

Uma API RESTful para gerenciar a vacinação com Spring Boot 2

Douglas De Jesus Fonseca

Curso de Análise e Desenvolvimento de Sistemas
Instituto Federal da Bahia - Campus Salvador
Salvador, Brasil
douglas.vasco79@gmail.com

Flavia Maristela Santos Nascimento

Curso de Análise e Desenvolvimento de Sistemas
Instituto Federal da Bahia - Campus Salvador
Salvador, Brasil
flaviamnsn@ifba.edu.br

Resumo—A vacina é uma das formas utilizadas para o combate das doenças bacterianas ou virais, podendo levar o indivíduo à óbito, de acordo com os riscos. Para que haja um gerenciamento de estoques e campanhas das vacinas, foi desenvolvido uma API RESTful para que possa ser fornecida uma interface para sistemas multiplataforma. Neste trabalho, descrevemos o desenvolvimento de uma API baseada no framework Spring Boot 2 que fornecerá módulos que buscam minimizar questões que poderiam ser complexas para a criação de um Api Rest robusta, como por exemplo APIs da Google. O objetivo deste trabalho, é gerenciar o fluxo de vacinação. Sabendo que para que haja um controle eficiente de estoque de vacinas, é necessário que a API consiga controlar as aplicações nos pacientes de destino, controlar as validades das doses, aplicações indevidas, além de emitir relatórios que servirão como base para campanhas de vacinação posteriores. Possibilitando também, informar aos pacientes notificações via e-mail e/ou SMS (Short Message Service) as doses e as campanhas de acordo com a sua faixa etária. Além disso, fornecer dados da carteira de vacinação para o controle do paciente e evitando perdas, ao contrário da carteira vacinal no formato de papel.

Index Terms—Vaccine - management - stock - control - campaign - API RESTful

I. INTRODUÇÃO

A imunização é o método mais eficaz de atenuar a ocorrência de doenças preveníveis através da vacinação. Sendo assim, a vacinação é um relevante instrumento de saúde pública no cuidado e controle de doenças imunopreveníveis. Neste contexto, e em virtude do crescimento acelerado, as **Tecnologias de Informação e Comunicação (TIC)**, estão se tornando como recursos essenciais na área da saúde [19] [20].

Os sistemas de saúde aplicam as TIC de inúmeras maneiras para ofertar distintos indivíduos envolvidos, assim como profissionais quanto pacientes/clientes, as ferramentas fundamentais para melhorar a performance dos atendimentos de saúde e de todas as funções de sustentação do serviço. A TIC Saúde, tem como finalidade a busca e a análise do estágio de adoção de TIC em estabelecimentos de saúde brasileiros, verificando o acesso e a utilização das TIC nas entidades públicas e privadas de saúde. Além também, de destacar a ampliação da saúde digital, que deve ser vista como uma política pública indispensável e imediata para aprimorar a condição de acesso e qualidade dos cuidados à saúde. Os resultados de 2019 mostram um avanço, sendo que 82% das instituições têm sistemas

eletrônicos para registro de informações, ao mesmo tempo que, em 2018 esse número era de 73%. Vale salientar que as menores proporções de instituições com sistema eletrônico foram na esfera pública [20].

As estratégias de vacinação no Brasil ocuparam um lugar de destaque e alcançaram elevados índices de eficiência, servindo como exemplo para outros países, visto que, as campanhas contra a varíola, poliomielite e o sarampo, proporcionaram eficiência na cobertura vacinal coordenadas pelo Ministério da Saúde [2]. A necessidade de manter o histórico de registro de campanhas é um dos pontos de grande importância para que não haja a perda de registros, que servirão como auxílio nas atualizações dos cadastros de dados e aprimoramento das técnicas para serem usadas em campanhas posteriores. O uso de sistemas de informação em uma campanha de vacinação contribui para a mitigação dos efeitos causados pela perda de registros e falta de controle adequado das aplicações dos pacientes.

Com o avanço da adoção das TIC pelas entidades públicas e privadas de saúde e a grande maioria dos estabelecimentos de saúde do Brasil utilizando computadores e internet em suas atividades [20]. Uma API (Interface de Programação de Aplicações) de gerenciamento de vacinação, possibilitará um serviço centralizado para qualquer tipo de plataforma de aplicação, sendo ele um software desktop, para o windows ou linux, a um aplicativo mobile (smartphones). Uma API Rest é um estilo de desenvolvimento web services baseado em um protocolo HTTP, um dos mais utilizados no mundo para a construção de serviços web [1].

A partir dessas informações, vários sistemas foram desenvolvidos visando oferecer à sociedade informações mais gerenciáveis sobre a vacinação no país. No entanto, não há, até o presente momento, uma API RESTful que consiga detalhar informações sobre vacinas disponíveis e indisponíveis de acordo com a necessidade, controlar o vencimento de um determinado lote de vacina, antes que possam ser usadas. Esse controle de lotes é feito desde a chegada do mesmo até a distribuição de forma coordenada entre as unidades cadastradas, evitando seu desperdício e compras desnecessárias, além de mensurar os resultados alcançados. Para o controle de lotes de vacinas, o estoque deve ter uma quantidade máxima e mínima de doses para não resultar em perdas futuras, por falta

de uso, e conservação correta, antes do uso das vacinas em determinadas campanhas [3].

O presente trabalho apresenta uma API RESTful com Spring Boot 2 que auxilie no gerenciamento do fluxo de vacinação. A API proporciona um serviço para aplicações multiplataforma, auxiliando no controle e gerenciamento de estoque, controle de validades dos lotes e os fornecedores desses lotes. Além de gerar relatórios que poderão mensurar as suas respectivas vacinas aplicadas a partir de um calendário de vacinação. Juntamente, possibilitando informar os pacientes sobre as próximas doses das campanhas pertinentes, sendo enviadas notificações através de SMS e e-mail, sobre as próximas doses do mesmo e informações sobre o cartão de vacina virtual. Desta forma, auxiliará as entidades públicas e/ou privadas, evitando a descentralização das informações obtidas em campanhas de vacinação. Com isso, sendo necessário o detalhamento de registros de forma confiável, e isso é como a garantia da qualidade, padrões de modelo, precisão das transações de estoque evitando falhas segundo diretrizes postuladas pelo **EVSM (Gerenciamento Eficaz de Armazenamento de Vacinas)**. Além de substituir os cartões de vacina físicos em papel que são os únicos registros dos históricos de vacinas da população brasileira. Estes por sua vez são perdidos, desgastam-se, são esquecidos e formam uma base de dados difícil de controle e análise [3] [4].

A API proposta neste trabalho busca também detalhar informações sobre vacinas disponíveis e indisponíveis de acordo com a necessidade. Para um gerenciamento de estoque eficaz, é indispensável um monitoramento que assuma critérios de qualidade para um fornecimento constante, nas unidades de saúde da população-alvo, além de controlar o vencimento de um determinado lote de vacina, antes que possam ser usadas. Esse controle é feito desde a chegada dos lotes até a distribuição de forma coordenada entre as unidades cadastradas, evitando seu desperdício e compras desnecessárias, além de mensurar os resultados alcançados. [3] [19].

Um sistema de controle de estoque precisa ter princípios consolidados, e para que isso ocorra é necessário o detalhamento de registros de forma confiável, e isso é como a garantia da qualidade, padrões de modelo, precisão das transações de estoque evitando falhas segundo diretrizes descritas pelo EVSM (Gerenciamento Eficaz de Armazenamento de Vacinas). Por mais que haja um manual para um gerenciamento de estoque eficiente, existem muitos países que usufruem de sistemas manuais que são sujeitos facilmente a falhas, trazendo uma má gestão para a distribuição de vacinas [3] [19].

II. REFERENCIAL BIBLIOGRÁFICO

Para entender melhor o cenário no qual o trabalho está situado, é de extrema importância conhecer algumas definições em que o mesmo se fundamenta. Logo, esta sessão tem como propósito enumerar conceitos que serviram como base para o mesmo.

A. Vacinação

A vacinação se define por ser a exposição determinada, através de injeção, ingestão ou aspiração de um produto atóxico que estimula o indivíduo a fabricar anticorpos. Caso o indivíduo seja exposto mais uma vez ao patógeno (que é nocivo para a saúde e causa doenças) contra o qual foi vacinado, a reexposição sucede em uma resposta periférica que compreende na proliferação de outras células e a formação de anticorpos, sendo assim, protegendo o indivíduo contra o progresso da doença [21] [30] [31].

B. Campanhas de Vacinação

As primeiras campanhas de vacina oral contra a Poliomielite iniciaram em 1961. Logo após, em 1962, foi o início da campanha contra a varíola. E em 1964 foi introduzida a vacina contra o sarampo [29]. Em seguida, em 1967, A Organização Mundial da Saúde (OMS) lança a Campanha de Erradicação Mundial da Varíola, sendo assim em 1971 surgiram os últimos casos de varíola. No mesmo período da Campanha de Erradicação Mundial da Varíola foi estabelecido o Plano Nacional de Controle da Poliomielite. [36] [8]

As campanhas de vacinação no Brasil obteve maior destaque servindo como exemplo para outros países pela grande cobertura vacinal, exemplos dessas campanhas ao longo da história são a da varíola e poliomielite. Apesar de ter êxito, uma das consideráveis dificuldades foi a capacidade de manter as evidências que possam ser usufruídas posteriormente em outras campanhas. Um exemplo da precariedade da sistematização foi na Revolta da Vacina, que serviria como base para o início da campanha da poliomielite que seria erradicada em 1995. Ao longo dos anos, os métodos de arquivamento e a documentação do Ministério da Saúde evoluíram, porém muitos conhecimentos não foram aproveitados no período das grandes campanhas de vacinação do século XX [2].

C. Programa Nacional de Imunização (PNI)

Após o grande sucesso das campanhas de vacinação da varíola, a Campanha de Erradicação da Varíola na década de sessenta, o Ministério da Saúde criou o Programa Nacional de Imunização (PNI) em 1973. O PNI é encarregado pelo combate pautado a inúmeras doenças imunopreveníveis, como a hepatite A, difteria, coqueluche, febre amarela, poliomielite, entre outras. Além também, de ser tornar uma referência mundial no cuidado dessas doenças, sendo um modelo sugerido pela Organização Mundial da Saúde. [37]

O PNI entrega mais de 300 milhões de doses anualmente, sendo de 44 distintos imunobiológicos, englobando vacinas, soros e imunoglobulinas. Além de contar com 42 Centros de Referência em Imunobiológicos Especiais (CRIE) e aproximadamente 34 mil salas de vacinação, que acomodam indivíduos portadores de situações clínicas especiais. Vale salientar, que o PNI também utiliza diversos métodos de vacinação, abrangendo a vacinação de rotina, campanhas, bloqueios de vacinas. [19]

D. Sistema Nacional de Vigilância Epidemiológica de Eventos Adversos Pós-Vacinação (SNVEAPV)

A vigilância epidemiológica tem a finalidade de fornecer as orientações técnicas necessárias para que os profissionais de saúde, que têm a incumbência de determinar sobre a definição de ações de controle de doenças, tornando disponíveis informações sobre as ocorrências das doenças, além dos fatores que condicionam, em uma determinada área geográfica ou população delimitada. Além de ser um importante instrumento para o planejamento, organização e operacionalização dos serviços de saúde [9].

Todos os casos de eventos adversos pertinentes à vacinação devem ser notificados imediatamente, esclarecida, investigada para que não venha trazer riscos ao programa de imunizações e a segurança epidemiológica da população. Por este motivo, os sistemas de vigilância de eventos pós-vacinação encarregam-se da busca e investigação de todas as reações notificadas à autoridade sanitária local. Visando uma melhor confiança nos programas de imunizações, o monitoramento dos eventos adversos é de relevante importância para o cumprimento dos objetivos para o benefício da população vacinada [54].

E. Sistema de Informações do Programa Nacional de Imunizações (SI-PNI)

O SI-PNI permite o acompanhamento das pessoas vacinadas em várias regiões do Brasil, assim como a localização da pessoa vacinada, através dos dados cadastrais. Desde 2010 o sistema já está funcionando no Brasil, porém, a ampliação da utilização não foi impulsionada por falta de equipamentos em todas as salas de vacina do país. A partir do ano de 2012, o PNI por meio da Portaria 2.363/2012 repassou recursos para a adesão dos equipamentos nos municípios que aderiram à portaria. A partir da adesão, o Ministério da Saúde disponibiliza videoaulas no youtube para acessar o material didático no auxílio e treinamento [37] [41] [56] [57] [58].

O SI-PNI compõe 7 módulos com o objetivo de possibilitar a avaliação dos riscos causados por surtos ou epidemias, com base nos registros dos imunobiológicos aplicados e a quantidade da população vacinada, separados por faixa etária, em um determinado período, além do controle de estoque, distribuição e utilização. O sistema contém o controle e o registro dos EAVP, controle dos imunobiológicos especiais nos CRIEs, e emissão de relatórios para o mapeamento e análise das supervisões [29] [41].

