

Lucas Jardim da Silva

**Painel Pró-Mamá: Um facilitador para o canal de
comunicação do Programa Municipal de
Aleitamento Materno de Osório - RS**

Osório

2018

Lucas Jardim da Silva

**Painel Pró-Mamá: Um facilitador para o canal de
comunicação do Programa Municipal de Aleitamento
Materno de Osório - RS**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de Tecnólogo em Análise e Desenvolvimento de
Sistemas.

Orientador: Prof. Bruno Chagas Alves Fernandes

Coorientador: Prof. Diana Cabral Cavalcanti

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS

Campus Osório

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Osório

2018

Lucas Jardim da Silva

**Painel Pró-Mamá: Um facilitador para o canal de
comunicação do Programa Municipal de Aleitamento
Materno de Osório - RS**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de Tecnólogo em Análise e Desenvolvimento de
Sistemas.

Prof. Bruno Chagas Alves Fernandes
Orientador

Prof. Timoteo Alberto Peters Lange
Convidado 1

Prof. Vinicius Fritzen Machado
Convidado 2

Osório
2018

RESUMO

O Brasil ainda está em uma situação longe da recomendada pela Organização Mundial da Saúde: A amamentação exclusiva até o sexto mês e aleitamento materno com alimentos complementares até dois anos ou mais de idade. No município de Osório-RS, o Programa Pró-Mamã, em parceria com o IFRS-Campus Osório, criou um aplicativo com objetivo de disseminar as informações sobre o período do aleitamento materno e conscientizar a comunidade osoriense. No processo de desenvolvimento do aplicativo, observou-se a necessidade de existir um sistema informatizado que possibilitasse aos profissionais da equipe Pró-Mamã, por algum meio, inserir as informações pertinentes ao período de amamentação. Com esta premissa foi desenvolvido o Painel Pró-Mamã, um Painel Administrativo e uma API que foram responsáveis por fazerem a informação e o conhecimento chegarem nas pessoas. Para o desenvolvimento usou-se o framework PHP Laravel que fornece boa documentação e permitiu que o projeto escalasse de tamanho sem riscos. O framework se demonstrou muito eficaz e deu suporte durante todo o processo de desenvolvimento do sistema, desde a codificação até a implantação. O sistema encontra-se em funcionamento e vem sendo usado pela equipe Pró-Mamã. Esta insere informações, notifica a comunidade e recebe e responde dúvidas. O Painel Pró-Mamã, juntamente com o aplicativo, causam um impacto positivo no município de Osório-RS, uma vez que estimulam e promovem campanhas sobre o aleitamento materno.

Palavras-chave: Sistema de Gerenciamento de Conteúdo. CMS. API. Laravel. Aleitamento Materno.

ABSTRACT

Brazil is still in a situation far from that recommended by the World Health Organization: Exclusive breastfeeding until the sixth month and breastfeeding with complementary foods up to two years old or over. In Osório-RS, the Pró-Mamá Program, in partnership with IFRS-Campus Osório, created a mobile application aimed at disseminating information about the breastfeeding period and raising awareness among the Osório community. In the process of application development, it was observed the need to have a computerized system that would allow the professionals of the Pró-Mamá team, from somewhere, to insert information pertinent to the period of breastfeeding. With this premise was developed the Painel Pró-Mama, an Administrative Panel and an API that were responsible for making information and knowledge reach people. In development, the PHP Laravel framework was used. It provided good documentation and allowed the project to scale in size without risks. The framework proved to be very effective and supported throughout the entire system development process, from coding to deployment. The system is in operation and has been used by the Pró-Mamá team. It inserts information, notifies the community, and receives and answers questions. Painel Pró-Mamá, together with the mobile application, have a positive impact on the municipality of Osório-RS, since they stimulate and promote campaigns on breastfeeding.

Keywords: Content Management System. API. Laravel. Breastfeeding.

LISTA DE ILUSTRAÇÕES

Figura 1 – Uso de linguagens de programação do lado do servidor para sites	13
Figura 2 – MVC aplicado à Web	15
Figura 3 – Popularidade de Frameworks PHP para Projetos Pessoais	17
Figura 4 – Popularidade de Frameworks PHP para uso profissional	17
Figura 5 – Interesse ao longo do tempo dos principais frameworks PHP	18
Figura 6 – Gráfico de Representação do Experimento do Tempo de Execução Entre PHP Puro, CodeIgniter e Laravel	19
Figura 7 – Painel de Controle do OuvidorSUS	26
Figura 8 – Distribuição dos sites usando tecnologias CMS	27
Figura 9 – Cenário preexistente	33
Figura 10 – Cenário atual	34
Figura 11 – Esquema Representativo do Funcionamento do Sistema	36
Figura 12 – Modelo de Arquitetura de Software	40
Figura 13 – Modelo Lógico do Banco de Dados	42
Figura 14 – Modelo de Arquitetura de Software	43
Figura 15 – Relação da Classe Controladora com a Modelo e a Visualização	44
Figura 16 – Tela de Login	47
Figura 17 – Layout Fixo e Layout Dinâmico	48
Figura 18 – Tela de Listagem de Informações	48
Figura 19 – Botão de Adicionar	49
Figura 20 – Formulário de Adicionar Bairro	49
Figura 21 – Formulário de Adicionar Informação	50
Figura 22 – Adicionar Link Externo	51
Figura 23 – Adicionar Link Interno	51
Figura 24 – Pré-Visualização de Informação	52
Figura 25 – Tela Principal do menu Fale Conosco	53
Figura 26 – Tela de Responder uma Dúvida (Menu Fale Conosco)	53
Figura 27 – Tela de Dúvida respondida para o usuário no aplicativo	53
Figura 28 – Tela de Dúvida respondida para todos no aplicativo	53
Figura 29 – Formulário de Notificação	54
Figura 30 – Página Inicial do Painel	56
Figura 31 – E-mail de notificação de nova dúvida	57
Figura 32 – Método de Responder Dúvida de Usuário	58
Figura 33 – E-mail de Esqueci a Senha	59
Figura 34 – Tela de Alterar Senha	59

LISTA DE TABELAS

Tabela 1 – Tabela de Testes de Carga com os frameworks Symfony e Laravel	19
Tabela 2 – CMSs que possuem aplicativos móveis	28
Tabela 3 – Requisito Funcional 1	37
Tabela 4 – Requisito Funcional 2	37
Tabela 5 – Requisito Funcional 3	37
Tabela 6 – Requisito Funcional 4	38
Tabela 7 – Requisito Funcional 5	38
Tabela 8 – Requisito Funcional 6	38
Tabela 9 – Requisito Funcional 7	38
Tabela 10 – Requisito Funcional 8	38
Tabela 11 – Requisito Funcional 9	39
Tabela 12 – Requisito Funcional 10	39
Tabela 13 – Requisitos não funcionais	39
Tabela 14 – Modelo do Cadastro de Notificação com o coringa	55
Tabela 15 – Modelo de como a Notificação aparece no aplicativo	55
Tabela 16 – Respostas Padrões	61
Tabela 17 – Rotas da API	62

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CMS	Content Management System
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
Pró-Mamá	Programa Municipal de Aleitamento Materno da cidade de Osório-RS
REST	Representational State Transfer
SGBD	Sistema Gerenciador de Banco de Dados
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TI	Tecnologia da Informação
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	11
1.2	Justificativa	11
2	REFERENCIAL TEÓRICO	12
2.1	Aleitamento materno	12
2.2	Aplicação Web	13
2.3	CMS	14
2.4	Model-View-Controller - MVC	15
2.5	Frameworks	16
2.5.1	Laravel	20
2.6	Application Programming Interface - API	20
3	TRABALHOS RELACIONADOS	22
3.1	WordPress	22
3.2	Drupal	23
3.3	Joomla	24
3.4	Moodle	24
3.5	OuvidorSUS	25
3.6	Considerações	27
4	METODOLOGIA	30
4.1	Atividades	30
5	DESENVOLVIMENTO	32
5.1	Cenário preexistente	32
5.2	Cenário atual	33
5.3	Funcionamento do Sistema	35
5.4	Análise dos Requisitos do Painel Administrativo	37
5.4.1	Requisitos Funcionais	37
5.4.2	Requisitos Não Funcionais	39
5.5	Arquitetura de Software	40
5.6	Banco de Dados	41
5.6.1	Modelo Lógico	41
5.6.2	Diagrama de Classes	43

5.6.2.1	Triggers	44
5.6.3	Considerações	44
5.7	Configuração do Ambiente	45
5.7.1	Controle de Versão	45
5.7.2	Ambientes de Testes	46
5.8	Painel Administrativo	46
5.8.1	Informação	49
5.8.1.1	Links externos e internos	50
5.8.1.2	Pré-Visualizar	51
5.8.2	Fale Conosco	52
5.8.3	Notificação	54
5.8.4	Página Inicial (Home)	55
5.8.5	Envio de e-mail	56
5.8.5.1	Cadastro de Dúvida	56
5.8.5.2	Resposta de Dúvida	57
5.8.5.3	Esqueci a senha	58
5.8.6	Alterar Senha	59
5.9	API	60
5.9.1	Requisições e Respostas	60
5.9.2	Rotas da API	61
5.10	Implantação (Deploy)	63
5.10.1	Manutenção do Sistema	63
6	CONCLUSÃO	64
6.1	Trabalhos Futuros	64
	REFERÊNCIAS	66

1 INTRODUÇÃO

A situação do aleitamento materno no Brasil ainda está longe da preconizada pela Organização Mundial da Saúde: amamentação exclusiva até o sexto mês e aleitamento materno com alimentos complementares até dois anos ou mais de idade (Organização Mundial da Saúde, 1994). A duração do aleitamento materno pode ser favorecida ou restringida por fatores biológicos, culturais, relativos à assistência à saúde e socioeconômicos ([ADAIR; POPKIN; GUILKEY, 1993](#)).

Os profissionais de saúde, por meio de suas atitudes e práticas, podem influenciar positiva ou negativamente o início da amamentação e sua duração ([CARVALHAES; CORRÊA, 2003](#)). Por tais motivos, existem diversas ações de apoio à amamentação que visam incentivar e conceder as informações necessárias para o período de aleitamento. E, isto só é possível através de uma comunicação direta dos profissionais da saúde para com as mães.

O Programa Municipal de Aleitamento Materno da cidade de Osório (Pró-Mamá) é um programa de iniciativa do Núcleo de Apoio à Saúde da Família – NASF e da Secretaria de Saúde do município. Trata-se de um trabalho multidisciplinar que conta com psicóloga, nutricionista e fonoaudiólogas e tem como objetivo acompanhar mães e bebês nas questões do aleitamento materno e desenvolvimento infantil. Uma das prioridades do Pró-Mamá é estar mais perto destas mães, e isso ocorre através dos profissionais de referência em amamentação que atuam em todas as Unidades de Saúde - Estratégia de Saúde da Família/ESF. Estes profissionais atuam através de atendimentos, encaminhamentos, acompanhamento, grupos de orientação, visitas domiciliares, entre outros ([OSORIO.RS.GOV.BR, 2017](#)).

O aplicativo Pró-Mamá, que surgiu da parceria entre o IFRS-Osório e o programa, foi idealizado com o objetivo de facilitar o acesso à informação e fazer comunicação direta entre as mães e os profissionais devidamente capacitados da área da saúde. Durante o processo de desenvolvimento do aplicativo observou-se a necessidade de um ambiente que permitisse a comunicação e gerência do app. Em suma, precisava-se de um sistema que possibilitasse que a equipe Pró-Mamá pudesse alimentar a base de dados do sistema com informações relevantes. E, também, gerenciar toda a comunicação do usuário com o profissional capacitado, através de perguntas e dúvidas.

Visto a necessidade de haver um sistema para facilitar a comunicação do Pró-Mamá com a comunidade de Osório foi então que surgiu o Painel Pró-Mamá. Este trabalho se trata do desenvolvimento de um sistema de gerenciamento de conteúdo que serve como meio pelo qual a equipe do Pró-Mamá pode inserir e fazer a gestão das informações em saúde e prevenção relativos ao aleitamento materno.

O Painel Pró-Mamá é o conjunto do Painel Administrativo e uma API. O Painel, como

dito no parágrafo anterior, serve como meio pelo qual os profissionais do Pró-Mamã inserem as informações e conhecimentos em saúde. Já a API é o meio tecnológico pelo qual o aplicativo utiliza para resgatar essas informações cadastradas.

1.1 OBJETIVOS

Nessa seção são apresentados os objetivos geral e específicos do presente trabalho.

1.1.1 Objetivo geral

Desenvolver um sistema de gerenciamento de conteúdo para o programa Pró-Mamã, permitindo à equipe de profissionais do programa produzir e gerenciar informações relativas ao aleitamento materno.

1.1.2 Objetivos específicos

- Analisar os requisitos para desenvolvimento do Painele Administrativo;
- Projetar o Banco de Dados;
- Projetar a arquitetura de software do Painele Administrativo;
- Desenvolvimento do Painele Administrativo;
- Desenvolvimento da API;
- Manutenção do Sistema;

1.2 JUSTIFICATIVA

Visto a necessidade de um facilitador de comunicação entre os profissionais da saúde e às mães que estão em período de aleitamento materno, observou-se que o aplicativo do programa Pró-Mamã, juntamente com o Painele, causariam um impacto positivo na comunidade. A tecnologia como aliada às causas sociais é importante tema e oferece amplo campo de pesquisa e estudo.

O próprio Ministério da Saúde recomenda que os meios de comunicação apoiem à amamentação seja através da divulgação da importância do aleitamento materno, das experiências positivas sobre o aleitamento materno ou da estimulação e promoção de campanhas sobre aleitamento materno ([PORTALMS.SAUDE.GOV.BR](https://portalms.saude.gov.br), 2018).

Portanto, são justificativas como esta que guiam e embasam o presente trabalho. Busca-se explorar e desenvolver, através das metodologias da área de TI, um sistema que seja útil e que agregue à saúde pública, atendendo uma causa importante como o aleitamento materno.

2 REFERENCIAL TEÓRICO

Nessa seção são apresentados os fundamentos teóricos relevantes para o desenvolvimento do presente trabalho.

2.1 ALEITAMENTO MATERNO

A Organização Mundial da Saúde aconselha que as mulheres amamentem seus filhos exclusivamente com leite materno em seus seis primeiros meses de vida. A amamentação não é totalmente instintiva no ser humano, muitas vezes deve ser aprendida para ser prolongada com êxito, considerando-se que a maioria das nutrizes precisa de esforço e apoio constantes. Nesse sentido, as mulheres, ao se depararem pela primeira vez com o aleitamento materno, requerem que lhes sejam apresentados modelos ou guias práticos de como devem conduzir-se nesse processo, que na maioria das vezes tem como primeira referência o meio familiar, as amigas e vizinhança nos quais estão inseridas (MACHADO et al., 2004).

Segundo o Ministério da Saúde, os meios de comunicação podem apoiar o aleitamento materno das seguintes formas:

- Cumprindo a NBCAL¹ e a Lei 11.265², que regulamentam a publicidade de alimentos e produtos que concorrem com a amamentação.
- Divulgando a importância do aleitamento materno.
- Divulgando experiências positivas sobre o aleitamento materno.
- Estimulando e promovendo campanhas sobre aleitamento materno.

Nota-se que os três últimos itens enfatizam a necessidade de políticas e ações de apoio que divulguem e promovam o aleitamento materno. A disponibilidade do acesso às informações, feita pelos meios de comunicações, dão conhecimento e ressaltam à comunidade de que existe atenção e olhares ao período de amamentação por parte dos profissionais de Saúde.

O programa Pró-Mamã, por meio do aplicativo, amplia e fomenta os conhecimentos essenciais sobre o período de amamentação. O aplicativo e o Painei, em conjunto, são ferramentas que auxiliam na promoção da conscientização do aleitamento materno, fortalecendo as políticas de apoio e qualificando a atenção no município para uma importante área no desenvolvimento da vida.

¹ Disponível em <http://dab.saude.gov.br/portaldab/ape_promocao_da_saude.php?conteudo=norma>

² Disponível em <http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Lei/L11265.htm>

2.2 APLICAÇÃO WEB

Uma aplicação web é um sistema de informática. É qualquer programa de computador que realiza uma função específica através de um navegador como cliente (NATIONS, 2014). A aplicação é acessada através da internet ou aplicativos desenvolvidos utilizando tecnologias web como HTML, JavaScript e CSS.

As aplicações Web geralmente usam uma combinação de script do lado do servidor (ASP, PHP, etc) e script do lado do cliente (HTML, Javascript, etc.) para desenvolver o aplicativo. O script do lado do cliente lida com a apresentação das informações, enquanto o script do lado do servidor lida com todas as coisas difíceis, como armazenar e recuperar as informações (NATIONS, 2014).

A vantagem de desenvolver um sistema Web é que o desenvolvedor não precisa se preocupar com os dispositivos ou computadores que acessarão a aplicação pelo motivo de que esta só precisa de um navegador para ser acessada. A maioria das aplicações Web é baseada na arquitetura cliente-servidor, na qual o cliente insere informações enquanto o servidor armazena e recupera informações (NATIONS, 2014).

A linguagem utilizada no lado do servidor foi a linguagem PHP. Esta, é uma linguagem popular, amplamente utilizada e possui muita documentação disponível na internet. Também como mostrado na figura 1, o PHP é usado por 78,9% de todos os sites cujas linguagens de programação do lado do servidor são conhecidas (W3TECHS.COM, 2018).

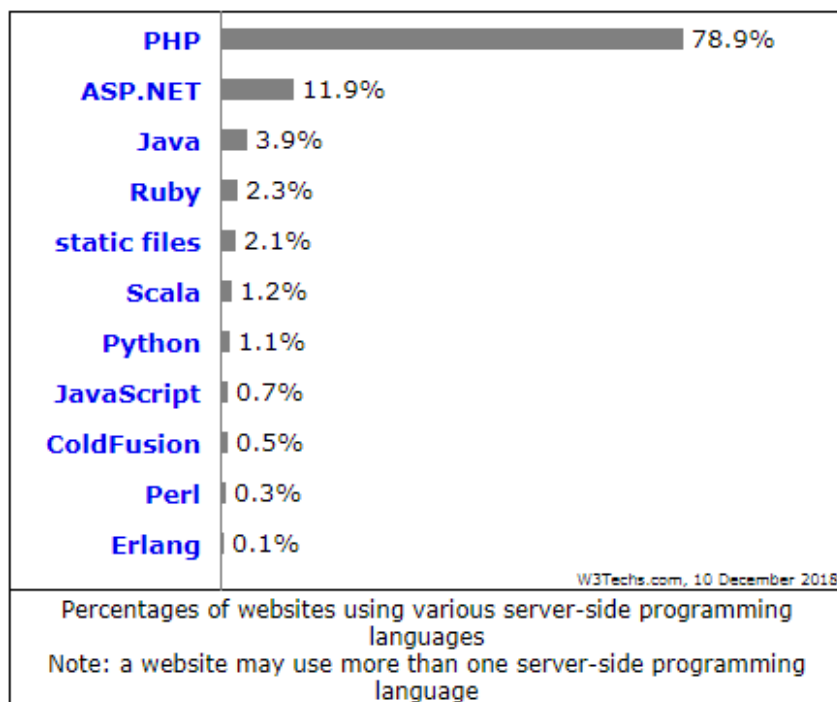


Figura 1 – Uso de linguagens de programação do lado do servidor para sites

Fonte: w3techs.com

A escolha por optar por desenvolver um sistema Web para o presente trabalho foi inevitável, uma vez que se encaixa perfeitamente no quesito de fácil e rápido acesso. Por se tratar de um painel administrativo, a interface foi desenvolvida seguindo os padrões de *Dashboards* que existem na internet: Uma barra lateral na esquerda com os itens de menu e o conteúdo à direita da tela.

