

GERENCIAMENTO DE ATENDIMENTO MÉDICO

Everton Correa Assis¹; Joice Leandra de Oliveira Bernardes²; Lucas Antonio Lisboa Reis³; Lucas Euripedes de Oliveira⁴; Paulo Limirio da Silva⁵.

^{1, 2, 3, 4, 5} Universidade de Uberaba

joiceoleandra@msn.com

Resumo

Com o desenvolvimento das tecnologias conhecidas nos dias atuais e visando os setores de saúde pública e privada tendo como prioridade a rapidez, o mercado da saúde vem sendo alvo de grande procura, por ser um ramo com oportunidade de crescimento. O sistema proposto neste artigo foi desenvolvido para a plataforma *web*, visando especialmente os usuários com pouco conhecimento de informática. Para tanto, foi concebido um sistema intuitivo utilizando os conceitos de usabilidade. Verificou-se que poucas empresas no ramo da saúde têm a oportunidade de utilizar algum tipo de plataforma de atendimento devido às atuais plataformas disponíveis não disponibilizarem vantagens como os nossos ou de difícil implementação levando em consideração a difícil usabilidade dos mesmos. A plataforma denominada como GLOBAL CLÍNICA oferece a opção dos utilizadores ao atendimento com segurança e sigilo, já que estamos falando sobre a saúde dos pacientes. A partir dos pressupostos de visibilidade e alcance de mercado, a plataforma propõe a facilidade de atendimento, controle de receitas, exames e atendimentos.

Palavras-chave: Saúde, usabilidade, atendimento, plataforma.

1 Introdução

O presente trabalho apresenta uma plataforma *on-line* de gerenciamento e atendimento médico deixando claro que a plataforma não é para atendimento a distância, ele é destinado a empresas no ramo de saúde e clínicas de atendimento

médico, de uma maneira ampla, porém muito intuitiva. O processo de atendimento pode se tornar algo exaustivo e consumir muito tempo de um médico caso seja feito de forma manual, por meio de fichas de atendimento e prontuários de papel, e lembrando que o(a) recepcionista, na maioria das vezes, depende de listas de chegada, separação e organização de prontuários no que acaba dificultando e atrasando o atendimento. Então, o objetivo da plataforma é auxiliar a demanda das clínicas fazendo com que tudo flua de forma rápida e interligada por meio da plataforma *on-line*, com ferramentas capazes de auxiliar no processo de atendimento de forma simples e segura. Atualmente há uma quantidade significativa de plataformas de atendimento, porém não possuem o foco ao atendimento médico principalmente se tratando de clínicas, pois, a Global Clínica possui lotações (cadastro de clínicas individuais) no qual cada clínica possui o seu usuário com todos os médicos cadastrados de forma individual. O intuito da plataforma é dar agilidade no processo de atendimento, cobrando um custo acessível para os possíveis interessados.

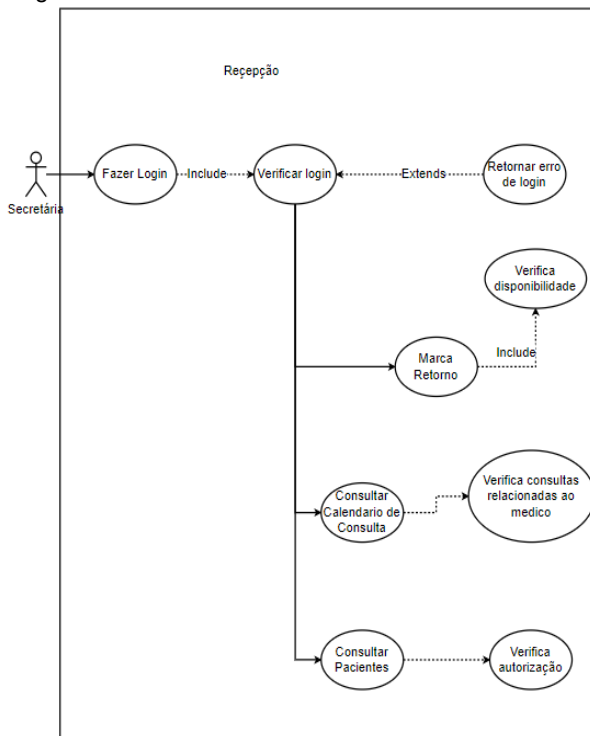
2 Materiais e Métodos

A elaboração de um *software* depende, primeiro, de boas ideias e planejamento daquilo que pode ser desenvolvido. O ideal é possuímos um planejamento muito bem estruturado e definido, para evitarmos problemas futuros.

