UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

PEDRO HENRIQUE SERPA IDA

SISTEMA PARA CASA DE ACOLHIMENTO

MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

GUARAPUAVA
2020

PEDRO HENRIQUE SERPA IDA

SISTEMA PARA CASA DE ACOLHIMENTO

Monografia de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Guilherme da Costa Silva
Universidade Tecnológica Federal do Paraná - Câmpus Guarapuava

Coorientador: Prof. Dr. Roni Fabio Banaszewski
Universidade Tecnológica Federal do Paraná - Câmpus Guarapuava

GUARAPUAVA
2020

Dedico este trabalho a Universidade Tecnológica Federal do Paraná - Campus Guarapuava e aos professores e servidores da instituição.

AGRADECIMENTOS

Agradeço em primeiro lugar a minha família por ter me dado forças e motivação em toda minha caminhada acadêmica.

Agradeço aos meus amigos e colegas por sempre estarem presentes ajudando e motivando das mais diversas formas para que eu chegasse até aqui.

Agradeço também ao Universo por ter me ajudada a ver soluções em problemas nos momentos difíceis e ter clareza os meus objetivos.

Agradeço a todos os professores da Universidade Tecnológica Federal do Paraná, em especial ao meu orientador Guilherme da Costa e Silva, que contribuiu na realização deste trabalho.

O mais competente não discute, domina a sua ciência e cala-se. (Voltaire).

RESUMO

IDA, Pedro. Sistema para Casa de Acolhimento. 2020. 24 f. Monografia de Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2020.

A gestão de casas de acolhimento ou lugares que possuem o mesmo fim, de ajudar pessoas que não possuem um local para permanecer por um curto período de tempo, seja por necessidade médica ou algo do gênero, ainda é feita de maneira manual em blocos de notas ou planilhas virtuais, tornando esse processo obsoleto no tempo em que vivemos. Desta forma, buscando suprir essa necessidade de um método de gerenciamento mais prático, dinâmico e centralizado, esse trabalho realiza o desenvolvimento de um software para tal fim. Para isso é desenvolvido um sistema que gerencia pessoas acolhidas, suas estadias e consumos durante o período de estadio, controle de doações e refeições cedidas diariamente.
Palavras-chave: Casa de acolhimento. Programa. Software. Gerenciamento.

ABSTRACT

IDA, Pedro. Title in English. 2020. 24 f. Monografia de Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2020.

The management of a host house or similares places that have the same objective, help people that do not have a place to stay for a short time, whether by clinical necessity or something else, still is doing by manual process with notepads or virtal spreadsheets, making this process obsolete in the time we live. So, seeking to supply this necessity of a practical, dynamic and centralized management method, this project proposes the development of a software for this reason. For this, will be developed a system that manage hosted people, the stays and they consumption during the stay period, donation control and meals provided daily.

Keywords: Host House. System. Software. Management.

LISTA DE FIGURAS

Figura 1 – Tela principal site IDES 4
Figura 2 – Nova reserva na plataforma Hqbeds 5
Figura 3 – Listagem de pacientes SisHOSP 5
Figura 4 – Metodologia ágil Scrum 7
Figura 5 – Linguagens mais populares 2015-2019 8
Figura 6 – Frameworks PHP 9
Figura 7 – Fluxo web 10
Figura 8 – Diagrama de Entidade Relacionamento 15
Figura 9 – Criação de novo usuário 16
Figura 10 – Lista de origens 16
Figura 11 – Criação de nova origem 16
Figura 12 – Criação de novo acolhido 17
Figura 13 – Criação de nova estadia 18
Figura 14 – Lista de refeições diárias 18
Figura 15 – Criação de nova refeição diária 18
Figura 16 – Dashboard 19

LISTA DE QUADROS

Quadro 1 – Requisitos não funcionais. 12
Quadro 2 – Requisitos funcionais. 13
Quadro 3 – Cronograma de atividades. 20

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
APOO	Análise E Projeto Orientado A Objetos
CSS	Folhas de Estilo em Cascata (do inglês Cascading Style Sheets)
HTML	Linguagem de Marcação de Hipertexto (do inglês Hypertext Markup Language)
PHP	Pré-processador de hipertexto (do inglês Hypertext Preprocessor)
SQL	Structured Query Language

SUMÁRIO

1 – INTRODUÇÃO **1**

2 – OBJETIVOS **2**

2.1 OBJETIVO GERAL 2

2.2 OBJETIVOS ESPECÍFICOS 2

3 – REVISÃO BIBLIOGRÁFICA **3**

3.1 SISTEMAS SIMILARES 3

3.2 REFERENCIAL TEÓRICO 6

4 – METODOLOGIA **11**

5 – DESENVOLVIMENTO **12**

5.1 LEVANTAMENTO DE REQUISITOS 12

5.2 MODELAGEM DO BANCO E PROTOTIPAÇÃO 14

5.3 DIVISÃO DAS TAREFAS 19

5.4 ESTUDOS DAS TECNOLOGIAS 19

6 – PLANEJAMENTO DO TRABALHO **20**

6.1 CRONOGRAMA 20

6.2 RECURSOS NECESSÁRIOS 20

7 – ANÁLISE E DISCUSSÃO DOS RESULTADOS **21**

7.1 REALIZAÇÃO DA COLETA DE REQUISITOS 21

7.2 PRINCIPAIS DADOS APÓS COLETA 21

7.3 APERFEIÇOAMENTOS NOS CONHECIMENTOS DAS TECNOLOGIAS . . 21

8 – CONCLUSÃO **22**

8.1 TRABALHOS FUTUROS 22

Referências **23**

1 INTRODUÇÃO

Atualmente, não é mais viável que a gestão de pessoas e/ou locais seja feita de maneira manual, seja em papel ou em planilhas, estas formas de controle, além de serem demoradas e exigirem atenção extra, podem gerar inconsistências em dados sensíveis, por exemplo, por meio da inserção de um valor errado.

Na área social, essa necessidade de uma gestão bem feita e organizada não é diferente, e em muitos casos ainda é feita de modo manual, pois não há oferta de sistemas no mercado que atendam a essa demanda mais específica. O que há disponível hoje, são sistemas mais robustos de gestão que atendem, primordialmente, empresas e indústrias. Dentro da área social, mais especificamente as casas de acolhimento, existem necessidades diferentes das oferecidas pelos sistemas de gestão existentes, por exemplo, a gestão de pessoas não é feita para gerir funcionários, suas atribuições, salários, etc, ao contrário, é utilizada para fazer o controle de pessoas que foram acolhidas na casa e pessoas que são voluntárias.

As casas de acolhimentos, alvos deste projeto, são locais disponibilizados para pessoas que, por algum motivo, não possuem um lugar para ficar durante um período de tempo. Por exemplo, o acompanhante de um paciente hospitalar que não pode pernoitar no hospital ou, ainda, convidados de algum evento religioso que não possuem onde passar a noite. Nessas situações, as pessoas, vindo de origens diversas, são orientadas a procurar uma casa de acolhimento, que irá ceder um local para dormir, as refeições diárias e o que mais estiver disponível com base nas suas necessidades. Estes locais, que em sua maioria possuem como financiadores, filantropos e/ou instituições religiosas, tem o objetivo de ajudar o próximo. Ainda, não é necessária uma automação ou controle mais tecnológico destes locais, porém o acúmulo de registros de hospedagens, compras, etc, em livros ou outros meios é obsoleto, logo um programa pode trazer ganho, ajudando na prestação de contas ou na angariação de fundos.

A proposta deste trabalho é preencher essa lacuna, com o desenvolvimento de um programa, ajudando no controle dos processos gerenciais desses locais, a fim de trazer mais integridade, segurança aos dados e praticidade, com a premissa que os usuários desse programa são pessoas, em sua maioria, sem conhecimentos práticos de informática, o uso deverá ser intuitivo e simples. Este requisito é o principal desafio para o desenvolvimento do programa, porque algumas funções ou interações que são claras para pessoas que têm mais contato com informática, certamente não são tão claras para quem pouco usa sistemas informatizados no seu dia a dia.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver um programa para auxiliar na administração de uma casa de acolhimento.

