

Aplicação Web em Saúde 4.0: Uma Plataforma Multiusuário Baseada em um Servidor HAPI FHIR

Maicon Francisco
*Instituto de Engenharia Biomédica
Universidade Federal de Santa
Catarina
Florianópolis, Brazil
ORCID: 0000-0002-8895-3249*

Igor Oliveira Vieira
*Instituto de Engenharia Biomédica
Universidade Federal de Santa
Catarina
Florianópolis, Brazil
ORCID: 0000-0003-4629-6970*

José Marino Neto
*Instituto de Engenharia Biomédica
Universidade Federal de Santa
Catarina
Florianópolis, Brazil
ORCID: 0000-0001-5090-8867*

Renato Garcia Ojeda
*Instituto de Engenharia Biomédica
Universidade Federal de Santa
Catarina
Florianópolis, Brazil*

Jefferson Luiz Brum Marques
*Instituto de Engenharia Biomédica
Universidade Federal de Santa
Catarina
Florianópolis, Brazil
ORCID: 0000-0001-8910-4735*

Abstract— This article presents an application in Healthcare 4.0 as a multi-user platform based on the HL7 FHIR (Health Level Seven Fast Healthcare Interoperability Resources) standard. A HAPI (HL7 Application Programming Interface) FHIR server was implemented and configured to meet the platform's access requests. A web application was designed and developed using the MVC design pattern (Model, View and Controller), which presents three interfaces that correspond to different user profiles so that different forms of data visualization are implemented for each interface.

Keywords — Healthcare 4.0, HL7 FHIR, HAPI FHIR, MVC design pattern, Multiuser Interface.

I. INTRODUÇÃO

A área da saúde vem mostrando constante evolução no desenvolvimento tecnológico e transformação digital. Novos conceitos em saúde e modelos baseados em inovação e utilização de tecnologias de comunicação e informação já estão sendo estudados e implementados. Estes conceitos seguem os princípios das aplicações de Indústria 4.0, adaptando-os para a indústria de saúde chamada de Indústria em Saúde 4.0, ou Saúde 4.0. O sistema de Saúde 4.0 pode ser definido como uma implementação da integração de plataformas de saúde de forma virtual, distribuída e em tempo real para pacientes e profissionais. Este conceito permite a integração, compartilhamento e otimização do uso dos serviços em saúde para melhorar seu gerenciamento e reduzir custos [1][2].

Este sistema de saúde faz uso de tecnologias como sistemas ciber físicos (SCF), computação em nuvem, IoT (*Internet of Things*), e sistemas de comunicação 5G. A implementação do paradigma de SCF ajudará softwares se adequarem a sistemas distribuídos e *Big Data* e inteligência artificial irão virtualizar as informações, permitindo a geração de diagnósticos de forma rápida e precisa [3].

A utilização de tecnologia da informação na saúde começou com a introdução de websites sobre saúde pública e gerenciamento de doenças, e então seguido pelos registros eletrônicos de saúde (RES), computação em nuvem e interações de aplicativos móveis. O crescimento da Saúde 4.0 traz todas as tecnologias acompanhadas de coleta de dados, desenvolvimento da inteligência artificial, e soluções sofisticadas para facilitar interfaces virtuais interativas. O objetivo da Saúde 4.0 é conectar pacientes, dispositivos

médicos, hospitais, clínicas, fornecedores médicos e farmacêuticos, organizações de pesquisa, e outros componentes relacionados ao ecossistema da saúde [1][3].

A aplicação de Saúde 4.0 é baseada em seis princípios da Indústria 4.0 [2][3][4]:

- Interoperabilidade: capacidade de trocar informações entre sistemas de saúde através de uma estrutura padronizada de comunicação;
- Virtualização: a aplicação da interoperabilidade para conexão entre diversos aparelhos de saúde permite criar a digitalização de informações, sistemas e processos;
- Descentralização: sistemas de saúde, com a aplicação de IA, irão apresentar e auxiliar na tomada de decisões e se auto controlar;
- Capacidade em tempo real: capacidade de adquirir dados, apresenta-los e tomar decisões em uma janela de tempo muito pequena;
- Serviços orientados: criação de softwares de serviço para interagir com aparelhos e sistemas médicos;
- Modularidade: sistemas com habilidades de melhorar módulos individuais para atender novos requisitos e reutilizar módulos para criar novos sistemas de saúde.