O foco foi desenvolver um sistema que satisfizesse os profissionais do Programa e ao mesmo tempo que fosse agradável de usar, sem complexidade e poluição gráfica. A plataforma Web provê muitas possibilidades e recursos, sendo assim de grande utilidade no desenvolvimento de sistemas que precisam se interconectar.

2.3 CMS

As pessoas usam uma grande variedade de ferramentas e métodos para processar informações e colocá-las em publicações. Desde cartões de anotações em papel até aplicações de banco de dados, as organizações encontraram maneiras de organizar e gerenciar informações (BOIKO, 2001).

CMS é um programa de servidor que armazena texto de páginas Web e publica detalhes, ou informações, em um banco de dados (SITELEADS.NET, 2018). Um CMS permite criar, organizar, publicar e apagar conteúdos de um site, como por exemplo as páginas de um blog (DRUBSCKY, 2018).

As páginas não podem ser carregadas até que seja solicitado pelo navegador do cliente e é construído sob demanda. Com o uso de WCMS muitos tipos de conteúdo podem ser organizados e publicado, e o layout, aparência e estrutura do site podem ser alterados com facilidade e rapidez, pois é baseado em modelos e, portanto, o conteúdo é separado da apresentação - ao contrário de um site codificado normal (PATEL; RATHOD; PRAJAPATI, 2011).

Um CMS pode ser estendido, o que significa que se pode adicionar novos recursos e funciona como e quando sua exigência muda ao contrário site estático e HTML. Isso significa que o CMS é um sistema dinâmico (PATEL; RATHOD; PRAJAPATI, 2011). Fornecem uma solução ideal, organizando informações e, principalmente, oferecendo a possibilidade de criar e gerenciar o conhecimento de uma empresa (BENEVOLO; NEGRI, 2007).

Ainda segundo (PATEL; RATHOD; PRAJAPATI, 2011), o CMS tem claras inúmeras vantagens sobre os sites estático usuais. Um CMS é a escolha preferida para aqueles que querem mudanças de conteúdo fáceis, controle simplificado de grandes quantidades de conteúdo, uma opção de plug-ins para realizar grande variedade de tarefas e acima de tudo a capacidade de fazer muito mais, muito mais fácil e muito mais rápido do que com qualquer outro sistema.

Este trabalho, portanto, utiliza os conceitos de CMS em sua estrutura uma vez que sua principal funcionalidade é permitir a inserção e gerenciamento das informações, pertinentes ao aleitamento materno, através de um sistema Web, ou painel administrativo.

2.4 MODEL-VIEW-CONTROLLER - MVC

A arquitetura Model-View-Controller (MVC) é uma arquitetura de software na qual os elementos de “visualização”, ou *view*, de uma aplicação são separados dos elementos da “camada de lógica de negócios”, denominada “modelo”, ou *model*. Na arquitetura MVC, um “controlador”, ou *controller*, determina quando o controle do aplicativo, durante sua execução, é passado para um elemento de visualização ou a um elemento de modelo (HAVIV; TRIPP; WEISMAN, 2014).

Ao construir aplicativos interativos, ou qualquer outro programa, a modularidade dos componentes traz enormes benefícios. Isolar as unidades funcionais umas das outras, tanto quanto possível, facilita para o desenvolvedor entender e modificar cada unidade em particular, sem ter que saber tudo sobre as outras unidades (KRASNER; POPE et al., 1988).

No contexto Web, a arquitetura MVC funciona como mostrado na figura 2. Um *request* é feito por uma aplicação cliente (geralmente um navegador Web) e é tratado por algum *controller* específico da aplicação. Esse *controller* pode acessar uma classe *model* para resgatar dados do Banco ou também pode renderizar uma *view* para o usuário.

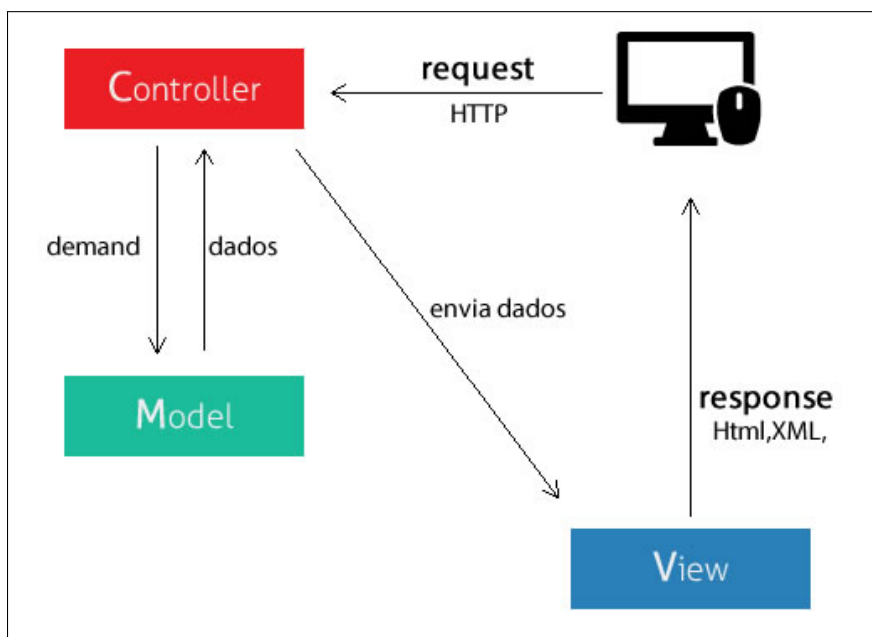


Figura 2 – MVC aplicado à Web

Fonte: <<https://tableless.com.br/mvc-afinal-e-o-que/>>

Portanto, o MVC é uma arquitetura que provê flexibilidade e reutilização de código no desenvolvimento. Por ser um padrão que divide as responsabilidades entre as camadas e organiza o fluxo de informações, o MVC se torna muito útil na construção de sistemas que escalam, ou que são propensos à mudança frequentemente.

O Laravel, que será comentado na subseção 2.5.1, é estruturado com os padrões MVC. Para o desenvolvimento do Painel era necessário que houvesse uma organização, um padrão a

ser usado. O MVC se mostrou um facilitador e auxiliador na construção do software e foi muito proveitoso durante todo o processo.

2.5 FRAMEWORKS

Basicamente, o Framework provê um template de base com diversas funções para um desenvolvedor, de modo que é desnecessário gastar tempo reproduzindo funções em diversos projetos (GAEA.COM.BR, 2018).

Segundo ([FAYAD; SCHMIDT; JOHNSON, 1999](#)), a utilização de frameworks apresenta os seguintes benefícios:

- Melhora a modularização – encapsulamento dos detalhes voláteis de implementação através de interfaces estáveis.
- Aumenta a reutilização – definição de componentes genéricos que podem ser replicados para criar novos sistemas.
- Extensibilidade – favorecida pelo uso de métodos hooks que permitem que as aplicações estendam interfaces estáveis.
- Inversão de controle – IoC – o código do desenvolvedor é chamado pelo código do framework. Dessa forma, o framework controla a estrutura e o fluxo de execução dos programas.

Frameworks para desenvolvimento na plataforma Web vêm sendo cada vez mais usados por programadores. Com códigos bem documentados, organização, segurança e utilidades, estes têm facilitado o trabalho dos desenvolvedores Web.

Segundo uma pesquisa de popularidade de frameworks PHP feita pelo website sitepoint em 2015, que contou aproximadamente com um número de 7800 participantes, o framework Laravel ganhou o título de Framework mais popular de 2015 ([SKVORC, 2015](#)).

Como é visto nas figuras 3 e 4, o Laravel é o mais utilizado tanto para uso profissional quanto para projetos pessoais. A diferença de votos entre o primeiro e segundo lugar nas pesquisas é grande, sendo de 592 votos na pesquisa de uso profissional e de 1107 votos na pesquisa de projetos pessoais.

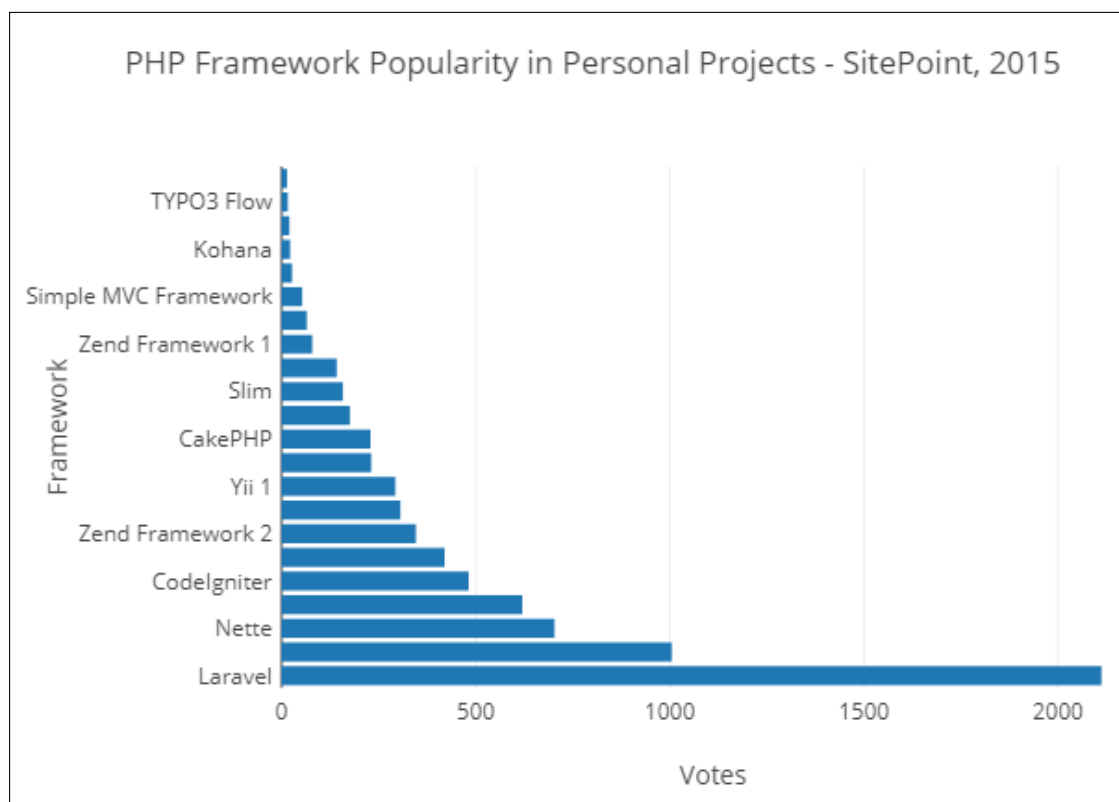


Figura 3 – Popularidade de Frameworks PHP para Projetos Pessoais

Fonte: sitepoint.com

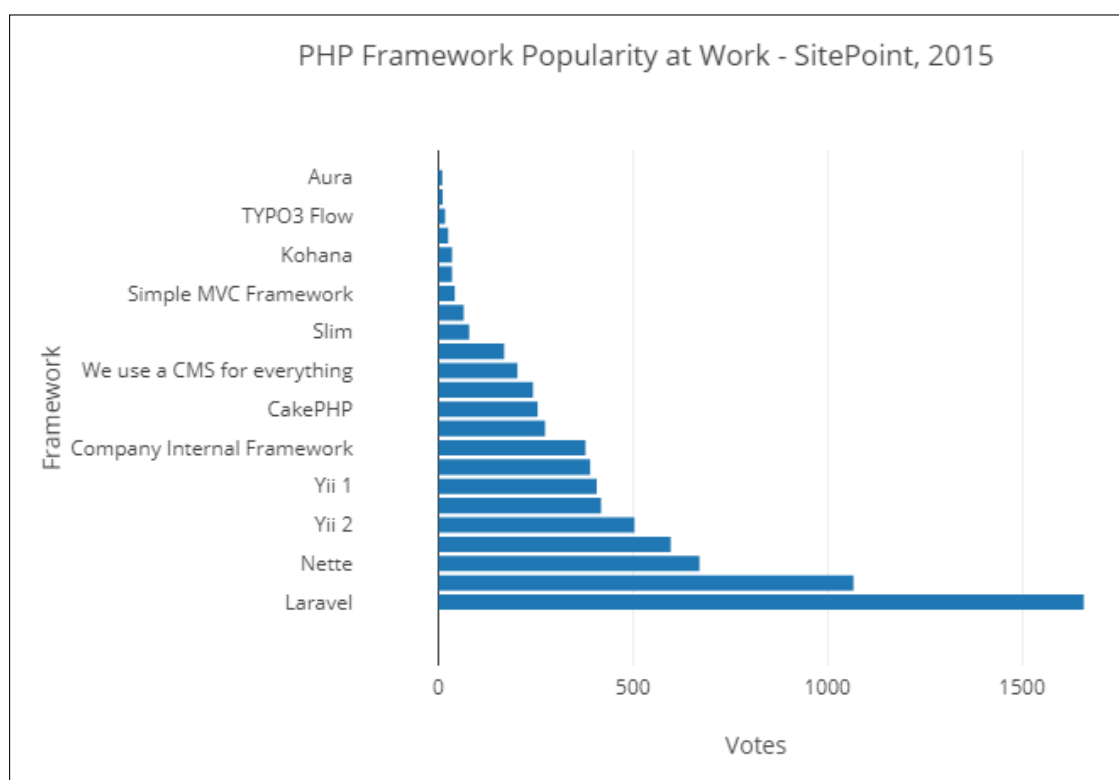


Figura 4 – Popularidade de Frameworks PHP para uso profissional

Fonte: sitepoint.com

O Laravel também aparece em primeiro nos *trends* de pesquisas do Google nos últimos 5 anos, como é visto na figura 5. O Google Trends reúne uma gigantesca base de dados para mostrar quais temas estão sendo pesquisados no momento, quais foram pesquisados nos últimos 30 dias (ou no último ano, ou nos últimos 5 anos, ou em qualquer data pré-definida desde 2004), quais temas relacionados à sua pesquisa as outras pessoas também estão buscando e outros *insights* (FARIAS, 2017).

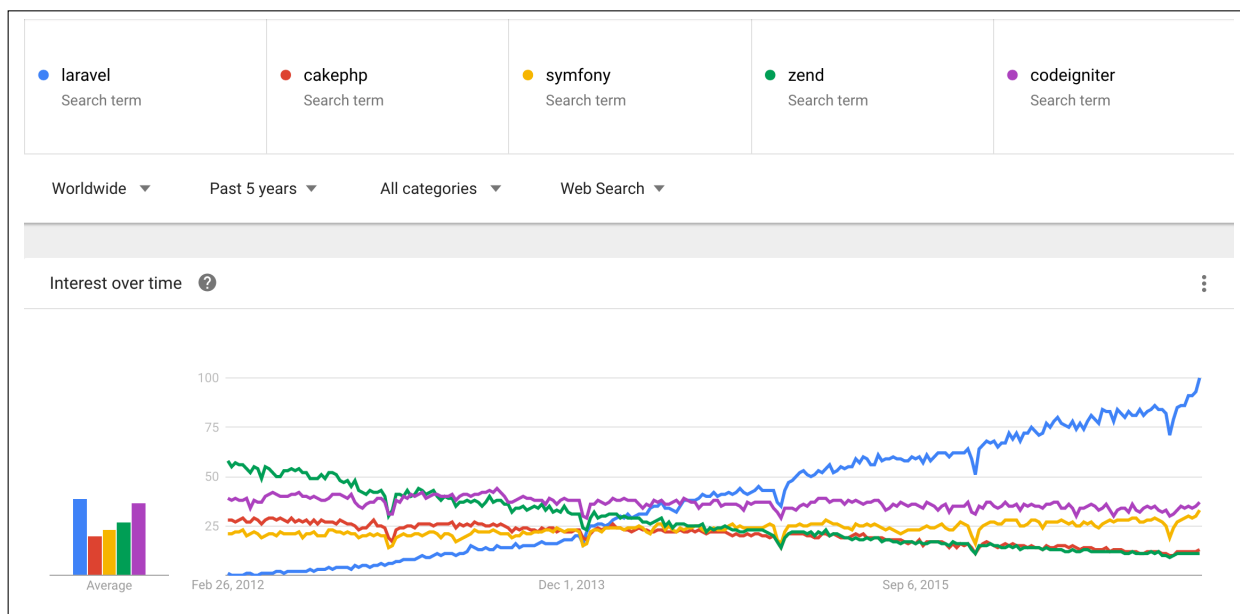


Figura 5 – Interesse ao longo do tempo dos principais frameworks PHP

Fonte: trends.google.com

De acordo com (DAS; SAIKIA, 2016), que fizeram um estudo comparando PHP procedural com CodeIgniter³ e Laravel, os autores chegaram à conclusão de que se uma aplicação de PHP fosse grande, o PHP puro ficava atrás em desempenho do que o Codeigniter, que por sua vez também não era capaz de executar tão eficientemente quanto o Laravel. Isto porque à medida que a aplicação ia ficando maior, o desempenho do Laravel ia melhorando e portanto, como resultado, a medição do tempo de execução dele era melhor do que os outros dois.

A figura 6 mostra os tempos de execução utilizando PHP puro, CodeIgniter e Laravel em um teste de desempenho de CRUD. Ou seja, quanto tempo levou para executar ações de *Create*, *Read*, *Update* e *Delete*, que na figura estão nas cores azul, vermelho, verde e lilás, respectivamente.