F. Manual de Normas e Procedimentos para Vacinação

O calendário Nacional de Vacinação tem por finalidade o controle, eliminação e erradicação de doenças imunopreveníveis. O calendário é definido pelo PNI com as orientações necessárias para crianças, adolescentes, adultos, gestantes, idosos e indígenas. Todas as vacinas que são fornecidas no serviço de saúde estão nos calendários de vacinação, no qual estão definidos os seguintes requisitos para o processo de criação [19] [55]:

- tipo da vacina [19] [55];

- número de doses do esquema básico e dos reforços [19] [55];
- idade para a administração de cada dose [19] [55];
- o intervalo entre as doses, caso tenha necessidade de mais de uma dose [19] [55];

O PNI estabelece os calendários de vacinação com diretrizes específicas para crianças, adolescentes, adultos, gestantes, idosos e indígenas. Os calendários estão devidamente regulados, na esfera do PNI, em todo o território brasileiro, sendo atualizados de acordo com o CGPNI (Coordenação Geral do Programa Nacional de Imunizações) [19] [55].

O registro individual das doses de vacinas administradas não pode ser feito sem um documento pessoal (cartão ou caderneta) e devidamente impresso pela unidade de saúde. Os serviços que já usam o SI-PNI ou qualquer outro sistema informatizado, esse registro deve ser feito diretamente no computador. Caso o computador da unidade não esteja disponível, é necessário o preenchimento de uma “Ficha de Registro do Vacinado” e depois encaminhado para uma secretaria municipal, ou distrito, ou qualquer instância definida pelo município. Além do registro individual, toda vacina administrada deve ser devidamente registrada no boletim diário e mensalmente definida no boletim mensal das doses aplicadas. Além das informações da movimentação de imunobiológicos e seus respectivos quantitativos mencionados anteriormente no SI-PNI [19].

1) *Avaliação dos indicadores de Imunização:* As informações coletadas pelo SI-PNI são processadas para que seja possível a avaliação do desempenho do PNI. Ou seja, para que seja possível avaliar uma situação vacinal de uma determinada área, em um tempo estipulado e por determinado imunobiológicos é utilizados os seguintes indicadores [19]:

- Coberturas vacinais: estima a proporção da população-alvo que foi vacinada que esteja hipoteticamente protegida por determinada doença. É calculado da seguinte forma [19]:

$$Cob./Vac = \frac{N}{Populacao} \times 100$$

Cob./Vac = Coberturas vacinais [19]

N = Número de doses aplicadas de determinada vacina [19]

Populacao = População-alvo da vacinação [19]

- Homogeneidade de coberturas vacinais: é um indicador generalizado e muito utilizado no contexto dos estados e do País. Tem a relevância em municípios com coberturas adequadas. Podendo ser resumido da seguinte forma [19]:

$$Homogeneidade = \frac{N}{M} \times 100$$

N = Número de municípios com coberturas adequadas para determinada vacina [19]

M = Número total de municípios [19]

- Taxas de Abandono (proporção de abandono de vacinas): tem por finalidade mensurar a aceitação do usuário/paciente ao programa de imunizações. Este indicador é aplicado para vacinas com esquema de multidoses. A referência do resultado é vista considerando as taxas inferiores a 5% como baixas, as maiores iguais a 5% e menores que 10% e as altas estão maiores e iguais a 10% [19].

$$Taxa = \frac{N - n}{N} \times 100$$

Taxa = Taxas de abandono [19]

N = Número de 1º doses [19]

n = Número de ultimas doses [19]

III. FUNDAMENTAÇÃO TÉCNICA

Para entender melhor o cenário no qual o trabalho está situado, é de extrema importância conhecer algumas definições em que o mesmo se fundamenta. Logo, esta sessão tem como propósito enumerar tecnologias que serviram como base para o mesmo.

A. JSON

O JSON é um formato de texto para serialização de dados, derivado do objeto literal de Javascript, definido na Programação ECMAScript Language Standard(ECMA-262). O JSON pode representar 4 tipos primitivos que são: strings, number, booleanos e nulos. Além de dois tipos estruturados: objetos e matrizes [67].

Uma estrutura de objeto é representada com um par de chaves em torno zero ou mais pares de nome/valor. Uma única vírgula separa um valor de um seguinte nome. Os nomes dentro do objeto devem ser únicos. Quando um objeto cujo nomes são todos únicos no sentido que todas as implementações de software que recebem esse objeto concordam com os mapeamentos nome/valor [67].

Um exemplo de objeto JSON:

```

1 {
2   "Pessoa": {
3     "nome": "exemplo",
4     "idade": 23,
5     "altura": 1.78,
6     "email": "exemplo@exemplo.com.br"
7   }
8 }
9 
```

Figura 1. Exemplo de objeto JSON [67]

Uma estrutura de matrizes é representada como colchetes ao redor de zero ou mais valores ou elementos, onde os elementos são separados por vírgulas [67]. Abaixo está uma matriz contendo 2 objetos:

```

1 [
2   {
3     "nome": "exemplo",
4     "idade": 23,
5     "altura": 1.78,
6     "email": "exemplo@exemplo.com.br"
7   },
8   {
9     "nome": "exemplo2",
10    "idade": 25,
11    "altura": 1.88,
12    "email": "exemplo2@exemplo.com.br"
13  }
14 ]
15 
```

Figura 2. Exemplo de matriz de objetos JSON [67]

B. REST (Representational State Transfer)

Cada aplicativo e site é um serviço web, sendo a parte mais notável da Internet. Um dos maiores sucessos para o protocolo se deve ao sucesso da WEB, URI(Uniform Resource Identifier) e HTML, e posteriormente o XML. Existe uma infinidade de protocolos e padrões e a maioria desenvolvidos junto com o HTTP e XML, algumas vinculadas ao HTML, JSON(JavaScript Object Notation), documentos binários e textos [1] [63] [64].

Quando pensamos em criar um sistema que seja flexível para outros tipos de ambientes, consideramos a criação de um Web Services baseado em SOAP(Simple Object Access Protocol) por ser um mecanismo que utiliza XML para o tráfego de informações. Só que ainda existe uma certa complexidade e verbosidade para até mesmo uma simples consulta [1] [63].

Devido a grande complexidade do modelo SOAP, surgiu a necessidade de criar um sistema rápido, fácil e eficiente para reutilizar em ambientes que englobam um sistema web, mobile ou desktop. Com isso, surgiu o REST, que é um estilo de desenvolvimento de Web services com a tese de doutorado de Roy Fielding que também é o co-autor de um dos protocolos mais utilizados no mundo, o HTTP(HyperText Transfer Protocol).

O Rest usufrui de um termo chamado **resources** (recursos) que são representados por URI ou URL, que em termo de Web são praticamente a mesma coisa. A diferença é que o URI é utilizado para identificar qualquer **path** (caminho) de um conteúdo, e a URL é utilizada apenas para fornecer o caminho, sendo que a URL é apenas um localizador e forma de URI. Um **resource** é algo que pode ser armazenado em um computador e é representado como um fluxo de bits: um documento, uma linha em um banco de dados ou resultado da execução de um algoritmo [63] [64] [72].

O REST não é protocolo, mas sim um formato de arquivo ou estrutura de desenvolvimento. Ou seja, é um estilo de arquitetura para sistemas hipermídia distribuídos, que descreve os princípios de engenharia de software que orientam o REST e as restrições de interação escolhidas para reter esses princípios, ao mesmo tempo em que os compara às restrições de outros estilos de arquitetura. É um estilo híbrido derivado de vários estilos arquitetônicos baseados em rede [72].

Para o estilo REST um recurso é qualquer coisa importante o suficiente para ser referenciado como uma coisa em si, ou seja, é algo que pode ser armazenado em um computador como um documento eletrônico, um registro de banco de dados ou o resultado da execução de um algoritmo. O modelo Rest chama

esses recursos de Recursos de informação, porque sua forma nativa é um fluxo de bits. Uma restrição é que cada recurso tenha uma URL própria [64] [72]. Um exemplo claro é quando realiza-se uma requisição GET para um recurso, a resposta deve servir um documento que captura o recursos de uma forma útil para que possa ter uma representação atual do estado do recurso. O corpo da resposta pode descrever uma linha de banco de dados como um documento XML, um objeto JSON [64].

A primeira perspectiva abordada para o processo de criação do projeto REST é que começa com nada, ou seja, uma tela em branco, quadro branco ou prancheta de desenho. A partir dessa perspectiva constrói-se uma arquitetura de componentes familiares até que satisfaça as necessidades do sistema pretendido. A segunda perspectiva é que um design começa com as necessidades do sistema como um todo, sem restrições, e em seguida, identifica e aplica as restrições aos elementos do sistema. Onde a primeira perspectiva tem o foco a criatividade e a visão ilimitada, o segundo a contenção e a compressão do contexto do sistema. O **Null Style** é simplesmente um conjunto vazio de restrições, onde não há limites distintos entre os componentes, configurando-se como o ponto de partida para descrição do REST [72].

Ao separar as questões da interface do usuário das questões de armazenamento de dados, melhora-se a portabilidade da interface do usuário em várias plataformas, além de ter uma melhor escalabilidade simplificando os componentes do servidor. O modelo mais significativo para a Web, é que a separação permite que os componentes evoluam independentemente [72].

O termo serviço usado na arquitetura **client-server** refere-se à abstração de um recurso. O recurso pode ser de qualquer tipo e baseado no fornecido pelo servidor (serviço). Por exemplo, se o servidor fornece páginas da Web, ele é chamado de servidor da Web e, se o servidor fornece arquivos, é chamado de servidor de arquivos e assim por diante. Um servidor pode receber solicitações de qualquer número de clientes em um momento específico [75], um exemplo pode ser visto logo abaixo:

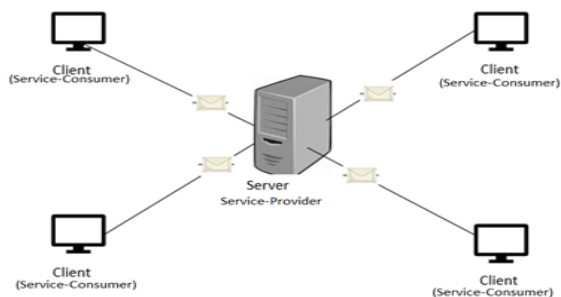


Figura 3. Client-Stateless-Server [72]

Uma característica que define o REST é a comunicação deve ser **Stateless** (sem estado) por natureza, de modo que cada solicitação do cliente para o servidor deve conter todas as informações necessárias para entender a solicitação e não pode

tirar proveito de nenhum contexto armazenado no servidor. O estado é mantido inteiramente no cliente [63] [72].

Essa restrição induz as propriedades de visibilidade, confiabilidade e escalabilidade. A visibilidade é aprimorada porque um sistema de monitoramento não precisa olhar para além de um único dado de solicitação para determinar a natureza completa da solicitação. A confiabilidade, pois facilita a tarefa de recuperação de falhas parciais. A escalabilidade é porque não ter que armazenar o estado entre as solicitações permitindo que o componente do servidor libere recursos rapidamente e simplifica ainda mais, pois o servidor não precisa gerenciar o uso dos recursos nas solicitações [63] [72].

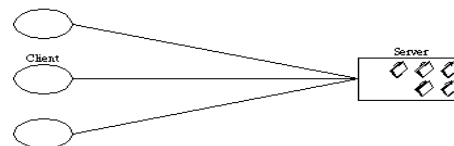


Figura 4. Client-Stateless-Server [72]

Para a eficiência da rede, o uso de **cache**, exige que os dados em uma resposta a uma solicitação sejam rotulados implícita ou explicitamente como armazenáveis ou não armazenáveis em cache. Caso uma resposta possa ser armazenada em cache, o cache do cliente terá o direito de reutilizar os dados de resposta para solicitações posteriores [72].

O cache é uma das restrições mais importantes da arquitetura da web. As restrições de cache instruem um servidor web a declarar a capacidade de cache dos dados de cada resposta. O armazenamento de dados de resposta em cache pode ajudar a reduzir a latência percebida pelo cliente, aumentar a disponibilidade e a confiabilidade gerais de um aplicativo e controlar a carga de um servidor web. Em uma palavra, o cache reduz o custo total da Web [75]

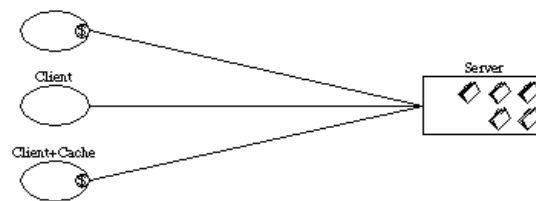


Figura 5. Client-Cache-Stateless-Server [72]

A vantagem de utilizar o **cache** é que é possível eliminar parcialmente ou completamente algumas interações, melhorando a eficiência, a escalabilidade e o desempenho percebido pelo usuário ao reduzir a latência média de uma série de interações. A desvantagem é que um cache pode diminuir a confiabilidade se os dados obsoletos dentro do cache diferem significativamente dos dados que já teriam sido obtidos se a solicitação tivesse sido enviada diretamente para o servidor [72].