As aplicações em Saúde 4.0 apresentam diversos desafios, como a privacidade dos pacientes, por este motivo os dados não podem ser armazenados em servidores públicos, ou, em alguns casos, os dados devem estar imediatamente acessíveis, sem atraso ou falha com a nuvem [5]. Outro grande desafio é a interoperabilidade entre sistemas e equipamentos, que aplicado este conceito, traz diversos benefícios, entre eles economia e a facilidade no compartilhamento e acessibilidade de dados.

O crescimento da quantidade e tipos de dados na área da saúde impulsionou o desenvolvimento de padrões para modelar e representar essas informações para criar a interoperabilidade entre diferentes sistemas informatizados em saúde. Alguns padrões foram desenvolvidos por organizações em saúde, tais como o *Health Level Seven (HL7)*, o *Observational Health Data Sciences and Informatics (OHDSI)*, o *Patient-Centered Outcomes Research Institute (PCORI)* [6].

A organização HL7 desenvolveu o padrão *Fast Healthcare Interoperability Resources* (FHIR) que combina as tecnologias modernas em evolução e as necessidades do mercado, e vem sendo aceito como a próxima geração de padrão para compartilhamento de informações de saúde entre sistemas e organizações [6][7].

Este trabalho tem o objetivo de propor uma aplicação de Saúde 4.0 como uma plataforma multiusuário para um servidor baseado no HL7 FHIR. A plataforma irá criar um controle de acesso aos dados do servidor e fornecer as informações condizentes com cada perfil pré definido pelo sistema. Dessa forma a interface dos usuários irá entregar informações de forma organizada, particularizada, estruturada e segura.

II. MATERIAIS E MÉTODOS

A. Arquitetura de Software

O sistema proposto é desenvolvido como uma aplicação web e utiliza o padrão de design MVC (*Models-Views-Controller*). O MVC consiste em três camadas, o “*Models*” é o objeto de aplicação, quem irá manipular os dados, a “*View*” é a interface apresentada ao usuário, e o “*Controller*” irá definir o que o usuário irá ver de acordo com os seus *inputs*. Este padrão de design permite grande flexibilidade, replicabilidade, clareza e segurança [8].

As camadas de *View* e *Model* devem responder as requisições exigidas pela camada de *Controller*, que será a camada que o usuário irá realizar seus *inputs*, como representado na Fig. 1, onde as setas representam as trocas de informações.

O projeto irá apresentar 3 perfis de usuários: pesquisador, profissional da saúde e gestor. Os dados visualizados por cada perfil de usuário dependem das aplicações e projetos incluídos no servidor. Assim a camada *Model* irá realizar as requisições ao servidor de acordo com as permissões do perfil do usuário, o *Controller* irá requisitar as *Views* para que o usuário possa visualizar então os dados.

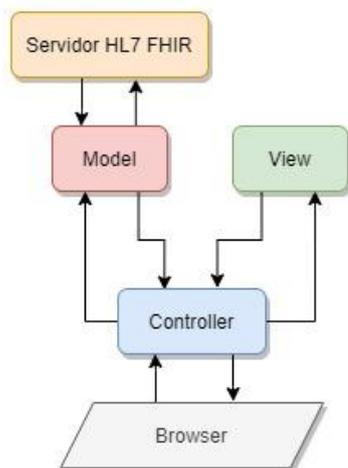


Fig. 1. A camada de interação do usuário, representada pelo *browser*, sempre será diretamente com o *Controller*, o qual irá realizar os comandos e requisições que deverão ser respondidas pela *View* e *Model* [adaptado de [8]].

B. Interface Multiusuário

A plataforma do usuário Pesquisador contém informações mais brutas dos dados e nenhuma identificação dos pacientes, com a possibilidade de exportá-los, e assim trabalhar

conforme sua necessidade. O usuário pode escolher quais dados deseja obter, podem ser eletrocardiograma, eletroencefalograma, dados respiratórios, e aplicar filtros, tais como, idade, gênero, comorbidade, etc.