Visando a estruturação, as ferramentas de modelagem foram utilizadas para representar a forma como o sistema irá funcionar. Elas também são utilizadas como um mapa para demonstrar de forma mais clara o fluxo dos processos. Para o

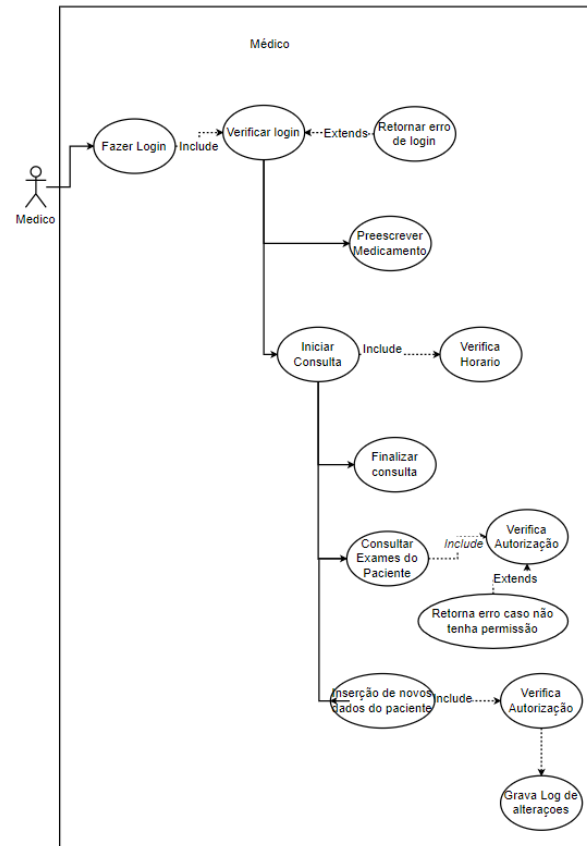
desenvolvimento da plataforma, objeto deste artigo, foi utilizado a linguagem UML, que de modo geral auxiliou no entendimento e na elaboração do sistema e da sua interface. As Figuras 1, 2 e 3 apresentam as etapas iniciais e essenciais para o desenvolvimento da plataforma, sendo ele, o diagrama de caso de uso, no qual são representadas as ligações e como funcionará cada etapa para cada usuário.

Figura 1 – Caso de uso



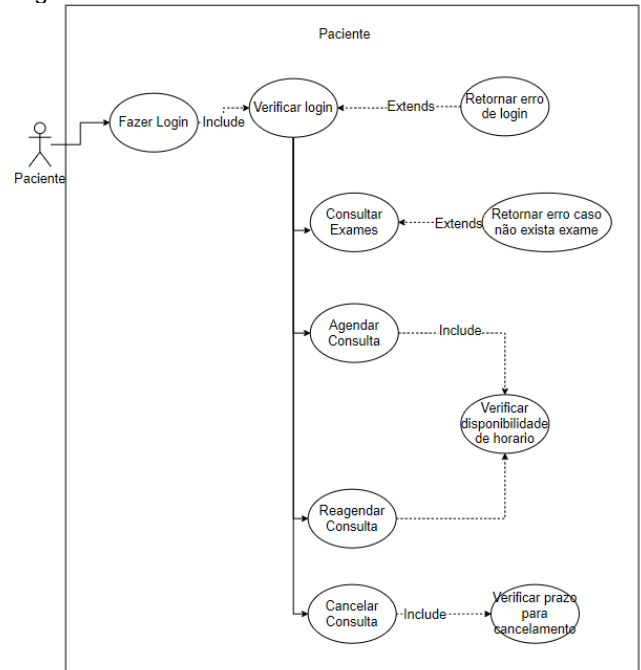
Fonte: Do autor, 2021

Figura 2 – Caso de uso



Fonte: Do autor, 2021

Figura 3 – Caso de uso



Fonte: Do autor, 2021

Possuindo o caso de uso, os passos seguintes à criação da plataforma tornam-se mais favoráveis, uma vez que foi definida uma referência mais próxima da realidade,

favorecendo a compreensão de todo o processo.