Para a captura do estado atual do recurso, os componentes REST executam ações utilizando uma **representação** que é transferida entre os componentes participantes. Uma

representação é uma sequência de bytes, que incluem um documento, arquivo e entidade. Uma representação consiste em dados, metadados que descrevem os dados, que estão no formato de pares nome/valor, onde o nome corresponde a um padrão de definição da estrutura e semântica do valor [63] [72].

O formato de dados de uma representação é conhecido como tipo mídia. Uma representação pode ser incluída em uma mensagem que é processada pelo destinatário de acordo com os dados de controle da mensagem e a natureza do tipo de mídia. Alguns tipos de mídia destinam-se ao processamento automatizado, alguns destinam-se a ser renderizados para visualização por um usuário e alguns são capazes de ambos. Os tipos de mídia compostos podem ser usados para incluir várias representações em uma única mensagem [63] [72].

C. RESTful, Arquitetura Orientada a Recursos (ROA)

Arquitetura Orientada a Recursos também é RESTful, porém REST não é uma arquitetura: é um conjunto de critérios de design. Não existe uma arquitetura REST, ou seja, é muito geral e não está vinculado só a WEB, nada sobre REST depende da mecânica do HTTP ou das estruturas URIs. As melhores práticas do RESTful são semelhantes às ROA [63].

Em arquiteturas RESTful, as informações do método vão para o método HTTP. Em outras arquiteturas, as informações de escopo vão para a URI. Se o método HTTP não corresponder às informações do método, o serviço não é RESTful, e se as informações de escopo não estão no URI, o serviço não é orientado a recursos. Ou seja, no RESTful insere as informações do método no método HTTP e as informações do escopo no URI [63].

Logo abaixo podemos ver algumas características do ROA

- 1) **Endereçabilidade:** Capacidade de uma aplicação expor recursos por meio de URIs. [63].
- 2) **Statelessness:** capacidade de isolar cada requisição HTTP sem manter estado no servidor [63] [72].
- 3) **Representações:** nos serviços RESTful, representações são hipermídias: documentos que contêm não apenas dados, mas links para outros recursos [63] [72].

É importante uma uniformidade em cada serviço RESTful usando a interface HTTP da mesma maneira em toda a WEB. Sem uma interface uniforme, haveria uma multiplicidade de métodos sem um determinado padrão, ou seja, cada serviço estaria de uma forma diferente tornando difícil a compreensão e comunicação [63] [75].

Vejam alguns conceitos e propriedades do ROA utilizado na API:

- **Resource providers:** os provedores de recursos expõem os recursos para que os consumidores de serviço invoquem os serviços com métodos HTTP [75];
- **Resource:** Um recurso é uma referência explícita a uma entidade que pode ser identificada e atribuída e, mais importante, referenciada como um recurso [75];
- **Resource representation:** A representação do recurso é a informação útil sobre o estado atual de um recurso, espe-

cificado com um formato específico, em uma linguagem [75];

- **The uniform interface:** Todo serviço precisa usar a interface HTTP da mesma forma como GET, POST, PUT, DELETE e assim por diante. A interface uniforme significa simplesmente usar alguma nomenclatura comum que é interpretada da mesma maneira na web [75].

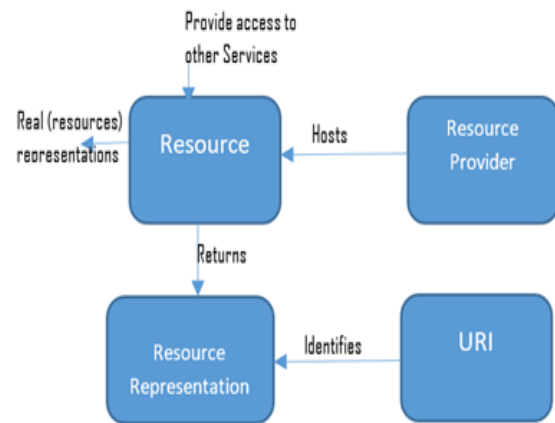


Figura 6. Design Diagrama baseado em ROA [75]

O diagrama apresentado na Figura 6 é a estrutura convencional do ROA e fornece a ideia de como os recursos são consumidos pelos consumidores do serviço.

D. Segurança API

O modelo tradicional de autenticação cliente-servidor, o cliente solicita um recurso de acesso restrito (recurso protegido) no servidor através das credenciais do proprietário do recurso. Com a finalidade de fornecer acesso a aplicativos de terceiros para recursos restritos, o proprietário do recurso compartilha suas credenciais com o terceiro. Criando vários problemas e limitações [65]:

- Os aplicativos de terceiros são necessários para armazenar o recurso das credenciais do proprietário para uso futuro, normalmente uma senha em texto [65].
- Os servidores são obrigados a suportar autenticação de senha, apesar das fragilidades de segurança inerentes às senhas [65].
- Os aplicativos de terceiros obtêm acesso ao recurso protegidos do proprietário, permitindo que os recursos sem qualquer restrição [65].
- Os proprietários dos recursos não podem invalidar o acesso a um terceiro de forma individual sem revogar o acesso a todos os outros terceiros [65].

Muitas APIs REST expõem recursos associados a um cliente e/ou usuário específico. Por exemplo, os documentos de uma API REST podem conter informações privadas e seus controladores podem expor operações destinadas a serem executadas por um público restrito. As regras nesta seção tratam da proteção dos recursos confidenciais de uma API REST [74].

A estrutura de autorização do OAuth 2.0 permite que um terceiro aplicativo possa obter acesso a um serviço HTTP, seja em nome de um proprietário de recurso orquestrando uma interação de aprovação entre o proprietário do recurso e o serviço HTTP, ou permitindo que o aplicativo de terceiros para obter o acesso em seu próprio nome. No OAuth, o cliente solicita o acesso aos recursos controlados pelo proprietário do recurso e hospedado pelo servidor de recurso. Invés do proprietário usar as credenciais para acessar o recurso, o cliente obtém o token de acesso, que é uma string denotando um escopo específico, tempo de vida e outros atributos de acesso. Tokens de acesso são emitidos para o cliente de terceiros por um servidor de autorização com a aprovação do proprietário do recurso. O Cliente usa o token de acesso para acessar os recursos protegidos hospedados pelo servidor de recursos [65] [74].

O Auth2 define 4 funções para a iteração:

- **Resource owner** (proprietário do recurso): uma entidade capaz de conceder acesso a um recurso protegido. Quando o proprietário do recurso é uma pessoa, é referido como um usuário final [65] [74].
- **Resource owner** (servidor de recursos): o servidor que hospeda os recursos protegidos, capaz de aceitar e responder a solicitações de recursos protegidos usando tokens de acesso [65] [74].
- **Client** (Cliente/WEB/Mobile/Desktop): um aplicativo que faz solicitações de recursos protegidos em nome do proprietário do recurso e com a sua autorização [65] [74].
- **Authorization Server** (Servidor de Autorização): o servidor emitindo tokens de acesso ao Client após ter sucesso na autenticação [65] [74].

Fluxo de interação entre as quatro funções e inclui as seguintes etapas:



Figura 7. Fluxo OAuth2

(A) O Client solicita autorização do **Resource owner**. A solicitação de autorização pode ser feita diretamente ao **Resource owner**, ou de preferência indiretamente por meio de **Authorization server** como intermediário [65] [74].

(B) o **Client** recebe uma concessão de autorização, que é uma credencial que representa a autorização do **Resource owner** [65] [74]. (C) o **Client** solicita um token de acesso autenticando com o **Authorization Server** e apresentação da concessão de autorização [65] [74]. (D) o **Authorization**

Server autentica o cliente é válida a concessão de autorização e, se válida, emite um token de acesso [65] [74]. (E) o **Client** solicita o recurso protegido do **Resource server** e autentica apresentando o token de acesso [65] [74]. (F) o **Resource server** valida o token de acesso e, se válido, atende ao pedido [65] [74].

Token de acesso (**Access Token**) são credenciais usadas para acessar recursos protegidos. O Token de acesso é uma **string** que representa uma autorização emitida para o Client. A **string** geralmente é opaca para o **Client**. Tokens representam escopos específicos e duração de acesso, concedidos pelo **Resource owner**, e reforçada pelo resource server e **Authorization Server** [65];

E. Message Broker

Message Broker é uma tecnologia de comunicação entre serviços, aplicativos, sistemas que auxilia na conversão de mensagens entre sistemas independentes baseado em um protocolo de mensagens. O protocolo em questão deste projeto é o **AMQP (Advanced Messaging Queuing Protocol)**, que determina os dados que devem ser transmitidos na rede, que por sua vez permite troca de mensagens assíncronas. Os message brokers conseguem, de maneira confiável, validar, armazenar, rotear e conceder mensagens aos destinos adequados. Estas mensagens funcionam como uma estrutura em filas, que ordenam e armazenam as mensagens até que um aplicativo ou sistema venha consumi-las [79] [80].

O servidor de mensageria utilizado é o **RabbitMQ**, por sua simplicidade e ser open source. No servidor em questão existem 3 principais componentes e 3 entidades envolvidas, que interagem no modelo **AMQP**, que são [79] [80]:

- **Exchange**: componente recebe as mensagens e faz o roteamento para as filas (message queues) [79] [80].
- **Message Queue**: componente que mantém as mensagens até que o consumidor venha a consumi-las [79] [80].
- **Binding**: componente que faz o relacionamento das exchanges e as queues, podendo definir regras de roteamento [79] [80].
- **Publisher (Producer)**: entidade responsável por enviar as mensagens para as filas [79] [80].
- **Consumer**: entidade encarregada de consumir as mensagens nas filas [79] [80].
- **Broker**: entidade servidora de filas. No caso do servidor que implementa a especificação AMQP [79] [80].

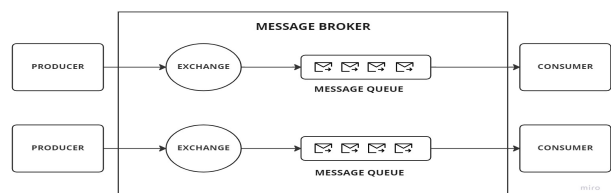


Figura 8. Fluxo RabbitMQ [80]

IV. TRABALHOS RELACIONADOS

A maioria das soluções pertinentes ao gerenciamento de vacinas, relatam a importância da existência de uma carteira digital que possibilite o controle das doses tomadas, a necessidade do usuário/paciente conseguir receber notificações sobre as campanhas de vacinação. A importância da substituição das cadernetas de vacinação de papel por cadernetas digitais, é um dos pontos mais mencionados como solução para diminuir o risco de perdas deste documento facilitando também as aplicações das doses.

A. Sistema Gerenciador de Vacinação

O trabalho de [5] tem a finalidade do desenvolvimento de um sistema WEB de gerenciamento de vacinação baseado na linguagem Java. Um dos pontos mais importantes desse software é de informar de antemão aos pacientes cadastrados das próximas doses de vacinas a serem administradas, e também, disponibilizar ao operador do sistema um histórico consistente dos dados cadastrados para não haver repetição de doses aplicadas. O sistema possibilita através de um portal o efeito de gerenciar as vacinas para o controle de estoque das unidades de saúde tanto públicas quanto privadas. Tanto o funcionário quanto o paciente terá um acesso ao sistema por meio da WEB, separado por níveis de acesso, sendo que cada usuário terá permissão para determinada funcionalidade. No nível de acesso do funcionário da unidade de saúde, o mesmo poderá acessar o histórico do paciente informando as vacinas pendentes, as que foram administradas e as que irão ser aplicadas. Para o paciente terá informação dos calendários e das doses que precisam ser tomadas, substituindo a carteira de vacinação no formato papel para o digital, além de conter notificações por meios de SMS (Short Message Service) e e-mail referentes às campanhas de vacinação [5].

Segundo [5] a escolha da linguagem foi justificada por ter uma grande proporção nas empresas e bem consolidada no mercado e a possibilidade de implementar o projeto com software gratuito (freeware). A estrutura do sistema foi toda fundamentada na tecnologia em JSP (JavaServer Page) que é baseada em Servlets (Java Servlets) que auxilia na criação de páginas dinâmicas em HTML (HyperText Markup Language) estilizadas com CSS (Cascading Style Sheets) JavaScript.