Os filtros aplicados servem para montar um *bundle* para a requisição do servidor. Um *bundle* significa um conjunto de requisições de Recursos do servidor em uma única requisição [9]. O usuário pode ainda selecionar um paciente (identificado por um número aleatório) para analisar apenas os seus dados.

Desta forma um Pesquisador irá sempre selecionar um projeto, realizar os filtros, selecionar ou não um paciente e baixar os dados (Fig. 2). Observamos que essa ação realiza duas requisições para o servidor.

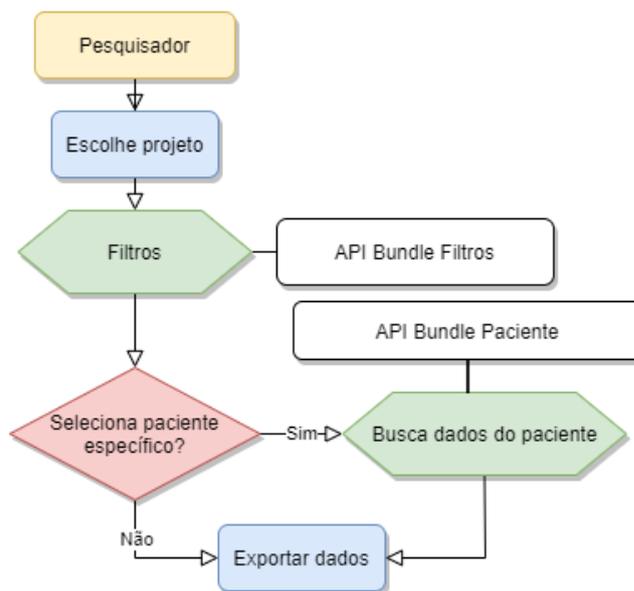


Fig. 2. Fluxograma da interface de Pesquisador. Nesta interface existem duas requisições para o servidor HL7 FHIR, uma onde é criado um *Bundle* através dos filtros selecionados, e uma para requisitar as informações de um paciente específico.

O perfil de Profissional da Saúde foi pensado para entregar informações mais diretas. Inicialmente o profissional iria procurar um paciente pelo CPF ou RG, e ter acesso a todas as informações básicas do paciente, como nome completo, idade, gênero, endereço, comorbidades, entre outros. Para cada paciente o profissional poderá ver análises e estudos incluídas no servidor. Por exemplo, um paciente que realizou um eletrocardiograma, e foram analisados batimento cardíaco médio, variabilidade de frequência cardíaca, análise espectral com o método CZF, análise através de um Plot de Lorenz, entre outros métodos, podem ser acessados pela plataforma.

Nesta plataforma, existe um campo de observações, onde o profissional poderá fazer comentários sobre o paciente, e um campo de “Medicamentos” onde ele poderá consultar quais medicamentos o paciente está utilizando, e até realizar modificações.

O perfil de Gestor realiza diversas análises, principalmente gráficas, com o intuito de auxiliar o usuário a obter informações para identificar o sistema de saúde, e assim traçar metas e soluções. Por exemplo, um gestor pode querer saber a quantidade de pessoas com diabetes mellitus na região de Florianópolis, traçar essa visualização por idade, gênero, e

observar projeções criadas pela plataforma com base nos dados, e assim criar políticas e ações para tratá-las.

C. Princípios do padrão HL7 FHIR

O padrão FHIR é fundamentalmente baseado no conceito de “Recurso”, que é um módulo de dados fundamental que pode ser acessado usando uma URL via REST (*Representational State Transfer*). Por exemplo, um dado de Paciente é formado agregando recursos FHIR como paciente, condição, procedimento, medicamentos, observação, imunizações, e plano de saúde. De forma geral, um recurso FHIR pode ser visto como uma peça de “Lego”, e você pode juntar essas peças para criar um *output* final. Dessa forma um registro médico de paciente pode ser acessado como um conjunto de recursos FHIR através de uma URL. Quando os dados são acessados via URL, os recursos são comunicados nos formatos XML ou JSON [9].

Os acessos ao servidor devem ser realizados por requisições RESTful, e as informações são entregues no formato de uma API seguindo o padrão HL7 FHIR.