³ Disponível em <<https://codeigniter.com/>>

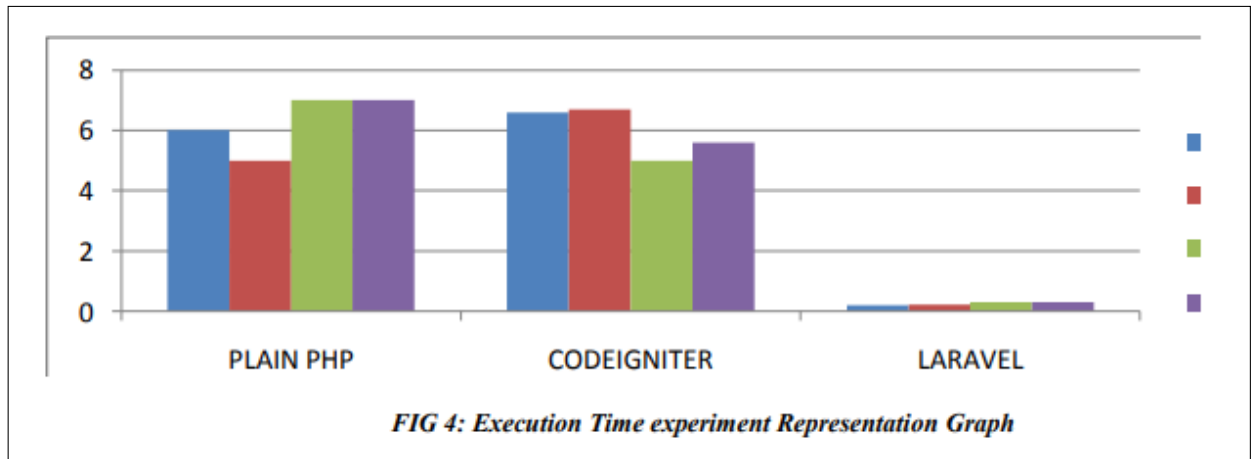


Figura 6 – Gráfico de Representação do Experimento do Tempo de Execução Entre PHP Puro, CodeIgniter e Laravel

Fonte: (DAS; SAIKIA, 2016)

Na internet existem vários comparativos entre os frameworks PHP mais populares. Um caso de teste feito pela (THINKMOBILES.COM, 2018) entre os framework Laravel e o Symfony⁴ buscou demonstrar a performance de ambos os kits de desenvolvimento. Foi feito um aplicação *single-page* simples para cada framework contendo uma única ação.

A tarefa foi extrair 1000 itens de um banco de dados (cada um com título, preço e estoque únicos), fazer 1000 pedidos para todos eles, atualizar quantidade em estoque no banco de dados e calcular o preço total do pedido.

Foram realizados testes de simulação de carga (usando o Apache Benchmark) com os dois maiores servidores Web da Internet: Apache e Nginx. Os resultados são vistos na tabela 1, onde o Laravel responde mais rapidamente que o Symfony em todos os testes.

Tabela 1 – Tabela de Testes de Carga com os frameworks Symfony e Laravel

Testing	Symfony (seconds)	Laravel (seconds)
Apache (10 requests, 1 user)	15.35	6.50
Apache (1 000 requests, 10 users)	638.56	202.66
NginX (10 requests, 1 user)	7.53	6.44
NginX (1 000 requests, 10 users)	658.73	211.85

Fonte: thinkmobiles.com

Já (RISHABHISOFT.COM, 2015) cita que o Laravel possui uma ampla gama de recursos como PHP nativo, mecanismo de modelagem leve chamado “Blade”, roteamento RESTful, entre outros. Processos comuns como sessões, cacheamento e roteamento são simplificados para o desenvolvedor. Como ele é construído sobre os componentes robustos do Symfony, o Laravel possui uma base sólida de código confiável e testado.

⁴ Disponível em: <<https://symfony.com/>>

Assim, segundo (RISHABHSOFT.COM, 2015), caso seja necessário desenvolver aplicações de nível corporativo ou APIs JSON de pequena escala, o Laravel pode ser usado. Ele é flexível para todos os tipos e tamanhos de projetos.

Após os estudos dos frameworks PHP mais populares do mercado e a análise das comparações feitas entre eles, entendeu-se que precisava-se de uma tecnologia que fosse popular, que suportasse a escalabilidade da aplicação e que desse suporte à construção de APIs.

Portanto, o Laravel, escolhido aqui, atendeu todos os requisitos e foi adotado para ser ferramenta auxiliar no desenvolvimento. A sua popularidade é comprovada pela sua primeira posição no Google Trends e por pesquisas populares que demonstram que tanto empresas quanto pessoas utilizam o framework, fazendo ele ser o mais famoso do mercado. As análises de comparação de desempenho aqui citadas também mostram que o framework suporta escalabilidade de desenvolvimento sem perder performance.

E, não menos importante, o framework possui toda programação pronta no que se diz respeito à comunicação entre diferentes sistemas, o que inclui o desenvolvimento de APIs. É a API que permite a comunicação entre o aplicativo Pró-Mamá e o Painel, possibilitando que o aplicativo consuma as informações necessárias.

2.5.1 Laravel

Laravel é um framework PHP livre e de código aberto que foi criado por Taylor Otwell e planejado para o uso em desenvolvimento de aplicações Web seguindo o padrão de projetos Model-View-Controller (MVC) (LARAVEL.COM, 2018).

O Laravel fornece componentes baseados em drivers, na qual os desenvolvedores podem usar para construir aplicativos de seu próprio jeito. Devido a esta característica, conseguiu aumentar sua popularidade e na sua versão 4 se tornou o framework com mais estrelas no Github, atestando sua notoriedade (BEAN, 2015).

O objetivo do framework, segundo o que consta na página inicial de seu website, é “tornar o processo de desenvolvimento agradável para o desenvolvedor sem sacrificar a funcionalidade do aplicativo”.

2.6 APPLICATION PROGRAMMING INTERFACE - API

API é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços (FOLDORG, 1995). É uma forma de integrar sistemas, possibilitando benefícios como a segurança dos dados, facilidade no intercâmbio entre informações com diferentes linguagens de programação e a monetização de acessos (FERNANDES, 2018).

Para a comunicação de dados entre aplicativo e o sistema Web, a API funciona como intermediadora, fazendo as entregas dos dados para o app quando requisitadas. É importante e fundamental, pois esta que permite que o aplicativo ganhe conteúdo dinâmico, que são as informações cadastradas pela equipe multiprofissional do Pró-Mamã.

3 TRABALHOS RELACIONADOS

Ao desenvolver este trabalho foram feitas pesquisas para investigar a existência de sistemas que se aproximassem ao funcionamento do Painel Pró-Mamá. Este trabalho se trata do desenvolvimento de um sistema informatizado para administração de conteúdo do aplicativo Pró-Mamá.

Neste capítulo serão apresentados e detalhados os sistemas WordPress, Drupal, Joomla, Moodle e OuvidorSUS. Todos estes são CMS (*Content Management System*), ou seja, são Sistemas de Gerenciamento de Conteúdo. Ao final do capítulo, na última seção, são feitas algumas considerações e exemplificado comparativos entre os sistemas. Também é explicado o porquê não foi utilizado nenhum dos trabalhos relacionados como ferramenta facilitadora para o canal de comunicação do Pró-Mamá.

3.1 WORDPRESS

O WordPress é um projeto de código aberto usado para criar sites, blogs ou aplicativos. Ainda segundo o site, 32% da internet usa o WordPress, de blogs amadores até os maiores sites de notícias. É um sistema de gerenciamento de conteúdo (CMS) para web, voltado principalmente para criações de páginas online e blogs ([WORDPRESS.ORG](https://wordpress.org/), 2018).

O WordPress surgiu em 2003. Conforme ([WORDPRESS.ORG](https://wordpress.org/), 2018), a necessidade de um sistema de publicação pessoal elegante e bem estruturado era clara mesmo naquela época. Hoje, o WordPress é baseado em PHP e MySQL, e licenciado sob a GPLv2. Contribuidores do WordPress trabalham no mundo inteiro e têm dedicado incontáveis horas para construir uma ferramenta que democratize o processo de publicação. O WordPress é um programa de código aberto que é tanto livre quanto grátis.

Entre as vantagens do Wordpress está a sua comunidade vasta e atuante. Uma imensa quantidade de pessoas mantém e lança atualizações do sistema constantemente na internet. Isto garante aos utilizadores do sistema uma maior segurança. Por ser um sistema muito utilizado, maiores são as tentativas das pessoas de invadirem esses sistemas. Entretanto, como há um frequente número de atualizações, os software possui as ferramentas necessárias para bloquear essas tentativas de invasão.

Outro ponto importante também de falar é que o Wordpress também possui seu aplicativo postado nas lojas de aplicativos PlayStore e AppStore. Assim como o Android, o aplicativo é um projeto de código aberto e qualquer um pode adicionar contribuições. Entre as funcionalidades que o aplicativo possui incluem: Configurar e gerenciar vários blogs, moderação de comentários, criar e editar publicações, incluindo categorias, tags e fotos, criar e editar páginas e receber notificações de novos comentários na barra de notificação do Android ([WIMBERLY](https://www.wimberly.com/), 2018).

O aplicativo do WordPress, entretanto, é focado na gerência do site e conteúdo do usuário. É mais um meio de gerenciar o conteúdo, que é feito através do app. Portanto, o Painel Pró-Mamá se diferencia no quesito de que o sistema gerenciador de conteúdo Web (Painel Administrativo) é acessado apenas por navegadores Web para fornecer conteúdo para ser visualizado no aplicativo. Ao contrário do WordPress, na qual o sistema gerenciador de conteúdo pode ser acessado por navegadores e por aplicativo, fornecendo conteúdo para ser visualizado exclusivamente por navegadores Web.

3.2 DRUPAL

O Drupal é um software de gerenciamento de conteúdo. É usado para criar muitos dos sites e aplicações que são usados todos os dias por pessoas. O Drupal tem recursos como fácil criação de conteúdo, desempenho confiável e excelente segurança. Mas o que o diferencia é sua flexibilidade; a modularidade é um dos seus princípios fundamentais. Suas ferramentas ajudam a criar conteúdo versátil e estruturado. O projeto Drupal é um software open source e portanto qualquer pessoa pode baixar, usar, trabalhar e compartilhar ele com outras pessoas. É distribuído sob os termos da GNU General Public License (GPL) ([DRUPAL.ORG](http://drupal.org), 2018).

Como é típico dos CMSs, um administrador de site configura inicialmente um site instalando o aplicativo Web principal do Drupal e faz a escolha de uma grande coleção de módulos. Os administradores do site não escrevem código, pois isso é feito pelos desenvolvedores do módulo. Depois que o site foi configurado, o Drupal permite que usuários não técnicos adicionem conteúdo e lidem com a rotina tarefas de manutenção ([CORLOSQUET et al., 2009](#)).

Conforme ([CORLOSQUET et al., 2009](#)) cada item de conteúdo no Drupal é chamado de nó. Nós geralmente correspondem mais ou menos diretamente para as páginas de um site. Os nós podem ser criados, editados e excluídos pelo conteúdo autores. Alguns módulos estendem os nós, por exemplo, um módulo de taxonomia permite atribuição de nós a categorias.

Ainda em conformidade com ([CORLOSQUET et al., 2009](#)), o Content Construction Kit (CCK) é um dos módulos mais populares e poderosos usado em sites Drupal. Permite que os administradores do site definam tipos de nós, chamados tipos de conteúdo e para definir campos para cada tipo de conteúdo. Campos podem ser campos de texto simples, datas, uploads de arquivos ou referências a outros nós, etc. Tipos de campos adicionais podem ser adicionados via módulos. Ao definir tipos de conteúdo e campos, o administrador do site deve fornecer ID, rótulo e descrição para tipos e campos de conteúdo. Além disso, o CCK permite para especificar as seguintes restrições nos campos: (i) Cardinalidade: os campos podem ser opcionais ou necessário, e pode ter uma cardinalidade máxima (ii) Domínio: os campos podem ser compartilhados entre um ou mais tipos de conteúdo; (iii) Intervalo: os campos podem ser do tipo texto, inteiro, decimal, float, data, anexo de arquivo ou referência de nó; campos de referência do nó podem ser restritos a nós de tipos de conteúdo específicos; campos de texto podem ser restritos a uma lista fixa de valores de texto.

O Drupal, assim como o Joomla e o WordPress, é um CMS cujo objetivo é a criação de páginas e aplicações Web, além da possibilidade de gerar conteúdo. O Drupal está entre os três principais CMS de código aberto ([PATEL; RATHOD; PRAJAPATI, 2011](#)).

3.3 JOOMLA

Joomla! é um sistema de gerenciamento de conteúdo (CMS) gratuito e de código aberto para publicação de conteúdo da web. Ele é construído em uma estrutura model-view-controller de framework para web que pode ser usado independentemente do CMS que permite a criação de aplicações robusta ([JOOMLA.ORG, 2018](#)).

Para ([RAHMEL, 2010](#)), o Joomla é uma das ferramentas CMS mais utilizadas devido sua facilidade de uso, as características profissionais e a interface intuitiva que contribuem para sua aceitação por instituições empresariais e educacionais. A ferramenta surgiu a partir do projeto Mambo, CMS desenvolvido pela Miro, companhia sediada em Melbourne, na Austrália. Problemas internos fizeram com que parte da equipe abandonasse o projeto e, a partir daí, em setembro de 2005, surgiu um novo gerenciador de conteúdo, o Joomla ([COELHO et al., 2011](#)).

O Joomla é usado no mundo todo em todos os tipos de site. Por exemplo, web sites e portais, intranets e extranets corporativas, revistas on-line, jornais e publicações, e-commerce e reservas on-line, aplicações do governo, pequenos sites de negócios, web sites sem fins lucrativos e organizacionais, portais comunitários, sites da escola e igreja, homepages pessoais ou familiares ([JOOMLA.ORG, 2018](#)).

Assim como o Painel Pró-Mamá o Joomla! é um sistema para gestão de conteúdos. O ponto é que assim como o Wordpress, o Joomla! é um sistema voltado para criação de páginas web e blogs. São sistemas mais abrangentes, genéricos. Atendem as necessidades das pessoas que querem publicar um site rapidamente ou também publicar informações, notícias ou qualquer outro conteúdo. O presente trabalho diferencia-se pelo fato de atender um contexto mais específico, o de facilitar o canal de comunicação do Programa Pró-Mamá da cidade de Osório através de um sistema de gestão de conteúdos para propagar conhecimento em saúde e prevenção.

3.4 MOODLE

O Moodle é um SGC - Sistema de Gerenciamento de Cursos. SGC são aplicações Internet/Intranet, rodam em um servidor e são acessados por um navegador web. O servidor está, normalmente, localizado em um departamento ou centro de processamento de uma Universidade, mas pode estar localizado em qualquer lugar do mundo ([FILHO, 2005](#)).

O Moodle é um ambiente virtual de aprendizagem usado nas instituições de ensino onde o professor pode criar um curso ou disciplina virtual onde somente alunos com acesso podem acessar. Nesse espaço online, os professores podem criar atividades, postar conteúdo, inserir informações, avaliar os alunos etc.

Esse software funciona em qualquer sistema operacional que suporte a linguagem PHP e é construído em módulos, o que permite adicionar, configurar ou remover funcionalidades. São as seguintes as ferramentas encontradas no Moodle: página para perfil dos alunos, inserção de avatares, fóruns, calendário, gestão de conteúdo, página de perguntas mais frequentes, criação de grupos, questionários e pesquisas, blogs, wikis, bancos de dados, sondagens, chat, glossários, ferramenta para construção de testes, avaliação em par e diários. Além disso, há as ferramentas administrativas, que permitem configurar o AVA, ativar edição, designar funções, atribuir notas, criar grupos, fazer backup, restaurar, importar, reconfigurar e emitir relatórios (PAIVA, 2010).

As principais vantagens do Moodle são principalmente em virtude de ser ele um sistema aberto, baseado em uma forte filosofia educacional, com uma comunidade de usuários crescente dia a dia que contribui para o desenvolvimento e apoio a novos usuários (FILHO, 2005). O Moodle atualmente também conta com aplicativo na PlayStore e AppStore, o que facilita o acesso aos cursos apenas tendo a URL do Moodle e seu login.

Fazendo um comparativo com o Painel Pró-Mamã, os dois sistemas tem o objetivo de gerenciar conteúdo e ajudar a uma causa social. O Moodle foca no ensino e aprendizado pelo meio virtual, sendo as disponibilizadas as informações e facilitando ao aluno que ele possa acessar os cursos de onde estiver. Já o Painel Pró-Mamã é focado em disseminar os conhecimentos sobre aleitamento materno para um território específico, a cidade de Osório. O objetivo do Painel é ser um meio pelo qual a equipe do Pró-Mamã possa estar inserindo as informações sobre a amamentação e fazendo esse conhecimento chegar mais longe.

3.5 OUVIDORSUS

O sistema OuvidorSUS (PORTALMSSAUDE.GOV.BR, 2017) é um sistema informatizado elaborado pelo Departamento de Ouvidoria Geral do SUS e desenvolvido pelo Departamento de Informática do SUS (DATASUS).

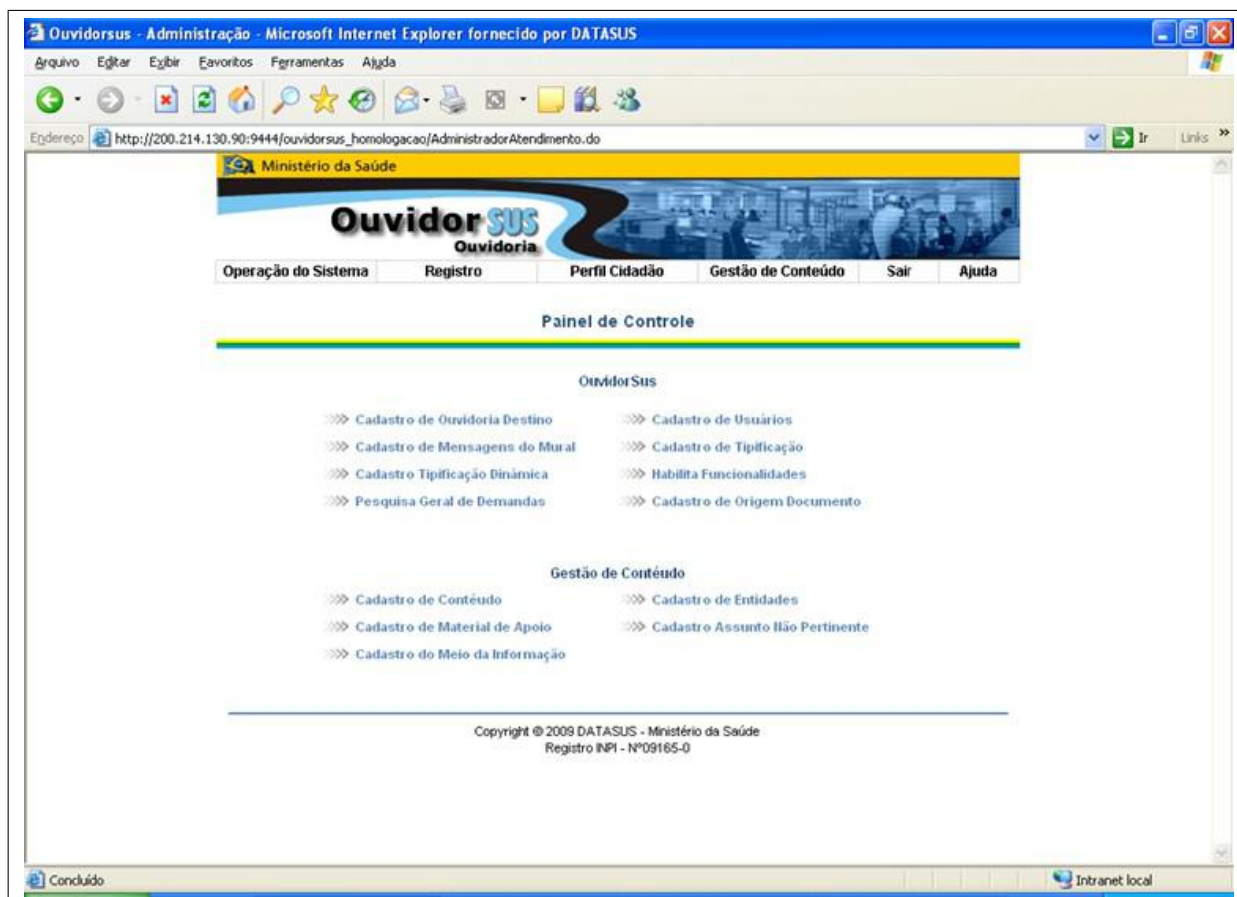


Figura 7 – Painel de Controle do OuvidorSUS

Fonte: <<http://slideplayer.com.br/slide/3393916/>>

O sistema permite a disseminação de informações, o registro e o encaminhamento das manifestações dos cidadãos. Possibilita ainda a troca de informações entre os órgãos responsáveis pela gestão do SUS, para adoção das providências cabíveis diante das manifestações recebidas.

