
 - Medicamentos:**
 - C. C. DOR... (partially visible)
 - SERINGA DESC. 10ML S/ AGULHA E PONTA SLIP
 - Água Destilada AMP 10ml, SOL. INJ
 - AGULHA DESCART. 40X12) C,DISP,SEGURANÇA
 - Medic. Oncolôg:** Nenhum item adicionado. Clique ao lado, no nome "Medic. Oncolôg", para adicionar.
 - Hemocompone...:** Nenhum item adicionado. Clique ao lado, no nome "Hemocomponentes", para adicionar.
 - Exame de Imu...:** Nenhum item adicionado. Clique ao lado, no nome "Exame de Imunologia", para adicionar.
 - Exames Laboratô:** Nenhum item adicionado. Clique ao lado, no nome "Exames Laboratoriais", para adicionar.
 - Materiais:** Nenhum item adicionado. Clique ao lado, no nome "Materiais", para adicionar.
 - Orientações:** Nenhum item adicionado. Clique ao lado, no nome "Orientações", para adicionar.
 - Procedimentos:** Nenhum item adicionado. Clique ao lado, no nome "Procedimentos", para adicionar.
- Prescrições:** A right-hand sidebar shows a list of prescriptions with dates and details:
 - 10/11/2015
 - MEDICO CARDIOLOGIA 10/11/2015 14:06
 - 29/09/2015
 - 28/09/2015
 - 27/09/2015
 - 26/09/2015
 - 25/09/2015
 - 24/09/2015
 - 23/09/2015
 - 22/09/2015
 - 21/09/2015

The bottom of the interface features a navigation bar with icons for various functions like 'Admissão', 'Evolução', 'Processos Cirurgico', 'Documentos Clínicos', 'Interconsulta', 'Registro Clínico', 'Encaminhar Paciente', 'Prescrição', 'Enfermagem', 'Resultado de Exames', 'Relatório Ficha de Parto', and 'Lista Paciente por Especialid...'. The version number 'Versão: SMA-PEP.02.050.2' is also visible in the top right corner.

Fonte: do Autor.

FIGURA 5 - Sistema de Acolhimento

The image shows a web browser window displaying a patient intake system. The browser's address bar shows the URL `http://pepweb:8082/mvsacr/#`. The browser's menu bar includes "Arquivo", "Editar", "Exibir", "Favoritos", "Ferramentas", and "Ajuda". Below the menu bar, there are sections for "Favoritos", "Sites Sugeridos", and "Galeria do Web Slice". The main content area features a header with the logo "MVSACR" and the text "Sistema de Acolhimento com Classificação de Risco". To the right of the header, a patient information box displays a blue person icon and the text "PACIENTE T1", "0 Ano 9 Meses", and "MASCULINO". Below the header, there are two icons: a list icon and an eye icon. The main content area is titled "» Lista de Pacientes » Resumo Acolhimento". On the left side, there is a vertical sidebar with four icons and labels: "Dados Paciente", "Abordagem Inicial", "Classificação", and "Situação". The main content area contains a form with the following fields: "Abordagem Inicial" (with a blue arrow icon), "Queixa Principal: * DOR DE ABDOMINAL" (with a blue document icon), "Alergia: NAO", and "Observação:".

Fonte: do Autor.

1.2.3 Acesso móvel às informações de saúde do paciente utilizando computação ubíqua

Neste trabalho o autor propõe o desenvolvimento de uma arquitetura integrando sistemas móveis de consulta às informações de pacientes para ambientes hospitalares, com objetivo de intentar um recurso diferenciado na identificação do contexto. O sistema destaca informações relevantes para o diagnóstico ou tratamento de doenças. Somando-se à proposta o uso de um padrão de comunicação médica interoperável, o Health Level Seven.

1.3 Fundamentação teórica

Este capítulo trata das fundamentações teóricas utilizadas para desenvolver este trabalho.

1.3.1 Orientação a objetos

A Orientação a Objetos é um paradigma amplamente utilizado, o principal motivo para isso são as questões de segurança e reaproveitamento de códigos, algo muito importante no desenvolvimento de aplicações modernas.

Esse paradigma foi criado para tentar aproximar o mundo real do mundo virtual, ou seja, simulando o mundo real dentro do computador.

A principal diferença dentre os outros paradigmas está no conceito de herança, que é a capacidade de um novo objeto tomar atributos e operações de um objeto existente.

Sommerville (2007, p. 208) afirma que “análise orientada a objetos concentra-se no desenvolvimento de um modelo orientado a objetos do domínio da aplicação. Os objetos nesse modelo refletem as entidades e as operações associadas ao problema a ser resolvido”.

1.3.2 MySQL

O MySQL foi criado pelos desenvolvedores David Axmark, Allan Larsson e Michael “Monty” Widenius, na década de 90, após precisarem de uma interface SQL compatível com as rotinas ISAM que utilizavam em suas aplicações e tabelas.

É um servidor de bancos de dados SQL, muito rápido, multitarefa e multiusuário. No dia 16 de Janeiro de 2008, a MySQL AB, desenvolvedora do MySQL foi adquirida pela Sun Microsystems, por US\$ 1 bilhão, um preço jamais visto no setor de licenças livres. No dia 20 de Abril de 2009 a Oracle compra a Sun Microsystems e todos o seu produtos, incluindo o MySQL.

MySQL é de Licença Dupla. Os usuários podem escolher entre usar o programa como um produto Open Source sob os termos da GNU General Public License ou podem comprar uma licença comercial.

Um dos motivos para utilizar este SGBD é a sua ampla utilização em sistemas web e o fato de que, quando um projeto open source atinge um número grande de usuários, que é o caso do MySQL, se torna muito difícil descontinuí-lo, mesmo que o autor original perca o interesse, sempre existirão voluntários dispostos a mantê-lo.

1.3.3 Java

Patrick Naughton, Mike Sheridan e James Gosling iniciaram em 1991 a criação da tecnologia Java. No início esta tecnologia não tinha como propósito a programação e sim uma ferramenta de programação do projeto chamado The Green Project da Sun Microsystems.

Somente em 1995, em uma conferência, que o Java foi anunciado e chamou a atenção por causa do enorme interesse pela Web (Deitel, 2010, p. 6).

Programas em Java, por serem interpretados, são compilados para um bytecode, que é executado por uma máquina virtual, o que permite aos

desenvolvedores criarem um programa uma única vez e depois executar este em qualquer uma das plataformas suportadas pela tecnologia.

Por ser uma linguagem multiplataforma, Java é utilizado em vários ambientes de desenvolvimento. Os mais utilizados são o ambiente web, desktop e dispositivos celulares. Qualquer dispositivo que tenha uma JVM instalada, pode rodar programas escritos na linguagem Java. Isso torna a vida do programador mais fácil, pois pode reutilizar códigos de um ambiente para o outro.

1.3.4 MVC (Model – View – Controller)

MVC é o acrônimo de *Model-View-Controller* (em português, **Modelo-Visão-Controle**) o qual, como o próprio nome supõe, separa as camadas de uma aplicação em diferentes níveis e executam apenas o que lhe é definido e nada mais do que isso.

Relatando cada um dos objetos do padrão MVC tem-se em primeiro lugar o controlador (Controller) que é responsável por interpretar e repassar ao Model e/ou para a View as ordens recebidas do usuário através da apresentação apropriada. Já o modelo (Model) gerencia os elementos de dados, responde a perguntas e a instruções para mudar de estado. O modelo é a principal estrutura computacional da arquitetura é ele quem modela o problema que está se tentando resolver. Finalmente, a visão (View) é o agente responsável por apresentar as informações para o usuário através de gráficos e textos. A View não tem gerência alguma sobre o que a aplicação está fazendo, ela apenas recebe instruções do controle e informações do modelo e então as exibe. A visão se comunica com o modelo e com o controlador para reportar o seu estado.

1.3.5 HL7 Fhir

O HL7 (Health Level-7) é uma organização desenvolvedora de padrões certificada pelo American National Standards Institute (ANSI) na área de saúde. O HL7 define a estrutura de mensagens específica para dados clínicos. Foi fundado em

1987 para fornecer informações padronizadas sobre troca, transferência, compartilhamento, integração e recuperação de informações em saúde com esmero científico e técnicos para gestão clínica.

Já o FHIR é o mais recente protocolo de HL7 orientado a recursos que permite que a informação esteja distribuída. O FHIR é baseado na arquitetura Restfull e se tornou um padrão mais robusto evitando a necessidade de ferramentas complexas.

1.3.6 Padrão Data Access Object – DAO

O padrão DAO é um padrão de projeto que tem por finalidade abstrair e encapsular os mecanismos de acesso aos dados omitindo os detalhes da execução da origem dos dados, surgiu com a necessidade de separa a lógica de negócios da lógica da persistência de dados. Em outras palavras, separou-se os códigos SQL dos códigos de programação.

Dentre as maiores vantagens de se abstrair os comandos SQL dos códigos de programação é a facilidade da troca de banco, evitando transtornos na regra de negócio e oferecendo uma manutenção mais eficiente.

1.3.7 Prontuário eletrônico do paciente

O Prontuário Médico é obrigatório em todo atendimento. Anteriormente apenas realizado através de registro em papel, hoje pode ser feito por meio eletrônico, desde que sejam cumpridas as exigências legais e éticas. O Prontuário Eletrônico do Paciente (PEP – Representado pela Figura 6), é uma ferramenta fundamental na construção de um novo paradigma do sistema de informação médica. A implementação do PEP nas instituições de saúde abriu uma discussão quanto aos processos de trabalho, rotinas médico-assistenciais, conceitos éticos, etc.

O PEP deve contemplar todos os atendimentos ambulatoriais, emergenciais e internações, permitindo o acompanhamento de cada evento com uma visão detalhada do histórico e da evolução clínica dos pacientes.

Permite o compartilhamento de informações entre diferentes profissionais de saúde tornando mais fácil e ampla uma abordagem legal para as ações médicas, pesquisas clínicas, estudos epidemiológicos, cuidado do paciente, bem como fonte de educação continuada e reciclagem médica.

FIGURA 6 - Prontuário Eletrônico do Paciente.

Fonte: do Autor.

1.3.8 Xml

XML (Extensible Markup Language) é uma linguagem de marcação que define um conjunto de regras para codificação de documentos, foi criado em meados da década de 1990. O *World Wide Web Consortium* (W3C) começou a trabalhar em uma linguagem de marcação que combinasse a flexibilidade da SGML (Standard

Generalized Markup Language) com a simplicidade da HTML(*HyperText Markup Language*). Fornece uma plataforma para definir elementos de marcação e gerar uma linguagem personalizada. Tem foco na simplicidade, generalidade e usabilidade, sendo usado para diversos serviços na web. A Figura 7, exemplifica um arquivo XML.