Dando sequência é pensar de forma que facilite a compreensão, o uso e a comunicação entre recepção, médico e paciente. Nesta pesquisa, as telas elaboradas para a criação do protótipo foram pensadas para facilitar a compreensão do usuário à utilização do sistema de forma limpa e agradável a fim de evitar a poluição visual. Para desenvolver o protótipo do *software*, é necessário reunir os conhecimentos sobre atendimento, demanda de pacientes, lógica de programação e conhecimentos específicos da área de saúde, ou seja, conhecimentos. É necessário definir uma sequência de funções que o sistema deverá realizar, de forma resumida seria mapear as funções necessárias, considerando os passos do processo de cadastros, atendimento e visualização dos atendimentos e a capacidade de resposta do programa, o que determinou a necessidade da elaboração do fluxograma apresentado anteriormente.

Dando sequência, estruturou-se a ideia do sistema nomeado como Global Clínica, constituído de *login* de acesso, sendo ele da recepcionista, médicos e pacientes, cadastro da clínica, cadastro de pacientes, busca de pacientes cadastrados, visualização de prontuários/atendimentos, realização de exames e de receitas. O sistema ainda possibilita a visualização do histórico de atendimentos. O desenvolvimento das telas de cadastro, será inicializada a partir da elaboração do processo de cadastros. Para isso, foi necessário o desenvolvimento de uma base de dados com os cadastros de pacientes e usuários por recepcionistas ou usuários administradores (médicos, gerente de clínicas).

O banco de dados escolhido para o desenvolvimento foi o banco MySQL, devido ao fato dele ser gratuito, possuir consistência, alta performance, confiabilidade além de ser fácil de usar.

Várias linguagens possuem acesso ao banco de dados MySQL:

- C, C++, Java
- PHP, Python, Perl
- Ruby
- .NET, Oracle Forms, Microsoft ASP
- Delphi, Kylix
- LISP, Lasso, Pike, TCL, Guile, REXX, Dylan (LINKOFICIAL, 2017).

Para o desenvolvimento da plataforma, foi utilizado no *Front-end* o *framework* Laravel, esse *framework* proporciona um desenvolvimento de forma mais organizada, utilizando os melhores conceitos de orientação a objetos, além do processo de desenvolvimento ser mais rápido devido sua estrutura, que por sua vez, tem como suas principais funções o desenvolvimento ágil, uma boa legibilidade do código e facilidade na integração com diversos serviços como, armazenamento na nuvem, envio de *e-mails*, notificações em tempo real, etc.

É uma coleção de vários elementos e funções personalizáveis para projetos da web de código aberto (opensource), empacotados previamente em uma única ferramenta. O container de serviços do Laravel é uma ferramenta poderosa para gerenciar dependências de classes e executar injeção de dependência. A injeção de dependências é uma frase sofisticada que essencialmente significa isso: as dependências de classe são “injetadas” na classe por meio do construtor ou, em alguns casos dos métodos “setter”. (OTWELLI, 2019 apud CAVOLI; FURTADO, 2019, p. 5).

Para Citrus7 (2017, p. 1) o “*Front-end* é a primeira camada com a qual nos deparamos ao acessarmos um *site*, uma intranet ou mesmo um sistema *web*”.

O padrão MVC¹ É o utilizado no desenvolvimento do *framework* Laravel, no qual as regras de negócio ficam bem divididas e organizadas da maneira correta.

¹ MVC (Model View Control) “facilita a troca de informações entre a interface do usuário aos dados no banco”. (ZUCHER, 2020, p. 1).

A arquitetura MVC auxilia os desenvolvedores a construir aplicações separando seus principais componentes, a manipulação e armazenamento dos dados, as funções que irão trabalhar com as entradas dos dados e a visualização do usuário. A arquitetura MVC específica onde cada tipo de lógica deve estar localizado na aplicação. (LEMOS *et al.*, 2010, p. 17).

Outra funcionalidade que ele nos oferece é o trabalho utilizando Migrations², facilitando assim o controle e versionamento do banco de dados, em que qualquer mudança que ocorra na estrutura do banco deve ser documentada em uma *migration*.

A escolha da linguagem PHP foi devida ser uma linguagem voltada para desenvolvimento *web*, facilitando assim a criação de *sites* e favorecendo a conexão entre servidores e a interface do usuário.

O PHP é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML. (PHP, 2021, p. 1).