2.2 OBJETIVOS ESPECÍFICOS

- Realizar o levantamento e análise dos requisitos necessários ao desenvolvimento do programa proposto;
- Analisar técnicas, padrões e ferramentas para serem utilizadas no desenvolvimento do programa;
- Desenvolver modelos e diagramas, de forma a dar forma e estrutura ao programa proposto;
- Gerenciar e executar o desenvolvimento do programa utilizando princípios da metodologia ágil;
- Obter conhecimento na utilização de planilhas para importação de dados em programas.
- Implementar/codificar o programa, privilegiando o simples e intuitivo para solução do problema;
- Implementar de testes automatizados para garantir estabilidade no programa;
- Implantar o programa, de forma a corrigir análises e usos projetados.

3 REVISÃO BIBLIOGRÁFICA

Até o presente momento, não foram encontrados sistemas gestores ou similares que sejam diretamente ligados ao objetivo de administrar uma casa de acolhimento ou lugares que tenham mesmo propósito.

Os programas que se aproximam do contexto atual desta proposta são destinados, em sua maioria, a atender Casas Lar, que são abrigos para crianças, adolescentes, mulheres que não possuem um local para viver devido a agressões sofridas em seus antigos lares, abandonados pelos pais ou outros cujas circunstâncias inviabilizam que voltem para seu lar. A diferença entre os programas desses locais e o desta proposta é que aqueles são focados em mostrar o local do abrigo, mostrar seus programas e receber doações para o financiamento do local, diferentemente deste que precisa ser focado na gestão e controle interno do local. A seguir, na próxima seção, são apresentados alguns programas que possuem funcionalidades que se assemelham às necessidades que o programa proposto deva atender.

3.1 SISTEMAS SIMILARES

Um programa que pode ser relacionado à proposta neste trabalho é o da Irmandade do Divino Espírito Santo (IDES, 2020), uma instituição religiosa que tem como foco o auxílio de crianças, adolescentes, jovens e suas famílias. O programa possui um fórum para o conhecimento do que é feito pela instituição, programas presentes e um espaço de doações financeiras para o desenvolvimento e mantenimento dos programas. Ele possui uma página inicial que dá acesso a todos os recursos conforme a Figura 1.

Figura 1 – Tela principal site IDES
Fonte: IDES (2020)

Outro, o HQBeds (2020), é um sistema gestor de pousadas que tem o foco no gerenciamento de reservas de quartos, vendas e controle de hóspedes. Esse sistema possui diversas funções que suprem a necessidade de todo tipo de hospedagem, sua principal característica e função é o controle de clientes e hospedagens seguindo a Figura 2, mantendo histórico de hospedagens, clientes ativos, inativos e proibidos de novas estadias.

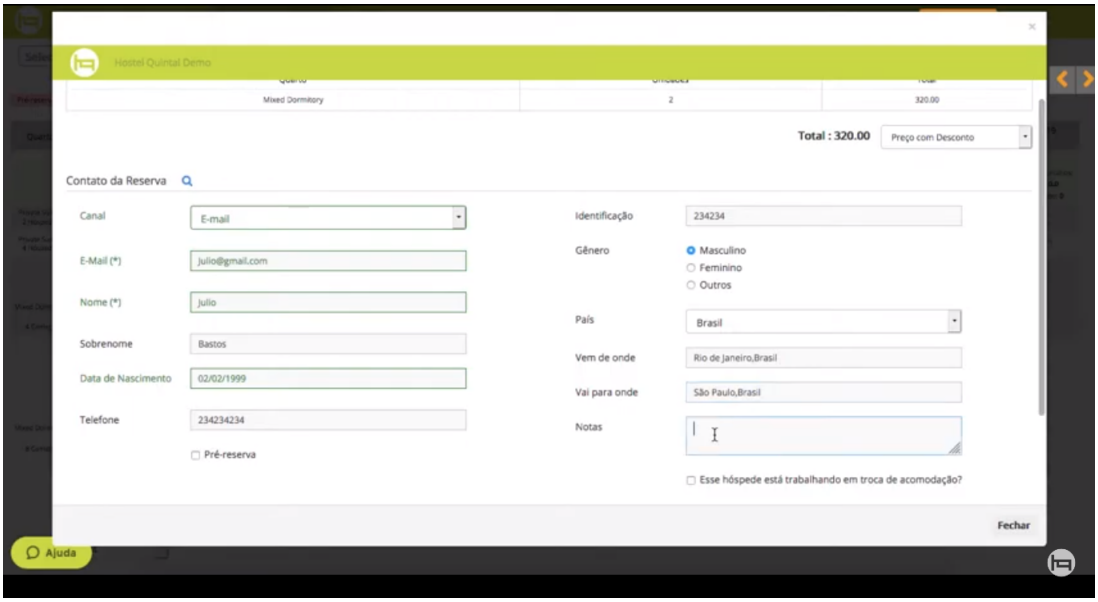

Figura 2 – Nova reserva na plataforma Hqbeds
Fonte: HQBeds (2020)

Na mesma ideia de casas de acolhimento, há um sistema voltado para casas de repouso de idosos, o SisHOSP (2020). Esse sistema possui muitas funções voltada para a área médica de cuidados com os idosos com prontuários, exames, contratos e afins. A similaridade com o projeto é o tratamento dos clientes como acolhidos no sistema e o controle dos mesmo, tendo o mesmo intuito de atender pessoas que necessitam de algum tipo de local ou cuidado específico, assim como pode ser verificado na Figura 3.

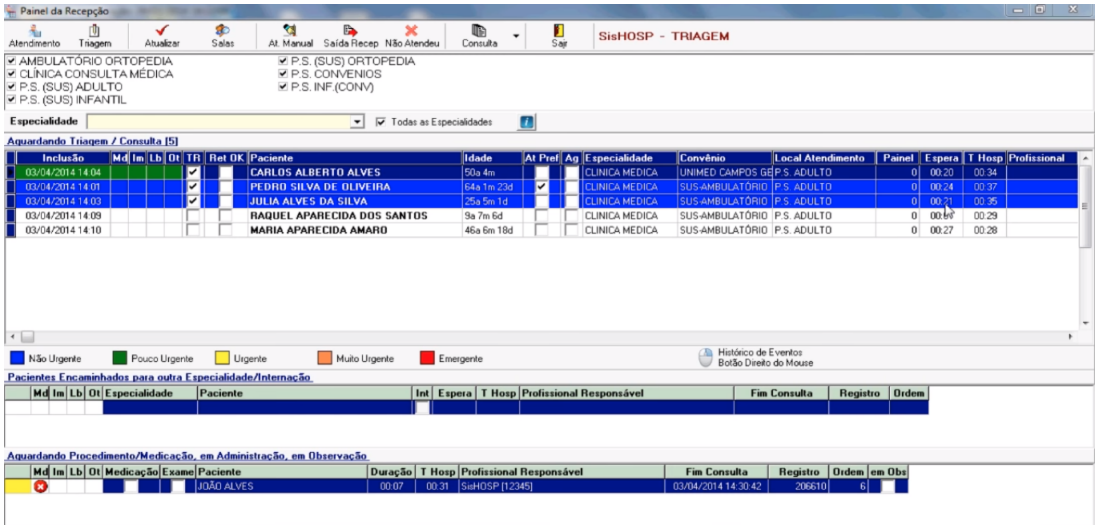

Figura 3 – Listagem de pacientes SisHOSP
Fonte: SisHOSP (2020)

Cada um dos programas citados acima possuem suas próprias funções e características para auxiliar nos processos referentes ao seu contexto. Nesse programa desenvolvido foi buscado

inspiração no site (IDES, 2020) para trazer o lado acolhedor e fraterno do programa, buscando a partir dele usar uma interface simples. A partir do programa da (HQBEDS, 2020) foi utilizado como base para a criação de estadias, a área de de hospedagens, abstraindo somente os campos relevantes para o programa. Do programa da (SISHOSP, 2020) foi utilizado a ideia do tratamento dos pacientes, que no caso são os acolhidos, para trabalhar com o controle dos mesmos de forma a trabalhar somente com dados estritamente necessários para realizar uma acolhida. Não foi utilizado inspiração em interface do último programa para essa área por conta da mesma ser muito poluída e demandar de conhecimentos aprimorados do programa para que seja utilizado.