FIGURA 7 - Exemplo XML

```
<Location xmlns="http://hl7.org/fhir">
  <id value="45348"/>
  <meta>
    <versionId value="1"/>
    <lastUpdated value="2019-10-11T09:35:03.452+00:00"/>
    <source value="#ea23c7981cf245f5"/>
  </meta>
  <identifier>
    <type>
      <coding>
        <system value="http://terminology.hl7.org/CodeSystem/v2-0203"/>
        <code value="SNO"/>
        <display value="Serial Number"/>
      </coding>
    </type>
    <value value="G21.R2.B2"/>
  </identifier>
  <name value="G21"/>
  <description value="G2D"/>
  <physicalType>
    <coding>
      <system value="http://terminology.hl7.org/CodeSystem/location-physical-type"/>
      <code value="bu"/>
      <display value="Building"/>
    </coding>
  </physicalType>
  <managingOrganization>
    <reference value="Organization/185"/>
  </managingOrganization>
</Location>
```

Fonte:http://hapi.fhir.org/read?serverId=home_r4&encoding=xml&pretty=true&resource=Location&action=read&id=45348&vid=1

1.3.9 Json

JSON (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto, cada valor apresentado tem um nome ou rótulo que descreve seu significado. Este formato é derivado da forma utilizada pelo JavaScript para representar informações.

Um objeto JSON pode representar qualquer tipo de informação. Representa no formato chave/valor, as chaves representam os nomes dos atributos da classe e os valores são os valores do objeto, conforme exemplo da figura 8.

Os tipos de dados básicos do JSON são:

1. **String** - separados por aspas duplas ou simples. Ex. "Palavra" ou 'Palavra';
2. **Número** - pode ser inteiro ou real e não usa aspas. Quando for do tipo real deve-se usar o caractere . (ponto) para separar a parte inteira das casas decimais. Ex. 1 ou 23.454;
3. **Booleano** - tipo lógico normal, valores true ou false;
4. **Nulo** – valor nulo. Ex. "nome": null.

FIGURA 8 - Exemplo JSON

```

{
  "resourceType": "Location",
  "id": "45348",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2019-10-11T09:35:03.452+00:00",
    "source": "#ea23c7981cf245f5"
  },
  "identifier": [
    {
      "type": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
            "code": "SNO",
            "display": "Serial Number"
          }
        ]
      },
      "value": "G21.R2.B2"
    }
  ],
  "name": "G21",
  "description": "G2D",
  "physicalType": {
    "coding": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/location-physical-type",
        "code": "bu",
        "display": "Building"
      }
    ]
  },
  "managingOrganization": {
    "reference": "Organization/185"
  }
}

```

Fonte:

http://hapi.fhir.org/read?serverId=home_r4&encoding=xml&pretty=true&resource=Location&action=read&id=45348&vid=1

1.3.10 Maven

Apache Maven ou simplesmente Maven, é uma ferramenta de integração de projetos, responsável por gerenciar dependências, controlar versão de artefatos, dentre outras. Desenvolvida pela Apache, está atualmente na sua versão 3.

Tornou-se uma ferramenta bastante utilizada por sua configuração ser mais simples que a ferramenta Ant, a qual deu origem ao projeto. O Maven tem sua configuração baseada em XML. Tratando-se de uma ferramenta de automação de projeto, ele é quem descreve todo o processo de construção de um software. O Maven

gerencia as dependências de um projeto, seus módulos e componentes e sequencia de construção através de uma estrutura conhecida como POM (Project Object Model). A característica mais marcante do Maven é a possibilidade de trabalhar em rede e poder baixar as dependências diretamente de um repositório para o seu núcleo.

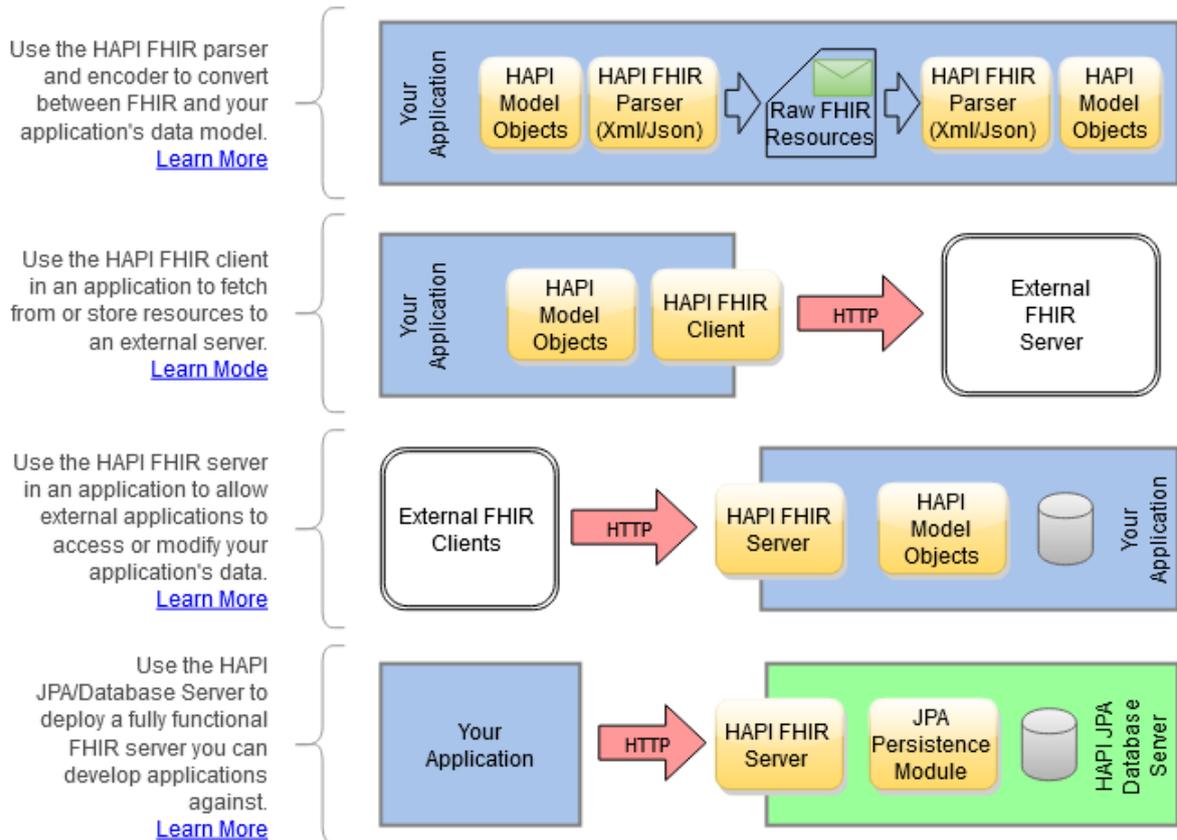
1.3.11 Hapi Fhir

A HAPI (HL7 Application Programming Interface) (UNIVERSITY HEALTH NETWORK, 2001) é licenciada por Mozilla Public License e General Public License (GNU) é uma API para programação do protocolo HL7. Ela tem um conjunto de classes e interfaces que ajudam na criação de aplicativos com suporte a mensagens HL7.

Esta biblioteca pode ser utilizada para a construção e troca de mensagens FHIR entre a sua aplicação e servidores ou clientes externos, por exemplo, através dos formatos JSON ou XML.

A figura 9 exemplifica as camadas da API HAPI FHIR e a figura 10 uma mensagem codificada em JSON.

FIGURA 9 - HAPI FHIR



Fonte: <https://hapifhir.io/>

FIGURA 10 - HAPI FHIR JSON

Response	✓ HTTP 200 OK
Response Headers	x-request-id: w8qmB9y411DDW9d9 date: Tue, 29 Oct 2019 17:49:54 GMT last-modified: Tue, 17 Sep 2019 22:13:29 GMT server: nginx/1.14.0 (Ubuntu) transfer-encoding: chunked x-powered-by: HAPI FHIR 4.1.0-SNAPSHOT REST Server (FHIR Server; FHIR 4.0.0/R4) content-location: http://hapi.fhir.org/BaseR4/Patient/149/_history/1 etag: 0/"1" connection: keep-alive content-type: application/fhir+json; charset=utf-8
Result Body JSON resource (810 bytes)	Raw Message <pre>{ "resourceType": "Patient", "id": "149", "meta": { "versionId": "1", "lastUpdated": "2019-09-17T22:13:29.609+00:00", "source": "#rad32f63b118b3" }, "text": { "status": "generated", "div": "<div xmlns='http://www.w3.org/1999/xhtml'><div class='hapiHeaderText'>Bob ALEXANDER </div><table class='hapiPropertyTable'><tbody><tr><td>Identifier</td><td>C7b2679e-a1dd-4cdf-bf16-be75a9431d30</td></tr></tbody></table></div>" }, "identifier": [{ "type": { "coding": [{ "system": "http://hl7.org/fhir/v2/0283", "code": "MR" }] }, "value": "C7b2679e-a1dd-4cdf-bf16-be75a9431d30" }], "family": [{ "family": "Alexander", "given": ["Bob"] }] }</pre>
Result Narrative	Bob ALEXANDER Identifier:C7b2679e-a1dd-4cdf-bf16-be75a9431d30

Fonte:

http://hapi.fhir.org/read?serverId=home_r4&encoding=json&pretty=true&resource=Patient&action=read&id=149&vid=1

1.3.12 Heterogeneidade

Segundo Coulouris, a Internet possibilita que os usuários em qualquer parte do mundo acessem e executem serviços por meio de um conjunto heterogêneo de computadores. A heterogeneidade é a variedade, a diferença de sistemas e de meios de comunicação existentes seguindo os seguintes aspectos:

- Implementação do diferente;
- Linguagens de programação;
- Hardware dos computadores;
- Sistemas operacionais;
- Redes.

Para tentar resolver o problema da heterogeneidade define-se uma camada de software intermediária, o middleware, que é um conjunto de funcionalidades e padronizações que atua entre a aplicação e o sistema operacional. Oferece uma

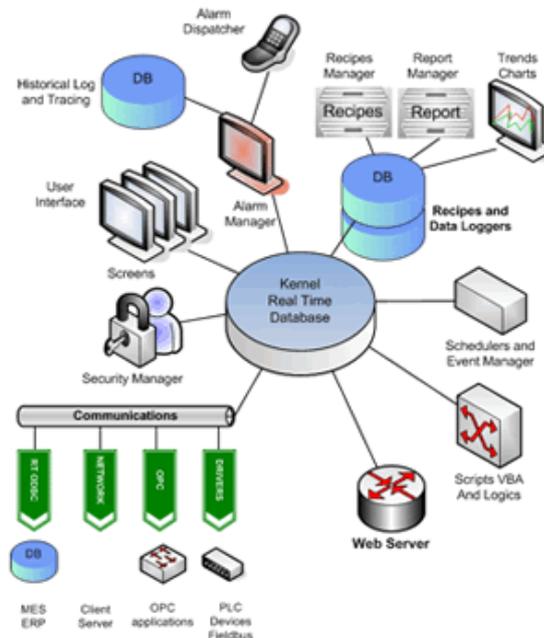
abstração para a comunicação e representação dos dados, permitindo que diferentes aplicações se comuniquem.