Os objetivos do sistema OuvidorSUS são: a) atuar como ferramenta no processo de descentralização do Sistema Nacional de Ouvidorias do SUS; b) facilitar a democratização de informações em saúde; c) agilizar o processo de recebimento, encaminhamento, acompanhamento e resposta das manifestações recebidas; d) gerar relatórios gerenciais que auxiliem na melhoria contínua do Sistema Único de Saúde.

Um dos pontos positivos do OuvidorSUS é que ele possui níveis de acessos diferentes para os gestores, o que gera uma melhor organização e segurança ao sistema. Outro ponto positivo é que existem prazos já fixados para atender as demandas. Portanto, dependendo da urgência da demanda, ela demorará menos tempo para ser concluída do que uma não urgente. É uma ideia muito boa para que não se acumule demandas no sistema, prezando a conclusão do processo de comunicação com o cidadão.

Já os pontos negativos e que mais afetam o sistema do OuvidorSUS é a questão de sua interface com pouca usabilidade. Dispõe de muitas telas e muitos campos, o que torna o uso do

sistema complexo para acessá-lo. O fato do sistema já ser um pouco antigo também é um fator que explica sua interface pesada e o grande número de páginas.

Portanto, uma comparação que se faz do sistema OuvidorSUS com o Painei Pró-Mamá é que o sistema de ouvidoria do SUS, por receber uma grande quantidade de demandas, aparenta ser um software complexo e pesado que exige um treinamento por parte dos utilizadores para usá-lo. Quanto ao sistema Web do Pró-Mamá o objetivo é que os utilizadores e profissionais da saúde possam se adaptar ao sistema sozinhos, pois a proposta é que seja um sistema simples para uso e que cumpra a proposta de espalhar as informações para a comunidade do município de Osório em particular.

3.6 CONSIDERAÇÕES

Dentre os trabalhos aqui citados nesse capítulo vale ressaltar alguns dados. O site BuiltWith é uma ferramenta de geração de perfis de websites, geração de leads, análise competitiva e inteligência de negócios que fornece adoção de tecnologia, dados de comércio eletrônico e análise de uso para a Internet.

Segundo uma análise de (BUILTWITH.COM, 2018), os três CMS mais utilizados atualmente são WordPress, Drupal e Joomla!. A figura 8 demonstra o gráfico da distribuição para sites usando tecnologias CMS. O WordPress lidera com 51% de uso seguido do Drupal com 5% e o Joomla! aparece com 3% em terceiro lugar. O Moodle, que não aparece no gráfico possui um total de 0,56% de uso, ficando em 12º lugar.

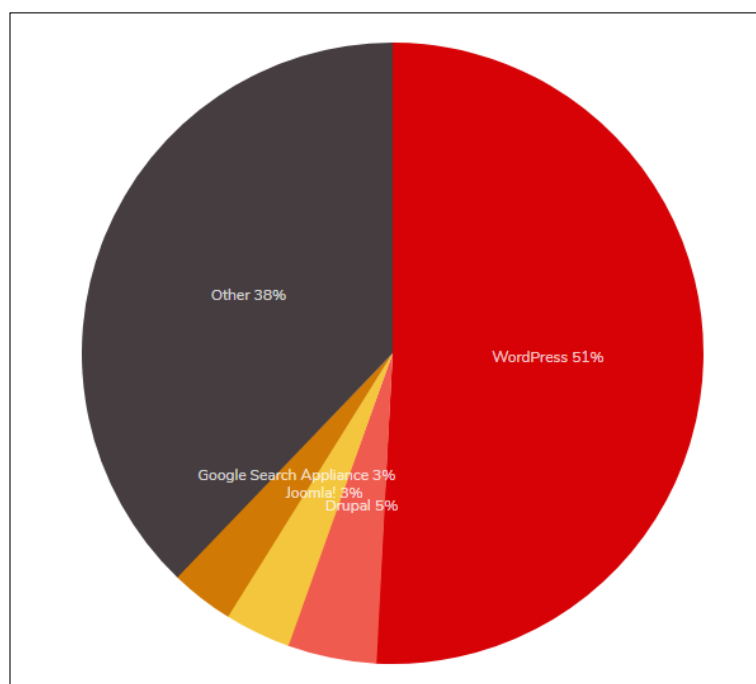


Figura 8 – Distribuição dos sites usando tecnologias CMS

Fonte: <builtwith.com>

De acordo com uma análise de desempenho de sistemas de gerenciamento de conteúdo feita por (PATEL; RATHOD; PRAJAPATI, 2011), comparando os sistemas Joomla, Drupal e WordPress, é provado que o Joomla, Drupal e WordPress são os CMS mais eficientes comparados com os outros, pois mostra maiores números de instalações, bons suportes de documentação e aparecem entre os primeiros no rankings de páginas.

O presente trabalho é desenvolvido utilizando as linguagens que são usadas pelos principais sites CMS hoje, sendo elas respectivamente PHP e MySQL. Entretanto, os CMS mais utilizados hoje em dia são voltados para criação de páginas Web e fornecer conteúdo para Web. São sistemas mais genéricos, que atendem diversos tipos de usuários e problemas. Seus conteúdos são acessados através de navegadores de internet, como por exemplo o Google Chrome e o Mozilla Firefox.

Tabela 2 – CMSs que possuem aplicativos móveis

	Joomla	Drupal	WordPress	Moodle
Possui Aplicativo Móvel nas lojas	Não	Não	Sim	Sim

Fonte: Autoria Própria

Como é visto na tabela 2, somente WordPress e Moodle possuem aplicativos publicados nas lojas de aplicativos (PlayStore e AppStore). O aplicativo do WordPress possibilita ao usuário escrever, editar e publicar posts em seu site. Já o aplicativo do Moodle é utilizado para navegar no conteúdo de cursos (mesmo offline), receber notificações instantâneas de mensagens e eventos, ver as notas dos cursos, entre outras funcionalidades.

Ambos aplicativos acessam as contas já existentes nos sistemas para gerenciar seus conteúdos. O Painel Pró-Mamá, entretanto, se diferencia exatamente nisso. O cadastro no Painel Administrativo é restrito apenas a equipe do Pró-Mamá, pois somente eles podem acessar o sistema Web. O cadastro feito pelo aplicativo é outro e portanto um usuário que se cadastrou no aplicativo não pode acessar o Painel e vice-versa. O usuário do aplicativo consome as informações somente pelo aplicativo.

O Painel Pró-Mamá é um sistema informatizado feito especificamente para o Programa Pró-Mamá, na qual tinha suas exigências e necessidades específicas. O uso de um dos CMS (WordPress, Drupal, Joomla) aqui citados para gestão do conteúdo não se aplicaria ao projeto. O Painel, aqui desenvolvido, atende a necessidade de publicar e gerir conteúdo exclusivamente para o aplicativo Pró-Mamá. É um sistema que busca cumprir com os requisitos que traçados pela equipe no início do projeto.

Já o OuvidorSUS, uma comparação que se faz com o Painel Pró-Mamá é que o sistema de ouvidoria do SUS, por receber uma grande quantidade de demandas, aparenta ser um software complexo e pesado que exige um treinamento por parte dos utilizadores para usá-lo. Quanto ao

sistema Web do Pró-Mamá o objetivo é que os utilizadores e profissionais da saúde possam se adaptar ao sistema sozinhos, pois a proposta é que seja um sistema simples para uso e que cumpra a proposta de espalhar as informações para a comunidade do município de Osório em específico.

Concluindo, o presente trabalho trata-se de um sistema de gestão de conteúdo com foco específico para fornecer e administrar conteúdo para o aplicativo Pró-Mamá. CMSs como Joomla, Drupal e WordPress são grandemente utilizados por pessoas para criação de seus próprios sites e conteúdos para Web. Não são voltados para produzir conteúdo para aplicativos móveis, como é o caso do Painel Pró-Mamá que propaga informações científicas em saúde e prevenção.

4 METODOLOGIA

4.1 ATIVIDADES

O primeiro passo do presente trabalho foi fazer um levantamento dos requisitos e necessidades dos profissionais do Programa que usariam o sistema Web. As pesquisas de levantamento se caracterizam pela entrevista, ou até um diálogo, com as pessoas cujo comportamento se deseja conhecer. Basicamente, procede-se a solicitação de informações a um grupo significativo de pessoas acerca do problema estudado para em seguida, mediante análise quantitativa, obter as conclusões correspondentes aos dados coletados (CARLOS, 1999).

A partir dos requisitos definidos, utilizou-se a metodologia de modelo incremental de *software* no processo de desenvolvimento do sistema. O modelo de processo incremental combina elementos dos fluxos de processos tanto lineares quanto paralelos. No primeiro incremento de um produto que utiliza o modelo incremental temos apenas o essencial do produto, ou seja, os requisitos básicos que devem ser atendidos para o *software* entrar em operação. Após o término do primeiro incremento o cliente utiliza e avalia esse incremento fornecendo posteriormente um resultado ou *feedback*. Com base nesse resultado fornecido pelo cliente o próximo incremento é planejado considerando a modificação do primeiro incremento, caso seja necessário, de acordo com o *feedback* do cliente. Após a liberação de cada incremento é realizado esse mesmo processo até que o produto esteja completo (PRESSMAN, 2005).

Visto isso, foi planejado com o grupo do Pró-Mamã que acontecessem reuniões periódicas para que houvesse um acompanhamento da construção tanto do Ambiente Web quanto do aplicativo. Assim, a equipe profissional poderia se habituar com ambos sistemas e também poderia testá-los para dar os retornos aos desenvolvedores.

Vale salientar que o perfil do grupo de profissionais são das mais diversas áreas, como Psicologia, Nutrição e Fonoaudiologia. Portanto, nenhum dos integrantes do grupo tem muito contato com tecnologias na área de informática. Tais dados são importantes para traçar a complexidade do sistema, que logicamente precisa que seja algo simples, porém funcional e possuir uma boa interface.

Após a etapa de análise de requisitos foi projetado como seria o Banco de Informações e a arquitetura de *software*. O modo como foi discutido isso foi através de uma revisão e análise da complexidade e o funcionamento do sistema.

Após então as etapas de requisitos e projeção de sistema começou a fase de desenvolvimento de uma primeira versão do Paineiro Administrativo. Esta dispunha de uma interface simples e objetiva. Foi adotado o Framework PHP Laravel como *back-end* e Bootstrap como *front-end* para auxiliar no desenvolvimento do sistema.

Ainda nesta fase de desenvolvimento foram implementadas também as rotas da API, essenciais para que o aplicativo já pudesse consumir as informações vindas da Equipe Pró-Mamã, inseridas por meio do Painel Administrativo. O mapa de rotas (Tabela 17) foi documentado devidamente e facilitou os desenvolvedores a melhor se comunicarem.

Em paralelo a todas essas fases acontecia simultaneamente a Manutenção do Sistema. A manutenção é um padrão de processo muito utilizado em ciclos de vida de desenvolvimento de *software* iterativas, na qual muitas mudanças ocorrem. O Painel Pró-Mamã passou por várias versões e testado diversas vezes antes da fase final de implantação (*deploy*).

A cada reunião eram definidos novos requisitos e anotados alguns pontos que deviam ser melhorados. Portanto, o diálogo direto com os envolvidos do projeto foi um método eficiente que ajudou na correção e manutenção do sistema. O *feedback* também foi importante para que o processo pudesse ser cada vez mais melhorado. O *feedback* é o mecanismo fundamental que permite que o cliente conduza o desenvolvimento diariamente e garanta que a equipe direcione as suas atenções para aquilo que irá gerar mais valor (TELES, 2004).

Já na etapa de conclusão do sistema aconteceu então o Lançamento oficial do Painel Administrativo, junto com o aplicativo. O lançamento marcou o fim das etapas de desenvolvimento. Entretanto, a manutenção é uma fase que não se encerra por se tratar de um *software*. *Softwares* são e devem ser continuamente mantencionados visando o pleno uso por parte do utilizador.

5 DESENVOLVIMENTO

Neste capítulo será apresentado como ocorreram as fases do processo de desenvolvimento. O Painel Pró-Mamá nada mais é do que um ambiente Web que compreende um Painel Administrativo e uma API para consumo das informações. É o painel que permite à equipe de profissionais da saúde disseminar as informações sobre aleitamento materno para o município de Osório e atender as dúvidas da comunidade. E a API é a parte que faz a entrega dessas informações até o aplicativo multiplataforma do Pró-Mamá.

Antes de explicar o desenvolvimento é necessário entender o cenário preexistente do Pró-Mamá, observando seus objetivos e entidades envolvidas. Após o entendimento do cenário preexistente será discutido o cenário atual, que é o encaixe do Painel com o aplicativo no cenário, investigando o que essa mudança proporciona e o impacto que isso gera.

Posteriormente será então visto o passo a passo do desenvolvimento do Painel. Desde a análise de requisitos até a manutenção do sistema. O objetivo deste capítulo é fornecer o detalhamento da construção de cada elemento produzido para o sistema Web.

5.1 CENÁRIO PREEXISTENTE

O Programa Municipal de Aleitamento Materno de Osório, de iniciativa do NASF e da Secretaria de Saúde do município, possui uma equipe de profissionais composta pelas fonoaudiólogas Ana Paula Sudbrack e Celina Rech Maggi, a nutricionista Gislaine Ferreira e a psicóloga Viviane Heckler.

O Pró-Mamá promove a formação de um profissional de referência em cada equipe de Estratégia de Saúde da Família – ESF, com o objetivo de atuar de forma mais próxima ao território e aos usuários, principalmente nos primeiros dias de vida, intervindo o mais breve possível. Tais profissionais são capacitados constantemente pela equipe multidisciplinar do Pró-Mamá. Foi realizada uma capacitação inicial com estes profissionais no intuito de balizar os conteúdos técnico-científicos e, atualmente, acontecem encontros bimestrais para discussão de casos e educação permanente com base na literatura científica. Estas referências atuam diretamente no acompanhamento da díade mãe-bebê em seu território adscrito através de visita domiciliar e atendimentos na Unidade de Saúde. Os casos mais complexos são discutidos e encaminhados para a equipe multiprofissional do Pró-Mamá que é composta por fonoaudiólogo, nutricionista e psicólogo.

Outras ações desenvolvidas pelo programa compreendem grupos, orientação, sala de espera, visitas domiciliares, inter-consultas e discussão de casos no intuito de que a introdução de fórmulas infantis ou leite de vaca seja a última opção.

A Figura 9 mostra como que ocorre hoje o processo do Pró-Mamá. A Secretaria da Saúde, por meio do Programa, atinge a comunidade de Osório. O Pró-Mamá, por meio de seus profissionais capacitados, orienta, consulta e visita as mães que estão em período de amamentação.

Portanto, nota-se que todo trabalho de atendimento e acompanhamento das mães e seus filhos acontece ainda de maneira presencial. Exige ou um encontro, ou uma reunião, ou uma visita domiciliar, para se sanar as dúvidas, esclarecimentos, obter conhecimentos. Desta maneira, é observado que o preexistente processo de divulgação de informações sobre o aleitamento materno sofre uma limitação quanto à propagação das informações a nível do município. Neste cenário, a comunidade de Osório como um todo, não é alcançada. Apenas uma parte dela.



Figura 9 – Cenário preexistente

Fonte: (CAVALCANTI et al., 2018)

5.2 CENÁRIO ATUAL

O cenário atual é uma soma ao Programa já existente. O trabalho em conjunto do aplicativo do Pró-Mamá com o Painel Pró-Mamá é ferramenta auxiliadora na divulgação e promoção do aleitamento materno no município de Osório, atingindo uma maior parcela de pessoas.

A figura 10 demonstra o cenário com o sistema e aplicativo integrados ao cenário já existente. O Programa, através do Painel Administrativo, inclui e recebe informações pertinentes que irão ser transmitidas aos aplicativos. A comunidade do município, com o *app* baixado, receberá estas informações, poderá submeter dúvidas e dar o *feedback* ao Pró-Mamá. Neste cenário, não só uma parcela da população é atingida, mas todo o município pode ter acesso às notícias e conhecimentos sobre o aleitamento.