O modelo de sistema REST foi descrito por Fielding no ano 2000, mais tarde este modelo foi chamado de arquitetura RESTful, e se tornou amplamente adotado como a abstração de informação dominante da WordWideWeb. A maioria das plataformas da Internet migraram para os serviços RESTful e saíram de outras opções de arquitetura [10].

As vantagens práticas da arquitetura RESTful inclui interfaces mais leves que permitem rápida transmissão e processamento de dados, mais adequado para dispositivos mobile e tablet. Interfaces RESTful também permitem um desenvolvimento mais rápido por ter estruturas mais simples [11].

No HL7 FHIR todos os recursos devem ter uma URL que identifica o recurso e especifica onde é acessado. Por exemplo, quando um recurso é acessado via FHIR RESTful API, a URL do recurso é “[base]/[resourceType]/[id]” onde “resourceType” e “id” vêm do recurso [12].

D. Aplicação do Servidor HL7 FHIR

Existem diversos servidores públicos, *open sources*, que podem ser baixados e implementados localmente, ou mesmo acessados online. Os desenvolvedores podem se conectar com esses servidores e utilizar suas APIs, e todos eles suportam acesso não autenticado. Os responsáveis por essas aplicações tem o objetivo de fornecer uma ferramenta para que desenvolvedores possam criar ou modificar recursos para realizar testes de implementações [13].

Um desses servidores muito utilizado é o HAPI (HL7 *Application Programming Interface*), uma implementação completa do HL7 FHIR em JAVA. O projeto é primeiramente suportado pela *University Health Network* (UHN), e é formado por um time de programadores com histórico e reconhecimento por produzirem softwares *open source* relacionados a projetos integrados ao HL7 [13]. O projeto tem vários benefícios, entre eles um servidor *open source*, excelente documentação, comunidade grande e colaborativa, e frequentemente é atualizado (arrumando bugs, adicionando recursos, e expandindo recursos para novos padrões e recursos FHIR) [14]. HAPI FHIR é geralmente a ferramenta mais utilizada para desenvolvedores, e tem sido usado em diversas aplicações, desde soluções mobile até implementações pelos maiores provedores de serviços em nuvem [13].

O sistema proposto utiliza um servidor HAPI FHIR utilizando a versão HL7 FHIR R4. Nesta aplicação temos acesso a todos os Recursos FHIR, acessos através da API RESTful, e a liberdade de criar nossos próprios Recursos para atender os formatos de dados dos projetos de interesse para visualizar através da plataforma proposta.

III. RESULTADOS

Os resultados apresentados foram realizados e testados localmente.

A. O Servidor HL7 FHIR

O servidor HAPI FHIR foi testado utilizando o Recurso de “Paciente”, sendo um dos 11 recursos padronizados pelo HL7 FHIR até a escrita deste artigo. Os dados foram escritos manualmente no servidor e acessados através da plataforma.

A comunicação entre a plataforma e o servidor acontece através de requisições, GET ou POST, que são respondidas pela API RESTful do servidor. A requisição é enviada via protocolo *http* e o servidor responde em um formato JSON.

B. A Plataforma

A plataforma contém 3 interfaces, de forma que, para o Recurso “Paciente”, cada uma irá apresentar os dados de forma diferente. Na Fig. 3 podemos observar a distribuição das informações que o Recurso “Paciente” possui, e quais dados as interfaces terão acesso. Por exemplo, para o usuário Pesquisador, não serão apresentadas as informações de “nome” e “CPF/RG” dos pacientes, apenas o identificador e informações que possam ser de interesse. O usuário Gestor irá visualizar apenas dados que possam trazer informações, como por exemplo número de pessoas por gênero e idade. No usuário Profissional temos acesso a todos os dados, pois é necessário a identificação do paciente no momento de uma consulta médica.