1.3.13 Sistemas distribuídos

“Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens” Couloris et al. 2007.

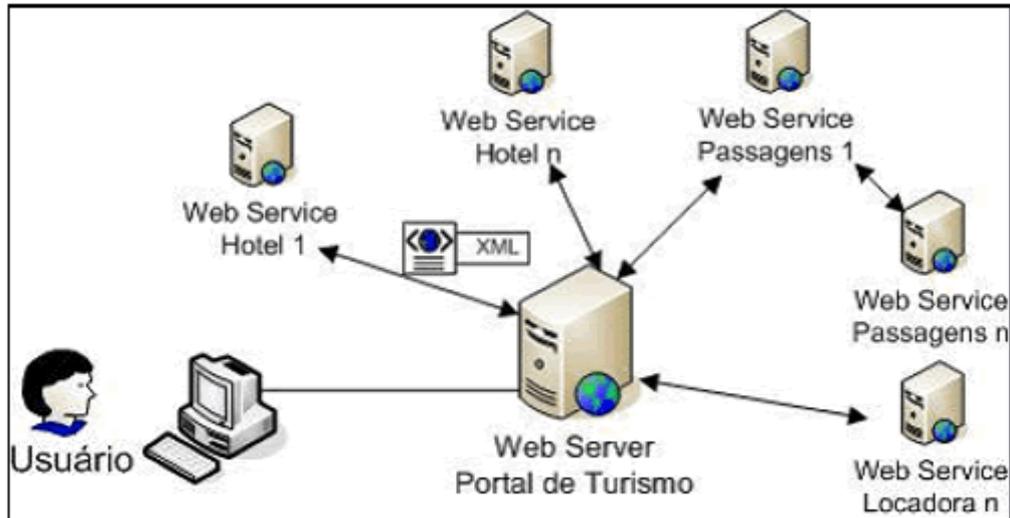
Desta forma, entende-se como um sistema distribuído um conjunto de computadores independentes que se apresenta ao usuário como um sistema único. É uma junção de computadores e softwares interconectados por uma rede, resultando em uma aplicação integrada, representado abaixo na figura 11 e 12.

FIGURA 11 - Sistema Distribuído.



Fonte: <https://tecnologia.culturamix.com/dicas/o-que-e-um-sistema-distribuido>

FIGURA 12 - Sistema Distribuído 2.



Fonte: <https://tecnologia.culturamix.com/dicas/o-que-e-um-sistema-distribuido>

1.3.14 Soap

SOAP (**Protocolo Simples de Acesso a Objetos**), protocolo baseado em XML, consiste na comunicação de mensagens entre aplicações via HTTP. Normalmente utilizado em Web Services é independente de plataforma e linguagem.

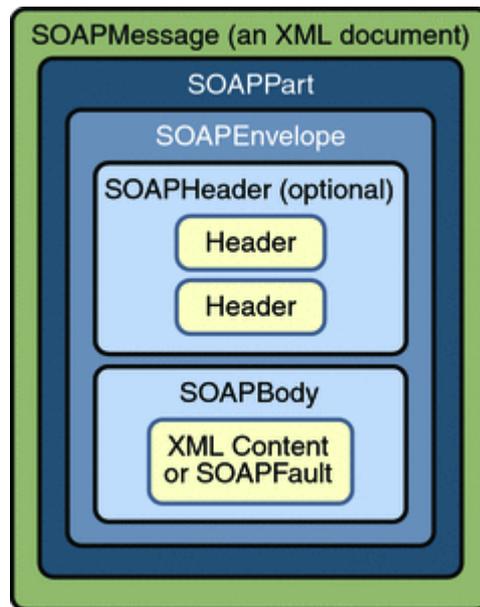
O SOAP é um protocolo projetado para invocar aplicações remotas através de RPC ou trocas de mensagens, em um ambiente independente de plataforma e linguagem de programação. (SIMPLE, 2011)

Uma mensagem SOAP consiste basicamente dos seguintes elementos:

- Envelope: Representa a mensagem, é o elemento principal do XML.
- Header: Permite a adição de características genéricas à mensagem SOAP de forma descentralizada.
- Body: Contém a informação a ser transportada.

Abaixo, na figura 13, uma representação desses elementos.

FIGURA 13 - SOAP.



Fonte: <https://zarelli.wordpress.com/2012/03/22/como-funciona-o-soap-protocolo-simples-de-acesso-a-objetos/>

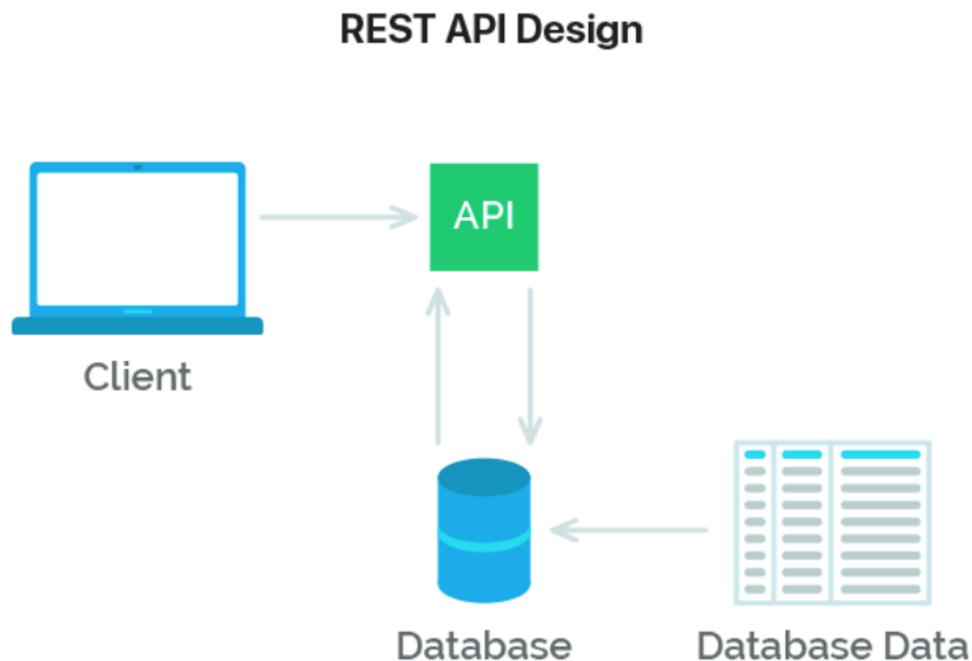
1.3.15 Rest/Restfull

Termo definido por Roy Fielding em sua tese de mestrado, REST é uma arquitetura cliente-servidor. No REST, cada requisição de dados contém todas as informações necessárias para responder a solicitação, diminuindo o trabalho do servidor. Não há necessidade de manter o estado das solicitações, permitindo ao servidor desalocar rapidamente os recursos já utilizados e ainda simplificar a implementação.

RESTful é uma implementação simples de um web-server utilizando o HTTP e os princípios REST.

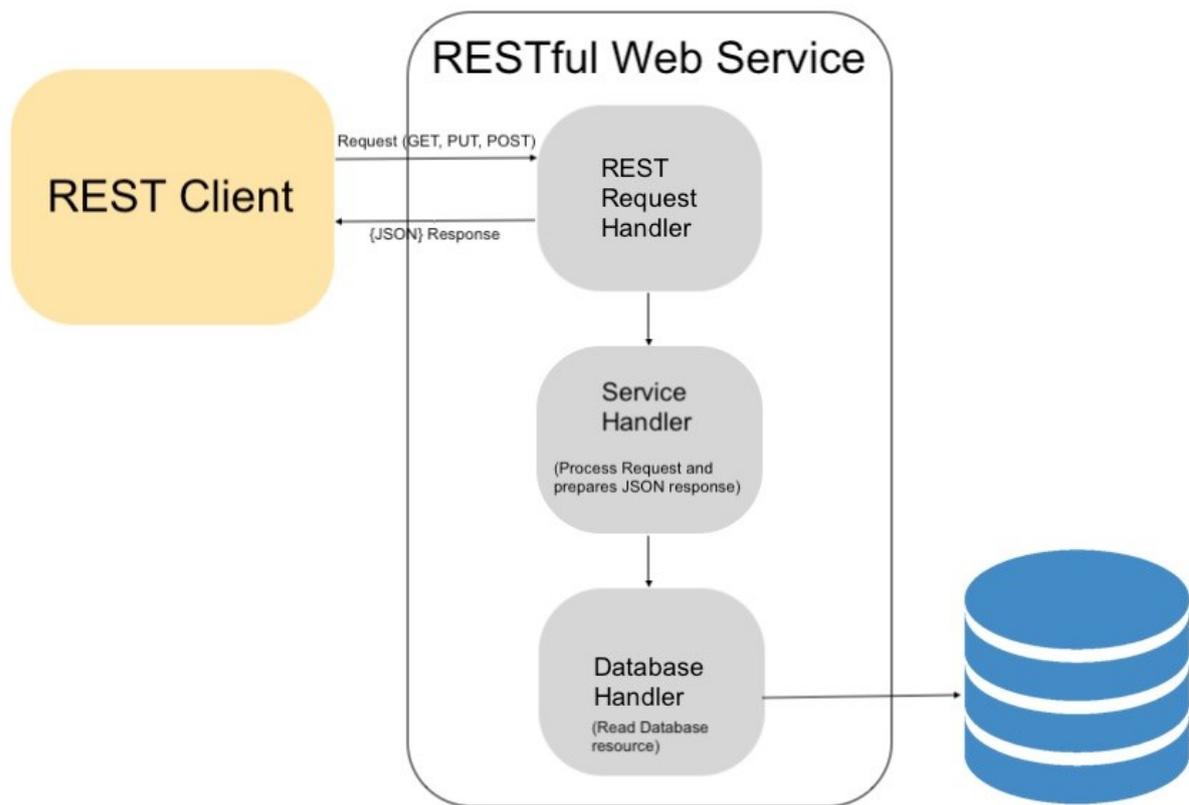
As figuras 14 e 15 demonstram a diferença e o funcionamento das duas arquiteturas.

FIGURA 14 - REST API



Fonte: <https://sites.psu.edu/annaarsiriy/2019/02/10/restful-api/>

FIGURA 15 - RESTful



Fonte: <https://phpspot.com/php/php-restful-web-service/>

1.3.16 Spring Tool Suite

É uma ferramenta para o desenvolvimento de aplicativos, torna a construção rápida e descomplicada. Remove muito do código padrão e da configuração associada ao desenvolvimento da web, apresenta um modelo de programação moderno, agilizando o desenvolvimento de aplicativos, APIs REST e sistemas bidirecionais baseados em eventos.