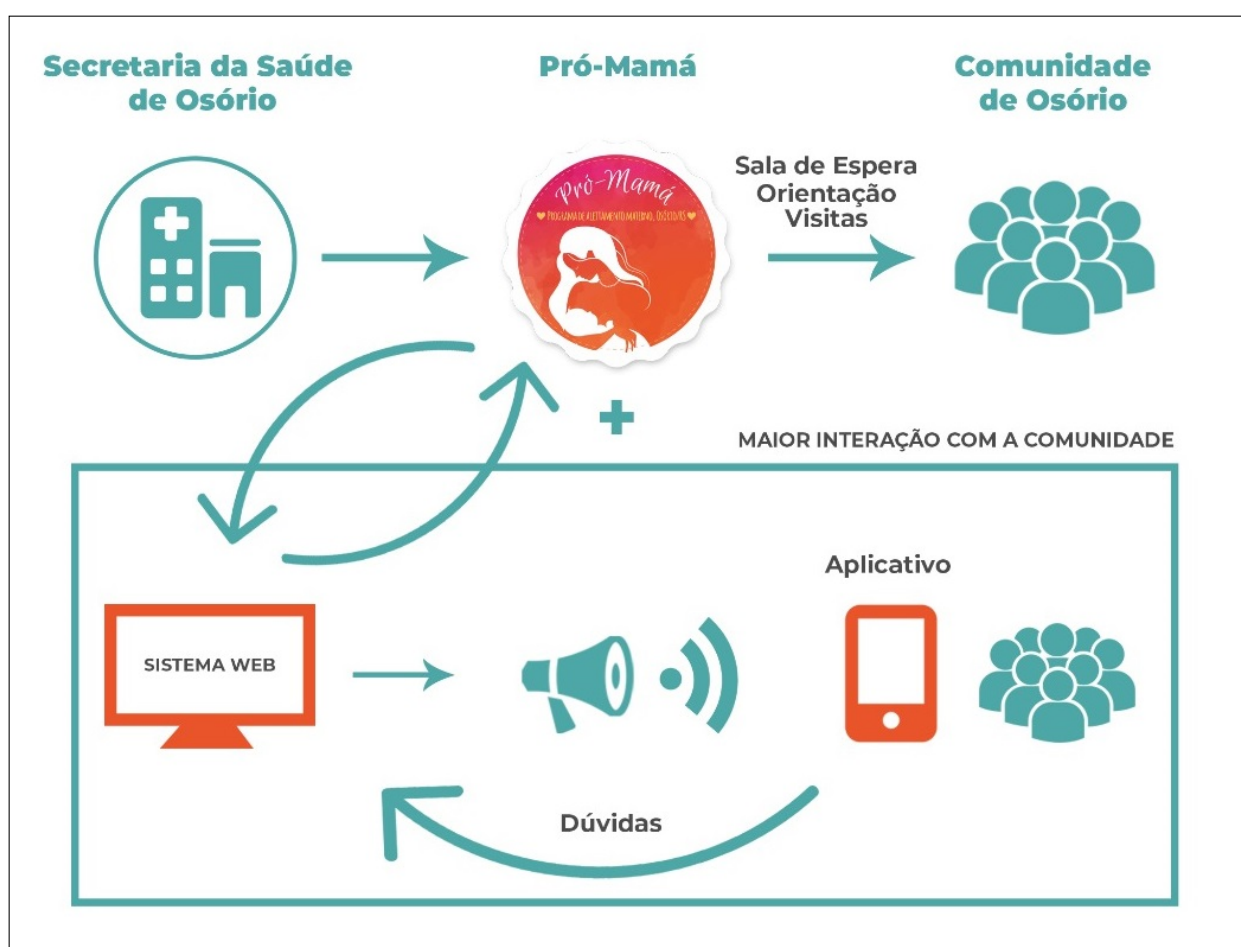


Figura 10 – Cenário atual

Fonte: (CAVALCANTI et al., 2018)

5.3 FUNCIONAMENTO DO SISTEMA

Como dito anteriormente, o sistema como um todo engloba o aplicativo móvel multiplataforma Pró-Mamá, desenvolvido pelo acadêmico José Paulo Oliveira Filho, e pelo Painel, que compreende API, Banco de Dados e Painel Administrativo, desenvolvido pelo autor.

Na figura 11, observa-se o esquema representativo do funcionamento. Com um servidor centralizado no núcleo de Tecnologia de Informação da Secretaria Municipal da Saúde de Osório, todas as regras de negócio, Painel Administrativo, API e Banco de Dados se encontram lá. A comunicação entre aplicativo Pró-Mamá e servidor se dá através da API, que atua como intermediadora no processo: Recebe as requisições, processa, consulta o Banco de Dados e retorna um resposta com as informações desejadas.

Já a comunicação entre o Cliente Web (um navegador Google Chrome, por exemplo) e o Servidor acontece através do Painel Administrativo. Através deste, os utilizadores do sistema (Equipe Pró-Mamá) podem alimentar o Banco de Dados com as informações e assim, fornecerem conteúdo para ser exibido no aplicativo. O painel é construído no intuito de facilitar e ampliar o canal de comunicação do Pró-Mamá, fazendo com que as informações alcancem um maior escopo da população.

Quanto às tecnologias utilizadas, pode-se notar ao lado de cada entidade, dentro do servidor, suas nomenclaturas. A API foi utilizada com os princípios da arquitetura REST, o banco de dados usado foi o MySQL e por último o Painel Administrativo, que é uma aplicação Laravel, na linguagem PHP.

Ao receber uma requisição, a API consulta o banco de dados, traz os dados que foram solicitados e os retorna na notação JSON, que é uma formatação leve de troca de dados. O aplicativo interpreta essa formatação e utiliza os dados da forma desejada.

Portanto, é visto que o funcionamento do sistema como um todo utiliza de tecnologias e padrões já muito utilizados na área da Tecnologia da Informação. O esquema representativo mostra de fato como acontece as transações e comunicação entre os dois sistemas (Web e Mobile). O aplicativo precisava de uma forma de consumir dados gerados pela equipe do Pró-Mamá e a solução foi utilizar tecnologias modernas e completas da área de desenvolvimento de sistemas. O detalhamento do processo de desenvolvimento será descrito em sequência.

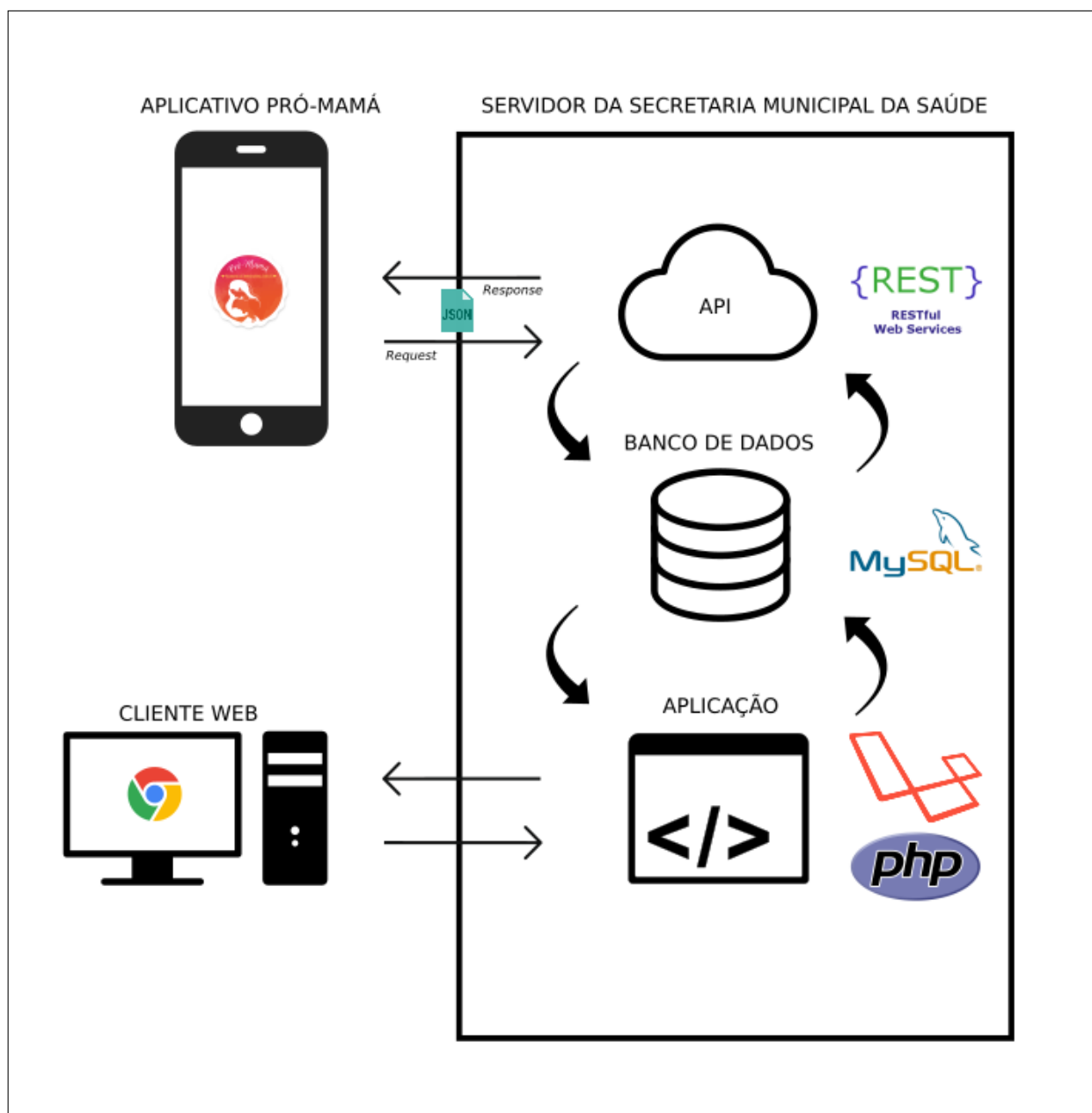


Figura 11 – Esquema Representativo do Funcionamento do Sistema

Fonte: (CAVALCANTI et al., 2018)

5.4 ANÁLISE DOS REQUISITOS DO PAINEL ADMINISTRATIVO

Nas primeiras reuniões que aconteceram entre os envolvidos do projeto foram definidos os requisitos do sistema. Através do interrogatório direto com a equipe, foram anotadas as principais funcionalidades que o Painel deveria ter, entre outras necessidades dos profissionais.

Após algumas reuniões com diálogos e discussões sobre o sistema, pôde então ser modelado os requisitos do Painel Administrativo. Nesta seção não constam os requisitos para construção da API, pois o desenvolvimento da mesma não foi baseado em requisitos solicitados pela equipe do Pró-Mamá e sim pelos requisitos do desenvolvedor do aplicativo. As informações detalhadas sobre a API constam na seção 5.9.

5.4.1 Requisitos Funcionais

Tabela 3 – Requisito Funcional 1

ID	Nome	Tipo	
RF01	Cadastrar Bairro	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar um Bairro. Solicita dados de : Nome do Bairro.			

Fonte: Autoria Própria

Tabela 4 – Requisito Funcional 2

ID	Nome	Tipo	
RF02	Cadastrar Posto de Saúde	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar um Posto de Saúde. Solicita dados de : Nome do posto, endereço e telefone. Cada posto possui um identificador único e possui uma referência de um bairro.			

Fonte: Autoria Própria

Tabela 5 – Requisito Funcional 3

ID	Nome	Tipo	
RF03	Cadastrar Informação	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar uma Informação. Solicita dados de : Título, corpo, idade alvo e outras opções da informação. Cada Informação possui um identificador único.			

Fonte: Autoria Própria

Tabela 6 – Requisito Funcional 4

ID	Nome	Tipo	
RF04	Cadastrar Notificação	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar uma notificação. Solicita dados de : Título, texto e idade alvo da notificação. Cada notificação possui um identificador único.			

Fonte: Autoria Própria

Tabela 7 – Requisito Funcional 5

ID	Nome	Tipo	
RF05	Cadastrar Dúvida Frequente	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar uma dúvida frequente. Solicita dados de : Título e texto da dúvida. Cada dúvida frequente possui um identificador único.			

Fonte: Autoria Própria

Tabela 8 – Requisito Funcional 6

ID	Nome	Tipo	
RF06	Cadastrar Termos de Uso	Essencial	Desejável
Descrição			
Permite o utilizador a cadastrar um título e um texto de Termos de Uso.			

Fonte: Autoria Própria

Tabela 9 – Requisito Funcional 7

ID	Nome	Tipo	
RF07	Responder dúvida do usuário	Essencial	Desejável
Descrição			
Menu Fale Conosco. Permite o utilizador a visualizar e responder a dúvida de um usuário. O utilizador pode escolher se quer que a dúvida respondida seja vista por todos ou só para o usuário que perguntou. Cada dúvida do usuário possui um identificador único.			

Fonte: Autoria Própria

Tabela 10 – Requisito Funcional 8

ID	Nome	Tipo	
RF08	Alterar Senha	Essencial	Desejável
Descrição			
Permite o utilizador a trocar a senha da conta. Solicita dados de: Senha atual, nova senha e confirmação da senha nova.			

Fonte: Autoria Própria

Tabela 11 – Requisito Funcional 9

ID	Nome	Tipo	
RF09	Envio de e-mail na interação do Fale Conosco	Essencial	Desejável
Descrição			
Possibilita que, quando chegue uma nova dúvida de usuário, seja enviado um e-mail para o utilizador do Painel. E quando o utilizador responder a dúvida é enviado um e-mail para o usuário que fez a pergunta.			

Fonte: Autoria Própria

Tabela 12 – Requisito Funcional 10

ID	Nome	Tipo	
RF10	Pré-Visualizar Informação	Essencial	Desejável
Descrição			
Possibilita que o utilizador do Painel tenha uma prévia, no cadastro de Informação, de como irá ficar o conteúdo na tela de um celular.			

Fonte: Autoria Própria

5.4.2 Requisitos Não Funcionais

Tabela 13 – Requisitos não funcionais

ID	Descrição
RNF01	O sistema deve ser desenvolvido na plataforma Web.
RNF02	O sistema precisa de internet para ser acessado.
RNF03	O sistema deve fornecer uma API para consumo das informações.
RNF04	O sistema deve utilizar o banco MySQL.
RNF05	O sistema deve estar sempre disponível (alta disponibilidade) para as requisições.

Fonte: Autoria Própria

5.5 ARQUITETURA DE SOFTWARE

Para o desenvolvimento da primeira versão da aplicação foram revisados os requisitos do sistema (referenciado no subcapítulo 5.3) em vista de estudar como o sistema poderia ser implementado.

Após uma revisão sistemática dos requisitos, pôde então ser elaborado uma prévia de modelo de arquitetura de software. A figura 12 foi uma representação de como aconteceria o fluxo de informações entre os diferentes dispositivos. A primeira coisa que se sabia é que o Painel, ou aplicação Web, iria ser acessado de duas formas: Através de um navegador Web ou via aplicativo Pró-Mamá (através da API). Ambas formas precisariam acessar o servidor Web na busca de consumir os dados do Banco. O navegador acessando através de uma interface Web (Painel Administrativo) e o aplicativo acessando por meio da API.

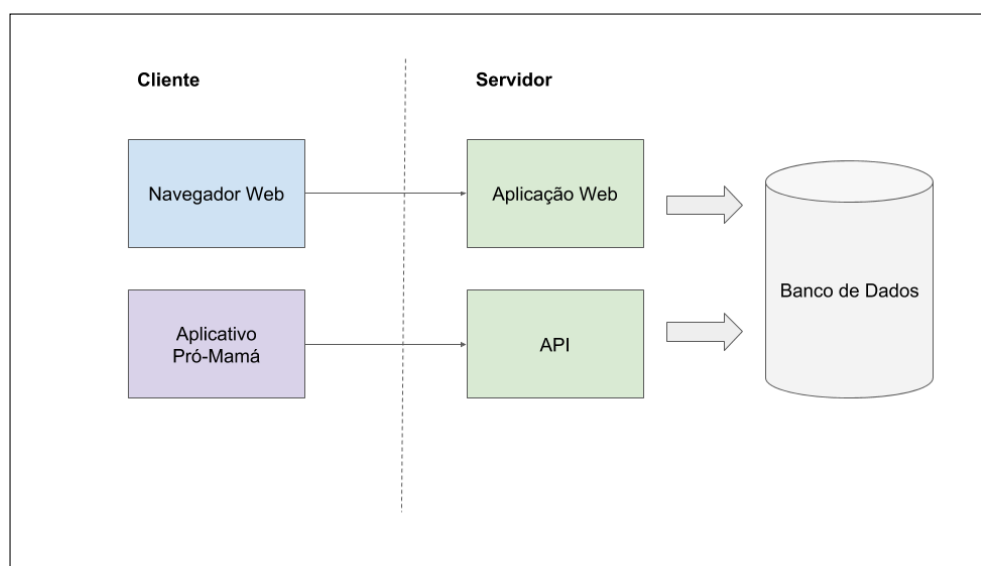


Figura 12 – Modelo de Arquitetura de Software

Fonte: Autoria Própria

Após esse planejamento de software, foi então decidido que o sistema iria ser implementado em linguagem PHP e com auxílio do Framework Laravel. Chegou-se à decisão pelo fato de o Laravel já incluir em sua biblioteca suporte fácil e prático nas rotas de APIs e todo o auxílio também para o desenvolvimento Web. Outros motivos também já foram mencionados no capítulo 2.5, que discutem sobre o quão importante é a ajuda de kits de desenvolvimento no processo de construção de um software.

Com as decisões feitas, o próximo passo no processo de desenvolvimento do sistema foi dado: Projetar o Banco de Dados. O próximo subcapítulo discute e busca detalhar como foi o processo de planejar e implementar o Banco.

5.6 BANCO DE DADOS

Para esta etapa de desenvolvimento foi analisado novamente os requisitos em busca de se anotar todas as entidades envolvidas. Uma entidade é uma representação de um conjunto de informações sobre determinado conceito do sistema. Após a listagem de entidades, buscou-se procurar relações entre elas, de forma que pudesse ser construído o modelo lógico do banco de dados e o diagrama de classes.

A primeira seção a ser apresentada será a de modelo lógico, onde conterà um diagrama e uma explicação sobre o mesmo. Já na segunda seção será detalhado o diagrama de classes e suas relações com o modelo MVC. E por último será falado do uso de *Triggers* no banco de dados.