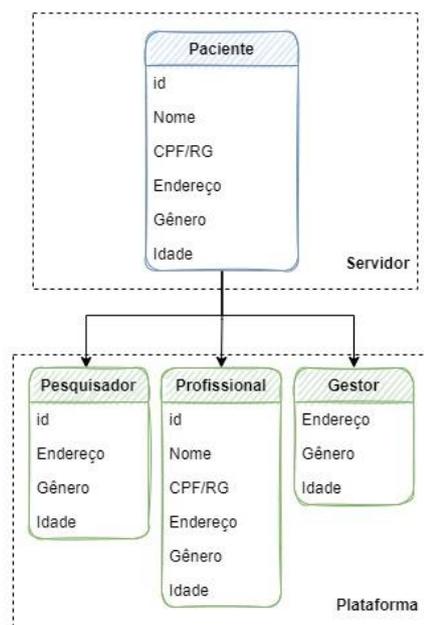


Fig. 3. As informações apresentadas em cada interface da plataforma dependerá do perfil do usuário, assim, mesmo acessando ao mesmo Recurso, como o Recurso Paciente, cada interface terá um acesso limitado aos dados.

Este fluxograma apresenta o formato de visualização da plataforma. Os dados coletados e incluídos no servidor serão distribuídos pelas interfaces, de forma que a informação seja

direcionada para cada perfil de usuário. Na Fig. 4 podemos observar o formato da tela de um paciente a partir de um usuário como Profissional da Saúde.

Cada interface contém um menu superior onde o usuário pode abrir seu perfil e alterar informações de login e senha, e um menu na lateral esquerda, onde poderá acessar informações específicas de cada projeto vinculado com o servidor HAPI.

As interfaces foram criadas utilizando a mesma lógica MVC. Foram criadas classes para o bloco de *Controller* e *Model*, e as *Views* são modelos HTML, a interação dessas classes acontece conforme o modelo de MVC já descrito. Na Fig. 5 podemos ver a diagramação de classes para a visualização de pacientes através da interface de um profissional da saúde, e da mesma forma foram construídos para os outros perfis de usuários.



Fig. 4. Interface de Profissional da Saúde em uma tela de paciente.

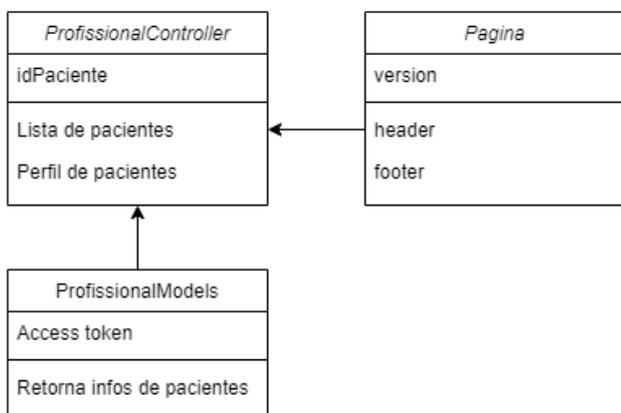


Fig. 5. Diagramação de classes para um perfil de Profissional da Saúde acessando os dados de pacientes. As classes *ProfissionalController* e *Pagina* fazem parte do bloco *Controller* e *ProfissionalModels* do *Model*, e a classe *Pagina* é uma classe que fornece o *header* e *footer* de uma página HTML.

IV. DISCUSSÃO

O servidor HAPI FHIR apresentou ser uma excelente ferramenta como implementação rápida de um servidor HL7 FHIR. Os testes foram realizados na mesma máquina do servidor, ou seja, testes através da nuvem são necessários para

simular um ambiente real e observar seu comportamento. Outras configurações são necessárias para serem realizadas no servidor e onde ele será implementado, como o sistema de segurança. Na documentação do HL7 FHIR [15] diz que ele não é um protocolo de segurança, nem define qualquer funcionalidade à segurança, mas o define protocolos de troca e modelos de conteúdo que precisam ser usados com vários protocolos de segurança definidos em outro lugar. Um exemplo são as trocas de dados que devem ser protegidas usando TLS (*Transport Layer Security*), como *https*, sistemas de autenticação usando o OAuth, controle de acesso e autorização, entre várias outros.

A função de controle de acesso do HL7 FHIR tem uma aplicação interessante para acrescentar no projeto, pois permite criar regras para os usuários, assim cada perfil terá acesso somente ao que for definido pelos projetos. Por exemplo, no recurso “Paciente”, na Fig. 3 observamos que apenas o perfil Profissional da Saúde terá acesso a informação de “nome”, ou seja, no servidor podemos implementar uma regra que apenas este perfil terá acesso a essa informação.