1.3.17 Postman

O Postman é uma plataforma de desenvolvimento de APIs que tem por objetivo facilitar aos desenvolvedores criar, compartilhar, testar e documentá-las. Permite aos usuários criar e salvar solicitações HTTP/ e HTTPs simples e complexas, bem como ler suas respostas.

Os recursos do Postman otimizam a colaboração para que se possa criar APIs melhores, mais rapidamente.

2 DESENVOLVIMENTO

2.1 Bando de Dados

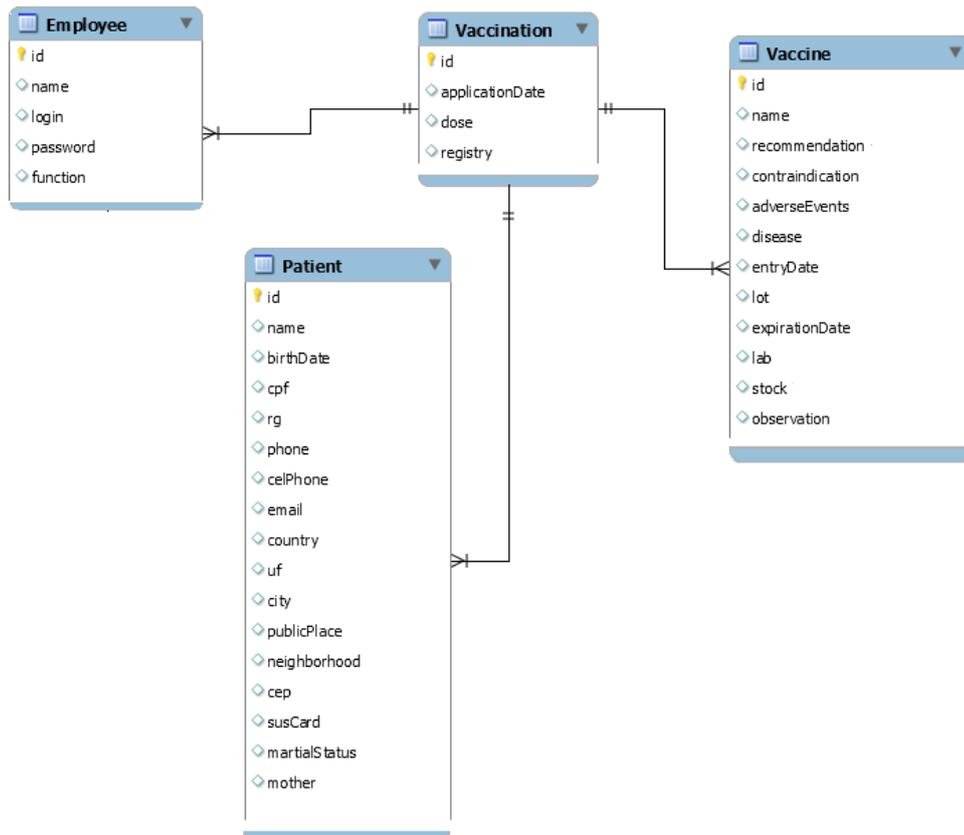
Todo software tem seu início pelo projeto. O escopo foi norteado pelos objetivos gerais e específicos do Trabalho de Conclusão de Curso.

Após análises, os requisitos são detalhados com todas as descrições e tudo o que o sistema realizará e que foi definido no escopo do projeto, esquematizados pelo Diagrama de Casos de Uso, referenciado no Apêndice A, que é uma orientação lógica para o desenvolvimento.

2.1.1 Modelo Lógico

Foram coletadas informações sobre as necessidades de negócios chegando ao modelo lógico representado pela figura 22.

Figura 16 - Modelo Lógico



Fonte: do Autor

2.1.2 Modelo Físico

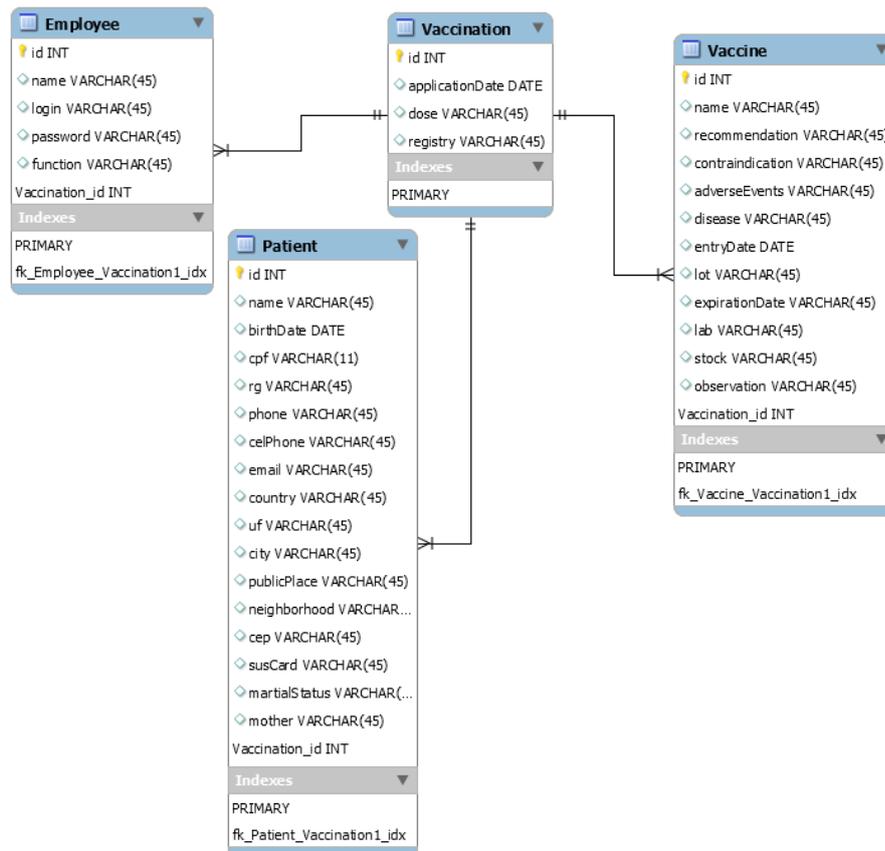
Baseado na modelagem lógica, a modelagem física, conforme figura 23, lida com o design do banco de dados real com base nos requisitos reunidos.

As informações coletadas foram convertidas em modelos relacionais e de negócios.

Chaves primárias e chaves estrangeiras foram definidas para fornecer restrições e ligações.

A modelagem física é baseada no Banco de dados Mysql.

Figura 17 - Modelo Físico

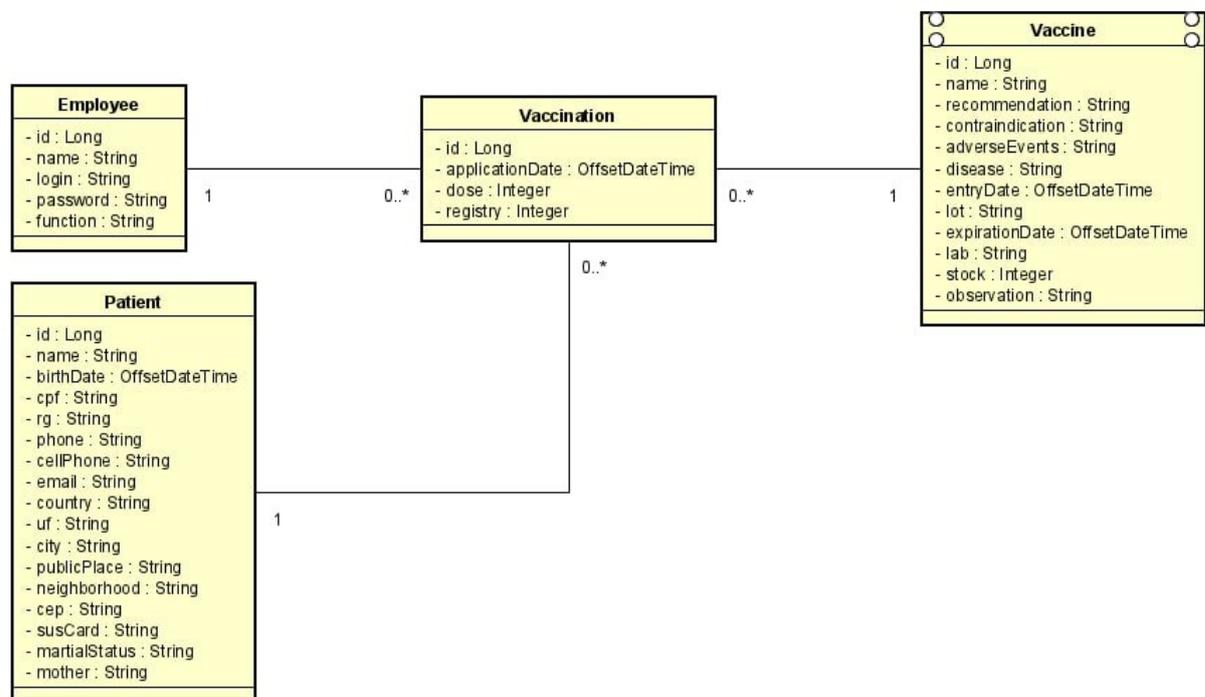


Fonte: do Autor

2.1.3 Diagrama de Classe

Fornecer uma visão estática do conjunto de classes do sistema proposto, componente muito importante em qualquer projeto orientado a objetos. Aqui representado pela figura 24.

Figura 18 - Diagrama de Classe



Fonte: do Autor

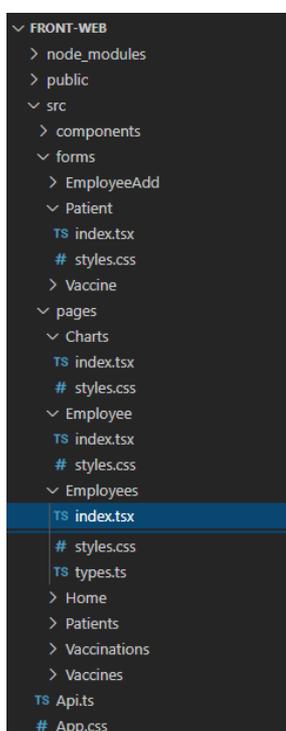
2.2 Aplicativo Web

2.2.1 FrontEnd ReactJS

O FrontEnd da aplicação foi desenvolvido com auxílio desta biblioteca JavaScript pois a mesma é muito popular, de código aberto e permite reuso de componentes desenvolvidos em outras aplicações e que usem a mesma função.

O React tem como um de seus objetivos facilitar a conexão entre diferentes partes de uma aplicação web, seu funcionamento acontece através de componentes. Divide uma tela em diversos componentes para trabalhar sobre eles de maneira individual. Os componentes são utilizados para reaproveitamento de código e padronização de interface. Tornando-o uma tecnologia muito flexível para a solução de problemas e para a construção de interfaces reutilizáveis, uma vez que cada um destes componentes pode ser manipulado de maneira distinta.