5.6.1 Modelo Lógico

Um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD (HEUSER, 2009). Como o brModelo oferece a possibilidade de transformar um modelo conceitual em um modelo lógico, foi realizada então a conversão de modelos e atribuído os campos para cada tabela. Obteve-se o seguinte esquema pelo brModelo:

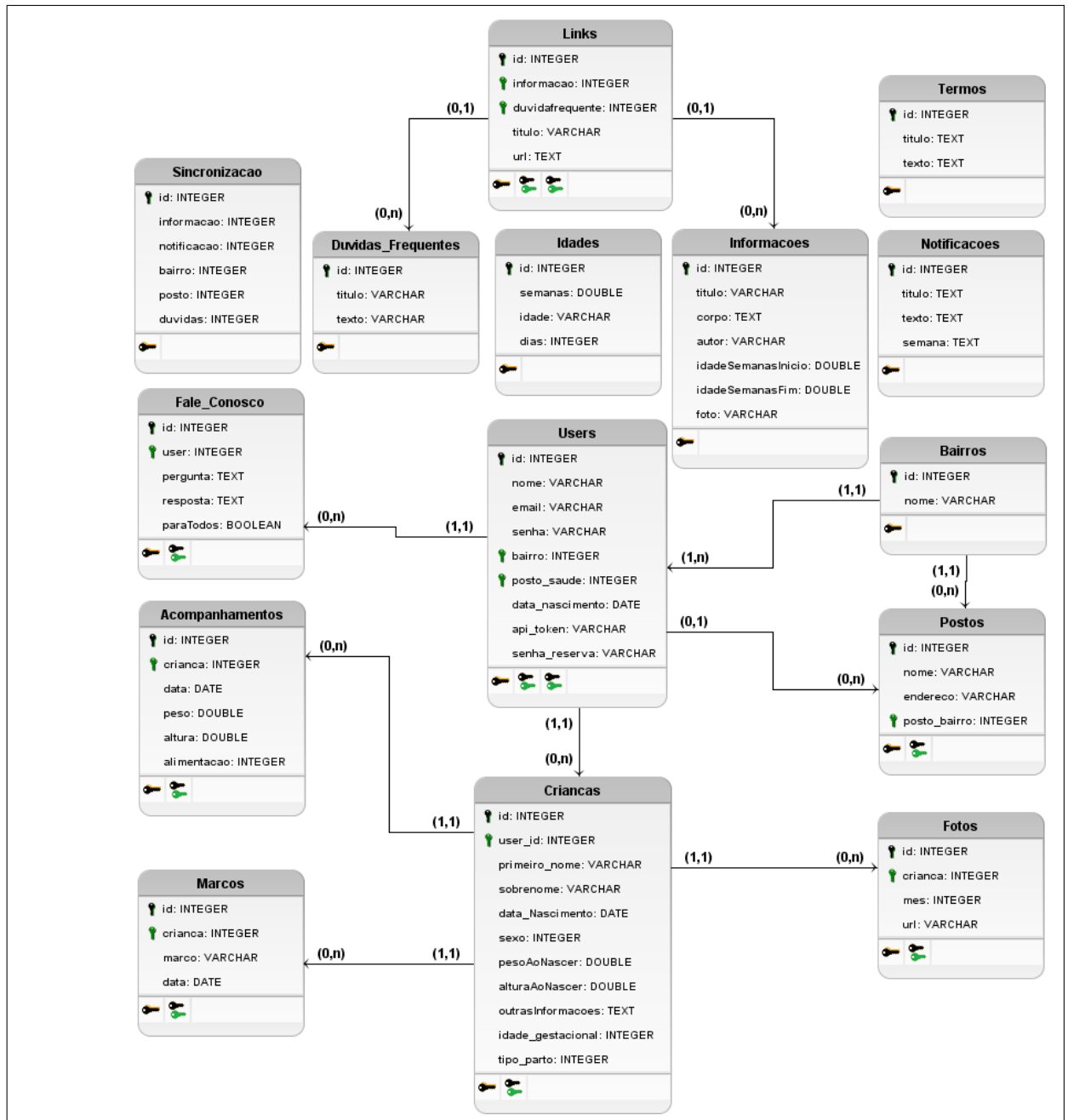


Figura 13 – Modelo Lógico do Banco de Dados

Fonte: Autoria Própria

Com o modelo lógico desenhado, alterações começaram a serem aplicadas. Como por exemplo a atribuição de campos, chaves primárias e chaves estrangeiras para as tabelas. A figura 13 apresenta o esquema final, com os atributos e chaves já configurados. Com tudo nomeado e referenciado corretamente, só o que precisou fazer foi gerar o modelo físico, que nada mais é do que o código de criação do banco de dados. Com o código em mãos restou apenas executá-lo em um SGBD e o banco já estava pronto para uso.

5.6.2 Diagrama de Classes

O diagrama de classes (Figura 14) representa as interações estáticas e as classes envolvidas no sistema, permitindo também identificar as hierarquias das classes, representadas por heranças e agregações. Este diagrama permite visualizar os dados que serão armazenados e manipulados pelo sistema (SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008).

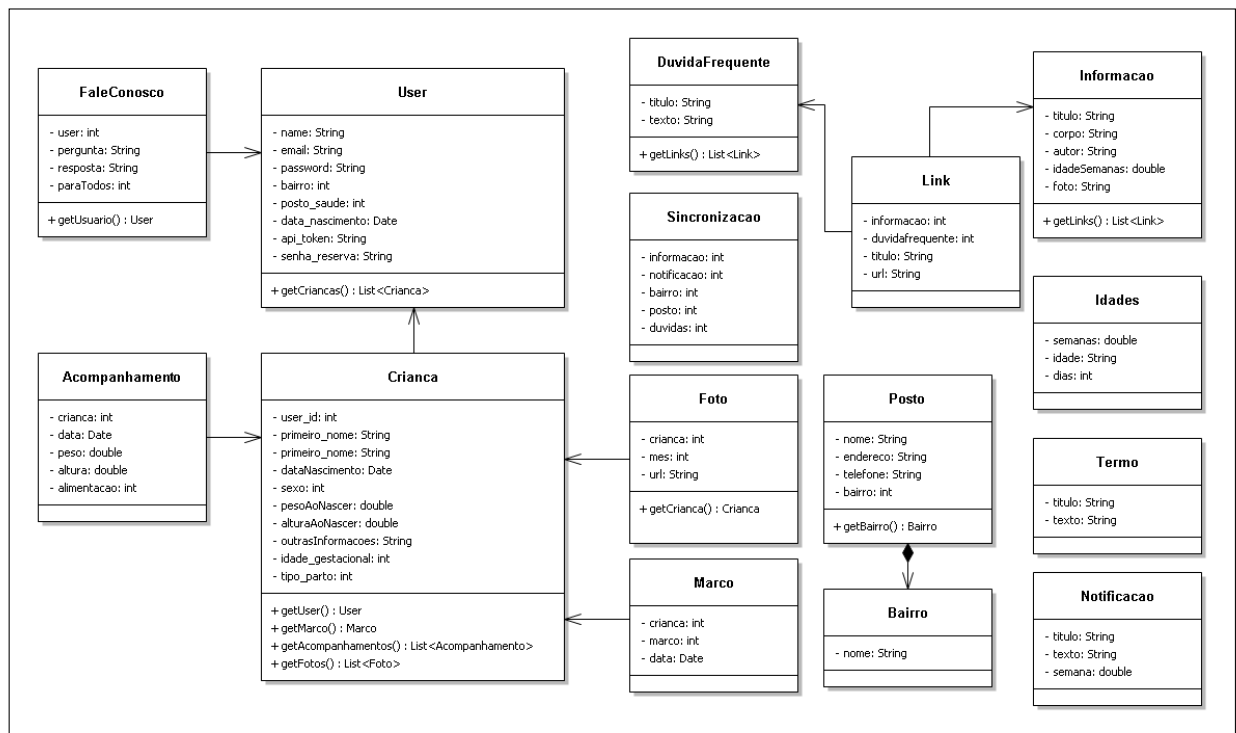


Figura 14 – Modelo de Arquitetura de Software

Fonte: Autoria Própria

Na figura 14 são exibidas as classes da camada Modelo usadas pelo Painel. São essas as classes que são “linkadas” com sua respectiva tabela do banco de dados e que permitem, através delas, inserir, recuperar, editar e deletar dados.

Na camada de Controle encontram-se as classes que tratam as requisições vinda do cliente (navegador Web ou aplicativo Pró-Mamá) e que retornam algum tipo de informação para ele. Essa informação pode ser uma *View* (HTML, CSS) ou um JSON (API).

Um exemplo de classe da camada de Controle pode ser visto na figura 15, onde é exibido o *controller* da classe Informação, no meio. Como é visto, esta classe possui todos os métodos de controle necessários para invocar sua classe modelo e fazer alguma ação. O primeiro método, *index*, corresponde a página inicial do menu Informações no Painel Administrativo, onde é retornada a *View* “listar” contendo a tabela com os registros de informações.

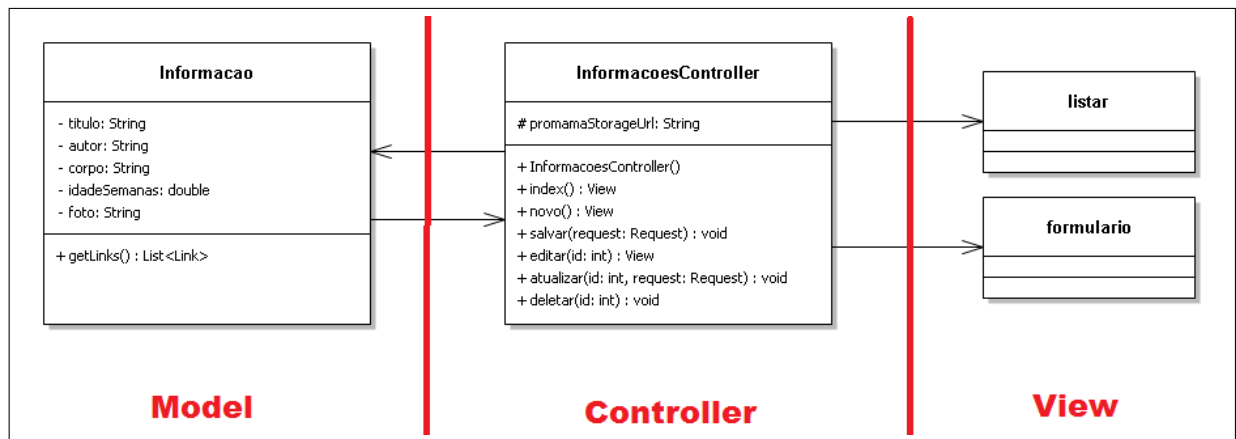


Figura 15 – Relação da Classe Controladora com a Modelo e a Visualização

Fonte: Autoria Própria

Para cada classe modelo, portanto, existe uma classe controladora. As demais classes controladoras do sistema seguem o mesmo padrão da figura 15. O padrão, assim sendo, nada mais é do que a arquitetura MVC. As classes de controle possuem os métodos de CRUD, invocam sua respectiva *Model* para gerir os dados e renderizam uma *View*.

A classe controladora de API, entretanto, é exceção desse padrão citado no parágrafo anterior. Esta classe não retorna *Views* e sim um JSON. Essa classe é responsável por tratar as requisições vindas do aplicativo móvel Pró-Mamá, processá-las e retornar uma resposta.

5.6.2.1 Triggers

É importante mencionar também que durante o processo de desenvolvimento do software observou-se necessidade da criação de gatilhos, ou *triggers*, no Banco de Dados para ajudar os desenvolvedores com as tarefas repetitivas.

No processo duas *triggers* foram criadas, a de funcionalidade de Esqueci a Senha e a de Sincronização. A primeira foi criada pelo motivo de que quando o usuário definia sua nova senha, devia ser apagada do Banco a senha reserva que estava lá só para ele fazer o processo de recuperação da conta. Basicamente precisava-se de um disparo automático para setar como nulo o campo de senha_reserva no registro do usuário.

A segunda função, de Sincronização, foi criada por motivos de que precisava ser incrementado na tabela de Sincronização um contador para cada alteração feitas nas outras tabelas. A tabela de Sincronização, no contexto geral, é usada para que o aplicativo que estava sem internet possa sincronizar as informações de seu Banco de Dados local com o do servidor.

5.6.3 Considerações

Concluindo, projetar o banco de dados foi uma tarefa primordial e necessária no processo de desenvolvimento, uma vez que o aplicativo Pró-Mamá e o Painel Administrativo iriam com-

partilhar do uso deste. Foi um planejamento que precisou usar uma visão técnica e analítica, em busca de soluções concretas.

5.7 CONFIGURAÇÃO DO AMBIENTE

Antes de iniciar a fase de implementação foi configurado todo o ambiente para o Laravel e suas dependências. O servidor que hospedou a aplicação durante a fase de desenvolvimento foi a própria máquina *Desktop* do autor. O sistema operacional utilizado no desenvolvimento foi o Windows 7 Ultimate. Para hospedar o projeto foi baixado o XAMPP, um pacote com os principais servidores de código aberto do mercado. Foi extremamente útil por não ter que precisar instalar um servidor de cada vez no sistema.

Seguindo a documentação do Laravel¹, portanto, foi instalado o mesmo na máquina. Com o Framework instalado só o que precisou ser feito para criar um novo projeto foi rodar um comando pelo terminal. O seguinte comando foi dado: *laravel new promama*. Após o disparo do comando, as dependências foram baixadas e uma chave aleatória para aplicação foi gerada (padrão de segurança do Laravel para dar segurança às sessões dos usuários).

Em seguida, algumas configurações precisaram ser feitas. Como por exemplo, colocar as credenciais do servidor de Banco de Dados, credenciais do servidor de e-mail, nome da aplicação, localidade e fuso horário. Essas informações foram cruciais de serem definidas devido ao fato do funcionamento do sistema depender dessas configurações.

5.7.1 Controle de Versão

Para gerenciar as versões do projeto foi utilizado o Git, um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.

O Laravel, em seus projetos, já inclui suporte para versionamento do Git. Isto pelo fato de o projeto já vir com um arquivo *.gitignore*, que é uma lista que ignora qualquer arquivo que ainda não foi acompanhado pelo repositório.

A interface utilizada para gerenciar o versionamento foi o Github Desktop. O Github é um serviço web que oferece diversas funcionalidades extras aplicadas ao Git. É um grande repositório de projetos versionados com Git.

Com o projeto configurado e versionado, a etapa de configuração do ambiente para desenvolvimento estava pronta. O passo seguinte foi adotar ferramentas de testes para ajudar no desenvolvimento da aplicação, antes do início da implementação.

¹ Disponível em <<https://laravel.com/docs/5.6>>

5.7.2 Ambientes de Testes

Para fazer testes com as funcionalidades do Painei usou-se duas ferramentas diferentes: O Postman e o MailTrap. Essas ferramentas foram usadas para testar, respectivamente, as rotas da API e os envios de e-mail.

O Postman é uma aplicação que permite realizar requisições HTTP a partir de uma interface simples e intuitiva, facilitando o teste e depuração de serviços REST. Já o MailTrap é um servidor de SMTP *fake* para testar, visualizar e compartilhar e-mails enviados a partir dos ambientes de desenvolvimento e preparação, sem enviar spams a clientes reais.

Esses softwares foram de grande auxílio no processo de desenvolvimento do Painei Pró-Mamá. Os testes eram necessários de serem realizados e as ferramentas aqui citadas ofereceram grande ajuda através de suas interfaces simples.

5.8 PAINEL ADMINISTRATIVO

Com a etapa de pré-implementação pronta, deu-se início ao desenvolvimento. Foram criada as classes modelos e controladoras de cada entidade do banco e também definidas as rotas Web.

As classes modelos foram implementadas primeiro. Elas são importantes por terem a ligação com sua respectiva entidade do Banco de Dados. No Laravel, para criar uma classe modelo, precisa-se abrir o terminal na pasta raiz do projeto e dar o seguinte comando: *php artisan make:model NomeDaClasseModelo*. Um padrão que o Laravel recomenda é sempre criar a classe com nomenclatura no singular e a tabela no banco no plural.

Depois de rodar os comandos e criar todas as classes modelos, o próximo passo foi criar as classes controladoras. Essas classes, no contexto Web, carregam consigo todos os métodos necessários para as “coisas funcionarem”. É nessas classes que se fazem as consulta ao Banco de Dados, o retorno de uma view (HTML) ou um retorno de um JSON por exemplo.

Para criar uma classe controladora no Laravel é o mesmo processo da modelo, a diferença está na sintaxe do comando, que fica: *php artisan make:controller NomeDaClasseControladora*. Um padrão, de autoria própria, utilizado na nomenclatura, foi adicionar ao final de cada nome de classe a palavra *Controller*, como por exemplo: *BairrosController*. Isto facilita para o desenvolvedor quando vários arquivos estão abertos na IDE, permitindo ele se achar facilmente.

A interface escolhida para o layout do Painei Pró-Mamá foi um template de *Dashboard* feita com Bootstrap e disponibilizada na versão grátis com a licença MIT. A licença MIT permite que qualquer pessoa que obtém uma cópia do software e seus arquivos de documentação associados pode lidar com eles sem restrição, incluindo sem limitação os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, vender cópias do software. As únicas condições impostas para tanto são apenas manter o aviso de *copyright* e uma cópia da licença em todas as cópias do

software.

Como a *Dashboard* já fornecia uma página de Login, foi simples fazer a integração da interface com o sistema. O utilizador, para acessar o Painei, preenche os campos de e-mail e senha e clica em entrar. A classe controladora de Usuário faz a verificação da combinação das credenciais, consultando o Banco de Dados através da *model* de Usuario, e caso o Login esteja correto ele carrega o Painei, se não estiver é exibida uma mensagem de erro abaixo do campo de e-mail.

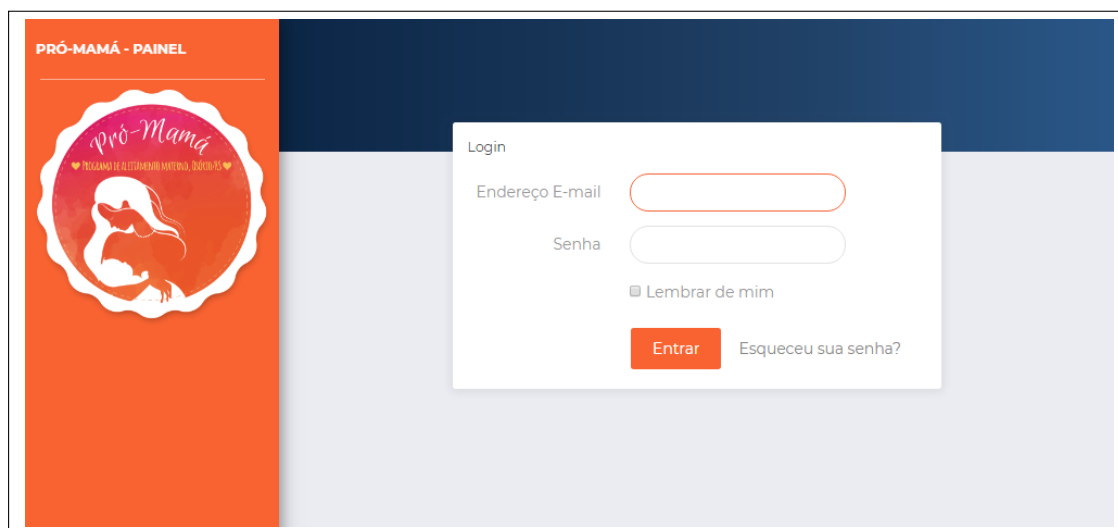


Figura 16 – Tela de Login

Fonte: Painei Pró-Mamá

Por ter uma boa página de documentação e ser feita usando o Bootstrap, a *Dashboard* foi adaptada rapidamente às necessidades do projeto. A primeira tarefa foi desmembrar o código HTML dela e criar a partir disso um layout fixo (figura 17) para ser compartilhado pelas outras páginas Web.