B. SICOVAC

[10] O SICOVAC tem a finalidade de gerenciar as vacinas nos postos de saúde do Distrito Federal. É um projeto social contendo uma arquitetura Cliente/Servidor o método de hospedagem é pelo protocolo de rede File Transfer Protocol (FTP), fazendo uso de banco de dados para o cadastro dos registros. Para o usuário ter acesso é necessário ter um computador com conexão à Internet através de um navegador web. O sistema em questão é baseado na linguagem JAVA WEB por ter vários recursos de segurança, fácil integração com recursos de rede, utilização do paradigma de Orientação a Objetos. O sistema obtém níveis de acesso para o controle da aplicação e segurança dos dados, o importante são os relatórios que

são emitidos para o mapeamento de vacinações para uma determinada população em um determinado período.

C. ImunizaBR

[11] O ImunizaBR é um aplicativo no formato mobile que tem por finalidade registrar as vacinas para que possam ser acessadas em um formato podendo ser smartphones ou tablets (aplicativos híbridos). O projeto tem o objetivo a preservação das cadernetas de vacinação que estão sujeitas a degradação ou até mesmo quando há o esquecimento do paciente ou a perda do mesmo. A necessidade de controlar as doses aplicadas é de extrema importância para evitar o desperdício das doses pela sua má administração, além de agendar a próxima dose, caso haja mais de uma dose, localizando os postos mais próximos de sua residência. O aplicativo é um gerenciador de estoque voltado ao usuário/paciente. A arquitetura do sistema é baseada em HTML utilizando o Framework Ionic que utiliza o Twitter bootstrap para camada de front-end e no AngularJs para implementação da camada de negócio.

D. Vacinação na Palma da Mão

O aplicativo Vacinação na Palma da Mão, foi criado com objetivo de ser um aplicativo para dispositivo móvel integrado ao SIPNI- Sistema de Informação do Programa Nacional de Imunização. O SIPNI que é utilizado para o cuidado na sala de vacinação, que continha um volume de dados considerável após a implantação nas unidades de saúde Unidades de Atenção Primária à Saúde - UAPS no Estado de Minas Gerais. O sistema em questão tem a funcionalidade também de um cartão virtual para lembretes e informações sobre as campanhas que poderão ocorrer de forma que os usuários/pacientes sigam em tempo real qualquer alteração [12].

Todo o aplicativo é composto por 2 módulos: o primeiro foi desenvolvido em ambiente desktop, fazendo uso de verificação das vacinas registradas no SIPNI e armazenamento no banco de dados online, baseado na linguagem Java que pode ser instalado no computador sistema operacional na sala de vacinação. O segundo módulo foi feito em ambiente WEB, garantindo a utilização do aplicativo móvel atualizando os registros no banco de dados online. [12]

O sistema VPM-Sinc é parte central do cartão de vacina digital e, inicialmente, foi desenvolvido para possibilitar a atualização do aplicativo móvel com registros vacinais do SIPNI. Composto de dois módulos, o VPM-Sinc Agent e o VPM-Sinc Core. O módulo VPM-Sinc Agent foi desenvolvido em plataforma desktop, linguagem de programação Java (por ser gratuita) e multiplataforma, o que permite sua instalação nos computadores da sala de vacinação do município do estudo, independentemente do sistema operacional utilizado. Nesse módulo, o objetivo é consultar os registros do cartão de vacina das pessoas no banco de dados do SIPNI, instalado no computador da sala de vacinação, e enviá-los automaticamente via internet para o banco de dados on-line do VPM-Sinc Core. [12]

O módulo VPM-Sinc Core foi desenvolvido em ambiente web, com tecnologia ASP.NET MVC, tendo o principal objetivo de disponibilizar os registros vacinais do SIPNI armazenados no banco de dados on-line pelo módulo VPM-Sinc Agent. O módulo VPM-Sinc Core tem como principal diferencial possibilitar a integração com o SIPNI em versão web por meio do Application Programming Interface (API) e, assim, disponibilizar os registros vacinais ao aplicativo móvel. Portanto, o VPM-Sinc é integrador de dados do SIPNI, em versão local ou web, o que viabiliza a exibição do cartão de vacina das pessoas no aplicativo móvel de forma automática. [12]

E. SISTEMA DE CONTROLE DE VACINAÇÃO

Para a criação do sistema, o autor observou que havia a necessidade de criar um software com o intuito de garantir um processo capaz de prevenir diversas doenças de maneira eficiente. Pelas pesquisas realizadas foram apuradas que no Brasil nunca houve uma atualização acerca dos registros de vacinações dadas aos pacientes, sempre foram documentadas em uma caderneta. Algumas redes já disponibilizam desde 2013, um aplicativo de vacinação, para android e iOS, liberando ao usuário o cadastro e atualização da sua própria carteira de vacinação. Informando ainda que, os dados em papel podem ser de fácil acesso e perdidos. A linguagem utilizada para o desenvolvimento do sistema foi em JAVA, possuindo muitos métodos utilitários, bastante usada, para unir com um banco de dados ou armazenamentos de dados, podendo ser adaptada ao HTML utilizando páginas JSP [13].

Em desenvolvimento web, existem alguns padrões de projeto, entre eles, o padrão MVC (Model-View-Controller). No padrão MVC, o sistema web é dividido em três partes principais: modelo, visão e controlador; e cada um desses elementos é desenvolvido para executar um papel específico na aplicação. O modelo corresponde à estrutura de objetos que diretamente manipulam os dados, lógicas e regras da aplicação. A visão é a representação visual dos dados trabalhados no modelo. E o controlador é a parte que faz o intermédio entre os demais elementos, modelo e visão, recebendo as requisições e manipulando os dados com o modelo para a visão de acordo com as regras de negócio. [13]

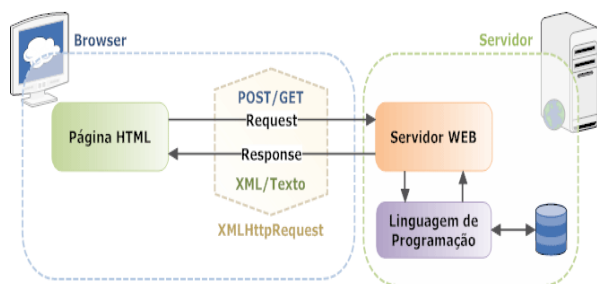


Figura 9. Fonte [13]

F. CONVAC

A finalidade do Trabalho é criar uma carteira de vacinação digital para o acesso aos registros de vacinação que é dito pelo autor como Cartões de Vacinação Inteligentes. O projeto proposto, denominado Sistema de Controle de Vacinação (CONVAC), abrange um sistema web, desenvolvido em JAVA EE, para o controle dos principais dados no processo de vacinação. Em conjunto com o sistema web, há uma API Java, capaz de disponibilizar a carteira de vacinação do paciente, via web, uma aplicação desktop, capaz de emitir e ler as informações de vacinação armazenadas em um cartão inteligente e a representação cartográfica de postos de saúde e moradores [14]. Portanto, o sistema é dividido nos seguintes módulos:

- Módulo Web: Cadastros e Consultas Gerais (Pacientes, Funcionários, Lotes de Vacinação, Níveis de Acesso, Vacinas e Postos de Saúde).
- Módulo Carteira de Vacinação Virtual: consiste em API Java, com as principais informações de vacinação do paciente, para acesso externo, via web.
- Módulo Cartão inteligente com a Carteira de Vacinação: aplicação desktop para emissão e leitura dos principais dados de vacinação armazenados no cartão.
- Módulo Representação Cartográfica de Postos e Moradores: possibilita a representação cartográfica da localização dos moradores e dos postos de saúde da cidade na qual o cidadão se encontra.

Módulo Web – Este módulo do sistema é destinado aos centros de saúde, no qual terão acesso ao cadastramento de novas informações, consultas das informações armazenadas e atualização de dados, em suma, através deste módulo o centro de saúde poderá manipular as informações registradas das vacinas e dos pacientes, como:

- Gestão de Vacinas: permite o cadastro e consulta das vacinas que chegam ao posto de saúde, são solicitadas as seguintes informações: nome da vacina e a importância da vacina, os campos com (*) são de preenchimento obrigatório.
- Gestão de Postos: permite o cadastro e consulta das informações referentes ao posto de saúde, como seu código, endereço, unidade e telefones.
- Gestão de Lotes: Cadastro de Lotes: permite o cadastro e consulta das informações referentes aos lotes de vacinação que chegam até o centro de saúde.
- Gestão de Usuários: permite o cadastro e consulta das informações do funcionário do posto de saúde. Neste cadastro, são definidos: o nível de acesso do usuário e a senha para acesso ao sistema, além da vinculação deste funcionário com um determinado posto de saúde.
- Gestão de Pacientes: permite o cadastro e consulta das informações dos pacientes, como nome, data de nascimento, e-mail, senha de acesso a carteira virtual de vacinação, bem como telefones, endereço, RG e CPF. [14]

Módulo Carteira de Vacinação Virtual – Esse módulo do sistema é destinado ao paciente, no qual ele terá acesso, via

web, a sua carteira de vacinação. A carteira de vacinação pode ser acessada por meio de qualquer navegador, preferencialmente pelo Google Chrome. Nesse módulo, o paciente fará a autenticação através de login e senha e encontrará seus dados pessoais (nome, Cadastro de Pessoa Física (CPF) e data de nascimento) e os dados vacinais cadastrados pelo centro de saúde. As informações que serão visualizadas pelo paciente, via web, são similares às que são encontradas, atualmente, na carteira de vacinação impressa. A carteira de vacinação virtual permite apenas a visualização para o paciente, pois como esclarecido anteriormente, de acordo com o Ministério da Saúde, a responsabilidade da atualização da carteira de vacinação é da unidade de saúde que realiza a aplicação da vacina. A Figura 3 ilustra a interface referente à sua carteira de vacinação virtual. [14]

Módulo Cartão inteligente com a Carteira de Vacinação – Esse módulo é destinado aos centros de saúde, no qual terão a possibilidade de realizar a leitura e gravação das informações registradas no cartão inteligente. O módulo é um complemento da carteira virtual de vacinação, portanto cada paciente receberá um smart card IDEalCitiz S3 1.6, com memória total de 80kb, único e intransferível, com as informações vacinais de maior relevância. É necessário destacar que o módulo em questão é uma aplicação desktop disponível apenas para os centros de saúde, pois esses possuem a responsabilidade da atualização de informações vacinais dos pacientes. A aplicação possui três telas: a de autenticação do usuário (login e senha), o cadastro do paciente e a de leitura e gravação de informações. A Figura 4 ilustra a interface gráfica para a gravação das informações no cartão inteligente. [14]

Módulo Representação Cartográfica de Postos e Moradores – Esse módulo do sistema é destinado aos órgãos gerenciadores para auxílio na tomada de decisão. O módulo objetiva disponibilizar, através de uma representação cartográfica (representação por mapas), os locais onde estão localizados os postos de saúde e os pacientes de cada posto. Para acessar o módulo, o funcionário deverá realizar a autenticação através de login e senha. Ao acessar o sistema, o usuário tem a possibilidade de mapear os moradores, os postos de saúde, separadamente e por último conjuntamente. [14]

G. Módulo de vacinação para integração ao SIGS

O sistema é um Módulo de vacinação para a Polícia Militar de Minas Gerais (PMMG), registrando todas as doses aplicadas em uma determinada campanha, contendo o histórico apropriado para seus pacientes, além de conter dados sobre a cobertura vacinal. Esse módulo em questão será integrado a um sistema já existente que é o SIGS (Sistema Integrado de Gestão à Saúde). O SIGS é software utilizado pelos profissionais do CBMMG (Corpo de Bombeiros Militar de Minas Gerais), possibilitando o registro de consulta, tratamentos, programas, prontuários eletrônicos [15].

H. Sistema Web para Controle de Vacinação de um Hospital

O Sistema Web de Controle de Vacinação de um Hospital Regional tem como utilidade um software que seja confiável e

seguro na aplicação de vacinas, pois é de extrema importância que em um sistema voltado para a área de saúde tenha o mínimo de erros possíveis, dando uma credibilidade maior hospital quanto a vida do paciente. Este trabalho descreve a importância da mudança dos cartões de vacinação em papel para os digitais, pois mitigam as perdas e os desgastes [16].

I. SIPNI

Programa Nacional de Imunizações (PNI), é responsável por reunir as informações de todos os programas de imunização que ocorre no Brasil, fazendo o controle das doses aplicadas e enviando ao Ministério da Saúde, recentemente utilizando o Sistema de Informação de Avaliação do Programa de Imunização (SI-PNI), em parceria com o Departamento de Informática do SUS (DATASUS) com a finalidade de substituir sistemas antes utilizados. Um dos principais objetivos é registrar os dados de cada indivíduo que reside no Brasil, facilitando a unificação dos dados para que possam reduzir erros no momento da imunização nas salas de vacinação, controle de estoque das vacinas, campanhas, sendo que ele é utilizado restritamente na sala de vacina. Consequentemente poderão emitir relatórios estatísticos que poderão auxiliar no controle da imunização. [17]

O sistema tem como tecnologia utilizada os software livres, ou seja, não proprietários, desenvolvido na linguagem de programação JAVA e banco de Dados PostgreSQL, e todo o software é em desktop, então existe a necessidade de instalar nas máquinas na sala de vacinação. Tendo em vista que a linguagem Java possibilita a instalação em sistemas operacionais diferentes. Os requisitos mínimos para a instalação do sistema são: Processador: Pentium III Hard disk; 20 GB (500 Mb Livres); Floppy disk: 1.44 MB; Unidade óptica; Leitor de CD-ROM ou Porta USB e Memória: 256 MB [17].