Figura 19 - FrontEnd



Fonte: do Autor

2.2.2 BackEnd Java JPA

Para o BackEnd foi utilizado JPA, que é um framework leve, baseado em POJOS (Plain Old Java Objects) para persistir objetos Java.

As tabelas foram mapeadas e acessadas em formato de objetos utilizando essa API, pois traduz um objeto Java para uma tabela de banco de dados e vice versa.

O JPA é composto por várias anotações, responsáveis por mapear os atributos de uma classe em uma tabela de banco de dados.

Cada entidade foi criada utilizando uma classe Java adicionando a ela os atributos ao seu conteúdo, bem como os getters e setters e um construtor padrão.

Conforme representado pelo código abaixo:

```
1 package work.tecnologia.pep.entities;
2 import java.io.Serializable;
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.Table;
8
9 @Entity
10 @Table(name = "employee")
11 public class Employee implements Serializable {
12     private static final long serialVersionUID = 1L;
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String name;
17     private String login;
18     private String password;
19     private String function;
20     public Employee() {
21 }
22     public Employee(Long id, String name, String login, String password, String function) {
23         this.id = id;
```

```
24         this.name = name;
25         this.login = login;
26         this.password = password;
27         this.function = function;
28     }
29     public Long getId() {
30     return id;
31     }
32     public void setId(Long id) {
33         this.id = id;
34     }
35     public String getName() {
36         return name;
37     }
38     public void setName(String name) {
39         this.name = name;
40     }
41     public String getLogin() {
42         return login;
43     }
44     public void setLogin(String login) {
45         this.login = login;
46     }
47     public String getPassword() {
48         return password;
49     }
50     public void setPassword(String password) {
51         this.password = password;
52     }
53     public String getFunction() {
54         return function;
55     }
56     public void setFunction(String function) {
57         this.function = function;
58     }
59     @Override
```

```

60     public int hashCode() {
61         final int prime = 31;
62         int result = 1;
63         result = prime * result + ((id == null) ? 0 : id.hashCode());
64         return result;
65     }
66     @Override
67     public boolean equals(Object obj) {
68         if (this == obj)
69             return true;
70         if (obj == null)
71             return false;
72         if (getClass() != obj.getClass())
73             return false;
74         Employee other = (Employee) obj;
75         if (id == null) {
76             if (other.id != null)
77                 return false;
78         } else if (!id.equals(other.id))
79             return false;
80         return true;
81     }
82     @Override
83     public String toString() {
84         return "Employee [id=" + id + ", name=" + name + ", login=" + login + ",
password=" + password + 85 ", function=" + function + "];
86     }
87 }

```

Para executar de maneira simples a manipulação da base de dados, inserir registros, atualizar, remover, etc. o Spring Data possui o JpaRepository, essa interface somente precisa ser configurada para saber a qual entidade JPA se refere. Dessa forma a aplicação terá disponível a implementação para os principais acessos a base de dados.

Os controllers foram criados utilizando a anotação 'Controller' do Spring Rest, que uma vez adicionada a uma classe Java, aceita um 'path' como parâmetro, disponibilizando esse componente para acesso HTTP.

No código abaixo, o '@RestController' será responsável por criar a API Rest, já o '@RequestMapping', indicará o path do serviço.

Foram mapeados os métodos dos controllers com a anotação '@GetMapping', seguido de um valor por parâmetro. Esta anotação refere-se a requisições HTTP GET, para as demais, apenas substituir o Get por Post, Put, Delete.

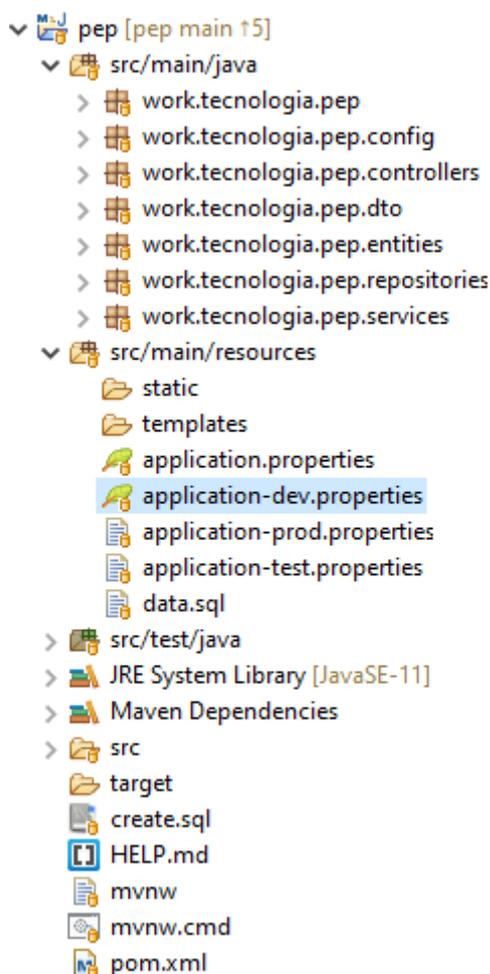
```
1 package work.tecnologia.pep.controllers;
2
3 import java.net.URI;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.data.domain.Page;
7 import org.springframework.data.domain.PageRequest;
8 import org.springframework.data.domain.Sort.Direction;
9 import org.springframework.http.ResponseEntity;
10 import org.springframework.web.bind.annotation.DeleteMapping;
11 import org.springframework.web.bind.annotation.GetMapping;
12 import org.springframework.web.bind.annotation.PathVariable;
13 import org.springframework.web.bind.annotation.PostMapping;
14 import org.springframework.web.bind.annotation.PutMapping;
15 import org.springframework.web.bind.annotation.RequestBody;
16 import org.springframework.web.bind.annotation.RequestMapping;
17 import org.springframework.web.bind.annotation.RequestParam;
18 import org.springframework.web.bind.annotation.RestController;
19 import org.springframework.web.servlet.support.ServletUriComponentsBuilder;
20
21 import work.tecnologia.pep.dto.EmployeeDTO;
22 import work.tecnologia.pep.services.EmployeeService;
23
24 @RestController
25 @RequestMapping(value = "/employees")
26 public class EmployeeController {
```

```
27
28     @Autowired
29     private EmployeeService service;
30
31 //     @GetMapping
32 //     public ResponseEntity<List<EmployeeDTO>> findAll() {
33 //         List<EmployeeDTO> list = service.findAll();
34 //         return ResponseEntity.ok().body(list);
35 //     }
36
37     @GetMapping
38     public ResponseEntity<Page<EmployeeDTO>> findAll(
39         @RequestParam(value = "page", defaultValue = "0") Integer page,
40         @RequestParam(value = "linesPerPage", defaultValue = "0") Integer
41 linesPerPage,
42         @RequestParam(value = "orderBy", defaultValue = "name") String orderBy,
43         @RequestParam(value = "direction", defaultValue = "ASC") String direction
44     ){
45
46     if(linesPerPage == 0) {
47         linesPerPage = Integer.MAX_VALUE;
48     }
49
50     PageRequest pageRequest = PageRequest.of(page, linesPerPage,
51 Direction.valueOf(direction), orderBy);
52
53 Page<EmployeeDTO> body = service.findAll(pageRequest);
54
55     return ResponseEntity.ok().body(body);
56 }
```

Para trafegar os dados de maneira organizada e facilmente gerenciados pelo Spring, foi utilizado o padrão de projetos DTO (Data Transfer Object), pois permitem que os dados de uma requisição sejam facilmente convertidos para uma classe Java. O que nada mais é que uma classe Java contendo atributos e getters e setters.

O sistema de arquivos mostrado na figura 26 demonstra como as pastas fazem referência à arquitetura MVC.

Figura 20 - BackEnd



Fonte: do Autor

3. RESULTADOS E DISCUSSÕES

A maior dificuldade em implementar um sistema de interoperabilidade é a quantidade de protocolos existentes para padronização. Esse excesso dificulta para que as unidades de saúde saibam identificar quais protocolos são fundamentais e quais podem ser dispensados.

O back-end desenvolvido resulta na independência da aplicação cliente, podendo ser desenvolvida em qualquer linguagem, bem como criada para qualquer aplicativo, tanto mobile quanto desktop.

Permite o compartilhamento de informações clínicas, no caso deste o compartilhamento de informações referente a vacinação, de forma simples e objetiva.

Como a informação é retirada de um único local e utilizado um padrão de interoperabilidade, a mesma tende a minimizar os erros e maximizar os cuidados médicos.

Desenvolvido sob o paradigma da Orientação Objeto, pode ser alterado e adequado facilmente para atender a demanda de novas informações do paciente.

Após analisar a padronização de dados HL7 verificou-se que o mesmo é um "livro de regras" com milhares de informações que estabelecem e detalham um framework para negociação e construção de mensagens, oferecendo técnicas para interoperabilidade.

Estudos do banco de dados MySQL o referendaram para utilização na aplicação, uma vez que há documentação, conectores e portabilidade para diversos sistemas operacionais, além da robustez.

Testes através do POSTMAN validaram os resources obtidos, conforme figura35, bem como as telas do sistema, referenciadas no Apêndice B.

Figura 21 - Postman

The screenshot displays the Postman interface with a GET request to `http://localhost:8080/patients` executed. The response is a JSON object representing a patient record. The interface includes a sidebar with collections, a main workspace for editing requests, and a bottom console for logs.

Request Details:

- Method: GET
- URL: `http://localhost:8080/patients`
- Params: Key, Value, Description

Response Body (Pretty):

```
1 {
2   "id": 4,
3   "name": "Deivid",
4   "sex": null,
5   "birthDate": null,
6   "cpf": "000.000.000-00",
7   "rg": "111111111111",
8   "phone": "(55) 5555-5555",
9   "cellPhone": "(55) 5555-5555",
10  "email": null,
11  "country": null,
12 }
```

Fonte: do Autor

4. CONSIDERAÇÕES FINAIS

Há diversas variações em implementações do HL7, uma bastante comum, a definição para a raça do paciente, que poderia ser uma abreviação ou um código exclusivo da instituição de saúde. Em determinada instituição de saúde, o código RC03 poderia definir a raça do paciente como parda, enquanto que em uma outra, este código poderia representar um estado do país.

Num possível aperfeiçoamento do sistema, a integração com dispositivos médicos seria de grande valia, tais como monitores cardíacos, equipamentos de diagnósticos por imagem, etc.

REFERÊNCIAS

APRENDER JPA: aprendizagem Simplesmente Fácil. **Tutorialspoint.com**, 2020. Disponível em: <https://www.tutorialspoint.com/pg/jpa/index.htm>. Acesso em: 4 ago. 2020.

AULA03-1 **Modelagem de Casos de Uso**. [S. I.], 12 abr. 2014. Disponível em: <http://www.facom.ufu.br/~bacala/DAW/>. Acesso em: 7 abr. 2020.