3.2 REFERENCIAL TEÓRICO

No desenvolvimento de sistemas, necessita-se certa criatividade e proatividade, já que o objetivo é sempre replicar coisas do mundo real para o virtual, visando facilitar e/ou auxiliar processos que podem ser burocráticos e demorados. Para cada tipo de problema atacado, há diferentes formas de serem resolvidos com diferentes tecnologias, linguagens de programação e ferramentas. Como todo processo para que haja um resultado concreto, é preciso utilizar um padrão e uma metodologia. No desenvolvimento de sistema, são utilizadas metodologias para gerenciar sua execução.

Uma das metodologias mais utilizadas é o Scrum, que foi criada por Jeff Sutherland e Ken Schwaber. Segundo o artigo da DesenvolvimentoAgil (2020), o Scrum consiste em otimizar o processo de desenvolvimento de um projeto dividindo-o em pequenas entregas com prazos delimitados, seja quinzenal, semanal e assim por diante, intercalando com reuniões da equipe para alinhamento das tarefas. Essas entregas são chamadas de Sprints, que são subdivididas em entregas menores a fim de tornar os processos mais independentes e fracionados para que seja possível realizar entregas mais frequentes para o cliente. Todas as funcionalidades são mantidas em uma lista chamada de Product Backlog antes de serem inseridas em uma Sprint para serem desenvolvidas. Todos os passos do fluxo de trabalho do Scrum podem ser vistos conforme a Figura 4.

No desenvolvimento de aplicações monolíticas, em que camadas de interface e lógica estão no mesmo lugar, geralmente, é utilizada uma linguagem de programação para implementar a lógica das regras de negócio. Já para criar as interfaces de interação com o usuário há algumas possibilidades para se trabalhar, podendo realizar a junção de várias ferramentas ou utilizar somente uma como para as regras de negócios. No momento existem algumas ferramentas front-end que se destacam, como por exemplo ReactJS, VueJs, JQuery, que são extremamente completas e cada uma possui sua especialidades buscando o melhor resultado para o programa.

Na maior parte dos sites ou programas web presentes na internet, é utilizado o framework front-end Bootstrap, que utiliza HTML, CSS e Javascript. O objetivo dele é otimizar o desenvolvimento através da adoção de uma estrutura única de codificação de folhas de estilo pra diminuir inconsistências entre as formas de codificar que variavam de cada desenvolvedor.

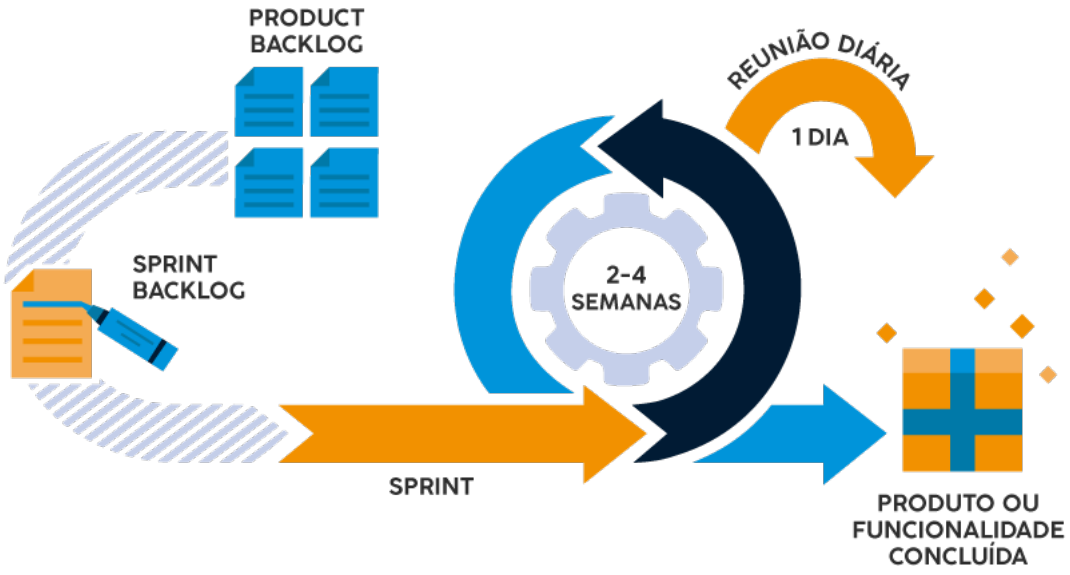

Figura 4 – Metodologia ágil Scrum
Fonte: Tecnicon (2020)

E a tentativa deu tão certo que eles perceberam o grande potencial da ferramenta, lançando-a no GitHub como um software open source (HOMEHOST, 2020).

O gerenciamento de requisições a um site web deve ser realizado por um servidor web, o qual mantém as páginas online para serem acessadas pelos usuários. Um dos servidores mais utilizados há tempos é o Apache, um servidor web open source mantido pela Apache Software Foundation e utilizado por mais de 45% dos sites da internet. Sua função é servir dados para os sites na internet. Para fazer isso, ele age como um intermediador entre o servidor físico e os computadores dos clientes (HOSTINGER, 2020).

No desenvolvimento web, umas das linguagens que mais se destaca por trazer bons resultados é o PHP. Ele foi criado em 1994 por Rasmus Lerdorf e se tornou muito utilizado no desenvolvimento pela pequena curva de aprendizagem e grande número de recursos que oferece. Ele é open source, geralmente utilizado junto com o HTML. Desta forma o PHP executado em um servidor possibilita a geração de conteúdo dinâmico que as páginas de um programa necessitam PHP (2020). Essa linguagem possui recursos muito completos que suprem as necessidades tanto de programas mais robustos, quanto sites básicos com menos lógica, por isso está entre as linguagens mais utilizadas conforme a Figura 5.

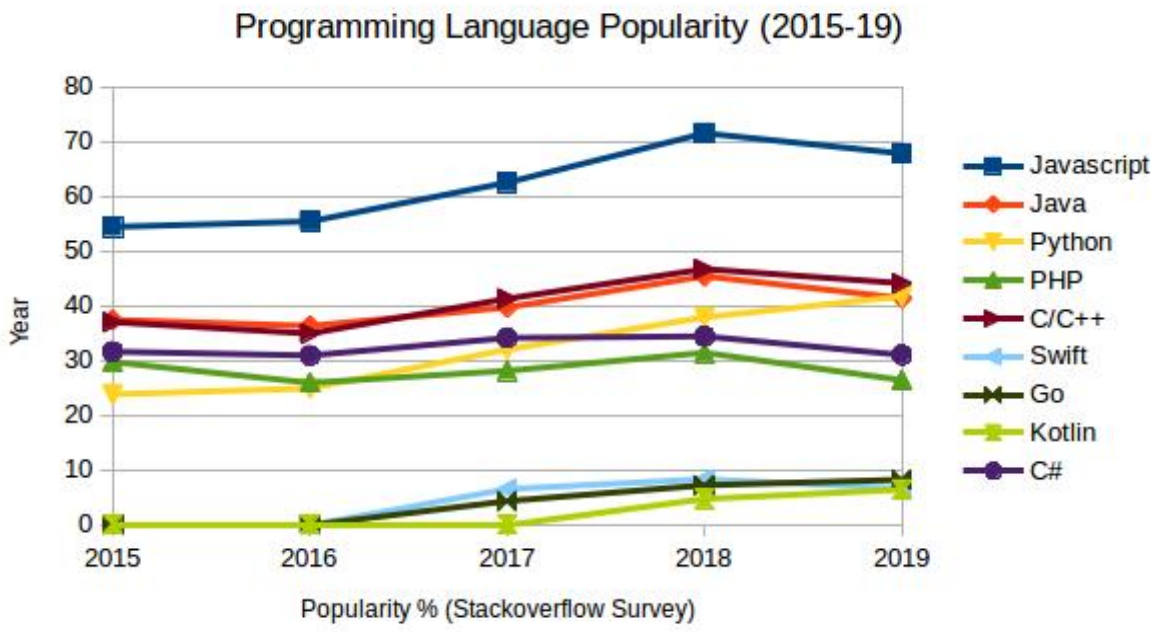

Figura 5 – Linguagens mais populares 2015-2019
Fonte: CodingInfinite (2020)