O desenvolvimento deste software é dividido em 7 módulos:

- API - Avaliação do Programa de Imunizações - Registra, por faixa etária, as doses de imunobiológicos aplicadas e calcula a cobertura vacinal, por unidade básica, município, regional da Secretaria Estadual de Saúde, estado e país. Fornece informações sobre rotina e campanhas, taxa de abandono e envio de boletins de imunização. Pode ser utilizado nos âmbitos federal, estadual, regional e municipal.
- EDI - Estoque e Distribuição de Imunobiológicos - Gerencia o estoque e a distribuição dos imunobiológicos. Contempla os âmbitos federal, estadual, regionais e municipais.
- EAPV - Eventos Adversos Pós-vacinação - Permite o acompanhamento de casos de reações adversas ocorridas pós-vacinação e a rápida identificação e localização de lotes de vacinas. Para as gestões federais, estaduais, regionais e municipais.
- PAIS - Programa de Avaliação do Instrumento de Supervisão - Sistema utilizado pelos supervisores e assessores técnicos do PNI para padronização do perfil de avaliação, capaz de agilizar a tabulação de resultados. Desenvolvido para a supervisão dos estados.

- PAISSV - Programa de Avaliação do Instrumento de Supervisão em Sala de Vacinação - Sistema utilizado pelos coordenadores estaduais de imunizações para padronização do perfil de avaliação, capaz de agilizar a tabulação de resultados. Desenvolvido para a supervisão das salas de vacina.
- AIU - Apuração dos Imunobiológicos Utilizados - Permite realizar o gerenciamento das doses utilizadas e das perdas físicas para calcular as perdas técnicas a partir das doses aplicadas. Desenvolvido para as gestões federais, estaduais, regionais e municipais.
- SICRIE - Sistema de Informações dos Centros de Referência em Imunobiológicos Especiais - Registra os atendimentos nos CRIEs e informa a utilização dos imunobiológicos especiais e eventos adversos.

J. SIVAC

O SIVAC é um software público on-line para monitorar e registrar a vacinação em todo território brasileiro, com a possibilidade de informar estatísticas precisas sobre as aplicações individualmente, gerenciando as campanhas por imunobiológico. Sendo dentre os softwares anteriores o mais completo pois disponibiliza informações sobre a distribuição das unidades e como eles distribuem suas doses entre as mesmas. O sistema controla também os estoques de vacinas que disponibilizaram a caderneta eletrônica com os padrões das Unidade Básica de Saúde, Unidade Saúde da Família, Policlínicas, dando agilidade no atendimento das campanhas de vacinação, tendo acesso às recomendações de áreas da com relatórios da cobertura vacinal. O sistema é compatível com os sistemas operacionais Linux/Ubuntu e Windows, onde é possível utilizá-lo com os navegadores IE, FIREFOX, CHROME, SAFARI e Opera. Foi implementado com Orientação a Objetos com as linguagens PHP(Hypertext Preprocessor), JAVASCRIPT, com CSS em conjunto com o HTML e o Ajax, e para o armazenamento de registros é utilizado o MYSQL. [18]

Abaixo pode ser visto um comparativo da disponibilidade dos trabalhos mencionados anteriormente e o trabalho em questão (API RESTful):

Projetos	Disponibilidade
Sistema Gerenciador de Vacinação	WEB
SICOVAC	WEB
ImunizaBR	MOBILE
Vacinação na Palma da Mão	MOBILE
Sistema de Controle de Vacinação	WEB
CONVAC	WEB/DESKTOP
SIGS(Sistema integrado de Gestão de Saúde)	WEB
SIVAC	WEB
SIPNI	DESKTOP
API RESTful para gerenciar vacinação	WEB/DESKTOP/MOBILE

V. PROPOSTA

A. Visão Geral

O presente projeto foi desenvolvido através de um levantamento bibliográfico baseado nos Manuais de Procedimentos de Vacinação e projetos já utilizados no SUS. Primeiro foi feito um levantamento dos principais dados que são utilizados no processo de controle e gerenciamento de vacinação, sempre respeitando os manuais emitidos pelo SUS. Após esta análise de dados foi feita uma comparação com os principais projetos realizados no território nacional para que seja elaborado um modelo de dados conciso. Porém, tendo em vista as necessidades particulares deste trabalho

A solução desenvolvida para esse trabalho é a criação de uma API RESTful para gerenciamento de vacinação que irá auxiliar as entidades públicas e privadas no controle dos registros relacionados à vacinação. Além também de poder fornecer recursos necessários para que o paciente tenha informações necessárias para o acompanhamento de suas doses de vacinas. A API é baseada no Spring Boot 2, uma tecnologia que utiliza a linguagem JAVA para desenvolvimento. O primeiro processo mostrará como foi feita a extração dos dados necessários para o mapeamento e modelagem do banco de dados da API, seguindo rigorosamente o Manual de Normas e Procedimentos para Vacinação recomendados pelo Ministério da Saúde e o SIPNI que mantém os registros e emissões de relatórios necessários para o mapeamento vacinal da população. Contando também com auxílio da API de Localização do IBGE para mapeamento das regiões brasileiras. Por fim, o segundo processo apresentará a construção base para o gerenciamento da API e todos os recursos necessários para a implementação.

A linguagem base de desenvolvimento da API é a JAVA, muito utilizado em sistemas WEB, sendo compatível com o framework Spring Boot 2. Este framework é um ecossistema, onde é subdividido em módulos, que será necessário o uso do módulo Spring Data como um gerenciador de persistência de dados, Spring Web que será o controlador web para a emissão e geração de dados que serão consumidos e fornecidos pela API. O módulo Spring WEB provê recursos em formato JSON para o fornecimento de serviços que podem ser disponibilizados para várias plataformas como: Android, Desktop e WEB. A comunicação com os clientes da API RESTful pode ser vista na Figura 10;

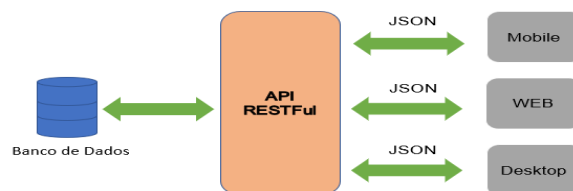


Figura 10. Fonte [13]

Como visto acima a comunicação é dividida da seguinte maneira:

- Banco de Dados: o SGBD em questão foi utilizado o MySQL.
- Mobile, WEB e Desktop: são os clientes da aplicação, ou seja, as plataformas que consumiram o serviços REST.
- JSON: são os tipos de dados principais que serão trafegados em rede dos clientes para a API e vice-versa.

A arquitetura da API RESTful desenvolvida com o ecossistema Spring Boot 2 tem o seguinte formato de acordo com a Figura 11:

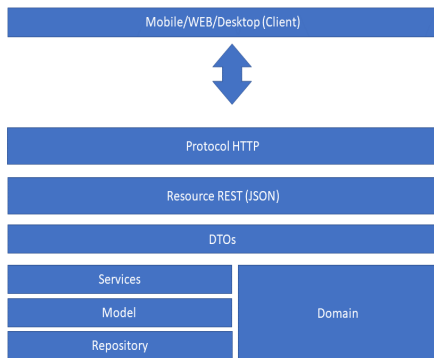


Figura 11. Arquitetura API

Toda a API é dividida em camadas. Neste esquema, a comunicação do cliente (Mobile/WEB/Desktop) é através do Protocolo HTTP, visto que é um protocolo de comunicação muito utilizado em sistemas distribuídos. A camada Resource é pautada também no protocolo HTTP e tem o seu contrato de implementação embasado no estilo arquitetural REST. Com isso, pode ser visto que tem uma referência ao JSON, sendo o formato principal de representação de recursos do REST. Apesar de ser o principal formato, com o Spring Boot 2, pode também fornecer recursos para que sejam exportados dados como XML e YAML. Esses formatos JSON, XML ou YAML são representados por modelos de recursos que são chamados de DTOs. A camada de DTOs fornece o modelo de recursos que estão entre os clientes e a API.

Os DTOs são enviados para a camada de Services, que por sua vez contém toda a regra de negócio da API, e onde também são convertidos os DTOs em Model para que possam fazer o acesso a dados na camada Repository. Essa camada Repository, por sua vez, tem a representação necessária para que possa ser feita as principais operações no banco de dados. Além disso, tem a possibilidade de ver no esquema o Domain, este domínio da API, ou seja, é o agrupamento de toda a estrutura de Services, Model e Repository.

A partir de análises dos trabalhos correlatos e do manual de vacinação, foi necessário subdividir a API em 6 módulos que são eles:

- **Segurança:** consistem em todo o controle de acesso a API baseado no módulo Spring Security do framework

Spring Boot 2. Nesta conterá as informações de permissões de acesso de usuário e funcionário.

- **Cadastro Geral:** compreende todos os dados básicos que auxiliam nos cadastros pessoais dos pacientes, funcionários, etnias, gêneros, CNAES (Classificação Nacional de Atividade Econômicas) e endereços. Sendo necessário a importação da API de Localização e a API Cnaes do IBGE.
- **Vacinação:** permite que seja gerenciado as campanhas, calendários de vacinação, vacinas, doses, aplicadas, notificações das próximas doses nos pacientes, informativos sobre o calendário de vacinação.
- **Estoque:** permite o controle de entradas e saídas de doses para unidades matriz e satélites alinhado com as aplicações de doses.
- **Relatórios:** compõem todo o mapeamento das regiões que foram vacinadas, histórico das campanhas, lotes de vacinas por respectivos fornecedores, por idade de aplicação de doses por vacinas.
- **Cartão Vacinal:** onde o paciente consegue visualizar as doses que foram tomadas contendo todo o histórico vacinal.
- **Core:** este módulo fornece funcionalidades utilitárias compartilhadas entre os outros módulos.

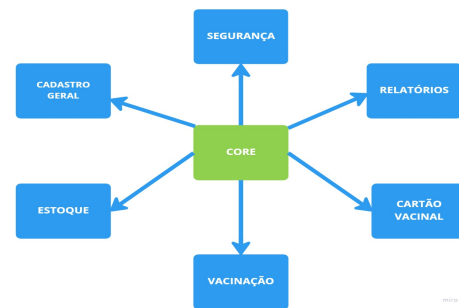


Figura 12. Módulos API

Além desses módulos baseados nos trabalhos correlatos (Seção IV) e o Manual de Vacinação (Seção II-G), foi também criado e aprimorado o modelo de dados da API, seguindo os Sistema de Informações do Programa Nacional de Imunizações (SI-PNI) descritos na Seção II-E, que constituem as informações mais relevantes para a criação da API deste trabalho.

Na Figura 13 pode ser visto uma parte do modelo de dados referentes às informações principais que serão cadastradas e consultadas dos pacientes:

O modelo apresentado na Figura 13 está diretamente conectado ao fluxo analisado com mais atenção no Manual de Vacinação da seção II-G. Com isso, pode ser feita a busca dos dados mais necessários para um cadastro de um paciente. Esses dados serviram posteriormente na emissão de relatórios dos pacientes. Para a criação desse modelo utilizada a linguagem SQL (Structured Query Language) que suporta múltiplos SGBs (Sistema Gerenciador de Banco de Dados). Visto que, para o gerenciamento e o geração de Scripts iniciais

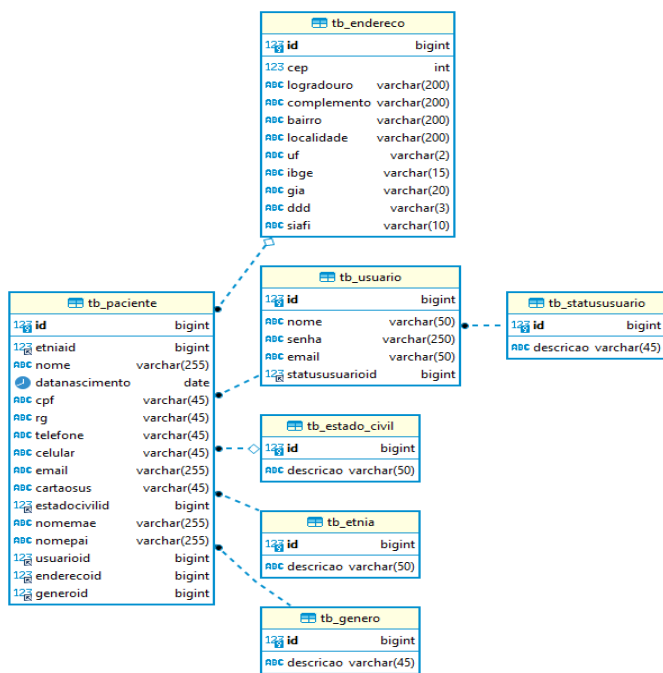


Figura 13. Modelo de dados Paciente

da API, foi utilizada uma ferramenta de migração de banco de dados de código aberto, o Flyway. Esta ferramenta pode ser integrada ao Spring Boot 2 e é compatível com scripts desenvolvidos em SQL.