A plataforma apresentou ter uma implementação intuitiva ao utilizar o método MVC, e os resultados iniciais mostraram respostas do servidor no formato JSON conforme o padrão HL7 FHIR, onde essas informações são tratadas e apresentadas à interface, conforme seu usuário de acesso.

A aplicação atendeu os princípios de Saúde 4.0 descrito em [2][3][4] ao utilizar o padrão HL7 FHIR, pois permite a conexão com outros dispositivos médicos, virtualização da informação, e a plataforma implementada em MVC apresentou modularidade de implementação e uma ferramenta de auxílio para tomada de decisão.

V. CONCLUSÃO

Neste trabalho foi descrito o desenvolvimento de uma aplicação em Saúde 4.0 como uma plataforma multiusuário de acesso a um servidor HL7 FHIR, sua implementação com o método MVC, comunicação padrão com a API RESTful do servidor, comportamento das interfaces e seus propósitos. Futuras pesquisas devem realizar mais testes com o servidor em nuvem, criar protocolos de segurança e implementar mais dados no servidor e configurar a interface para interpretá-los.

REFERÊNCIAS

- [1] Al-Jaroodi J, Mohamed N, Abukhousa E, Healthcare 4.0, IEEE Engineering Management Review, v. 47, n. 3, p. 24–28, 1 jul. 2019.
- [2] Chute C, French T, Introducing Care 4.0: An Integrated Care Paradigm Built on Industry 4.0 Capabilities, International Journal of Environmental Research and Public Health, v. 16, n. 12, p. 2247, 25 jun. 2019.
- [3] Khelassi, A, et al. Health 4.0: Applications, Management, Technologies and Review, Medical Technologies Journal, v. 2, n. 4, p. 262–276, 9 jan. 2019.
- [4] Thuemmler C, Bai C, Health 4.0: How virtualization and big data are revolutionizing healthcare, Cham: Springer International Publishing, 2017.
- [5] Pace P, et al. An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0, IEEE Transactions on Industrial Informatics, v. 15, n. 1, p. 481–489, 1 jan. 2019.
- [6] Hong N, et al. Visual FHIR: An Interactive Browser to Navigate HL7 FHIR Specification, Proceedings - 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017. Institute of Electrical and Electronics Engineers Inc., 8 set. 2017.
- [7] Janki Z R, et al. Authorization solution for full stack FHIR HAPI access. IEEE 30th Jubilee Neumann Colloquium, NC 2017. Institute of Electrical and Electronics Engineers Inc., 17 jan. 2018.

- [8] Gamma E, Helm R, Johnson R, Vlissides J, Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos, tradução Luiz A. Meirelles Salgado, Porto Alegre: Bookman, 2007.
- [9] Resource Bundle – Detailed Descriptions. Disponível em: <<https://www.hl7.org/fhir/bundle-definitions.html>>. Acessado em: 05 de julho de 2021.
- [10] Saripalle R, Fast health interoperability resources (FHIR): Current status in the healthcare system, International Journal of E-Health and Medical Communications, [S. l.], v. 10, n. 1, p. 76–93, 2019. DOI: 10.4018/IJEHMC.2019010105.
- [11] Bender D, Sartipi K, HL7 FHIR: An agile and RESTful approach to healthcare information exchange. In: Proceedings Of Cbms 2013 - 26th Ieee International Symposium On Computer-Based Medical Systems 2013. [s.l.: s.n.] p. 326–331. DOI: 10.1109/CBMS.2013.6627810.
- [12] FHIR Overview - Developers. Disponível em: <<http://hl7.org/fhir/overview-dev.html>>. Acessado em: 10 de julho de 2021.
- [13] Benson T, Grieve G, Implementing FHIR, Principles of Health Interoperability, [S. l.], p. 397–416, 2016. DOI: 10.1007/978-3-319-30370-3_22.
- [14] Hussain M A, Langer S G, Kohli M, Learning HL7 FHIR Using the HAPI FHIR Server and Its Use in Medical Imaging with the SIIM Dataset, Journal of Digital Imaging, [S. l.], v. 31, n. 3, p. 334–340, 2018. DOI: 10.1007/s10278-018-0090-y.
- [15] FHIR Security. Disponível em: <<http://hl7.org/fhir/security.html>>. Acessado em: 18 de julho de 2021.