DAMAS, Luís. **SQL, Structured query language**; tradução Elvira Maria Antunes Uchôa. 6ª Edição, Rio de Janeiro: LTC, 2007.

DIAS, Claudia Augusto. **Terminologia: conceitos e aplicações**. Ciência da Informação, Brasília, v.29, n. 1, p. 90-92, jan/abr 2000. Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/270> >.

HAPI Fhir. [S. I.], 28 jan. 2020. Disponível em: <https://hapifhir.io/>. Acesso em: 10 fev. 2020.

HEALTH LEVEL SEVEN INTERNATIONAL [Internet]. **Ann Arbor (MI): HL7 International**, 2014. Disponível em: < <http://www.hl7.org/index.cfm> >. Acesso em: 15 fev. 2020.

MANZANO, José Augusto N.G. **MySQL 5.5 interativo: guia essencial de orientação e desenvolvimento**. 1ª edição, São Paulo: Érica, 2011.

OLIVEIRA, V. S. **Buscando interoperabilidade entre diferentes bases de dados: o caso da Biblioteca do Instituto Fernandes Figueira**. 2005. 109 f. Dissertação (Mestrado em Saúde Pública)- Fundação Oswaldo Cruz, Escola Nacional de Saúde Pública Sérgio Arouca, Rio de Janeiro, 2005. Disponível em: <<http://www.arca.fiocruz.br/handle/icict/1318>>.

L., Andrei. **O QUE é React e Como Funciona?**. [S. I.], 21 maio 2019. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>. Acesso em: 21 nov. 2020.

PROJETO ReactJS. [S. I.], 1 jan. 2021. Disponível em: <https://pt-br.reactjs.org/>. Acesso em: 1 fev. 2021.

QUAL a diferença entre modelagem conceitual, lógica e física?. [S. I.], 1 nov. 2018. Disponível em: <https://pt.stackoverflow.com/questions/294699/qual-a-diferen%C3%A7a-entre-modelagem-conceitual-l%C3%B3gica-e-f%C3%ADsica>. Acesso em: 11 fev. 2021.

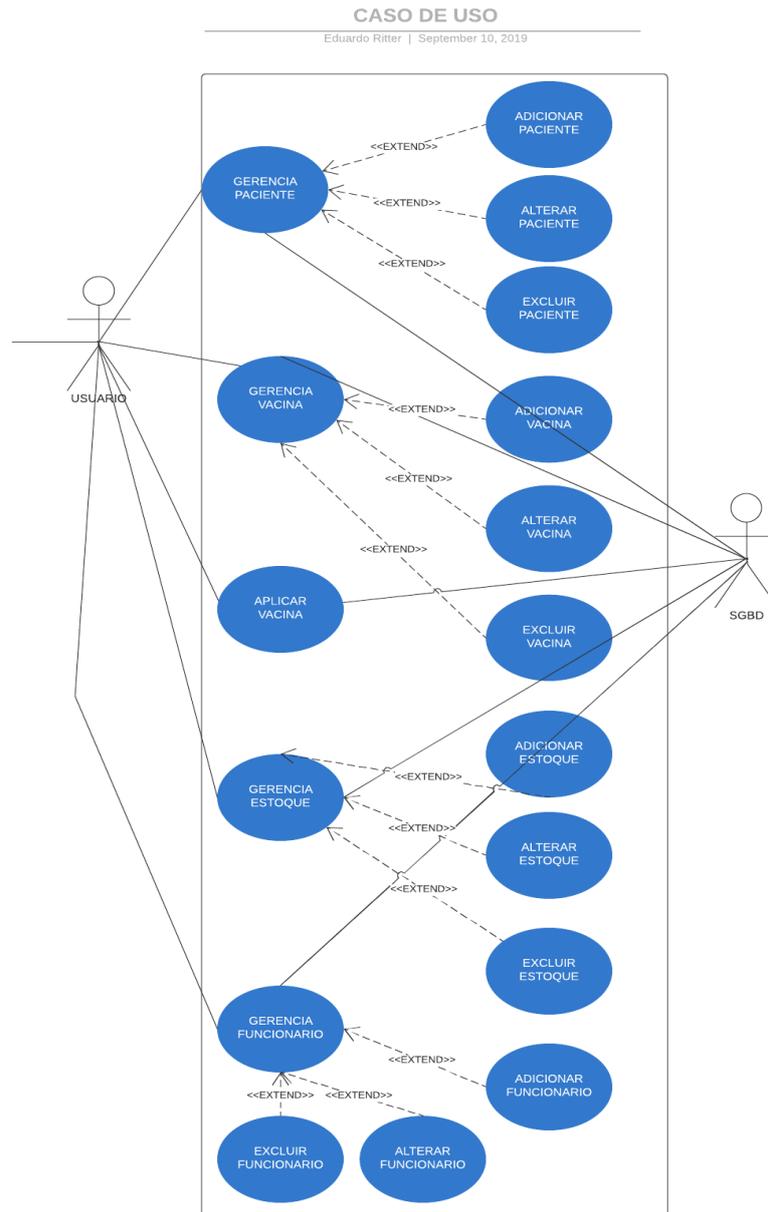
ROCHA DE SOUZA, Givanaldo. Engenharia de Software. **Diagrama de Caso de Uso**, [S. l.], p. 1-45, 12 abr. 2014. Disponível em: <https://docente.ifrn.edu.br/givanaldorochoa/disciplinas/engenharia-de-software-licenciatura-em-informatica/diagrama-de-caso-de-uso-tecnico>. Acesso em: 7 abr. 2020.

FLORES, Gabriela. **Tutorial de utilização do padrão HL7 FHIR**, [S. l.], p. 1, 22 set. 2018. Disponível em: <https://medium.com/@floresgabriela/tutorial-de-utiliza%C3%A7%C3%A3o-do-padr%C3%A3o-hl7-fhir-para-troca-de-mensagens-de-sistemas-em-java-335907ca71c6>. Acesso em: 15 jun. 2020.

UDEMYY: Java Persistence: Hibernate and JPA Fundamentals. [S. l.], 2 abr. 2020. Disponível em: https://www.udemy.com/course/hibernate-and-jpa-fundamentals/?matchtype=b&msclkid=083089d8ab731fdc89abe9f614c7d74b&utm_campaign=BG-LongTail_la.EN_cc.ROW&utm_content=deal4584&utm_medium=udemyads&utm_source=bing&utm_term=._ag_1210562715354467_.ad_.kw_%2BJPA+%2BTutorial_de_c_dm_pl_ti_kwd-75660480903335_li_685_pd_. Acesso em: 4 jan. 2021.

APÊNDICE A

Diagramas Geral de Casos de Uso



Especificações de Casos de Uso

Nome do caso de uso	Gerenciar Paciente
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode incluir, alterar, excluir ou consultar um paciente
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Inserir dados (nome, CPF, cartão SUS) referente ao paciente a ser localizado ou efetua a busca	
	Retorna o(s) paciente(s) encontrados(s)
	Se inserir novo cadastro, executar caso de uso Adicionar Paciente.
	Se alterar cadastro, executar caso de uso Alterar Paciente.
	Se excluir cadastro, executar caso de uso Excluir Paciente
	Excluir Paciente.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Adicionar Paciente
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode adicionar um paciente
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o cadastro de um novo paciente	
	Apresenta formulário de cadastro do paciente
Informa os dados necessários e cadastra	
	Valida os dados e cadastra novo paciente no SGBD.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos
Informa paciente já existente	
	Sistema Informa que o paciente já está cadastrado. Retorna a tela de cadastro

Nome do caso de uso	Alterar Paciente
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode alterar um paciente
Pré-Condições	Existir ao menos um paciente
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o paciente para alterar	
	Apresenta formulário do paciente
Modifica os dados do paciente	
Confirma	
	Valida os dados e salva no SGBD
	Caso de uso se encerra

Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Excluir Paciente
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode excluir um paciente
Pré-Condições	Existir ao menos um paciente
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o paciente para excluir	
	Apresenta tela de exclusão
Confirma exclusão	
	Remove o paciente do SGBD
	Caso de uso se encerra

Nome do caso de uso	Gerenciar Vacinas
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode incluir, alterar, excluir ou consultar uma vacina.
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Insere dados referente a vacina a ser localizada	
	Retorna a(s) vacina(s) encontrada(s).
	Se inserir novo cadastro, executar caso de uso Adicionar Vacina.
	Se alterar cadastro, executar caso de uso Alterar Vacina.
	Se excluir cadastro, executar caso de uso Excluir Vacina.

	Excluir Vacina.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Gerenciar Estoque
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode incluir, alterar, excluir ou consultar um paciente
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Inserir dados (nome, vacina) referente ao estoque a ser localizado ou efetua a busca	
	Retorna o(s) estoque(s) encontrados(s)
	Se inserir novo cadastro, executar caso de uso Adicionar Estoque.
	Se alterar cadastro, executar caso de uso Alterar Estoque.
	Se excluir cadastro, executar caso de uso Excluir Estoque.
	Excluir Estoque.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Adicionar Estoque
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode adicionar uma nova vacina
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o cadastro de uma nova vacina.	
	Apresenta formulário de cadastro do estoque
Informa os dados necessários e cadastra	
	Valida os dados e cadastra o estoque no SGBD.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos
Informa estoque já existente	
	Sistema Informa que o paciente já está cadastrado. Retorna a tela de cadastro

Nome do caso de uso	Alterar Estoque
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode alterar um estoque de vacina.
Pré-Condições	Existir ao menos um paciente
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o estoque para alterar	
	Apresenta formulário do estoque.
Modifica os dados do estoque.	
Confirma	
	Valida os dados e salva no SGBD
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Excluir Estoque
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode excluir uma vacina
Pré-Condições	Existir ao menos um estoque de vacina cadastrado
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o estoque para excluir	
	Apresenta tela de exclusão
Confirma exclusão	
	Remove o estoque do SGBD
	Caso de uso se encerra

Nome do caso de uso	Adicionar Vacina
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode adicionar uma nova vacina
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o cadastro de um novo paciente	
	Apresenta formulário de cadastro do paciente
Informa os dados necessários e cadastra	
	Valida os dados e cadastra novo paciente no SGBD.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos
Informa paciente já existente	
	Sistema Informa que o paciente já está cadastrado. Retorna a tela de cadastro

Nome do caso de uso	Alterar Vacina
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode alterar uma vacina.
Pré-Condições	Existir ao menos um paciente
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita a vacina para alterar	
	Apresenta formulário da vacina
Modifica os dados da vacina	
Confirma	
	Valida os dados e salva no SGBD
	Caso de uso se encerra

Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Excluir Vacina
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode excluir uma vacina
Pré-Condições	Existir ao menos uma vacina
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita a vacina para excluir	
	Apresenta tela de exclusão
Confirma exclusão	
	Remove a vacina do SGBD
	Caso de uso se encerra
Fluxo de Exceção	
Informar sobre vacina associada	
	Sistema informa que a vacina não pode ser excluída devido a doses já aplicadas