Uma visão geral dos módulos pode ser representada pelo diagrama de pacotes, onde cada pacote tem uma estrutura básica definida. Não excluindo a possibilidade de ter outros pacotes, como na Figura 14:

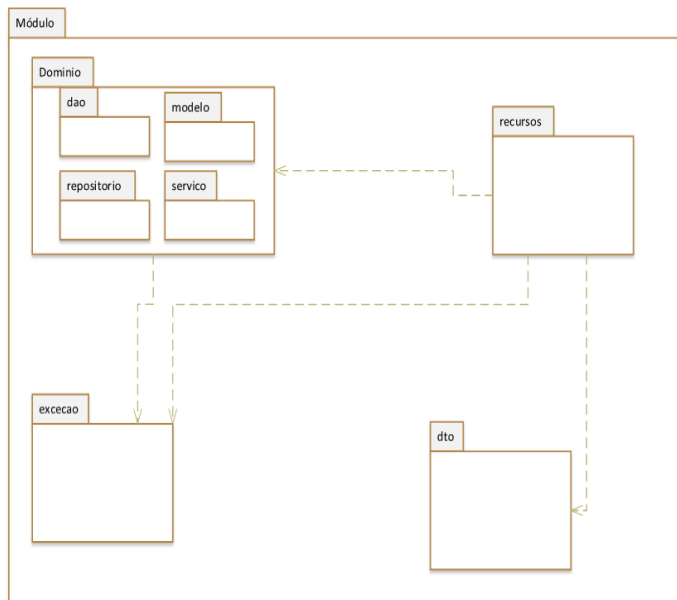


Figura 14. Diagrama de Pacotes

Cada módulo tem pacotes básicos, como visto na Figura 16, que por sua vez são: **recursos**, **dto**, **execcao** e **dominio**. O pacote de **dominio** tem 4 subpacotes que são: **dao**, **modelo**, **repositorio**, **servico**. O pacote de **recursos** recebe a chamada de um cliente (Mobile, Desktop ou WEB). Cada recurso tem uma permissão pré-estabelecida como será mencionada na subseção do Módulo de Segurança. Ou seja, para que a chamada desse recurso seja executada com sucesso é necessário ter o **perfil** indicado de acesso.

O pacote de **recurso** tem uma dependência do pacote de **dto**, pois é através das representações dos **dtos** que será feita uma determinada requisição. A partir dessa requisição, é feita uma chamada no pacote de **servico**, que contém toda a regra de negócio e é processada a requisição. Podendo ou não consultar a camada de **repositorio**, que está diretamente ligada a camada de banco de dados

B. Módulo Segurança

Este módulo contém todo o fluxo de permissão de acesso aos recursos da API. Para sua implementação foi utilizado o protocolo de autorização OAuth2, mencionado na seção III-D. O próprio Spring Boot 2 fornece recursos abstraindo a implementação deste protocolo. Para que todo este fluxo funcione foi preparado um modelo de dados na Figura 15.

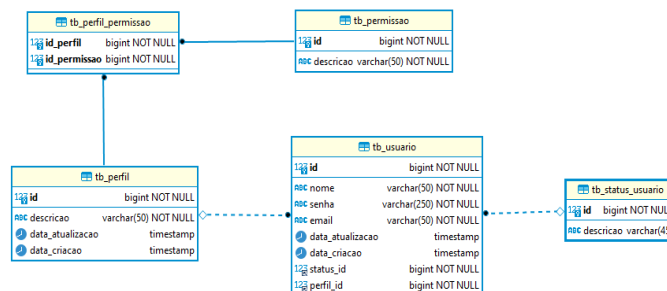


Figura 15. Modelo de dados do Módulo Segurança

O modelo mencionado anteriormente tem por finalidade controlar as permissões dos usuários por determinado módulo e funcionalidade. A tabela **tb_perfil** determina se o usuário é um **Funcionário** ou um **Paciente**. O Paciente inicialmente terá acesso apenas ao módulo do Cartão Vacinal e dados pessoais do mesmo. Vale ressaltar que a API não está fixa só a esses tipos de perfis, pois fornece recursos pertinentes a cadastrar, remover, alterar dados relacionados ao tipo de usuário (perfis) e suas respectivas permissões.

C. Módulo Cadastro Geral

Este módulo é o principal motor da API, pois é nele onde é feito o cadastro de todos os dados pertinentes ao paciente, funcionário, gênero, etnia, cnae, estado cível, unidades (unidades que são postos de imunização). Com a sua flexibilidade será possível um usuário com o **perfil** de funcionário: editar, incluir, pesquisar ou remover dados de acordo com a permissão.

Para este módulo foi necessário um estudo dos campos principais dos cadastros geral de vacinação, baseado na seção

II. Além de ser necessário utilizar a **API de Localidades do IBGE** para a busca dos dados de Estado e Municípios do território nacional, pertinentes aos funcionários, pacientes e unidades. Além dos Estados e Municípios pode ser feita a consulta do CEP para o cadastro de endereços anteriormente mencionados. Foi feita também, como mencionado anteriormente, a importação dos dados da **API de Cnaes do IBGE**. Porém, só foram importados os cnaes referentes à seção ‘Q’ que são atividades relacionadas à saúde.

Para este módulo foi necessário um estudo dos campos principais dos cadastros geral de vacinação, baseado na seção II. Além de ser necessário utilizar a **API de Localidades do IBGE** para a busca dos dados de Estado e Municípios do território nacional, pertinentes aos funcionários, pacientes e unidades. Além dos Estados e Municípios pode ser feita a consulta do CEP para o cadastro de endereços anteriormente mencionados

D. Módulo Vacinação

O Módulo de Vacinação consiste no gerenciamento dos dados referentes às vacinas, campanhas, calendários de vacinação, doses e a aplicação das vacinas nos pacientes. O usuário que tiver as devidas permissões poderá alterar, adicionar, pesquisar, remover e editar esses dados de acordo com a permissão. Este módulo é responsável por notificar os pacientes através de **SMS** ou/e **e-mail** sobre as campanhas de vacinação que estão em vigor, possibilitando manter informados e atualizados do mesmo.

Outra importante notificação é a possibilidade de avisar ao paciente qual a data da próxima dose que deverá ser aplicada. Ou seja, se um paciente **X** estiver participando de uma campanha que aplique a vacina **Y**, e esta vacina tenha a necessidade de aplicação de **2 doses**, dependendo da idade do paciente, o paciente será avisado da data de início da **1º dose**. Dependendo da data que foi aplicada a **1º dose**, será calculado de acordo com as regras da campanha em questão, a data da aplicação da **2º dose**. Sendo assim, tanto a data do início da primeira dose e a data da segunda dose, o paciente será notificado via **SMS** e/ou **e-mail** das respectivas datas.

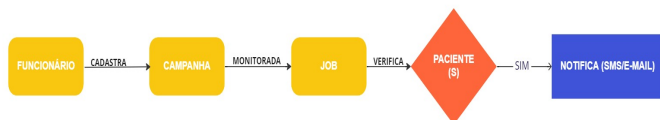


Figura 16. Job Notificação

O fluxo das notificações é realizado por **JOBS**. A princípio o funcionário cadastra uma campanha, e o **JOB** específico

monitora as campanhas ativas e verifica os pacientes pertinentes dessas campanhas. Esses pacientes são determinados pelos critérios de faixa etária da campanha. Se o paciente se enquadra no critério, é enviado uma notificação via **SMS** e/ou **e-mail** com as informações necessárias. Uma outra notificação importante são os informativos dos calendários de vacinação que só são enviados apenas por **e-mail**. Vale salientar que para que todo fluxo de notificação referentes às doses funcione, é de extrema importância a utilização dos recursos de **vacinas aplicadas**. Pois, com os registros das vacinas aplicadas dos pacientes (doses) é possível calcular e enviar a notificação das próximas doses.

Para que o processamento desses **e-mails** e **SMSs** ocorram de maneira mais segura e eficiente, foi necessário utilizar o serviço de **Message Broker** mencionado na seção **III-E**. Ou seja, para cada registro de paciente encontrado é gerado uma mensagem na fila do **RabbitMQ** com os dados do paciente. Sendo assim, uma fila do paciente é responsável pelos **SMSs** e outra fila pelos **e-mails**.

Toda a estrutura de envio de **e-mail** faz uso do **Spring Mail**, um módulo do **Ecossistema Spring Boot**, que tem todos os recursos necessários para a implementação do mesmo em Java. O **SMTP (Simple Mail Transfer Protocol)** utilizado para envio de e-mail é o Gmail, pois é de simples utilização. Já para o envio de **SMS**, foi contratado o serviço da **API COMTELE**, também pela fácil utilização de sua **API**.

E. Módulo Estoque

Um dos principais módulos é o de estoque, pois é nele que são registradas as entradas dos lotes das vacinas de acordo com os seus fornecedores e suas unidades de imunização. É responsável por controlar as doses disponíveis e indisponíveis para que seja possível aplicar as vacinas nos pacientes. A **API** também fornece recursos para verificar a validade de cada lote para que possa ser mantida a integridade da imunização. Este módulo possibilita para o lote, fornecedores e o estoque de vacinas a funcionalidade de excluir, remover, alterar e editar os dados de acordo com a necessidade.

F. Módulo Relatórios

O Módulo de relatórios viabiliza dados necessários para emissão de relatórios de todas as principais informações pertinentes à vacinação. Está diretamente ligada a todos os outros módulos. Pois, podem emitir dados do módulo de Cadastro Geral, Vacinação e Estoque. Os relatórios que podem ser emitidos de acordo com os módulos são:

- **Cadastro Geral:** pacientes, funcionários, unidades.
- **Vacinação:** total de vacinados por estados, municípios, por períodos, por idade, por vacinas, doses, total de vacinados por ano e mês. Além de poder emitir sobre as abstenções.
- **Estoque:** vacinas disponíveis e indisponíveis, lotes, fornecedores e unidades.

G. Módulo Core

Este módulo fornece funções e processamentos utilitários do resto de todos os módulos. O **JOB** principal verifica de

tempos em tempos a busca dos pacientes que precisam fazer receber as imunizações e enviar para fila no RabbitMQ, que posteriormente o módulo de Vacinação enviará as devidas notificações que estão na fila.

H. Módulo Cartão Vacinal

Neste módulo é para informações dos usuários/pacientes que poderão consultar as doses que foram aplicadas e registradas no histórico da API.

VI. VALIDAÇÃO

Para validar a API foi necessário a criação de uma aplicação WEB como um cliente da API deste trabalho, construído com o framework Angular(desenvolvido pela Google). Entretanto, antes mesmo que fosse construída as telas da aplicação foi necessário utilizar um framework de geração de documentação para API, chamado **Swagger UI** (Framework para visualização de API REST). Nele é possível conhecer todos os recursos que são expostos pela API. O **Swagger** pode ser visto na Figura 17:

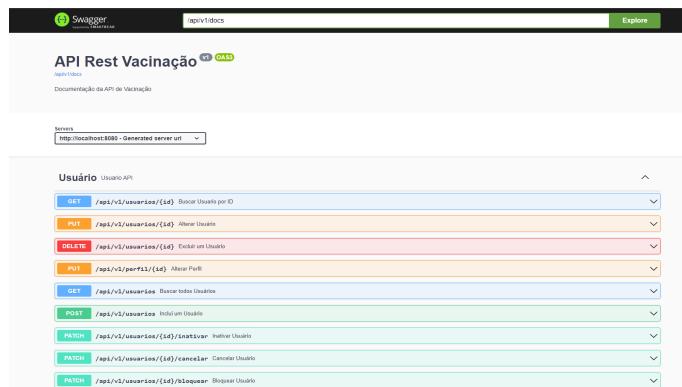


Figura 17. Swagger UI da API

Uma das principais telas é a **Tela de Cadastro de Estoque**, pois é o ponto de partida para o gerenciamento da vacinação. Nela contém dados de entradas dos os lotes, fornecedores das suas respectivas vacinas. Onde também pode ser informada a data de entrada, a Unidade (Local de Vacinação), e a quantidade total do lote. A tela pode ser vista na Figura 18.