Como o PHP é uma linguagem de código aberto, facilita que o código seja melhorado constantemente. Não é uma linguagem difícil de se aprender, sendo considerada uma das linguagens mais fáceis de se programar. E por se tratar de uma aplicação que trabalha com cadastros, o PHP além de ter um alto desempenho, também suporta uma grande quantidade de dados, conseguindo executar múltiplas funções e recursos ao mesmo tempo. Além de ser compatível com os principais bancos de dados da atualidade.

Entre os principais bancos de dados utilizados, o PHP é compatível com:

- Oracle
- MySQL
- Interbase

- SQLite
- Sybase

Para a criação da API foi utilizado o *framework* Lumen, que é um *micro framework* baseado em Laravel, sendo uma API, um conjunto de regras que compõem uma interface, possibilitando a comunicação entre diversos tipos de aplicações, seja ela *mobile*, *desktop* ou *web*, fazendo com que elas se comuniquem entre si ao consumir essa API.

De forma simplificada uma API recebe uma requisição que vem de uma aplicação, processa a requisição feita pela aplicação fazendo chamadas no banco de dados e recebendo o retorno, logo após devolve um resultado para a aplicação. Podemos comparar a API com um garçom (API), que recebe um pedido do cliente (aplicação), leva esse pedido até a cozinha para preparo (Banco de dados), após o pedido ficar pronto o garçom (API) pega o prato e leva novamente para o cliente.

O Lumen possui uma gama de funcionalidades que auxiliam e controlam de forma muito confiável as requisições realizadas pelas aplicações que consomem a API, ela utiliza o método de autenticação *Bearer Token*, onde um *token* é enviado no cabeçalho da requisição HTTP validando o acesso do usuário na API. Esse método de validação por *token* é utilizado, pois, não é possível e nem recomendado em uma API guardar sessão de usuário assim como fazemos nos navegadores, então quando um usuário realiza uma requisição, ele recebe um *token* e assim faz a verificação e o mesmo, autoriza suas requisições na API.

3 Resultados

O desenvolvimento do trabalho possibilitou avaliar algumas das linguagens e APIs e, assim, aplicar as que atendessem às necessidades da aplicação.

Tendo o desafio de construir todo um planejamento tecnológico partindo pela busca das melhores propostas e usar as que

² "Migrations trabalha na manipulação da base de dados: criando, alterando ou removendo e também manter um histórico de alterações que a base de dados vai sofrendo ao longo do tempo."

(EDUARDO, 2016, p. 1).

fossem melhor para o desenvolvimento do *software*, visto que existem uma extensa quantidade de linguagens e APIs, e cada uma delas possuindo características próprias. O desenvolvimento deste artigo se deu no momento que o mundo passava por uma pandemia de Covid-19 e, neste período, muitas pessoas ficaram receosas de sair de suas casas e, com essa aplicação, podemos facilitar que o usuário tenha acesso e agende o horário de sua consulta ou retorno de acordo com a necessidade, localização, convênio e disponibilidade do médico.

4 Discussão

O desafio para desenvolver a plataforma foi a estruturação para efetuar de forma limpa a ligação entre todas as funções levando em consideração que cada parte depende de um “pai” para dar sequência a outros processos. Quando falamos de inovação, na área de saúde existem diversos paradigmas que devem ser quebrados, afinal são dados e informações que serão confidenciais que deverão ser mantidas em nuvem. Por se tratar de dados pessoais, podem conter informações das quais o paciente queira manter em sigilo até mesmo de pessoas próximas, por isso manter esses dados em segurança é primordial, tanto de invasores que podem obter esses dados mas também por conta da intimidade do paciente. O acesso às informações é cada vez mais importante para a área de saúde. Muitas vezes precisamos mudar de médico e geralmente é necessário o novo médico fazer toda a análise novamente do paciente, tal fato pode tornar o retorno um pouco mais demorado do que deveria. Tendo em mãos, ou em um sistema, o médico pode utilizar o relatório de saúde de seu paciente, pedir exames específicos e não pedir certos tipos de exames que podem constar no cadastro do cliente como realizados com outro médico. As consultas aconteceriam de forma mais rápida, pois o paciente consegue visualizar as datas disponíveis e os médicos que atendem nesta determinada data sem a necessidade de estar ligando na clínica ou

