

SOFTWARE DE GESTÃO PARA HOTEL GERIÁTRICO (ILPI)

Eduardo Nunes Prado de Melo¹; Maria Carolina Vieira Novais²; Paulo Sérgio Costa Magalhães³;

Paulo Limirio da Silva⁴

1, 2, 3, 4 Universidade de Uberaba

eduardonunes@edu.uniube.br; paulo.limirio@uniube.br;

Resumo

Este trabalho tem como objetivo apresentar um *software* de gestão para Instituições de Longa Permanência para Idosos (ILPI) com a intenção de proporcionar aos hotéis que aderirem aos serviços do *software* um controle de gestão e logística de qualidade, para compor um ótimo funcionamento dos hotéis em benefício dos clientes e pacientes. Assim, como aproximar os familiares de seus entes, via interface *online* que o *software* oferece para interação e análise da qualidade atribuída para com os residentes do local.

Palavras-chave: Administração; Monitoramento; ILPI;

1 Introdução

A presença familiar e de amigos para aqueles que residem em um hotel ILPI sempre se fez necessário, especialmente para aqueles que se encontram em circunstâncias mais solitárias. A qualidade atribuída a esses residentes também deve ser revisada e aprimorada para que assim possam receber um melhor atendimento.

Este hiato acadêmico motivou a iniciativa para criar esse projeto, sendo importante ressaltar a visão dos autores para com o tema e discussão em questão. Visão está em atingir qualidade e excelência no atendimento àqueles que residem em hotéis ILPI, aproximar a família mesmo estando em espaço e tempo diferentes através da tecnologia e proporcionar a essas empresas aprimoramento na área de gestão e logística com maior facilidade. Segundo Magalhães e Lemgruber (2017), o Brasil, assim podemos considerar que, ainda está se desenvolvendo em relação ao tema em questão de acordo também com a psicologia nas ILPI e o envelhecimento bem-sucedido.

A solução para o problema da solidão dos idosos e daqueles que optam por esse serviço como também a resolução da gestão das empresas ILPI. A pesquisa promove a difusão do tema, estabelecendo os critérios básicos para a seleção de amostra.

Como primeiro trabalho de pesquisa a tratar do tema deferido, acreditamos que o pioneirismo pode incentivar futuros alunos e empreendedores a se interessarem por explorar o tema fortemente presente no nosso cotidiano.

O objetivo geral deste trabalho foi desenvolver um *software* de gestão para ILPI com a intenção de proporcionar a essas empresas que aderiram aos serviços do *software* um controle de gestão e logística de qualidade, para compor um ótimo funcionamento dos hotéis em benefício dos clientes e pacientes. Assim, como aproximar os familiares de seus entes, via interface *online* que o *software* oferece para interação e análise da qualidade atribuída para com os residentes do local.

2 Materiais e Métodos

No processo de desenvolvimento do software em questão utilizou-se para *backend* o *framework Laravel* através do editor *PhpStorm* licenciado pela Uniube para estudantes e *WAMPSEVER* como servidor local de *php*. Como gerenciador de dependências o *COMPOSER* e no *PhpMyAdmin* foi criado um banco de dados. Segundo Douglas e Marabesi (2017, p. 1), no *E-books Aprendendo Laravel: O framework PHP dos artesãos da web*, traz a nota sobre *Composer*:

Primeiro, se não utiliza o Composer em seu projeto, posso dizer que você tem coragem. Hoje em dia, a maioria, senão todas as bibliotecas PHP, tende a estar no controle do Composer, que nada mais é do que uma ferramenta de gerenciamento de dependências em *PHP*.