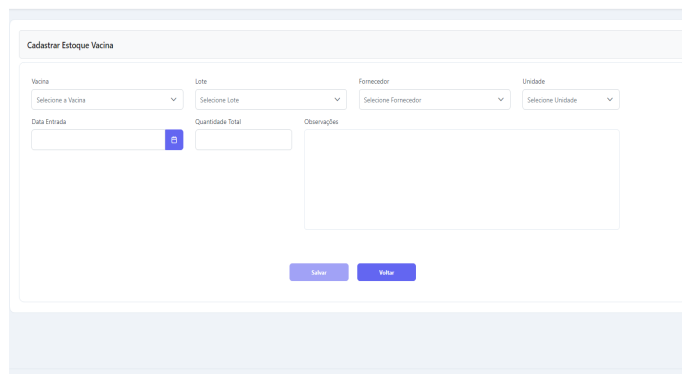


Figura 18. Tela de Cadastro de Estoque

Para que seja possível gerar registros de vacinação e deduzir as vacinas aplicadas no estoque, é necessário ter a Tela de Cadastro de Paciente, apresentada na Figura 19

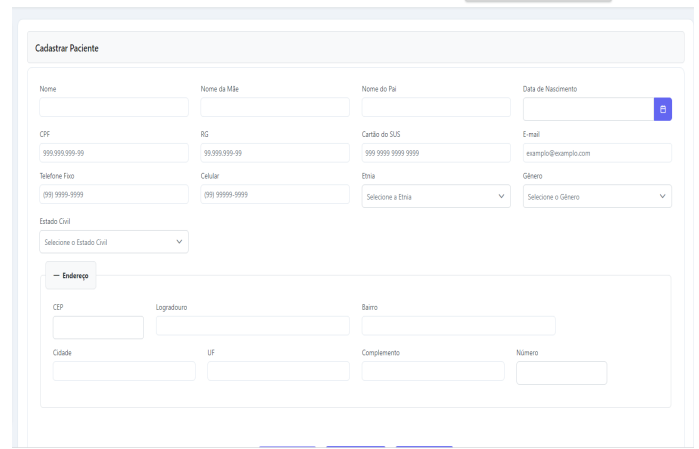


Figura 19. Tela de Cadastro de Paciente

A **Tela de Cadastro do Calendário de Vacinação** está diretamente ligada ao paciente. Visto que, é o Calendário que determina se o um determinado paciente, de acordo com os critérios estabelecidos no calendário, pode tomar a dose da vacina ou não. Esta tela pode ser vista na Figura 20.

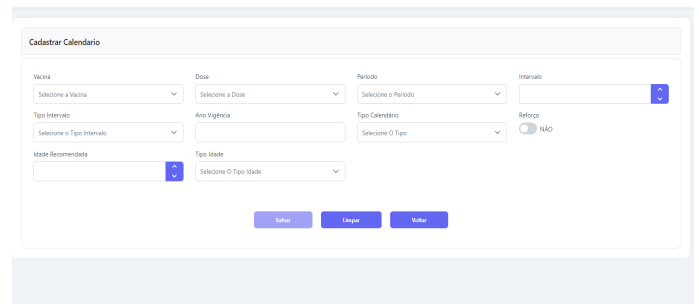


Figura 20. Tela de Cadastro de Calendário de Vacinação

Foi concluído que o Swagger UI da API estava em conformidade com as orientações fornecidas para o desenvolvimento de uma aplicação WEB, e posteriormente consumida, sendo possível a sua confiabilidade. Através das telas foi possível simular um gerenciamento de vacinação eficaz, conforme descrito para cada funcionalidade. Assim também, como todo o processo de notificação da API.

VII. CONCLUSÃO

É necessário observar a importância da adoção da TIC nas entidades públicas e privadas, proporcionando instrumentos fundamentais para melhorar a performance dos atendimentos e gerenciamento de saúde e de todas as funções de sustentação do serviço [20]. A busca por um conhecimento da área da vacinação é desafiadora. Pois, à medida que se busca informações confiáveis, percebe-se a precariedade no fornecimento do mesmo.

Contudo a meta deste trabalho não é fornecer uma fórmula que consiga mitigar todos os problemas. Mas, trazer a possibilidade de ampliação da tecnologia em uma área da saúde, tanto para as entidades envolvidas no gerenciamento, quanto para os indivíduos que irão usufruir os serviços finais.

VIII. TRABALHOS FUTUROS

Durante o desenvolvimento da solução proposta, observaram-se possibilidades de melhorias na implementação visando evoluir o Monitum. Nesse contexto, destacam-se como trabalhos futuros:

- Implementação de esquema de Importação de dados no formato XML, CSV ou JSON.
- Divisão dos módulos em microsserviços. Ou seja, cada módulo poderá ser um microsserviço independente.

REFERÊNCIAS

- [1] SAUDATE. Alexandre. REST. Construa API's Inteligentes de Maneira Simples (Português) Capa comum – 1 janeiro 2013 livro (SAUDATE, 2013).
- [2] PORTO, A. and PONTE, C. F.: Vacinas e campanhas: imagens de uma história a ser contada. *Histórias, Ciências, Saúde Manguinhos*, vol. 10 (supplement 2): 725-42, 2003 (PÔRTO E PONTE, 2003).
- [3] World Health Organization.(2006).Vaccine stock management : guidelines on stock records for immunization programme and vaccines store managers. World Health Organization. Acessado em: 17 de Março de 2021; disponível em: <https://apps.who.int/iris/handle/10665/69629> (World Health Organization, 2006).
- [4] World Health Organization (WHO). Immunization supply chain and Logistics. A neglected but essential system for national immunization programmes. WHO/IVB/14-05.2014. Acessado em 17 de Março de 2021 .Disponível em: https://www.who.int/immunization/documents/WHO_IVB_14.05/en/ .
- [5] P. Glauton L. Sistema de Gerenciamento de Vacinação, Curitiba, 2013. Acessado 17 de Março de 2021. Disponível em :https://pessoal.dainf.ct.utfpr.edu.br/criskochem/TCC_Glauton.pdf
- [6] A. Mayara. PG. Gerenciamento da sala de vacinas em unidades básicas de saúde- atuação do enfermeiro. Saúde coletiva, Salvador, 2018.
- [7] R. Rodrigo. Análise do Controle de Estoque: Um estudo de caso em uma Empresa Avícola do Sudoeste do Paraná. 2013. 55f. Trabalho de Conclusão de Curso (Curso Superior Bacharel em Administração), Departamentos Acadêmicos de Administração, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.
- [8] Nascimento, Dilene Raimundo do, As campanhas de vacinação contra a poliomielite no Brasil (1960-1990). *Ciência e Saúde Coletiva*. 2011. Acessado em 21 julho de 2021 Disponível em: <https://doi.org/10.1590/S1413-81232011000200013> .
- [9] HEUSER, Carlos A. Projeto de banco dados/ Carlos Alberto Heuser. 6 ed. - Porto Alegria: Bookman, 2009.
- [10] POOTZ, Douglas V, S e LEMES, Guilherme G. SICOVAC - SISTEMA DE CONTROLE DE VACINAÇÃO. Brasília, 2010.
- [11] PAPADÓPOLIS, Alexandre V e FONTES, Ronicley R. ImunizaBR: Aplicativo Móvel para Registro de Vacinação. Brasília, 2016.
- [12] Lopes, Jéssica P. VACINAÇÃO NA PALMA DA MÃO: CARTÃO DE VACINA DIGITAL PARA DISPOSITIVOS MÓVEIS. Divinópolis, 2019.
- [13] ALMEIDA, Yuri L. SISTEMA DE CONTROLE DE VACINAÇÃO. Rio de Janeiro, 2017.
- [14] Muratt et al. CARTEIRA DE VACINAÇÃO VIRTUAL. Goiânia, 2016.
- [15] PERPÉTUO, Samuel H M. Desenvolvimento do módulo de vacinação para integração em um Sistema de um Órgão Público Estadual. João Monlevade–MG, 2020.
- [16] CAMILO, Gabriel M M, Sistema Web para Controle de Vacinação de um Hospital. João Monlevade–MG, 2019.
- [17] SI-PNI, acessado em: 26 de fevereiro de 2021. Disponível em: <https://antigo.saude.gov.br/saude-de-a-z/vacinacao/si-pni>
- [18] SINVAC, acessado em: 03 de Março de 2021. Disponível em: <https://softwarepublico.gov.br/social/profile/sivac>
- [19] Ministério da Saúde BRASIL. Manual de Normas e Procedimentos para Vacinação, 2014, acessado em 19 de setembro de 2021 .Disponível em: https://bvsmms.saude.gov.br/bvs/publicacoes/manual_procedimentos_vac_inacao.pdf.
- [20] CGI.BR. Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos Estabelecimentos de Saúde Brasileiros-TIC Saúde 2019. São Paulo : Comitê Gestor da Internet no Brasil - CGI.br, 2020 , acessado em 19 de setembro de 2021. Disponível em: https://www.ceti.c.br/media/docs/publicacoes/2/20201123084414/tic_saude_2019_livro_eletronico.pdf
- [21] LUZ, Karine R.; SOUZA, Deborah C.; CICONELLI, Rozana M. Vacinação em Pacientes Imunossuprimidos e com Doenças Reumatológicas Auto-Imunes Revista Brasileira de Reumatologia,2007. Acessado em 10 de outubro de 2021. Disponível em :<https://www.scielo.br/j/rbr/a/3WTRPQWTLz3zhbBvR6zvbxJ/?lang=pt#> .
- [22] Silva Junior, Arnaud Marcolino da. Proposta de gestão on-line das informações de vigilância epidemiológica de eventos adversos pós-vacinação. 2010. 84 f. Dissertação - Escola Nacional de Saúde Pública Sergio Arouca, Fundação Oswaldo Cruz, Rio de Janeiro. Acessado em 10 de setembro de 2021. Disponível em :http://bvssp.icict.fiocruz.br/pdf/f/25880_silvajunioramm.pdf .
- [23] Centro Cultural Do Ministério Da Saúde. A História das Vacinas Brasil: Uma Técnica Milenar, Rio de Janeiro 2009. Acessado em 10 de setembro de 2021 . Disponível em :<http://www.ccs.saude.gov.br/revolta/pdf/M7.pdf> .
- [24] REZENDE, JM. À sombra do plátano: crônicas de história da medicina [online]. São Paulo: Editora Unifesp, 2009. Variola: uma doença extinta. ISBN 978-85-61673-63-5. Available from SciELO Books. Acessado em 15 de agosto de 2021. Disponível em: <http://books.scielo.org/id/8kf92/pdf/rezende-9788561673635-24.pdf>.
- [25] Fiocruz BRASIL. Como surgiram as vacinas?. Acesso em 16 de Agosto de 2021. Disponível em : <https://www.bio.fiocruz.br/index.php/pergunta-s-frequentes/69-perguntas-frequentes/perguntas-frequentes-vacinas/213-como-surgiram-as-vacinas>.
- [26] V.C. Maniero, M.O. Santos, R.L. Ribeiro, P.A.C. de Oliveira, T.B. Silva, A.B. Moleri, I.R. Martins, C.C. Lamas, and S.V. Cardozo. Dengue, chikungunya e zika vírus no Brasil: situação epidemiológica, aspectos clínicos e medidas preventivas. *Almanaque Multidisciplinar de Pesquisa*, 1(1), 2016. 4. Acessado em 19 de Agosto de 2021 . Disponível em : <http://publicacoes.unigranrio.edu.br/index.php/amp/article/view/3409/2114>.
- [27] Portal da Educação Brasil. Imunização no Brasil. Acessado em 02 de Setembro de 2021. Disponível em: <https://siteantigo.portaleducacao.com.br/conteudo/artigos/enfermagem/imunizacao-no-brasil/48339>.
- [28] R.G Souza. História do Brasil, Governo Rodrigues Alves. Acessado em 03 de Setembro de 2021. Disponível em: <https://brasilescola.uol.com.br/historiab/rodrigues-alves.htm>.
- [29] Programa Nacional de Imunizações 30 Anos/Ministério da Saúde, Secretaria de Vigilância em Saúde - Brasília, 2003. Acessado em 04 de Setembro de 2021. Disponível em: https://bvsmms.saude.gov.br/bvs/publicacoes/livro_30_anos_pni.pdf.
- [30] Lockshin MD: Vaccinations and Rheumatic Disease, 2011. Acessado em 04 de Setembro de 2021. Disponível em: https://www.hss.edu/conditions_vaccinations-rheumatic-disease.asp.
- [31] World Health Organization (WHO). Vaccines and immunization:What is vaccination?, 30 December 2020. Acessado em 04 de Setembro de 2021. Disponível em : <https://www.who.int/news-room/q-a-detail/vaccines-and-immunization-what-is-vaccination>.
- [32] Centro Cultural do Ministério da Saúde, Ministério da Saúde As Epidemias de Oswaldo Cruz, 2020. Acessado em 04 de Setembro de 2021. Disponível em :<http://www.ccms.saude.gov.br/noticias/epidemias-de-oswaldo-cruz>.
- [33] Fundação Oswaldo Cruz: uma instituição a serviço da vida, A trajetória do médico dedicado à ciência, publicado em maio de 2017. Acessado em 04 de Setembro de 2021. Disponível em : <https://portal.fiocruz.br/trajetoria-do-medico-dedicado-ciencia>.
- [34] UNA-SUS, Oswaldo Cruz; o sanitaria que mudou o Brasil, publicado em 5 de agosto de 2015. Acessado em 05 de Setembro de 2021. Disponível em : <https://www.unasus.gov.br/noticia/oswaldo-cruz-o-sanitaria-que-mudou-o-brasil>.
- [35] Fundação Oswaldo Cruz. A Revolta da Vacina , publicado em 25 de Abril de 2005. Acessado em 13 de Agosto de 2021 . Disponível em : <https://portal.fiocruz.br/noticia/revolta-da-vacina-2>.
- [36] Muniz, Érico Silva ,Memórias da erradicação da varíola. *Ciência e Saúde*