Nome do caso de uso	Gerenciar Funcionário
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode incluir, alterar, excluir ou consultar um funcionário
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Insere dados (nome, login, senha) referente ao funcionário a ser localizado ou efetua a busca	
	Retorna o(s) funcionário(s) encontrados(s)

	Se inserir novo cadastro, executar caso de uso Adicionar Funcionário.
	Se alterar cadastro, executar caso de uso Alterar Funcionário.
	Se excluir cadastro, executar caso de uso Excluir Funcionário.
	Excluir Funcionário.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Adicionar Funcionário
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode adicionar um funcionário
Pré-Condições	Nenhuma
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o cadastro de um novo funcionário	
	Apresenta formulário de cadastro do funcionário
Informa os dados necessários e cadastra	
	Valida os dados e cadastra novo funcionário no SGBD.
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos
Informa funcionário já existente	

	Sistema Informa que o funcionário já está cadastrado. Retorna a tela de cadastro
--	---

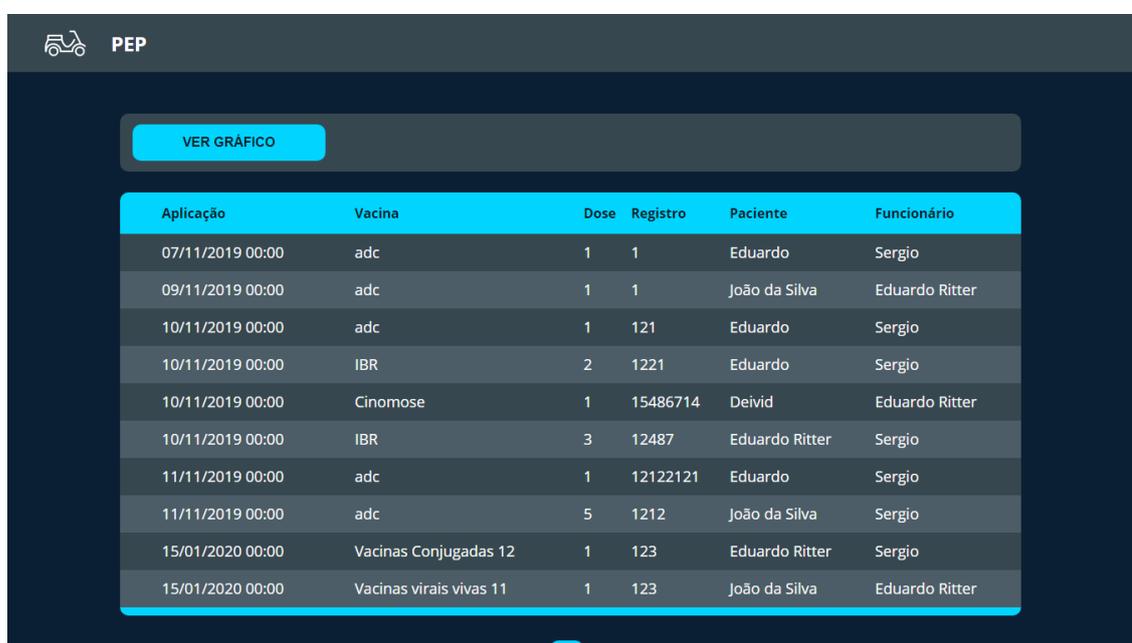
Nome do caso de uso	Alterar Funcionário
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode alterar um funcionário
Pré-Condições	Existir ao menos um funcionário
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o funcionário para alterar	
	Apresenta formulário do funcionário
Modifica os dados do funcionário	
Confirma	
	Valida os dados e salva no SGBD
	Caso de uso se encerra
Fluxo de Exceção	
Informar falta de dados ou dados incorretos	
	Sistema informa que os dados estão incorretos ou não fornecidos

Nome do caso de uso	Excluir Funcionário
Atores Envolvidos	Usuário, SGBD
Descrição	Descreve como o usuário pode excluir um funcionário
Pré-Condições	Existir ao menos um funcionário
Fluxo Básico	
Ações do Ator	Ações do Sistema
Solicita o funcionário para excluir	
	Apresenta tela de exclusão
Confirma exclusão	
	Remove o funcionário do SGBD
	Caso de uso se encerra

APÊNDICE B

Telas do sistema

Figura 22 - Tela Sistema 1



The screenshot displays the PEP system interface. At the top left, there is a logo and the text 'PEP'. Below this, a blue button labeled 'VER GRÁFICO' is visible. The main content is a table with the following columns: 'Aplicação', 'Vacina', 'Dose', 'Registro', 'Paciente', and 'Funcionário'. The table contains 12 rows of data, each representing a vaccination record.

Aplicação	Vacina	Dose	Registro	Paciente	Funcionário
07/11/2019 00:00	adc	1	1	Eduardo	Sergio
09/11/2019 00:00	adc	1	1	João da Silva	Eduardo Ritter
10/11/2019 00:00	adc	1	121	Eduardo	Sergio
10/11/2019 00:00	IBR	2	1221	Eduardo	Sergio
10/11/2019 00:00	Cinomose	1	15486714	Deivid	Eduardo Ritter
10/11/2019 00:00	IBR	3	12487	Eduardo Ritter	Sergio
11/11/2019 00:00	adc	1	12122121	Eduardo	Sergio
11/11/2019 00:00	adc	5	1212	João da Silva	Sergio
15/01/2020 00:00	Vacinas Conjugadas 12	1	123	Eduardo Ritter	Sergio
15/01/2020 00:00	Vacinas virais vivas 11	1	123	João da Silva	Eduardo Ritter

Fonte: do Autor

A figura 27 representa a tela de pacientes vacinados, com seus respectivos dados.

Figura 23 - Tela Sistema 2

```
{
  "content": [
    {
      "id": "1",
      "applicationDate": "2019-11-07T00:00:00-03:00",
      "dose": "1",
      "registry": "1",
      "employee": {
        "id": "1",
        "name": "Sergio",
        "login": "sergio",
        "password": "123456",
        "function": "Biomédico",
        "patient": {
          "id": "1",
          "name": "Eduardo",
          "sex": null,
          "birthDate": null,
          "cpf": "000.000.000-00",
          "rg": "71000111",
          "phone": "(55) 5555-5555",
          "cellPhone": "(55) 5555-5555",
          "email": null,
          "country": null,
          "uf": "RS",
          "city": "Santo Angelo",
          "publicPlace": "Crisantemos 482",
          "neighborhood": "Henezes",
          "cep": null,
          "susCard": null,
          "maritalStatus": null,
          "mother": null,
          "vaccine": ""
        },
        "entryDate": "2019-11-06T00:00:00-03:00",
        "lot": "2124sasas",
        "expirationDate": "2019-11-05T00:00:00-03:00",
        "lab": "1",
        "stock": "12121",
        "observation": "212"},
        {
          "id": "2",
          "name": "João da Silva",
          "sex": null,
          "birthDate": null,
          "cpf": "555.555.555-55",
          "rg": "707070707074",
          "phone": "(55) 5555-5555",
          "cellPhone": "(55) 5555-5555",
          "email": null,
          "country": null,
          "uf": "RS",
          "city": "Santo Angelo",
          "publicPlace": "Centro",
          "neighborhood": "Centro",
          "cep": null,
          "susCard": null,
          "maritalStatus": null,
          "mother": "Maria da Silva",
          "vaccine": ""
        },
        {
          "id": "3",
          "name": "Eduardo Ritter",
          "login": "eritter",
          "password": "eritter",
          "function": "TI",
          "patient": {
            "id": "2",
            "name": "João da Silva",
            "sex": null,
            "birthDate": null,
            "cpf": "555.555.555-55",
            "rg": "707070707074",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Centro",
            "neighborhood": "Centro",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": "Maria da Silva",
            "vaccine": ""
          },
          "entryDate": "2019-11-06T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-05T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "4",
          "name": "Sergio",
          "login": "sergio",
          "password": "123456",
          "function": "Biomédico",
          "patient": {
            "id": "1",
            "name": "Eduardo",
            "sex": null,
            "birthDate": null,
            "cpf": "000.000.000-00",
            "rg": "71000111",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Crisantemos 482",
            "neighborhood": "Henezes",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": null,
            "vaccine": ""
          },
          "entryDate": "2019-11-06T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-05T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "5",
          "name": "Eduardo Ritter",
          "login": "eritter",
          "password": "eritter",
          "function": "TI",
          "patient": {
            "id": "1",
            "name": "Eduardo",
            "sex": null,
            "birthDate": null,
            "cpf": "000.000.000-00",
            "rg": "71000111",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Crisantemos 482",
            "neighborhood": "Henezes",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": null,
            "vaccine": ""
          },
          "entryDate": "2019-11-06T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-05T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "6",
          "name": "Eduardo Ritter",
          "login": "eritter",
          "password": "eritter",
          "function": "TI",
          "patient": {
            "id": "4",
            "name": "Deivid",
            "sex": null,
            "birthDate": null,
            "cpf": "000.000.000-00",
            "rg": "111111111111",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Feiteiras",
            "neighborhood": "Vermelho",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": null,
            "vaccine": ""
          },
          "entryDate": "2019-11-10T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-09T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "7",
          "name": "Sergio",
          "login": "sergio",
          "password": "123456",
          "function": "Biomédico",
          "patient": {
            "id": "3",
            "name": "Eduardo Ritter",
            "sex": null,
            "birthDate": null,
            "cpf": "000.711.050-26",
            "rg": "7077597669",
            "phone": "(55) 3313-2000",
            "cellPhone": "(55) 98126-1800",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Tv Crisantemos, 482",
            "neighborhood": "Henezes",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": null,
            "vaccine": ""
          },
          "entryDate": "2019-11-11T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-10T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "8",
          "name": "Sergio",
          "login": "sergio",
          "password": "123456",
          "function": "Biomédico",
          "patient": {
            "id": "1",
            "name": "Eduardo",
            "sex": null,
            "birthDate": null,
            "cpf": "000.000.000-00",
            "rg": "71000111",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Crisantemos 482",
            "neighborhood": "Henezes",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": null,
            "vaccine": ""
          },
          "entryDate": "2019-11-06T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2019-11-05T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"},
        {
          "id": "9",
          "name": "João da Silva",
          "sex": null,
          "birthDate": null,
          "cpf": "555.555.555-55",
          "rg": "707070707074",
          "phone": "(55) 5555-5555",
          "cellPhone": "(55) 5555-5555",
          "email": null,
          "country": null,
          "uf": "RS",
          "city": "Santo Angelo",
          "publicPlace": "Centro",
          "neighborhood": "Centro",
          "cep": null,
          "susCard": null,
          "maritalStatus": null,
          "mother": "Maria da Silva",
          "vaccine": ""
        },
        {
          "id": "10",
          "name": "Eduardo Ritter",
          "login": "eritter",
          "password": "eritter",
          "function": "TI",
          "patient": {
            "id": "2",
            "name": "João da Silva",
            "sex": null,
            "birthDate": null,
            "cpf": "555.555.555-55",
            "rg": "707070707074",
            "phone": "(55) 5555-5555",
            "cellPhone": "(55) 5555-5555",
            "email": null,
            "country": null,
            "uf": "RS",
            "city": "Santo Angelo",
            "publicPlace": "Centro",
            "neighborhood": "Centro",
            "cep": null,
            "susCard": null,
            "maritalStatus": null,
            "mother": "Maria da Silva",
            "vaccine": ""
          },
          "entryDate": "2020-01-15T00:00:00-03:00",
          "lot": "2124sasas",
          "expirationDate": "2020-01-15T00:00:00-03:00",
          "lab": "1",
          "stock": "12121",
          "observation": "212"}
      ]
    }
  ]
}
```

Fonte: do Autor

Na figura 28, constam os dados exportados para JSON.