A linguagem deixou de ser utilizada no desenvolvimento a partir do zero e passou a ser utilizada com frameworks, assim estes tratam da parte verbosa e básica do funcionamento web, deixando a cargo dos desenvolvedores somente o que é realmente específico do programa em desenvolvimento, otimizando todo o processo e dando mais segurança e estabilidade ao programa. Dentre os frameworks PHP, pode ser citados os três mais utilizados: Laravel, CodeIgniter e Symfony, conforme a Figura 6.

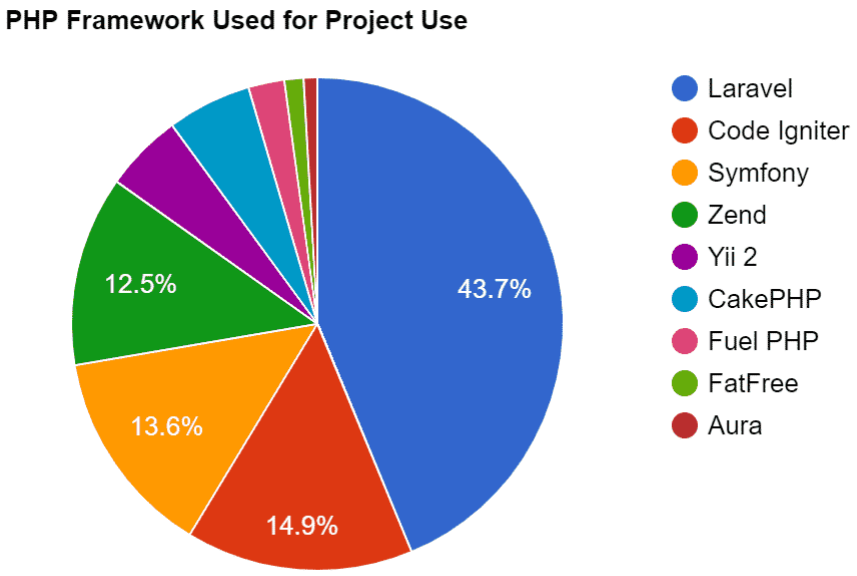

Figura 6 – Frameworks PHP
Fonte: easy2becoder (2020)

O *framework* Laravel possui uma comunidade grande e disponibiliza constantemente atualizações, proporcionando diversas funcionalidades e otimizações para programas de pequeno a grande porte.

O VueJS é um *framework* javascript componentizado, isto é, trabalha com componentes que podem ser reutilizados e divididos conforme sua responsabilidade. Ele é progressivo porque pode ser incrementado aos poucos mesmo que haja outro *framework* sendo utilizado. Ademais a comunidade dele é muito extensa assim como sua documentação.

A grande maioria dos programas tem como principal função a manipulação de dados, seja através da interação com usuários ou, ainda, através de comunicação direta com outros programas, em geral, solicitando e apresentando dados/informações. Estes dados, enquanto necessários e/ou úteis, precisam ser armazenados para posterior uso e/ou consultas. A forma comumente utilizada para isso é com o uso de bancos de dados.

Os bancos de dados podem ser divididos em dois tipos de arquiteturas: relacionais e não-relacionais. Os relacionais possuem estrutura divida em tabelas, possuindo relacionamentos com outras tabelas para mostrar a conexão existente entre os dados de tabelas distintas. Os não-relacionais trabalham com todos os dados interligados por chaves e valores, não possuindo separação.

Os bancos relacionais são utilizados na grande maioria dos programas existentes com banco de dados, devido ao seu desempenho para trabalhar com muitos dados e seus relacionamentos. Para trabalhar com esses bancos deve ser utilizado um Sistema Gerenciador de Banco de Dados (SGBD). São bancos relacionais, entre outros, o MySQL, MariaDB, PostgreSql e Oracle.

A manipulação dos dados, nestes bancos, ocorre através do uso de Standard Query Language (SQL). SQL é a linguagem padrão dos bancos de dados relacionais e com ela podemos realizar todas as ações necessários para manipulação desses dados armazenados. Apesar de ser um padrão, a forma de fazer algo em um determinado banco pode ser diferente em outro. Por exemplo, a declaração de funcionamento de gatilhos em tabelas e até mesmo a escrita de uma consulta pode variar conforme o banco de dados escolhido.

Em um programa web, as ações tomadas pelo usuário devem ser tomadas através do navegador, que é o meio de comunicação entre o usuário e o programa. O usuário realiza o acesso a uma página web, que pode ser desenvolvida utilizando as ferramentas citadas acima, e executa uma requisição ao servidor. Este então processa os dados enviados e encaminha-os esses dados ao interpretador da linguagem, neste caso o PHP sendo utilizado pelo Laravel ou outro framework, que executa as instruções referentes a requisição juntamente com os dados enviados pelo usuário, seja realizando uma consulta ou alteração no banco de dados, a solicitação de um arquivo ou até mesmo o disparo de um email. Após a ação ser realizada, o servidor deve retornar uma resposta ao usuário referente ao que foi requisitado, podendo ser um redirecionamento, uma listagem de dados ou uma confirmação se a ação foi bem sucedida. Essa resposta é interpretada no servidor e devolvida para o usuário em seu navegador. Esse fluxo está ilustrado na Figura 7.

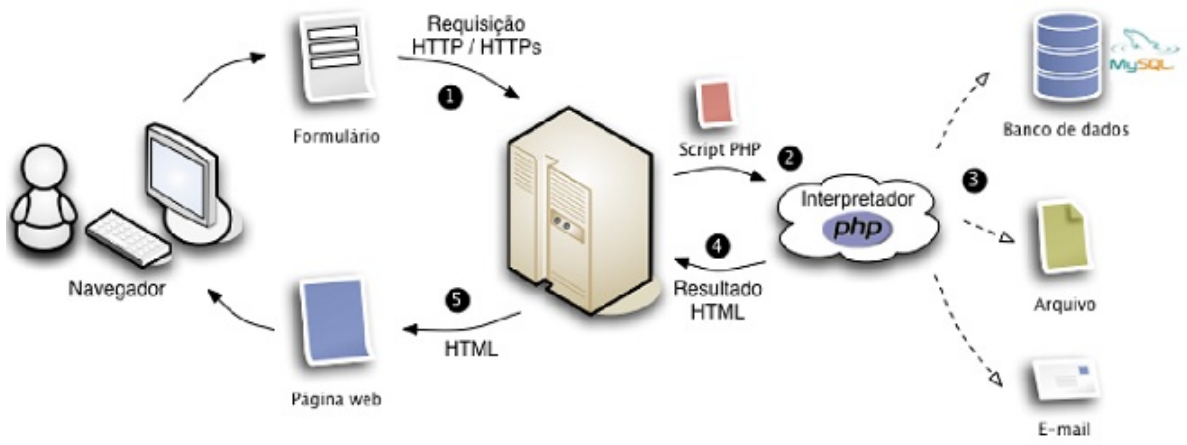

Figura 7 – Fluxo web
Fonte: MARINHO (2020)