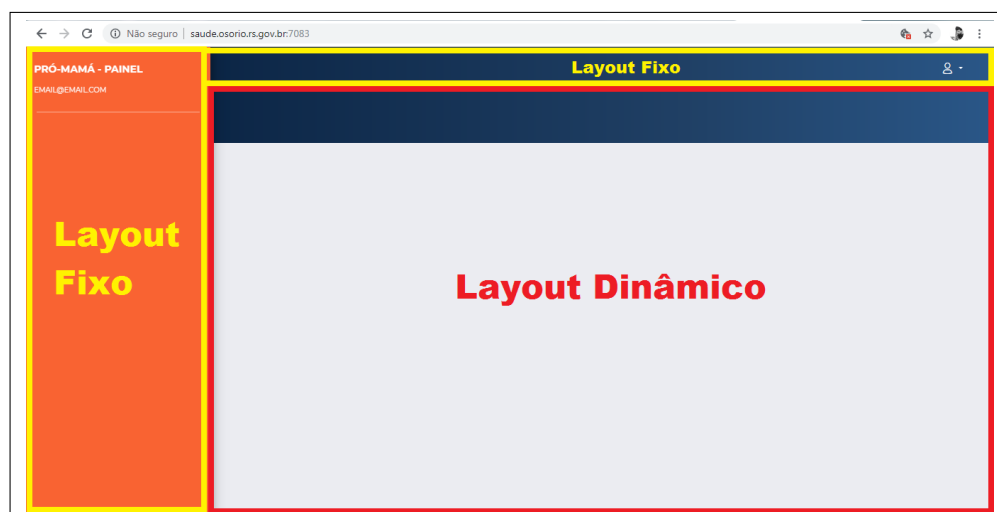


Figura 17 – Layout Fixo e Layout Dinâmico

Fonte: Painel Pró-Mamá

Neste layout fixo incluiu-se os menus Bairros, Postos, Informações, Notificações, Fale Conosco, Dúvidas Frequentes e Termos de Uso. Cada um desses itens era um link que levava a uma página diferente. Um exemplo de uma dessas páginas pode ser visto na figura 18, referente à tabela Informações.

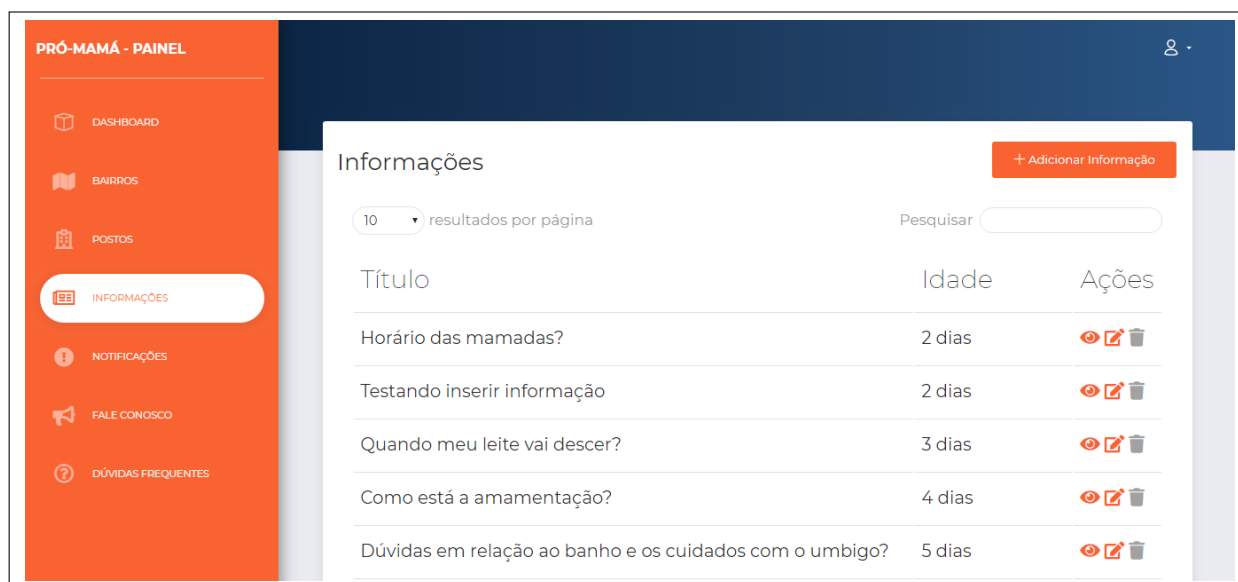


Figura 18 – Tela de Listagem de Informações

Fonte: Painel Pró-Mamá

Como dito no parágrafo anterior, para cada link do menu há uma página. Quando um item do menu é clicado, a primeira informação que vem à tela é uma tabela listando registros da tabela selecionada. E na parte superior direita de cada uma das tabelas possui um botão “Adicionar”, que redireciona para um formulário para cadastro de um registro.

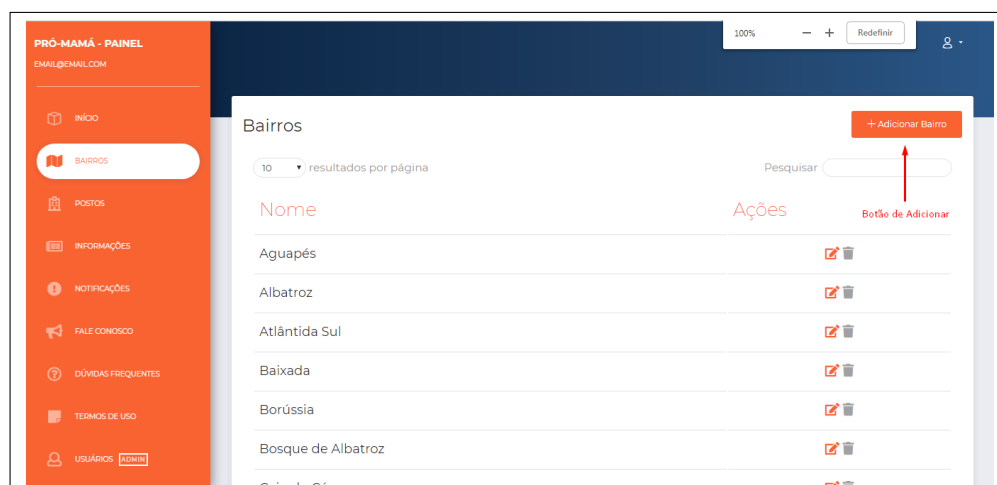


Figura 19 – Botão de Adicionar

Fonte: Painel Pró-Mamá

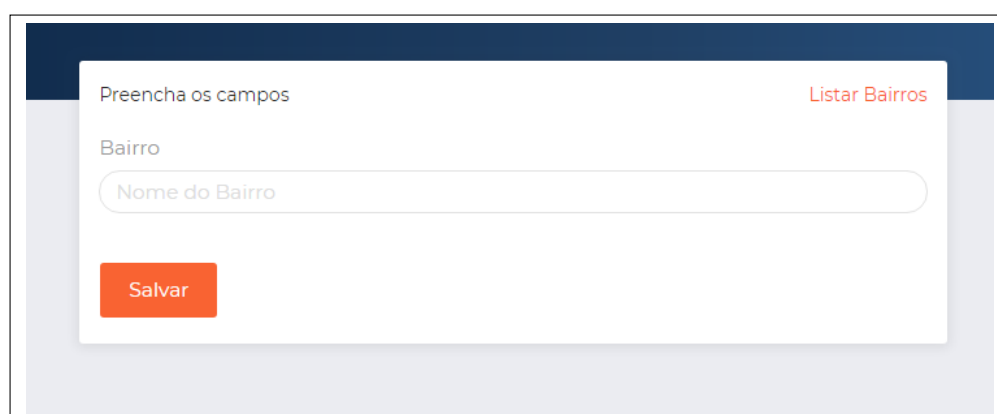


Figura 20 – Formulário de Adicionar Bairro

Fonte: Painel Pró-Mamá

Nos próximos subcapítulos serão circunstanciados as principais funcionalidades do Painel Administrativo. Como dito, constarão apenas as principais funcionalidades, pois Cadastro de Bairro, Posto de Saúde, Dúvida Frequente e Termos de Uso possuem semelhante funcionamento que é exibir conteúdo vindo do Banco em forma de tabelas de registros e possibilitar a inclusão, edição e exclusão dessas.

5.8.1 Informação

O menu Informação é uma das principais e mais importantes funções dentro do Painel Administrativo. É por meio desta funcionalidade que a equipe do Pró-Mamá tem a possibilidade de cadastrar informações que serão exibidas no aplicativo, alcançando determinadas idades.

O registro de informações foi um requisito que sofreu várias alterações durante o processo de desenvolvimento. O modelo de formulário feito no começo do projeto possuía campos de

Título, Autor, Corpo do Texto, Idade de Início, Idade de Fim e Foto. Entretanto, em reuniões com os profissionais do Pró-Mamã, foi discutido, por exemplo, a necessidade de haver campos para inserir Links externos da Web em uma informação. Também foi visto que precisava de Links para Dúvidas Frequentes.

Outro ponto que também foi alterado foi o das idades alvo. No começo havia um *range* de idades, na qual a informação cadastrada aparecia para crianças nesse intervalo de idade. Porém, isso também teve que ser alterado, fazendo com que cada informação tivesse apenas uma idade alvo em específico.

Essas foram as alterações sofridas no formulário de Informação durante o processo de desenvolvimento. O formulário final, mostrado na figura 21, foi testado e bem aceito pelos utilizadores do Painei.

A imagem mostra a interface de um navegador web acessando o sistema "PRÓ-MAMÁ - PAINEL". No topo, há uma barra de endereço com o URL "saude.osorio.rs.gov.br/7083/informacoes/novo". À esquerda, há um menu lateral com opções como INÍCIO, BARRIOS, POSTOS, INFORMAÇÕES (destacado), NOTIFICAÇÕES, FALE CONOSCO, DÚVIDAS FREQUENTES, TERMOS DE USO e USUÁRIOS. O formulário principal, intitulado "Preencha os campos", contém os seguintes elementos: um campo "Título" com uma sub提示 "Título da Informação (Máximo 60 caracteres)"; um campo "Corpo da Informação"; um campo "Idade" com uma sub提示 "Idade" e um menu suspenso atualmente mostrando "recém-nascido"; um link "Mais opções" com uma seta para baixo; e dois botões na base: "Salvar" em laranja e "Pré-visualizar" em cinza.

Figura 21 – Formulário de Adicionar Informação

Fonte: Painei Pró-Mamã

5.8.1.1 Links externos e internos

Como visto anteriormente, uma Informação precisava ter a opção de poder inserir Links, fossem eles externos (Web) ou internos (Dúvida Frequente). Por estes motivos que a tabela Link foi criada, como vista no subcapítulo 5.5.2. Portanto, para cada informação cadastrada é disponibilizado campos para inserir Links.

No formulário, para acessar tal recurso, é necessário clicar no item “Mais Opções” para ver os campos. Logo abaixo desse item aparecem dois textos, um embaixo do outro, que oferecem opção de um botão “+” para inserir um novo formulário de link no HTML.

Figura 22 – Adicionar Link Externo

Fonte: Painel Pró-Mamá

Figura 23 – Adicionar Link Interno

Fonte: Painel Pró-Mamá

Na figura 22 nota-se que para inserir um novo link externo necessita-se preencher um título e a URL. Já para inserir um link interno, figura 23, aparece a lista de Dúvidas Frequentes para a seleção. Caso o utilizador não queria mais inserir o Link é só clicar no “-” do lado direito do formulário.

Para implementar essa funcionalidade foi usada a linguagem JavaScript, uma linguagem de programação que permite criar conteúdo dinâmico no lado do cliente. Nesse contexto especificamente, a linguagem permitiu que formulários de Links dinamicamente fossem criados ou excluídos no HTML sem a necessidade de “pedir” nada para o servidor.

5.8.1.2 Pré-Visualizar

A opção Pré-Visualizar foi uma funcionalidade que surgiu como ideia do próprio autor para o contexto do cadastro da Informação. A função de Pré-Visualizar basicamente mostra uma prévia de como fica a atual informação sendo preenchida na tela de um smartphone.



Figura 24 – Pré-Visualização de Informação

Fonte: Painel Pró-Mamá

Este recurso permite que o utilizador que está cadastrando ou editando uma Informação possa ter uma ideia de como ficará composto os elementos na tela do *smartphone*. É uma função que busca aprimorar a interação entre utilizador e sistema, levando uma melhor experiência de uso pro usuário.

5.8.2 Fale Conosco

O menu Fale Conosco é outro importante item a se falar. Este mecanismo permite o utilizador do Painel, um profissional capacitado da área da saúde, a responder uma dúvida de um usuário. Ao clicar no menu “Fale Conosco”, o usuário é redirecionado para uma página que lista as dúvidas ainda não respondidas. Caso queira visualizar, editar ou até mesmo excluir dúvidas já respondidas, o utilizador do Painel só precisa clicar no link superior à direita da tabela.

Para responder uma dúvida ainda não lida, o utilizador do Painel deve clicar no ícone de responder, em laranja, assim como é mostrado na figura 25. Isso o levará para a página da dúvida a ser respondida, figura 26, contendo nome do usuário solicitante, data da solicitação e a pergunta em si. Logo abaixo dessas informações encontra-se uma área de texto para o profissional da saúde colocar uma resposta.

Outro fato importante de comentar é que ao responder uma dúvida, o profissional pode optar por selecionar a opção “Responder Dúvida Para Todos”, que faz com que a pergunta se torne pública, ou seja, todas as pessoas com o aplicativo podem ver aquela dúvida. Esta funcionalidade foi definida em reunião e foi discutido que isso seria muito útil para que certas dúvidas, pertinentes, já ficassem de mostruário para as pessoas.

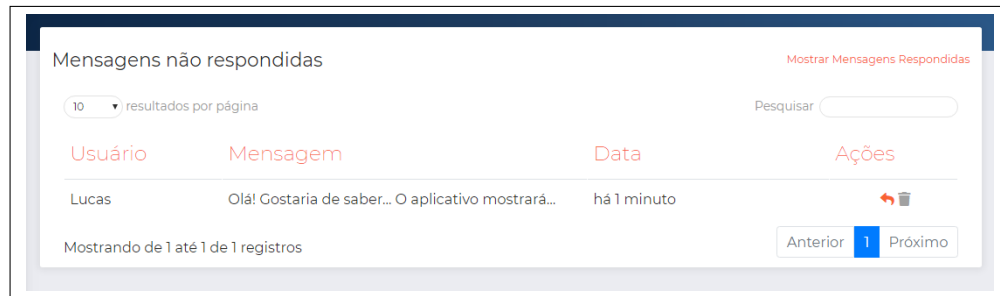


Figura 25 – Tela Principal do menu Fale Conosco

Fonte: Painel Pró-Mamá

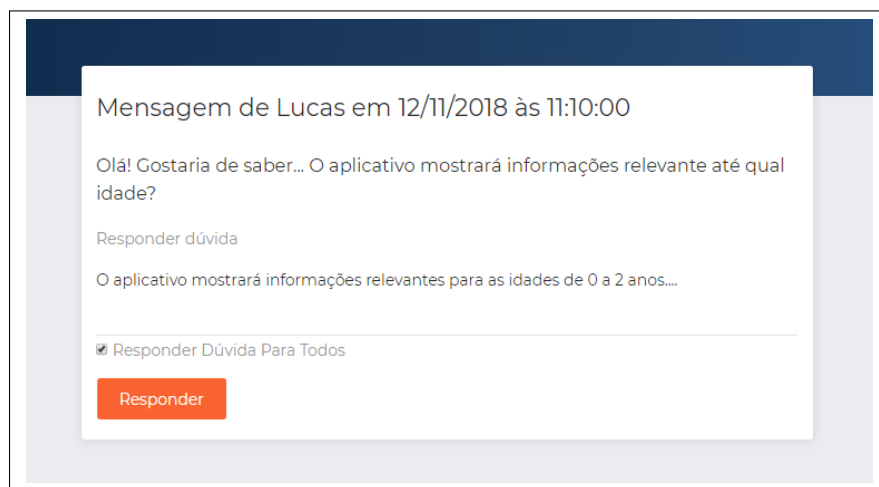


Figura 26 – Tela de Responder uma Dúvida (Menu Fale Conosco)

Fonte: Painel Pró-Mamá



Figura 27 – Tela de Dúvida respondida para o usuário no aplicativo



Figura 28 – Tela de Dúvida respondida para todos no aplicativo

5.8.3 Notificação

O menu notificação também será discutido aqui por ser importante ferramenta para o aplicativo. É através dessa funcionalidade que os utilizadores do Painel podem cadastrar notificações que irão aparecer no aplicativo do Pró-Mamá. A página principal, assim como as outras páginas, é uma lista de registros de notificações.

Para cada registro de notificação são exibidas opções para editar ou excluir a mesma. Já para cadastrar uma notificação deve-se clicar no botão “Adicionar Notificação” na parte superior direita da tabela. O utilizador será redirecionado para o formulário de cadastro, como mostra na figura 29, e preencherá os dados Título, Texto da Notificação e Idade Alvo e uma caixa de seleção de “Notificar o quanto antes”.


A imagem mostra um formulário de notificação com o título "Preencha os campos". O formulário contém os seguintes elementos: um campo de texto para "Título da notificação" com o placeholder "Título da notificação"; um campo de texto para "Texto da notificação" com o placeholder "Texto da notificação"; um campo de seleção para "Idade alvo:" com a opção selecionada "recém-nascido"; uma caixa de seleção desativada para "Notificar o quanto antes"; um link "Ajuda" com um ícone de seta vermelha; e um botão laranja "Enviar".

Figura 29 – Formulário de Notificação

Fonte: Painel Pró-Mamá

A caixa de seleção “Notificar o quanto antes” é uma opção disponibilizada para que quando o utilizador cadastre a notificação, esta apareça o quanto antes no aplicativo. Acontece da seguinte maneira: Se uma notificação for configurada para ser mostrada “o quanto antes”, o aplicativo irá agendar todas as notificações configuradas dessa forma para serem mostradas uma a cada hora. Ou seja, se existem três notificações configuradas dessa forma, o usuário irá recebê-las dessa forma, por exemplo:

- 10:00 - Recebeu a lista de notificações;
- 11:00 - 1ª notificação configurada para ser mostrada o quanto antes é mostrada;

- 12:00 - 2ª notificação configurada para ser mostrada o quanto antes é mostrada;e
- 13:00 - 3ª notificação configurada para ser mostrada o quanto antes é mostrada.

Do contrário, a notificação será mostrada ao usuário quando uma de suas crianças atingir aquela idade (as notificações de idades passadas não são mostradas). E um detalhe a ser notado é que cada criança recebe sua notificação, não importando quantas crianças estão cadastradas.

Para personalizar as notificações, pode-se usar as palavras coringa %ARTIGO%, %NOMEDACRIANCA% e %NOMEDOUSUARIO% no título e no texto da notificação. Como por exemplo, o título e texto cadastrado da maneira representada na tabela 14 irá aparecer no aplicativo como mostra na tabela 15.

Tabela 14 – Modelo do Cadastro de Notificação com o coringa

Título	Texto
Olá %NOMEDOUSUARIO%!	Como %ARTIGO% %NOMEDACRIANCA% está?

Fonte: Autoria Própria

Tabela 15 – Modelo de como a Notificação aparece no aplicativo

Título	Texto
Olá Renata !	Como a Júlia está?