Figura 24 - Tela Sistema 3

Vacina	Recomendação	Lote	Estoque	Vencimento
Vacinas Conjugadas 12				
Vacinas virais vivas 11				
Cinomose				
IBR				
Brucelose				
Doença de Marek				
Erisipela				
adc	asc	2124sasas	12121	2019-11-05T00:00:00-03:00
teste1	este1	12454-c	5	2022-01-01T00:00:00-03:00

Fonte: do Autor

A figura 29 representa a tela de cadastro das vacinas e seus respectivos dados. Já a figura 30, os dados em JSON.

Figura 25 - Tela Sistema 4

```
{
  "content": [
    {
      "id": 2, "name": "Vacinas Conjugadas",
      "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 3, "name": "Vacinas virais vivas",
      "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 4, "name": "Cinomose", "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 5, "name": "ISB", "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 6, "name": "Brucelose",
      "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 7, "name": "Doença de Harek", "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 8, "name": "Erisipela", "recommendation": "", "contraindication": "", "adverseEvents": "", "disease": "", "entryDate": null, "lot": "", "expirationDate": null, "lab": "", "stock": null, "observation": ""
    },
    {
      "id": 9, "name": "adc", "recommendation": "asc", "contraindication": "asc", "adverseEvents": "ssasa", "disease": "assa", "entryDate": "2019-11-06T00:00:00-03:00", "lot": "121454545", "expirationDate": "2019-11-05T00:00:00-03:00", "lab": "1", "stock": "12121", "observation": "212"
    },
    {
      "id": 9, "name": "testet1", "recommendation": "estet1", "contraindication": "estet1", "adverseEvents": "estet1", "disease": "estet1", "entryDate": "2021-01-01T00:00:00-03:00", "lot": "12454-c", "expirationDate": "2022-01-01T00:00:00-03:00", "lab": "Phyfyser", "stock": "5", "observation": "testet1"}
  ],
  "pageable": {
    "sort": {
      "sorted": true, "unsorted": false, "empty": false
    },
    "offset": 0, "pageNumber": 0, "pageSize": 12, "pageable": true, "unpaged": false, "last": true, "totalElements": 9, "totalPages": 1, "size": 12, "number": 0, "numberElements": 9, "empty": false
  }
}
```

Fonte: do Autor

Figura 26 - Tela Sistema 5

Cartão SUS	Nome	Mãe	Cidade
	Deivid		Santo Ângelo
	Eduardo		Santo Ângelo
	Eduardo Ritter		Santo Ângelo
	João da Silva	Maria da Silva	Santo Ângelo

Fonte: do Autor

A figura 31, representa a tela de consulta do cadastro de pacientes e a figura 32 os dados em JSON.

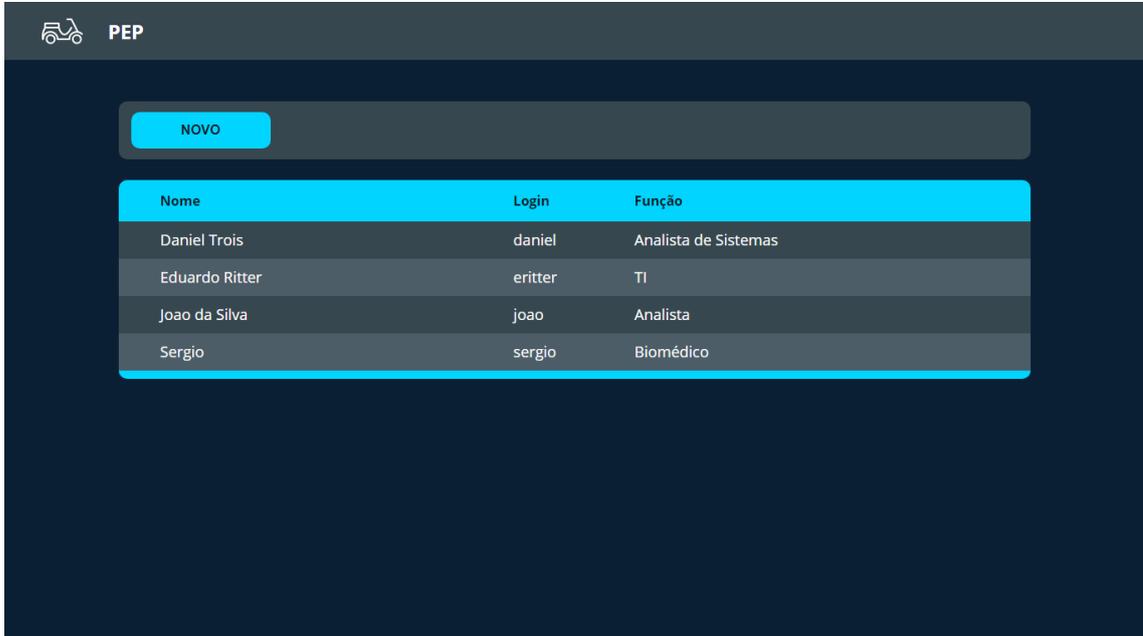
Figura 27 - Tela Sistema 6

```
{
  "content": [
    {
      "id": 4,
      "name": "Deivid",
      "sex": null,
      "birthDate": null,
      "cpf": "000.000.000-00",
      "rg": "11111111111",
      "phone": "(55) 5555-5555",
      "cellPhone": "(55) 5555-5555",
      "email": null,
      "country": null,
      "uf": "RS",
      "city": "Santo \u00c2ngelo",
      "publicPlace": "Feitic\u00e7\u00e3s",
      "neighborhood": "Vermelho",
      "cep": null,
      "susCard": null,
      "maritalStatus": null,
      "mother": null
    },
    {
      "id": 1,
      "name": "Eduardo",
      "sex": null,
      "birthDate": null,
      "cpf": "000.000.000-00",
      "rg": "711000111",
      "phone": "(55) 5555-5555",
      "cellPhone": "(55) 5555-5555",
      "email": null,
      "country": null,
      "uf": "RS",
      "city": "Santo \u00c2ngelo",
      "publicPlace": "Crisantemos 482",
      "neighborhood": "Menezes",
      "cep": null,
      "susCard": null,
      "maritalStatus": null,
      "mother": null
    },
    {
      "id": 3,
      "name": "Eduardo Ritter",
      "sex": null,
      "birthDate": null,
      "cpf": "000.711.050-26",
      "rg": "7077597669",
      "phone": "(55) 3313-2000",
      "cellPhone": "(55) 98126-1800",
      "email": null,
      "country": null,
      "uf": "RS",
      "city": "Santo \u00c2ngelo",
      "publicPlace": "Tv Crisantemos, 482",
      "neighborhood": "Menezes",
      "cep": null,
      "susCard": null,
      "maritalStatus": null,
      "mother": null
    },
    {
      "id": 2,
      "name": "Jo\u00e3o da Silva",
      "sex": null,
      "birthDate": null,
      "cpf": "555.555.555-55",
      "rg": "707070707074",
      "phone": "(55) 5555-5555",
      "cellPhone": "(55) 5555-5555",
      "email": null,
      "country": null,
      "uf": "RS",
      "city": "Santo \u00c2ngelo",
      "publicPlace": "Centro",
      "neighborhood": "Centro",
      "cep": null,
      "susCard": null,
      "maritalStatus": null,
      "mother": "Maria da Silva"
    }
  ],
  "pageable": {
    "sort": {
      "sorted": true,
      "unsorted": false,
      "empty": false,
      "offset": 0,
      "pageNumber": 0,
      "pageSize": 2147483647,
      "paged": true,
      "unpaged": false,
      "last": true,
      "totalElements": 4,
      "totalPages": 1,
      "size": 2147483647,
      "number": 0,
      "sort": {
        "sorted": true,
        "unsorted": false,
        "empty": false,
        "first": true,
        "numberOfElements": 4,
        "empty": false
      }
    }
  }
}
```

Fonte: do Autor

A tela de sistema abaixo, representada pela figura 33, mostra os usu\u00e1rios cadastrados e os respectivos dados. J\u00e1 a figura 34, os dados gerados para exporta\u00e7\u00e3o em JSON.

Figura 28 - Tela Sistema 7



The screenshot shows a dark-themed web interface. At the top left, there is a logo and the text 'PEP'. Below this, there is a prominent blue button labeled 'NOVO'. Underneath the button is a table with three columns: 'Nome', 'Login', and 'Função'. The table contains four rows of user data.

Nome	Login	Função
Daniel Trois	daniel	Analista de Sistemas
Eduardo Ritter	eritter	TI
Joao da Silva	joao	Analista
Sergio	sergio	Biomédico

Fonte: do Autor

Figura 29 - Tela Sistema 8

```
{
  "content": [
    {
      "id": 4,
      "name": "Daniel Trois",
      "login": "daniel",
      "password": "1anove",
      "function": "Analista de Sistemas"
    },
    {
      "id": 2,
      "name": "Eduardo Ritter",
      "login": "eritter",
      "password": "eritter",
      "function": "TI"
    },
    {
      "id": 3,
      "name": "Joao da Silva",
      "login": "joao",
      "password": "maria da silva",
      "function": "Analista"
    },
    {
      "id": 1,
      "name": "Sergio",
      "login": "sergio",
      "password": "123456",
      "function": "Biomédico"
    }
  ],
  "pageable": {
    "sort": {
      "sorted": true,
      "unsorted": false,
      "empty": false,
      "offset": 0,
      "pageNumber": 0,
      "pageSize": 2147483647,
      "paged": true,
      "unpaged": false,
      "last": true,
      "totalElements": 4,
      "totalPages": 1,
      "size": 2147483647,
      "number": 0,
      "sort": {
        "sorted": true,
        "unsorted": false,
        "empty": false,
        "first": true,
        "numberOfElements": 4,
        "empty": false
      }
    }
  }
}
```

Fonte: do Autor