4 METODOLOGIA

Este capítulo apresenta os procedimentos metodológicos para a resolução do problema apresentado. A metodologia seguirá os seguintes passos:

1. Para a obtenção dos requisitos será realizada uma consulta a Casa de Acolhida e retiros Nazaré da Igreja Santa Cruz em Guarapuava, para encontrar o maior número de informações sobre os dados relevantes e necessários que o programa deverá contemplar para uma melhor gestão do local;
2. Com base nos estudos e coletas, serão levantados os requisitos funcionais, permitindo entender as necessidades que serão abordadas neste projeto;
3. Após os requisitos funcionais serem encontrados serão feitos a modelagem do banco de dados e a prototipação das principais telas do programa;
4. Sabendo de todas essas informações, serão escritas todas as histórias e tarefas referentes ao projeto seguindo a metodologia do Scrum para subdividir e otimizar o desenvolvimento das funções;
5. Tendo os passos anteriores definidos, será aperfeiçoado o conhecimento no *framework* VueJs para melhorar as interações do usuário com o programa;
6. Para o desenvolvimento geral, será utilizado como base o *framework* PHP Laravel, devido aos conhecimentos prévios e recursos ofertados por tal *framework*;

5 DESENVOLVIMENTO

Neste capítulo serão descritas as fases de desenvolvimento das soluções das questões que foram levantadas os capítulos anteriores. Vale ressaltar que todas as etapas do processo demandaram muito tempo para serem concluídas.

5.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos do programa foi realizado através de entrevistas com os voluntários da casa de acolhimento juntamente com visitas feitas ao local para o melhor entendimento do ambiente no qual o programa será implantado. Com base as informações obtidas, foi possível definir os requisitos funcionais e não funcionais necessários conforme os Quadros 2 e 1.

Quadro 1 – Requisitos não funcionais.

Requisitos não funcionais
1. O sistema deve possuir uma interface simples e intuitiva

Quadro 2 – Requisitos funcionais.

Requisitos funcionais
1. O sistema deve dispor de login somente para administradores e voluntários
2. O administrador deverá ter permissão para cadastrar novas origens de acolhidos
3. O administrador deverá ter permissão para cadastrar novos acolhidos
4. O administrador deverá ter permissão para proibir um acolhido de ser recebido
5. O administrador deverá ter permissão para cadastrar novos administradores e voluntários
6. O administrador deverá ter permissão para cadastrar uma nova acolhida
7. O voluntário deverá ter permissão para cadastrar uma nova acolhida
8. O administrador deverá ter permissão para editar uma acolhida que está em aberto
9. O voluntário deverá ter permissão para editar uma acolhida que está em aberto
10. O administrador deverá ter permissão para editar a quantidade de refeições cedidas em um dia
11. O administrador deverá ter permissão para visualizar gráficos referentes as refeições cedidas
12. O voluntário deverá ter permissão para visualizar gráficos referentes as refeições cedidas
13. O administrador deverá ter permissão para importar planilhas de acolhidas realizadas antes da implantação do sistema
14. O administrador deverá ter permissão para realizar o controle de doações recebidas
15. O sistema deve possuir dois perfis de usuários: administrador e voluntário
16. O sistema deve não deve permitir duas acolhidas simultâneas do mesmo acolhido

5.2 MODELAGEM DO BANCO E PROTOTIPAÇÃO

Tendo todos os requisitos do programa, sendo eles funcionais ou não, bem definidos e descritos pelos usuários, foi possível elaborar o esquema completo do banco de dados para atender as necessidades de todas as funcionalidades requisitadas. O banco foi desenhado de forma a atender os requisitos do momento da implementação do programa, sem pensar nas melhorias futuras que podem ser feitas no mesmo conforme pode ser averiguado na Figura 8.

O banco de dados escolhido para o programa foi o PostgreSQL (POSTGRESQL, 2020) devido ao seu bom desempenho para trabalhar com dados relacionais, sua otimização em consultas e a possibilidade de utilização de plugins que melhoram as interações que são requisitadas pelos usuários.