- Coletiva. 2011. Acessado em 21 de Agosto de 2021. Disponível em: <https://doi.org/10.1590/S1413-81232011000200034> .
- [37] Ministério da Saúde BRASIL. SI-PNI - Sistema de Informação do Programa Nacional de Imunizações / DATASUS / MS. Brasil, Rio de Janeiro. Acessado em 23 de Agosto de 2021. Disponível em <http://pni.datasus.gov.br/apresentacao.asp>.
- [38] J.L. Benchimol. Febre amarela: A Doença e a Vacina, uma história inacabada. Fiocruz, 2001. Acessado em 12 de Outubro de 2021 . Disponível em : <https://static.scielo.org/scielobooks/4nktq/pdf/benchimol-9788575413951.pdf>.
- [39] Hochman, Gilberto .Vacinação, varíola e uma cultura da imunização no Brasil. Ciência e Saúde Coletiva. 2011. Acessado 12 de Agosto de 2021. Disponível em: <https://www.scielo.br/j/csc/a/YWJ7XqXpmNXNFtBtMbr8Sm/?lang=pt>.
- [40] Temporão, José Gomes .O Programa Nacional de Imunizações (PNI): origens e desenvolvimento. História, Ciências, Saúde-Manguinhos. 2003, v. 10. Acessado 04 Setembro 2021. Disponível em: <https://doi.org/10.1590/S0104-59702003000500008>.
- [41] Ministério da Saúde , DATASUS - Departamento de Informática do SUS, Brasília. Acessado 09 Setembro 2021. Disponível em : <http://sipni.datasus.gov.br/si-pni-web/faces/apresentacaoSite.jsf>
- [42] Ministério da Saúde, Secretaria de Vigilância em Saúde, Departamento de Imunização e Doenças Transmissíveis, Coordenação-Geral do Programa Nacional de Imunizações. Manual dos Centros de Referência para Imunobiológicos Especiais - CRIE, 5ª edição - Brasília/DF 2019. Acessado 12 Setembro de 2021. Disponível em: https://bvsm.sau.gov.br/bvsm/publicacoes/manual_centros_imunobiologicos_especiais_5ed.pdf
- [43] Fundação Nacional de Saúde, Recomendações de vacinação em pessoas infectadas pelo HIV, Brasília, 2002. Acessado em 15 de Setembro de 2021. Disponível em: https://bvsm.sau.gov.br/bvsm/publicacoes/funas_a/vacinacao_hiv.pdf
- [44] Qual a diferença entre AIDS e HIV. Acessado em 15 de Setembro de 2021. Disponível em : <https://www.pfizer.com.br/noticias/ultimas-noticias/qual-diferenca-entre-hiv-e-aids>
- [45] Fundação Médica Brasileira, Tratamento com Imunobiológicos: uma nova perspectiva para pacientes com doenças autoimunes. Disponível em: <http://portalfmb.org.br/2016/11/01/tratamento-com-imunobiologicos-uma-nova-perspectiva-para-pacientes-com-doencas-autoimunes>. Acessado em 15 de Setembro de 2021.
- [46] Observatório Juventude CT Fio Cruz, O que é Imunologia?. Acessado em 15 de Setembro de 2021. Disponível em: <http://www.juventudect.fiocruz.br/imunologia>
- [47] Universidade Federal do Rio Grande do Sul, Microrganismos. Acessado em 15 de Setembro de 2021. Disponível em: <http://www.ufrgs.br/patriomio geneticos/conceitos-e-definicoes/microrganismos>
- [48] Faculdade de Ciências Médicas-Unicamp, Imunologia Celular. Acessado em 16 de Setembro de 2021. Disponível em: <https://www.fcm.unicamp.br/fcm/cipoi/imunologia-celular/overview>
- [49] Ministério da Saúde, Secretaria de Vigilância em Saúde, Departamento de Vigilância Epidemiológica. Manual dos Centros de Referência para Imunobiológicos Especiais - Série A. Normas e Manuais Técnicos, 3ª edição - Brasília/DF 2006. Acessado 17 Setembro de 2021. Disponível em: https://bvsm.sau.gov.br/bvsm/publicacoes/manual_centro_referencia_imunobiologicos.pdf
- [50] World Health Organization, 2020, Como funcionam as vacinas. Acesso em 17 de Setembro de 2021. Disponível em: https://www.who.int/pt/news-room/feature-stories/detail/how-do-vaccines-work?gclid=Cj0KCQiAsqOMBhDFARiAFBtN3fV47f82s8e9WL-R-Zl3kDTZnI1FE19A98dBeMeVbbV776oCX9PssaAqRCEALw_wcB
- [51] Organização Pan-Americana da Saúde, 2021, Imunização. Acessado em 17 de Setembro de 2021. Disponível em: <https://www.paho.org/pt/topicos/imunizacao>
- [52] Instituto Butantan, 2021, Entenda como funciona a tecnologia de vírus inativado usada na CoronaVac. Acessado em 17 de Setembro de 2021. Disponível em: <https://butantan.gov.br/covid/butantan-tira-duvida-tira-duvida-noticias/entenda-como-funciona-a-tecnologia-de-virus-inativado-usada-na-coronavac>
- [53] Pfizer, Saiba tudo sobre vacinas. Acesso 18 de Setembro de 2021. Disponível em : <https://www.pfizer.com.br/sua-saude/vacinacao/tudo-sobre-vacinasc>
- [54] Ministério da Saúde, Secretaria de Vigilância em Saúde. Departamento de Imunizações e Doenças Transmissíveis. Manual de Vigilância Epidemiológica de Eventos Adversos pós-vacinação, 3ª edição, Brasília/DF, 2014. Acessado em 22 de Agosto de 2021. Disponível em: https://bvsm.sau.gov.br/bvsm/publicacoes/manual_vigilancia_epidemiologica_eventos_adversos_pos_vacinacao.pdf
- [55] Ministério da Saúde. Portaria nº 1.533, de 18 de agosto de 2016. Redefine o Calendário Nacional de Vacinação, o Calendário Nacional de Vacinação dos Povos Indígenas e as Campanhas Nacionais de Vacinação, no âmbito do Programa Nacional de Imunizações (PNI), em todo o território nacional. Acessado em 20 de Outubro de 2021. Disponível em: https://bvsm.sau.gov.br/bvsm/saudelegis/gm/2016/prt1533_18_08_2016.html
- [56] Ministério da Saúde, Secretaria de Vigilância em Saúde, Departamento de Informática do SUS - DATASUS. Sistema de Informação do Programa Nacional de Imunizações - SIPNI. Manual do usuário do SIPNI(Desktop), 2018. Acessado em 14 de Agosto de 2021. Disponível em: http://pni.datasus.gov.br/sipni/documentos/manual_sipni.pdf
- [57] Ministério da Saúde, Coordenação Geral do Programa Nacional de Imunizações, Secretaria de Vigilância em Saúde, Departamento de Informática do SUS - DATASUS. Manual do Sistema de Informação do Programa Nacional de Imunizações SIPNI, 2014. Acessado em 14 de Agosto de 2021. Disponível em : https://www.sau.gov.br/images/imagens_migradas/upload/arquivos/2015-02/manualsipni_20022014_final.pdf
- [58] Ministério da Saúde. Portaria nº 2.363, de 18 de outubro de 2012. Institui repasse financeiro do Fundo Nacional de Saúde aos Fundos de Saúde dos Estados, Distrito Federal e Municípios, por meio do Piso Variável de Vigilância e Promoção da Saúde, para fomento na implantação do Sistema de Informação do Programa Nacional de Imunizações (SI-PNI) e Sistema de Informação de Agravos de Notificação (SINAN), no âmbito das unidades de saúde . Acessado em 15 de Agosto de 2021. Disponível em : https://bvsm.sau.gov.br/bvsm/saudelegis/gm/2012/prt2363_18_10_2012.html
- [59] Ministério da Saúde Secretaria de Vigilância em Saúde Departamento de Vigilância Epidemiológica Coordenação Geral do Programa Nacional de Imunizações GT- Análise e Informação em Imunizações. Manual para registro de doses aplicadas no Sistema de Informação online de Avaliação do Programa de Imunizações – APIWEB, 2014. Acessado em 19 de Outubro de 2021. Disponível em: <http://pni.datasus.gov.br/DOWNLOAD/API/ManualAPIWEB.pdf>
- [60] Ministério da Saúde. Portaria nº 2.488, de 21 de outubro de 2011. Aprova a Política Nacional de Atenção Básica, estabelecendo a revisão de diretrizes e normas para a organização da Atenção Básica, para a Estratégia Saúde da Família (ESF) e o Programa de Agentes Comunitários de Saúde (Pacs). Acessado em 19 de outubro de 2021. Disponível em: http://bvsm.sau.gov.br/bvsm/saudelegis/gm/2011/prt2488_21_10_2011.html
- [61] Ministério da Saúde. Calendário Nacional de Vacinação. Acessado em 19 de outubro de 2021. Disponível em: https://www.gov.br/saude/pt-br/media/pdf/2021/junho/09/calendario-de-vacinacao-2020_gestantes.pdf
- [62] FIO CRUZ, Fundação Oswaldo Cruz, O que é IgG e IGM, 2020. Acessado em 19 de Setembro de 2021. Disponível em: <https://www.bio.fiocruz.br/index.php/br/noticias/1739-o-que-e-igg-e-igm>
- [63] RICHARDSON, L.; RUBY, S. RESTful Web Services: Editora O'Reilly, 1ª edição ,2007.
- [64] RICHARDSON, L.; AMUNDSEN, M. RESTful Web APIs: Editora O'Reilly Media, 1ª edição, 2013.
- [65] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>. Acessado em dezembro de 2021.
- [66] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, DOI 10.17487/RFC6750, October 2012, <https://www.rfc-editor.org/info/rfc6750>. Acessado em dezembro de 2021.
- [67] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>. Acessado em dezembro de 2021.
- [68] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>. Acessado em dezembro de 2021.
- [69] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>. Acessado em dezembro de 2021.

- [70] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540> . Acessado em dezembro de 2021.
- [71] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231> . Acessado em dezembro de 2021.
- [72] FIELDING, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Dissertação de Doutorado - University of California, Irvine, 2000. Disponível em http://www.ic.s.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. Acessado em dezembro de 2021.
- [73] THOMPSON, H. and C. Lilley, "XML Media Types", RFC 7303, DOI 10.17487/RFC7303, July 2014, <https://www.rfc-editor.org/info/rfc7303>. Acessado em dezembro de 2021.
- [74] MASSÉ, M. REST API Design Rulebook: Editora O'Reilly Media, 1ª edição, 2011.
- [75] SUBRAMANIAN, H; RAJ, P. Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs: Editora Packt Publishing, 1ª edição, 2019.
- [76] RED HAT. O que é arquitetura orientada a serviços (SOA)?. Publicado em: 27 de julho de 2020 .Disponível em: <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-service-oriented-architecture>. Acessado em dezembro de 2021.
- [77] IBM C. E. SOA (Arquitetura Orientada a Serviços) :Conheça a SOA (arquitetura orientada a serviços), um estágio importante na evolução do desenvolvimento e integração de aplicativos. Publicado em 17 de Julho de 2019. Disponível em : <https://www.ibm.com/br-pt/cloud/learn/soa>. Acessado em dezembro de 2021.
- [78] JOSUTTIS, M. N. SOA in Practice The Art of Distributed System Design : Editora O'Reilly Illustrated, 2007.
- [79] IBM. Message Brokers. IBM Cloud Education. Publicado em 23 de Janeiro de 2020; Disponível em: <https://www.ibm.com/br-pt/cloud/learn/message-brokers> . Acessado em Novembro de 2022.
- [80] RabbitMQ. AMQP 0-9-1 Model Explained. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/message-brokers> . Acessado em Novembro de 2022.