até mesmo indo ao local evitando assim filas e aglomerações, levando em consideração que os agendamentos são realizados remotamente o paciente receberá notificações caso a consulta seja confirmada ou não. Essa aplicação pode alcançar um nível muito mais rápido de atendimento, pois todos os dados do cliente estariam prontos e atualizados, como, por exemplo, plano de saúde, endereço, contato, data da última consulta e procedimentos realizados e até mesmo problemas de saúde como hipertensão, diabetes, etc... no qual o médico que faria a consulta poderia obter os dados necessários de forma rápida e sem precisar pedir uma nova bateria de exames. O paciente também poderá agendar suas consultas sem precisar ligar na clínica ou consultório, tudo seria feito de forma muito prática e ágil, por meio do aplicativo que traria essa funcionalidade para o paciente. Tarefas como retornos seriam marcados automaticamente e caso o paciente não tenha disponibilidade de comparecer nesta data o mesmo pode realizar a mudança de dia e horário.

5 Conclusão

Neste trabalho foram abordadas características de uma aplicação, e foi desenvolvido um sistema para gerenciamento de Atendimento Médico como foco principal o desenvolvimento de uma aplicação *web*, no qual foi possível aplicar as técnicas e linguagens de programação para internet, para que fosse realizado todo o acesso de forma *on-line*, mas não a consulta. De modo a facilitar o acesso, sejam eles administradores, pacientes, analistas ou quaisquer outros usuários dessa plataforma.

6 Referências

CAVOLI, Diego da Silva; FURTADO, Ludmila Breder. Support system: sistema de gerenciamento de atendimentos. 2019. 25 f. Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) - Centro Universitário de Manhuaçu, 2019.

CITRUS7. **O que é front-end e back-end?** São Paulo: [s. n.], 2021. Disponível em: <https://citrus7.com.br/artigo/o-que-e-front-end-e-back-end/>. Acesso em jun. 2021.

EDUARDO, Carlos. **Migrations:** o porque e como usar. São Paulo: Medium, 2016. Disponível em: <https://juniorb2s.medium.com/migrations-o-porque-e-como-usar-12d98c6d9269>. Acesso em: maio 2021.

FERREIRA, Kellison. **O que é PHP e por que você precisa conhecer essa linguagem de programação web.** Belo Horizonte: Rockcontent, 2019. Disponível em: <https://rockcontent.com/br/blog/o-que-e-php/>. Acesso em: jun. 2021.

LINKOFICIAL. **MySQL:** o que é, para que serve e suas vantagens. Vargem Grande do Sul: Link Oficial, 2017. Disponível em: <https://www.linkoficial.com.br/mysql-o-que-e-para-que-serve-e-suas-vantagens>. Acesso em: out. 2019.

LEMOS, Maxmilian Ferreira de *et al.* **Aplicabilidade da arquitetura MVC em uma aplicação web (WebApps).** 2013. 17 f. Trabalho de Conclusão de Curso (graduação em Computação) Universidade José do Rosário Vellano, 2013. Disponível em: <http://revistas.unifenas.br/index.php/RE3C/article/viewFile/54/13>. Acesso em: jun. 2021.

LONGEN, Andrei Silveira. **O Que é PHP?** Hostinger: Weblink, 2019. Disponível em: <https://www.weblink.com.br/blog/php/o-que-e-php-conheca/>. Acesso em: maio 2021.

PHP. **O que é o PHP?** [S. l., s. n.], 2021. Disponível em: https://www.php.net/manual/pt_BR/intro-what-is.php. Acesso em: jun. 2021.

PROGRAMADORA, Galinha. **Um pouco do framework PHP Laravel e sua estrutura de diretórios.** Rio de Janeiro:

Galinha Programadora, 2019. Disponível em: <https://www.galinhaprogramadora.com.br/2019/05/um-pouco-do-framework-php-laravel-e-estrutura-de-diretorios.html>. Acesso em: maio 2021.

SANTOS, Evandro; VASCONCELOS, Rodrigo Erasmo. Sistema de Gestão e Controle de Normas Regulamentadoras que auxilia os Profissionais de SESMT. 2019, 15 f. Trabalho de Conclusão de Curso (Análise de Desenvolvimento de Sistemas) Universidade Tecnológica Federal do Paraná, 2019. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/12601>. Acesso em: jun. 2021.

ZUCHER, Vitor. **O que é padrão MVC?** Entenda arquitetura de softwares! Paris: Le Wagon, 2020. Disponível em: <https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>. Acesso em: jun. 2021.