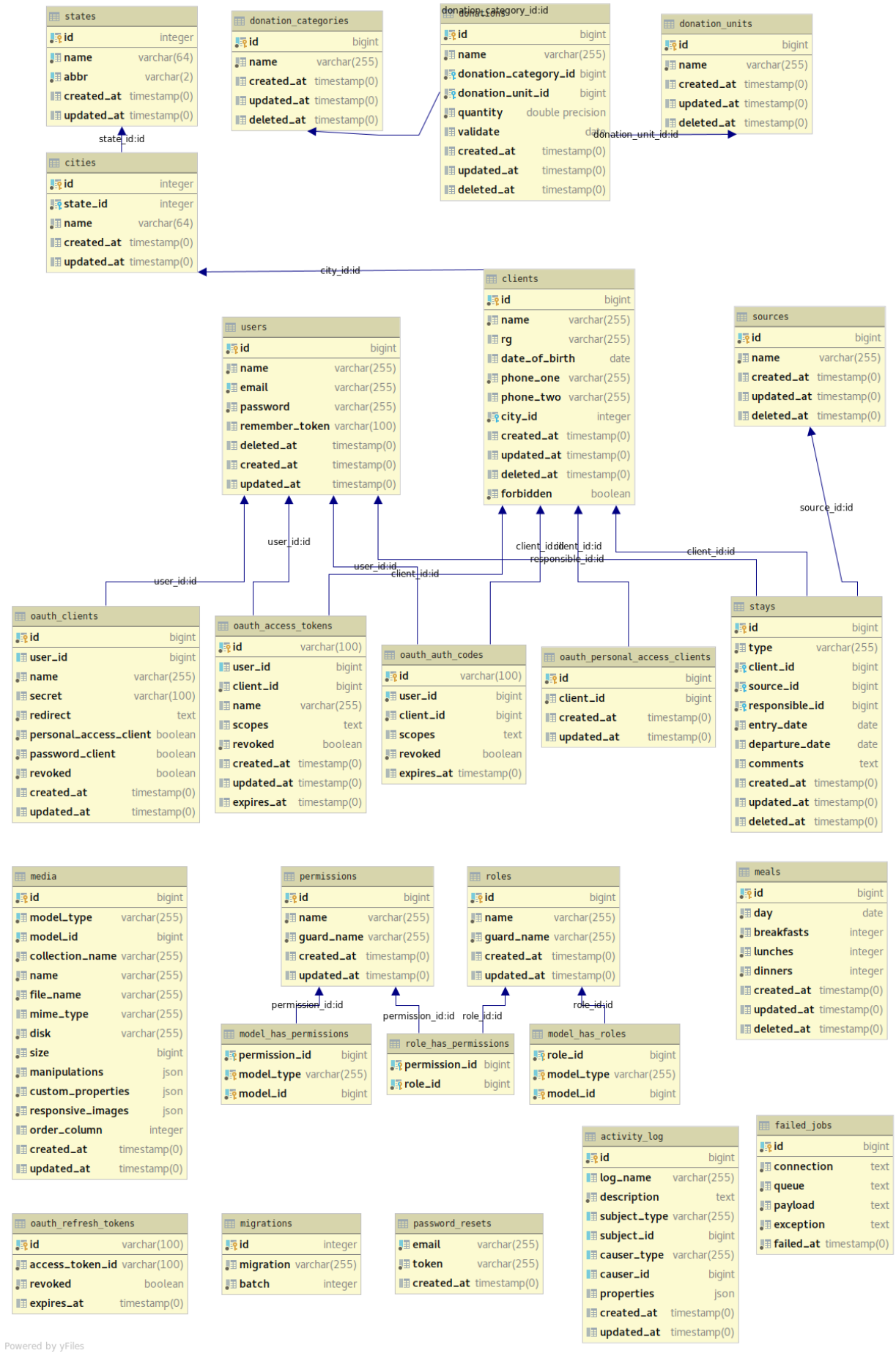

Figura 8 – Diagrama de Entidade Relacionamento

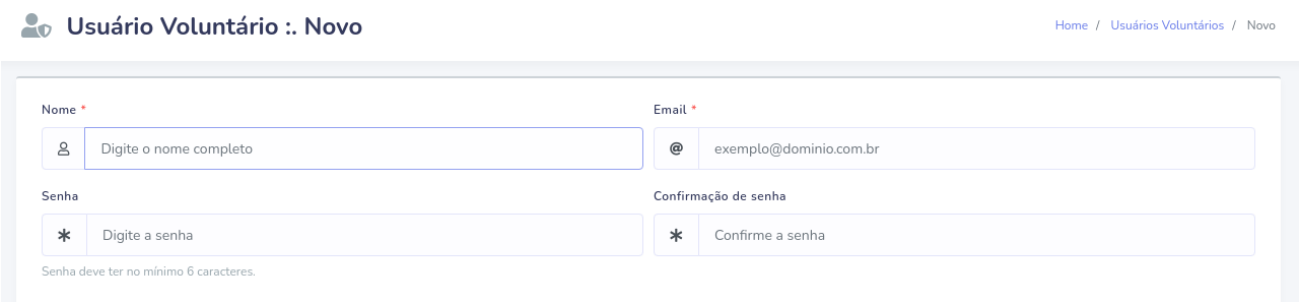

Figura 9 – Criação de novo usuário

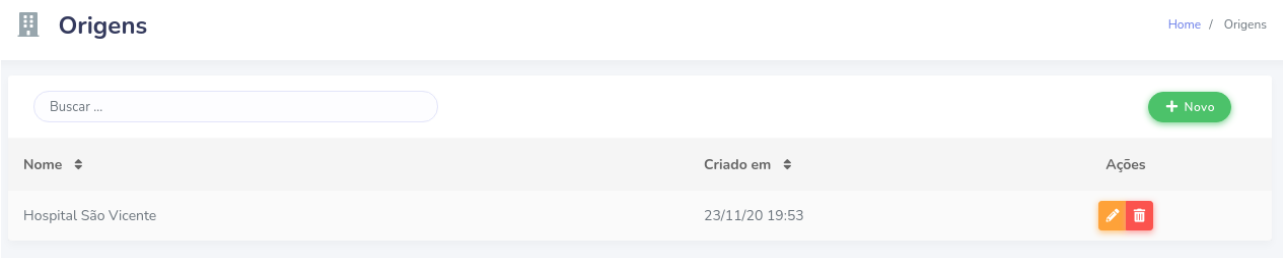

Figura 10 – Lista de origens

Figura 11 – Criação de nova origem

Para o desenvolvimento do layout do programa, foi utilizado o template web Stisla (STISLA, 2020) que provê uma interface muito simples e bonita com várias ferramentas visuais que suprem as necessidades do projeto para ser acessível.

Após a definição do banco de dados foi dado início ao processo de prototipação das telas do programa. Como os usuários do programas, em sua maior parte, não possuem muito contato com tecnologia, todas as telas devem possuir o mínimo de campos possíveis.

Como primeiro controle necessário, foram feitas as telas de controle de usuários administradores e voluntários do programa, diferenciando-os por suas permissões no programa. A tela de cadastro de usuário é bem simples, sendo a mesma tanto para administradores quanto para voluntários como pode ser visto na Figura 9

Após isso foi desenvolvido a área de origens de acolhidos, por ser básica, que devia possuir apenas o nome do local como pode ser visto nas Figuras 10 e 11.

Depois das origens foi desenvolvida a área de acolhidos que demandou um pouco mais de tempo para ser feita devido aos dados mais trabalhados, como a data de nascimento que foi feita com um calendário da biblioteca Vuejs-Datepicker (VUEJSDATEPICKER, 2020) para

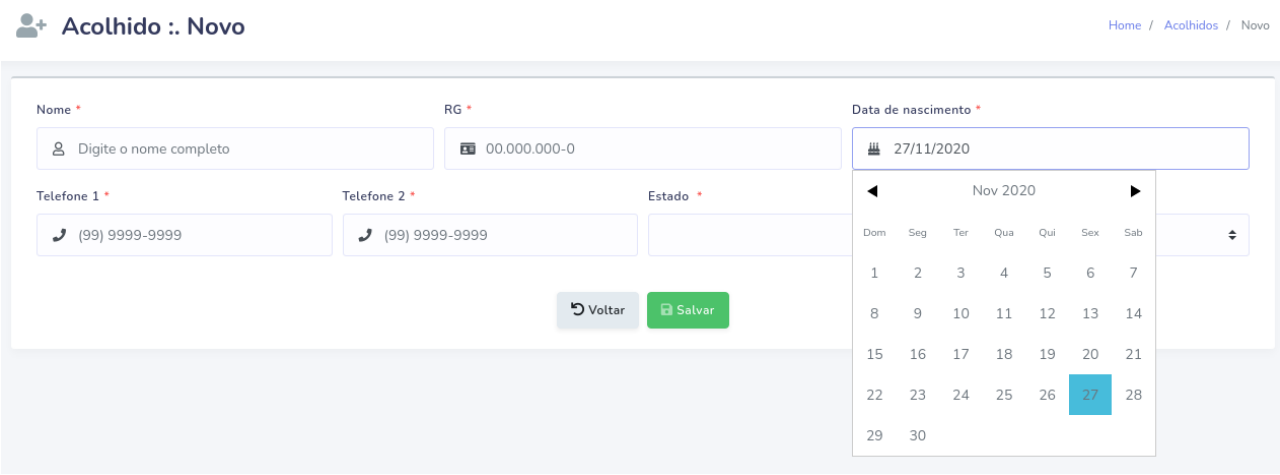

Figura 12 – Criação de novo acolhido

deixar mais amigável a interface e os campos de cidade e estado, que eram dependentes entre si e havia a necessidade de fazer requisições ajax para preenchê-los como pode ser visto na imagem 12.

Tendo as áreas de acolhidos, origens e usuário feitas, o principal recurso do programa que é a área de estadias estava pronta para ser feita, está que demandou a maior parte do tempo para ser elaborada e desenvolvida. Para selecionar o acolhido e o responsável por ele foi utilizado a biblioteca Select2 (SELECT2, 2020) para facilitar as buscas por nomes, para as datas de entrada e saída foi utilizado a mesma biblioteca Vuejs-Datepicker. O resultado da tela de criação de uma nova acolhida pode ser visto na Figura 13.

As estadias possuem um recurso de importação de planilha que foi desenvolvida para auxiliar na sincronização dos dados antigos da casa de acolhimento que foram passados para uma planilha, para que não fosse necessário realizar o cadastro manual de anos de acolhidas e acolhidos. A importação da planilha foi colocada para executar em segundo plano devido ao tamanho da planilha que poder ser extensa, evitando que os usuários necessitem esperar parados na tela até que a mesma seja finalizada. Nos últimos meses essas informações começaram a ser passadas para uma planilha em Excel para o melhor controle dos dados, isso facilitou para que fosse planejado uma funcionalidade neste programa para importar essa planilha, a fim de agilizar o processo de sincronização dos dados antigos com o novo programa.

A área de refeições é um pouco mais simples, não dependendo de uma lista convencional assim como nos outros recursos do programa, portanto foi utilizado o modelo de cartões na listagem para trazer uma interface mais amigável nesta tela como pode ser visto na Figura 14, já que não possuem muitos dados relacionados ao recurso de refeição.