PhpMyAdmin é uma ferramenta de *software* livre escrita em *PHP* que se destina a lidar com a administração de um servidor de banco de dados *MySQL* ou *MariaDB*, para o desenvolvimento desse projeto foi utilizado o *MySQL* que é um banco relacional. Pode-se usar o *phpMyAdmin* para executar a maioria das tarefas de administração, incluindo a criação de um banco de dados, a execução de consultas e a adição de contas de usuários. *Laravel* é uma estrutura pronta de arquivos e códigos, por via muito organizada que se pode utilizar para a linguagem *PHP*. Proporcionando produtividade e escalabilidade com uma proposta de se trabalhar com um código elegante. Com o *Laravel* é possível criar

aplicações robustas para o web seguindo uma linha simples, embora ele seja rico em recursos avançados. O *framework* possibilita aplicações para *backend* também *frontend* e para criações de *APIs* e oferece um suporte para desenvolver projetos de grande porte e complexos. Laravel é além de um *framework*, existe todo um conjunto, ecossistema, recursos e ferramentas disponíveis pagos e gratuitos que fazem parte do universo do Laravel.

A partir da versão 8.0 do *Laravel*, foi iniciado a parte de *Kits* de autenticação, o que nos possibilita autenticação *Breeze*, que é simplificada e *Jetstream*, uma autenticação de dois fatores o que o torna mais completo.

No desenvolvimento utilizamos a autenticação *Breeze*. Para rodar os comandos *npm* foi instalado o *NodeJS*.

Alguns dos diretórios que contém no *Laravel* são as pastas de autenticação, componentes, *layouts*, *app* e também os seus recursos que são: *css*, *js*, *lang* e *views*. Também foi utilizado um código para que a tela de login e cadastro esteja em português.

Para *frontend* utilizou-se o *framework* progressivo e reativo *Vue.js* para desenvolvimento web que foi adotado pelo *Laravel*, ele é focado na construção de *interfaces* do usuário, o código *html*, *css* ou *javascript* que é utilizado para manipular a interface dentro da *Vue.js*. Um *framework* progressivo é uma ferramenta que possibilita ser utilizada como Biblioteca ou *framework*. O *Vue.js* é um *framework* recente, lançado em fevereiro de 2014. As suas vantagens são: a redução de complexidade, ele possui a menor curva de aprendizado entre os *frameworks* de sua categoria, sendo mais simples, fácil, leve e flexível para executar as mesmas tarefas; documentação detalhada, possui uma ótima documentação e também na versão em português; a flexibilidade e adaptabilidade, o *Vue.js* faz uso de templates como experiência padrão mas também contém o suporte a *Render Functions* e até mesmo ao *LSX* como *react*, o que facilita a transição dos desenvolvedores habituados com outros *frameworks*, ou seja, o *Vue* é receptivo para os devs com experiências menores e flexível para com maiores experiências, possibilita criar os componentes utilizando *Pug*, *Sass*, *PostCss*, *Stylus*, *Babel*, *CoffeeScript*, *Typescript* entre tantos, assim permite adaptar o *Vue* para vários tipos de equipes e projetos; multiplataforma, pode-se utilizar o *Vue* no *Mobile* para criar aplicações híbridas e nativas, no *desktop* e web; comunidade para a comunidade pois não é mantida por uma grande organização.

Framework reativo diz respeito a sua arquitetura que atualiza a camada visual cada vez que tem uma mudança de estado, sendo assim, o *html* será atualizado toda vez que o valor

de uma variável de um componente sendo *view* for alterada. a reatividade dos *frameworks front-end* traz uma interação fluida e instantânea entre o usuário e aplicação. De acordo com Incau (2017, p. 4), no *E-book Vue.js: construa aplicações incríveis*, ele nos diz:

O *Vue.js* nos oferece um *CLI (Command Line Interface)* que nos ajuda a gerar nossos projetos. Eles nos proveem *templates* para diversos *workflows* diferentes. em poucos minutos, podemos ter uma aplicação rodando com diversas tarefas predefinidas, com *live-reload* e *linter*.

Para atualizações em tempo real utilizou-se o *Websocket*, que é um protocolo que torna possível a comunicação bidirecional entre um navegador web e o servidor, permite o transporte contínuo das informações entre os dois. faz parte das especificações do *html5* e é suportado pelas últimas versões dos navegadores.