A criação de um novo dia de refeições também não foi feita como nos outros recursos, para deixar mais dinâmico e rápido, foi utilizado um modal que contém um calendário para selecionar a data da refeição e os campos referentes aos tipos de refeições disponíveis (Figura 15). A criação ou edição de refeições foi feita via ajax para evitar recarregamento de página desnecessário e fluidez na utilização do recurso.

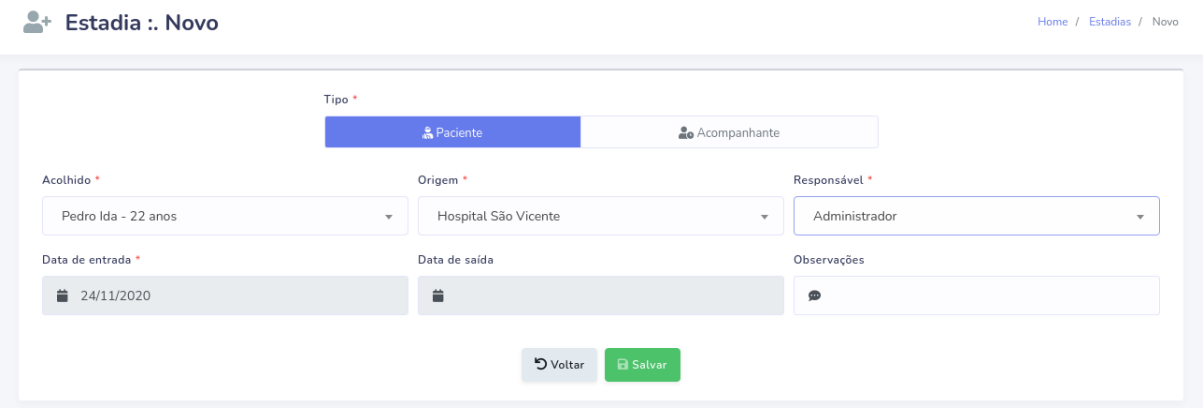

Figura 13 – Criação de nova estadia

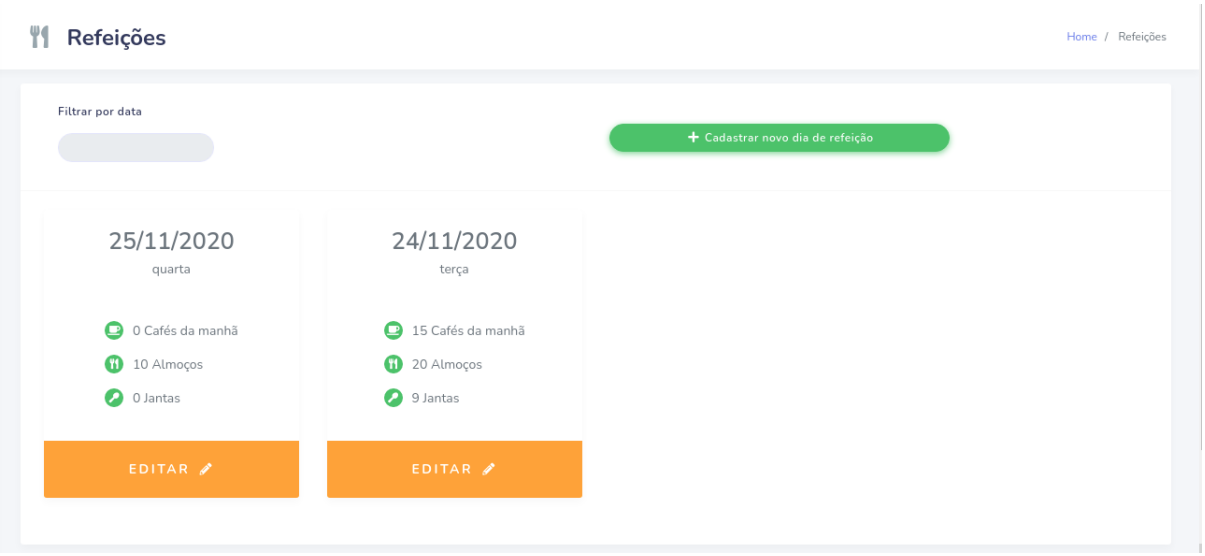

Figura 14 – Lista de refeições diárias

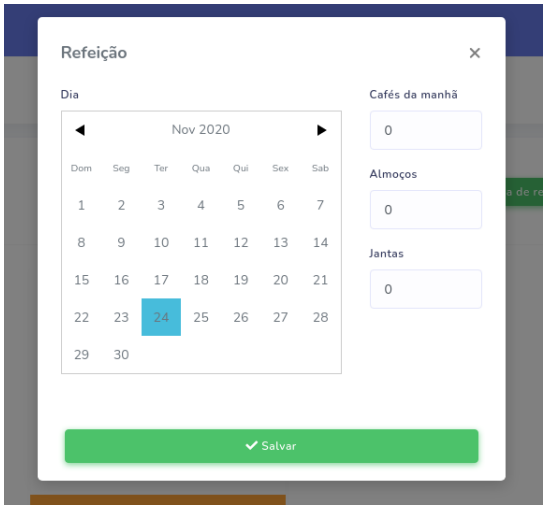

Figura 15 – Criação de nova refeição diária

A dashboard com os gráficos referentes aos acolhidos e as refeições foi a última tela a ser desenvolvida devido a necessidade dos outros recursos serem concluídos para que os dados

fossem gerados corretamente. A princípio ela possui dois gráficos: acolhidos e refeições. Esses gráficos possuem um filtro para a visualização da última semana, mês, ano ou todos os dados sem limite de data (Figura 16).

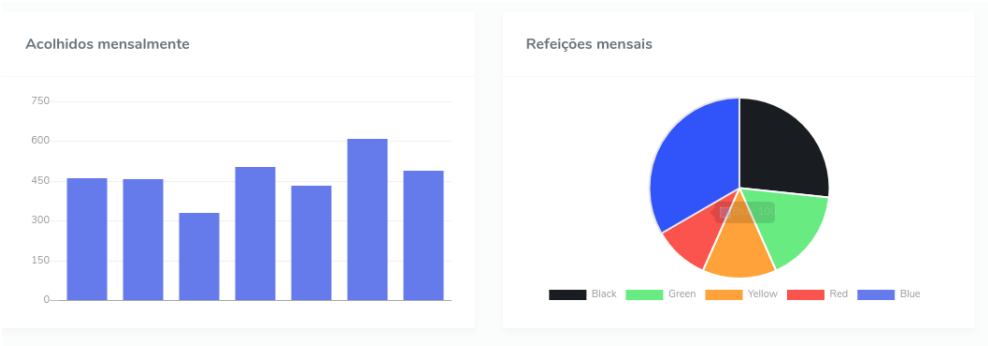

Figura 16 – Dashboard

5.3 DIVISÃO DAS TAREFAS

Conforme foi elaborado o banco de dados e priorizado os recursos do programa, foi montado o modelo Scrum para a realização das tarefas na cronologia em que foi realizado a prototipação das telas no capítulo anterior.

5.4 ESTUDOS DAS TECNOLOGIAS

Seguido junto com o desenvolvimento, foi dado início a um processo de estudo dos frameworks utilizados para que pudesse ser feita a melhor utilização dos mesmos, principalmente do VueJs para trazer mais fluidez na interface com o usuário. As documentações de ambos é bem clara, objetiva e possuem vários exemplos para tornar melhor a compreensão dos recursos ofertados.

6 PLANEJAMENTO DO TRABALHO

6.1 CRONOGRAMA

O planejamento do trabalho que será desenvolvido pelo aluno, ao longo do período letivo, está descrito no cronograma do Quadro 1. Neste cronograma constam todas as atividades com seus respectivos prazos para o cumprimento.

Quadro 3 – Cronograma de atividades.