A câmera *WebCam* foi utilizada para as simulações nos testes para verificar o funcionamento do *software* no seu módulo de monitoramento, logo as câmeras *DVRs* e *NVRs* são indicadas para os hotéis que aderirem ao *software* que deseja ativar o monitoramento. As câmeras *DVRs* são conectadas diretamente ao gravador, contudo o sistema *NVR* exige apenas que cada câmera se conecte à mesma rede lógica.

A forma de pagamento poderá ser realizada por boletos, depósitos bancários, transferência por *PIX* e por cartão de crédito, fica por escolha do hotel a plataforma de pagamento a oferecer para o cliente.

3 Resultados

O desenvolvimento do trabalho possibilitou avaliar as linguagens e APIs e, assim, adotar as que atendessem às necessidades da aplicação. Tendo o desafio de construir todo um planejamento partindo pela busca das melhores propostas e usando as que fossem melhores para o desenvolvimento do *software*.

O que auxiliou muito foi a linguagem *PHP* que é uma linguagem de código aberto, facilita que seja melhorado constantemente sendo uma linguagem fácil de se aprender e programar. Por se tratar de uma aplicação que também 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, o que colaborou bastante com o engajamento do aditamento do projeto.

Ao utilizarmos no *backend* o *framework Laravel*, este proporcionou um desenvolvimento de forma mais organizada, contribuindo nos 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.

4 Discussão

O desafio para desenvolver a plataforma foi a estruturação para efetuar de forma limpa a ligação entre todas as funções combinadas para a apresentação na banca, levando em consideração que cada parte depende de uma base hierárquica para dar sequência a outros processos.

Quando falamos de inovação, existem diversos paradigmas que devem ser rompidos nesse momento, afinal são dados e informações que serão armazenados e administrados que devem ter a sua pontualidade. Por se tratar de dados pessoais, como imagens dos pacientes e colaboradores do hotel, informações dos remédios corretos para cada idoso, a dieta correta, os horários de cada tarefa, como também toda parte administrativa e burocrática que o *software* facilitará para a empresa que o adquirir é necessária atenção para que um bom funcionamento seja executado.

Assim, outro ponto que se apresentou como desafiador foi a conexão das câmeras e como teste foi utilizado o *WebCam*, mas os primeiros testes obtiveram resistência para o seu funcionamento, porém com outras tentativas e pesquisas foi possível realizar a simulação do monitoramento com excelência. A conclusão do *software* está estipulada para o ano de 2023, pela questão do projeto ser mais complexo em sua totalidade proposta.

5 Considerações Finais

Em relação ao *software* desenvolvido, foi significativo ver seu resultado dentro de um prazo estipulado. Sendo assim, a base de conhecimento em relação ao processo de desenvolvimento de *software* para obter o desejado é de grande importância e principalmente pela sua manutenção e não sendo simplesmente uma codificação. Além de que nesse tempo, algo novo sempre é aprendido, como por exemplo, simular um monitoramento por *WebCam* em uma sala de aula *online*. Em relação ao tema, percebe-se a importância e necessidade de se administrar bem os dados e registrá-los com precisão para a segurança dos pacientes e para garantir a integridade do hotel.

Referências

DOUGLAS, Michael; MARABESI, Matheus; **Aprendendo Laravel**: o framework PHP dos artesãos da web. 1. ed. São Paulo: Novatec, 2017. Disponível em: https://play.google.com/store/books/details?id=xTY8DwAAQBAJ&rdid=book-xTY8DwAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport. Acesso em: out. 2021.

INCAU, Caio. **Vue.js**: construa aplicações incríveis. São Paulo: Casa do Código, 2017.

MAGALHÃES, Paula Raquelyne; LEMGRUBER, Karla Priscilla. Atuação do psicólogo nas instituições de longa permanência para idosos. **Revista Psicologia e Saúde em Debate**, Patos de Minas, v. 3, n. Supl. 1, p. 52-53, 2017. Disponível em: <http://www.psicodebate.dpgpsifpm.com.br/index.php/periodico/article/view/114>. Acesso em: out. 2021.