Atividades	**Ago**	**Set**	**Out**	**Nov**	**Dez**	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**
1. Escrita da proposta do TCC	X	X	X							
2. Defesa da proposta do TCC			X							
3. Revisão dos apontamentos da banca			X							
4. Revisão bibliográfica			X	X						
5. Elaboração do projeto de TCC	X	X	X	X						
6. Defesa do projeto de TCC				X						
7. Desenvolvimento do sistema				X	X	X	X			
8. Validação dos requisitos implementados				X	X					
9. Teste do programa resultante							X	X		
10. Redação da monografia do TCC								X	X	
11. Defesa da monografia do TCC										X

6.2 RECURSOS NECESSÁRIOS

Para o desenvolvimento deste projeto será necessário: um notebook com bom poder de processamento e memória, disponibilidade de internet nos momentos de desenvolvimento, um monitor extra para otimizar a visualização e escrita do programa.

7 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo serão apresentados os resultados, dificuldades e problemas que foram, até o momento, encontrados no desenvolvimento do projeto.

7.1 REALIZAÇÃO DA COLETA DE REQUISITOS

Na realização da visita ao local, foi possível visualizar o meio no qual o sistema deve funcionar e qual é o padrão de usuário que terá acesso. A experiência que foi ganha tendo um contato dessa maneira para obter os requisitos do sistema foi um desafio, tendo em vista que essa tarefa requer um olhar clínico para que seja possível abstrair o máximo de informações sobre o processo como um todo e não é uma tarefa que é feita corriqueiramente.

7.2 PRINCIPAIS DADOS APÓS COLETA

Após a coleta dos requisitos, foi possível desenhar o Diagrama de Entidade Relacionamento que atende as necessidades de forma simples e direta, sem levar em consideração as mudanças futuras que podem ocorrer no projeto, utilizando assim alguns conhecimentos que foram adquiridos nas aulas de APOO no curso conforme a imagem (Figura 8).

Com base nas informações coletadas e a estruturação do banco de dados, foi desenhado pensando na melhor experiência para os usuários, as principais telas que seriam utilizadas, o cadastro de uma nova estadia de um acolhido e o controle de refeições diárias. Foi percebido a necessidade de complementar as caixas de textos com ícones para a melhor associação e assimilação dos campos por parte dos usuários do sistema para acelerar o processo de aprendizagem do mesmo. Alguns conceitos que foram apresentados na matéria de IHC sobre a forma de interação de usuário leigos com sistemas puderam ser utilizados no projeto como pode ser visto nas Figuras 13 e 14.

7.3 APERFEIÇOAMENTOS NOS CONHECIMENTOS DAS TECNOLOGIAS

Para desenvolver o projeto é utilizado o framework PHP Laravel, o qual é muito robusto e possui inúmeras funcionalidades que poderiam ser utilizadas no desenvolvimento do projeto, portanto foi necessário aprofundar os conhecimentos nele, buscando ajuda na documentação que é extremamente clara e simples, assim como foi necessário estudar sobre o framework front-end VueJs para poder aproveitar ao máximo a flexibilidade e reutilização que ele pode proporcionar para a aplicação. Esta foi a parte mais menos complexa no processo do projeto devido a conhecimentos prévios nas tecnologias e linguagens das mesmas.

8 CONCLUSÃO

O objetivo deste trabalho é trazer uma solução muito simples e de fácil utilização para o gerenciamento da Casa de Acolhimento Nazaré, um programa que pode ser utilizado por pessoas que não possuem tanto conhecimento com tecnologia. Nesta solução é implementado um programa web, ou seja, que é acessível tanto através de um computador quanto de um smartphone. O programa deveria ser simples e possuir três funções primordiais: realizar o controle de acolhimentos e acolhidos, controle de refeições cedidas diariamente e controle de doações recebidas. A casa de acolhimento em questão possuía muitos registros de acolhimentos, realizados desde sua abertura, sendo mantidos apenas em cadernos de anotações, tornando praticamente inviável uma busca por alguma informação e até um risco para segurança dos dados. Com esse projeto foi possível entender melhor como funciona a coleta de requisitos no momento de iniciar o desenvolvimento de um software, a entender melhor o lado do usuário que vai utilizá-lo, podendo assim assimilar as ideias e propor soluções que facilitem os processos.

8.1 TRABALHOS FUTUROS

O programa possui algumas melhorias que foram colocadas como opcionais e com baixa prioridade de desenvolvimento pelos integrantes da instituição favorecida. Uma nova função do programa seria o controle de quartos disponíveis no recinto, já que no momento não há a necessidade de tal função, porém a longo prazo pode ser necessária caso o número de acolhidos aumente. Outra função que pode ser necessário com o passar o tempo é o controle de horário de disponibilidade de um quarto, já que um acolhido pode sair, por exemplo, antes do almoço e já liberar uma nova vaga a partir deste horário.

Referências

CODINGINFINITE. **CodingInfinite**. 2020. Disponível em: <https://codinginfinite.com/top-programming-languages-2020-stats-surveys/>. Acesso em: 16 de novembro de 2020. Citado na página 8.

DESENVOLVIMENTOAGIL. **DesenvolvimentoAgil**. 2020. Disponível em: <https://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 17 de novembro de 2020. Citado na página 6.

EASY2BECODER. **easy2becoder**. 2020. Disponível em: <http://easy2becoder.blogspot.com/2017/04/top-5-php-frameworks-for-developers.html>. Acesso em: 16 de novembro de 2020. Citado na página 9.

HOMEHOST. **homehost**. 2020. Disponível em: <https://www.homehost.com.br/blog/tutoriais/o-que-e-bootstrap/>. Acesso em: 21 de novembro de 2020. Citado na página 7.

HOSTINGER. **O que é Apache? Uma Visão Aprofundada do Servidor Apache**. 2020. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-apache>. Acesso em: 21 de novembro de 2020. Citado na página 7.

HQBEDS. **HQBeds**. 2020. Disponível em: <https://hqbeds.com/>. Acesso em: 10 de setembro de 2020. Citado 3 vezes nas páginas 4, 5 e 6.

IDES. **IDES**. 2020. Disponível em: <https://www.ides-sc.org.br/>. Acesso em: 10 de setembro de 2020. Citado 3 vezes nas páginas 3, 4 e 6.

MARINHO, F. **FREDERICO MARINHO**. 2020. Disponível em: <https://www.fredericomarinho.com/introducao-ao-desenvolvimento-web-com-php-aula-1-preparando-o-ambiente-para-in cliente-servidor-php/>. Acesso em: 16 de novembro de 2020. Citado na página 10.

PHP. **Manual PHP**. 2020. Disponível em: <https://www.php.net/manual/pt_BR/intro-whatis.php>. Acesso em: 21 de novembro de 2020. Citado na página 7.

POSTGRESQL. **PostgreSQL**. 2020. Disponível em: <https://www.postgresql.org/>. Acesso em: 27 de novembro de 2020. Citado na página 14.

SELECT2. **Select2**. 2020. Disponível em: <https://select2.org/>. Acesso em: 27 de novembro de 2020. Citado na página 17.

SISHOSP. **SisHOSP**. 2020. Disponível em: <https://sishosp.com.br/software-de-gestao-para-casa-de-repouso/>. Acesso em: 24 de novembro de 2020. Citado 2 vezes nas páginas 5 e 6.

STISLA. **Stisla**. 2020. Disponível em: <https://getstisla.com/blog/open-source>. Acesso em: 27 de novembro de 2020. Citado na página 16.

TECNICON. **Tecnicon**. 2020. Disponível em: <https://www.tecnicon.com.br/blog/411-Metodologia_Scrum_para_a_gestao_de_processos_ageis_na_industria>. Acesso em: 16 de novembro de 2020. Citado na página 7.

VUEJSDATEPICKER. **VuejsDatepicker**. 2020. Disponível em: <https://github.com/charliekassel/vuejs-datepicker>. Acesso em: 27 de novembro de 2020. Citado na página 16.