Fonte: Autoria Própria

Ao utilizar esta funcionalidade, deve-se prestar atenção no uso do coringa %ARTIGO%, que será substituído na frase final pelo artigo masculino “o” ou o artigo feminino “a” de acordo com o sexo da criança. E além do coringa do artigo, também deve-se prestar atenção no uso do coringa %NOMEDOUSUARIO%, pois o nome do usuário é opcional dentro do aplicativo e o usuário não precisa necessariamente fornecer esta informação para utilizar o aplicativo. Assim, se o nome não estiver preenchido na seção “Perfil da Mãe” dentro do aplicativo, na notificação o coringa será substituído por “mãe”. O uso de qualquer coringa se adapta ao início da frase, capitalizando a primeira letra.

A função de cadastro de Notificações foi importante no contexto do aplicativo por se tratar de um lembrete ao usuário do aplicativo. A ideia é que aquela ou aquele que baixou o aplicativo e cadastrou uma criança receba informações via notificações para checar o aplicativo e receber as informações pertinentes ao desenvolvimento da criança.

5.8.4 Página Inicial (Home)

A página inicial do Painei é onde o utilizador visualiza alguns cartões informativos sobre o mesmo. São três cartões que existem. O primeiro é o de dúvidas ainda não lidas, que serve

como lembrete para responder as dúvidas. O segundo é o de quantas pessoas se cadastraram no aplicativo, ao todo. E o terceiro e último é um *card* que possui uma tabela das últimas cinco informações postadas.

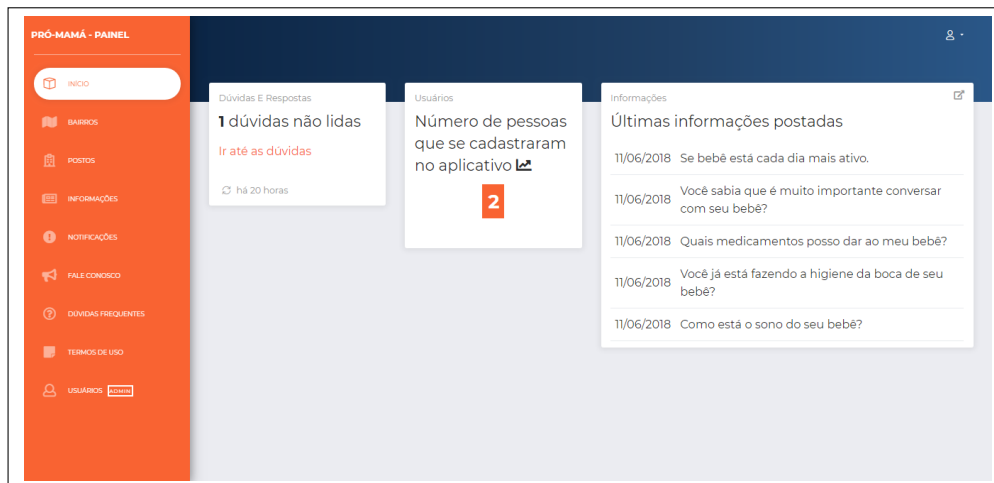


Figura 30 – Página Inicial do Painel

Fonte: Painel Pró-Mamá

5.8.5 Envio de e-mail

O Laravel, em seu arquivo de configuração, já traz as variáveis do servidor de e-mail consigo para serem preenchidas. Após colocar tais credenciais só o que se precisa fazer é criar uma classe de e-mail personalizada contendo um corpo, um assunto e um remetente como atributos dela. Com a classe feita, é só colocar onde quiser no código uma linha para o envio de e-mails.

No sistema são três possibilidades de envios de e-mail. A primeira é quando um usuário envia uma dúvida pelo aplicativo. A segunda possibilidade é quando o utilizador do Painel responde a dúvida e a última é a recuperação de senha do aplicativo. As três possibilidades são detalhadas na sequência.

5.8.5.1 Cadastro de Dúvida

Quando um usuário do aplicativo acessa o recurso de Fale Conosco, ele tem a possibilidade de fazer uma pergunta para a equipe multiprofissional do Pró-Mamá. Ao mandar a pergunta, o sistema dispara um e-mail para o endereço eletrônico “master”, que é de posse do Pró-Mamá. A funcionalidade é uma soma ao sistema, uma vez que o serviço de e-mail aqui serve mais como lembrete, ou notificação, para os envolvidos no projeto.



Figura 31 – E-mail de notificação de nova dúvida

Fonte: Mailtrap

5.8.5.2 Resposta de Dúvida

Na comunicação feita entre utilizador do Painele e usuário do aplicativo, através da função do menu Fale Conosco, observou-se que seria de grande utilidade também que houvesse um disparo de e-mail para o usuário que enviou a dúvida.

Quando um profissional responder a dúvida, o usuário receberá em seu e-mail o aviso de que ela foi respondida. No corpo do e-mail conterà a pergunta feita por ele e uma resposta junto. A figura 32 é uma captura de tela do e-mail que chega ao usuário.



Figura 32 – Método de Responder Dúvida de Usuário

Fonte: Autoria Própria

5.8.5.3 Esqueci a senha

Por último, a funcionalidade que também utiliza envio de e-mails é a opção “Esqueci a senha” que se encontra na tela de login do aplicativo Pró-Mamá. É importante deixar claro que essa é uma funcionalidade requisitada pelo aplicativo e não pelo Painele.

Esta opção visa ser um meio de ajuda ao usuário caso ele tenha esquecido sua senha de acesso. Ao clicar nesta opção, um e-mail é enviado ao endereço eletrônico registrado na hora do cadastro pela pessoa. As recomendações dadas são de esperar até 1 minuto e caso nada tenha chegado na caixa principal, checar a caixa de spam.

A pessoa irá receber uma senha provisória de acesso de 6 dígitos por e-mail e terá que logar no aplicativo com esta. Ao entrar, será solicitado a ela que seja cadastrada uma nova senha, definitiva. Ao cadastrar essa senha, o usuário está apto novamente a logar em sua conta, podendo das próximas vezes entrar normalmente com seu e-mail e senha.



Figura 33 – E-mail de Esqueci a Senha

Fonte: Mailtrap

5.8.6 Alterar Senha

A opção de Alterar Senha foi um dos últimos requisitos a serem definidos. A funcionalidade consiste basicamente em poder alterar a atual senha de acesso ao Painel. Os dados requeridos para alteração são Senha Atual, Nova Senha e Confirmar Nova Senha. Para se chegar nessa tela deve-se clicar no ícone na parte superior direita do Painel e clicar em “Alterar Senha”.

A screenshot of a web form titled "Alterar Senha". It contains three input fields with labels "Senha atual", "Nova Senha", and "Confirmar Nova Senha". Below the fields is an orange button with the text "Trocar senha".

Figura 34 – Tela de Alterar Senha

Fonte: Painel Pró-Mamá

5.9 API

A API é o canal de troca de mensagens pelo qual o aplicativo extrai recursos do Banco de Dados do Pró-Mamá. Pode-se dizer que ela é a intermediadora que recebe e faz a entrega dos dados solicitados. A API foi construída nos padrões REST, que faz a comunicação entre diferentes sistemas através dos métodos (POST, PUT, GET e DELETE) do protocolo HTTP. E como já dito no subcapítulo 5.3, a notação de troca de dados foi no formato JSON.

O Laravel possui dois arquivos para mapear as rotas: O “web.php” e o “api.php”. As rotas Web são todas as rotas que são acessadas por um cliente Web, como um navegador Google Chrome por exemplo. Já as rotas de API são específicas para a comunicação de diferentes dispositivos, como o de *smartphones* por exemplo. Para acessar as rotas de API, a requisição precisa conter um *token* de acesso, que está cadastrado em um registro de usuário no Banco de Dados. Ou seja, mesmo que alguém descubra o endereço da rota, precisará ter uma chave de acesso para consumir as informações do Banco.

Será mostrado, em sequência, as rotas de API implementadas na aplicação Laravel do Painel. Foi importante ter documentado estas rotas devido o processo de desenvolvimento do aplicativo depender de tais recursos para dinamizar o conteúdo da aplicação, como por exemplo mostrar as Informações para certa idade etc.

5.9.1 Requisições e Respostas

As requisições feitas à API, portanto, precisam ser autenticadas. Para autenticar a troca de informações, um campo `api_token` vai na URL contendo uma *String* identificadora daquele usuário. O usuário ganha um *token*, que é gerado aleatoriamente, quando se cadastra no sistema. Exemplo de URL:

<http://servidor.com/api/user?api_token=abcdefg123456>

As requisições e respostas são transmitidas no formato JSON. As respostas pela API seguem um padrão, tornando o processo de consumo das informações mais intuitivo e simples. A tabela 16 exibe os padrões de respostas mais utilizados nas trocas de informações.

Tabela 16 – Respostas Padrões

Tipo	Exemplo
1. Sucesso	<pre>{ "success": true, "message": "Mensagem", "id": "{Algum ID}" }</pre>
2. Erro	<pre>{ "success": false, "message": "Erro/Falta algo" }</pre>
3. Objeto	<pre>{ "id": 1, "email": "exemplo@mail.com" }</pre>
4. Array de objetos	<pre>[{ "id": 1, "email": "exemplo@mail.com" }, { "id": 2, "email": "exemplo@ex.com" }]</pre>

Fonte: Autoria Própria

5.9.2 Rotas da API

Neste subcapítulo o objetivo é visualizar as rotas de API, agrupadas por entidades, e buscar ver as relações dos métodos de HTTP com a finalidade daquela rota. Por exemplo, observa-se que para fazer uma inclusão no Banco de Dados usa-se o método POST, já um update, ou atualização, o método PUT.

As colunas nomeadas de Resposta que aparecem em sequência, indica qual o padrão de resposta que é obtido com a requisição daquela rota. Por exemplo, se uma rota de Usuário retorna o tipo 3 de resposta, sabe-se que ela irá trazer um objeto de Usuário com dados como nome, email, data de nascimento etc. Para se chegar qual padrão de resposta precisa-se checar a tabela 16.

Tabela 17 – Rotas da API

Rota	Descrição	Método	Campos	Resposta
api/user	Cadastrar Usuário	POST	email, password	Tipo 1 ou 2
api/user/{userID}/editar	Editar Usuário	PUT	userID na URL e qualquer campo de usuário via POST	Tipo 1 ou 2
api/user/{userID}/remover	Remover Usuário	DELETE	userID na URL	Tipo 1 ou 2
api/login	Login de Usuário	POST	email, password	Tipo 1 ou 2*
api/user/{userID}	Resgatar usuário por ID	GET	userID na URL	Tipo 3
api/informacoes	Obter todas as informações	GET	—	Tipo 4
api/criancas	Cadastrar Criança	POST	crianca_primeiro_nome, crianca_dataNascimento	Tipo 1 ou 2
api/criancas/{criancaID}/editar	Editar Criança	PUT	criancaID na URL e qualquer campo de criança via POST	Tipo 1 ou 2
api/criancas/{criancaID}/rmv	Remover Criança	DELETE	criancaID na URL	Tipo 1 ou 2
api/criancas/{criancaID}	Resgatar um objeto de criança	GET	criancaID na URL	Tipo 3
api/criancas/{userID}/user	Obter Todas Crianças de um Usuário	GET	userID na URL	Tipo 4
api/conversa	Cadastrar Dúvida	POST	user, pergunta	Tipo 1 ou 2
api/conversa-user	Resgatar dúvidas do usuario	GET	—	Tipo 4
api/conversa-todos	Obter dúvidas que vai Para Todos	GET	—	Tipo 4
api/postos	Obter todos os postos de saúde	GET	—	Tipo 4
api/duvidas-frequentes	Obter todas as dúvidas frequentes	GET	—	Tipo 4
api/sync	Sincronização dos Dados	GET	—	Tipo 3
api/notificacoes	Obter todas as notificações	GET	—	Tipo 4
api/marcos	Cadastrar Marco	POST	crianca, marco, data, extra	Tipo 1 ou 2
api/marcos	Obter marcos da criança	GET	crianca ID na url	Tipo 4
api/acompanhamentos	Cadastrar Acompanhamento	POST	crianca, data, peso, altura, alimentacao	Tipo 1 ou 2
api/acompanhamentos	Resgatar acompanhamentos por ID do usuario	GET	—	Tipo 4
api/file/filename	Resgatar um arquivo de foto	GET	Nome do arquivo na URL	Arquivo de Foto
api/uploadfoto	Fazer upload de arquivo de foto	POST	foto (tipo FILE)	Tipo 1 ou 2
api/fotos	Resgatar fotos da criança por ID do usuario	GET	—	Tipo 4
api/fotos	Cadastrar Foto	POST	crianca, mes, url	Tipo 1 ou 2
api/fotos/{fotoID}/editar	Editar Foto	POST	fotoID na URL	Tipo 1 ou 2
api/esqueciasenha	Recuperação de senha	POST	email	Tipo 1 ou 2

Fonte: Autoria Própria

5.10 IMPLANTAÇÃO (DEPLOY)

Ao fim da fase de Desenvolvimento, o Painel Administrativo e API estavam prontos para serem implantados no servidor da Secretaria Municipal da Saúde de Osório-RS. Nas últimas reuniões foram passados os dados de acesso a um servidor virtual com Ubuntu, um sistema operacional de software de código aberto construído a partir do núcleo Linux.

Após testar o acesso e conseguir de fato logar na máquina virtual deles, a primeira coisa que foi solicitada pelo grupo de desenvolvedores foi da liberação da porta 80 no servidor. A porta 80 é a responsável pela transferência do protocolo HTTP, fundamental para funcionamento da aplicação.

Com a liberação da porta, o próximo passo foi configurar o ambiente do Ubuntu para hospedar uma aplicação Laravel. Os passos foram os mesmos já citados na seção 5.7, de configuração de ambiente. A única diferença é que no servidor com o Ubuntu os programas foram instalados por linha de comando através do gerenciador de pacotes apt-get usado em distribuições Debian Linux.

Com o ambiente configurado e pronto para receber a aplicação, clonou-se então o projeto que estava no Github e alterada algumas permissões em pastas de armazenamento. No demais, a aplicação estava pronta para uso e disponível para as requisições.

O Laravel dispõe em sua documentação uma série de comandos e dicas úteis para o *deploy* de suas aplicações. Como por exemplo, rodar comandos para limpar o cache, limpar configurações e otimizar o tráfego das informações.

5.10.1 Manutenção do Sistema

Após o Lançamento do Painel Administrativo o foco deste trabalho visa manter as funcionalidades, corrigir os erros, procurar falhas, bugs e receber o *feedback* da equipe do Pró-Mamá. São processos como este que garantem a qualidade do software.

Devido a existência da documentação do seguinte trabalho quanto às tecnologias empregadas, o processo de manutenção se torna menos custoso e complexo. A documentação forma os alicerces para um desenvolvimento bem-sucedido e fornece um guia para a tarefa de manutenção do software (PRESSMAN, 1995).

6 CONCLUSÃO

Como visto, o objetivo deste trabalho foi desenvolver um sistema informatizado, especificamente um sistema de gerenciamento de conteúdo Web, na qual permitisse à equipe de profissionais do Pró-Mamá produzir e gerenciar informações em saúde com foco no aleitamento materno.

Foram investigadas tecnologias e suportes ao desenvolvimento Web para a construção de um Painel Administrativo e uma API. Após os estudos, verificou-se que o Framework Laravel atendia as exigências e necessidade que o projeto pedia, como por exemplo o suporte à escalabilidade. Com esse kit de ferramentas foi possível ter um processo de desenvolvimento estável e adaptado às mudanças, que eram constantes.

O objetivo do trabalho foi atingido, uma vez que o Painel Administrativo e a API encontram-se em funcionamento, hospedados no servidor da secretaria da saúde de Osório. O lançamento do Aplicativo juntamente com o Painel Administrativo e a API ocorreram no dia 7 de Agosto de 2018, no II Seminário Municipal da Amamentação em Osório-RS.

Os requisitos, levantados no início do desenvolvimento, foram todos alcançados. Desde os requisitos essenciais até os desejáveis. As reuniões que aconteceram com a equipe do Pró-Mamá foram importante para receber o feedback de uso do sistema, ou seja, entender como os utilizadores estavam se adaptando a cada requisito implementado.

Desde o lançamento do sistema, portanto, já pode ser visto os resultados deste trabalho. A equipe do programa, já familiarizada com o sistema, vêm inserindo informações, cadastrando notificações e respondendo dúvidas que chegam dos usuários. O aplicativo Pró-Mamá, juntamente com o Painel, estão causando um impacto positivo na comunidade de Osório. A informação e o conhecimento estão chegando mais longe.

Havia uma demanda a ser atendida que teve que ser analisada em busca de encontrar uma solução. A demanda era que o Pró-Mamá precisava ampliar e disseminar as informações sobre a amamentação na região de Osório. A solução foi desenvolver o Painel Administrativo como ferramenta, junto com o aplicativo, para divulgar e conscientizar as pessoas de um tema tão importante quanto o aleitamento materno.

6.1 TRABALHOS FUTUROS

Como o sistema desenvolvido atendeu todas as necessidades que se tinham, ficam algumas sugestões como trabalhos futuros para agregar ainda mais ao trabalho. Pelo fato do projeto estar disponível no Github¹, o código do Painel está aberto para ser acessado e baixado. Logo, fica livre

¹ Disponível em <<https://github.com/lucasjardi/promama>>

aqui àquele que quiser implementar os trabalhos futuros e colaborar com o projeto do Pró-Mamá.

Concluindo, para efeito de acrescentar os estudos e aplicação deste trabalho, recomenda-se os seguintes trabalhos futuros para serem desenvolvidos:

- Estudar maneiras de criar relatórios que permitam a visualização do desenvolvimento das crianças da região, seus históricos de alimentações etc;
- Investigar novas formas de implementar funcionalidades dentro do sistema, buscando atingir melhores desempenhos;
- Manter o sistema e dar suporte a equipe do Pró-Mamá quanto às suas dúvidas;
- Criação de projetos e trabalhos de pesquisas como esse, aliando a tecnologia às causas sociais;

REFERÊNCIAS

- ADAIR, L. S.; POPKIN, B. M.; GUILKEY, D. K. The duration of breast-feeding: how is it affected by biological, sociodemographic, health sector, and food industry factors? *Demography*, Springer, v. 30, n. 1, p. 63–80, 1993. Citado na página 10.
- BEAN, M. *Laravel 5 essentials*. [S.l.]: Packt Publishing Ltd, 2015. Citado na página 20.
- BENEVOLO, C.; NEGRI, S. Evaluation of content management systems (cms): a supply analysis. *Electronic Journal of Information Systems Evaluation*, v. 10, n. 1, 2007. Citado na página 14.
- BOIKO, B. Understanding content management. *Bulletin of the American Society for Information Science and Technology*, Wiley Online Library, v. 28, n. 1, p. 8–13, 2001. Citado na página 14.
- BUILTWITH.COM. *CMS Usage Distribution in the Top 1 Million Sites*. 2018. Disponível em: <<https://trends.builtwith.com/cms>>. Citado na página 27.
- CARLOS, G. A. Métodos e técnicas de pesquisa social. *São Paulo: Atlas*, 1999. Citado na página 30.
- CARVALHAES, M. A. d. B. L.; CORRÊA, C. R. H. Identificação de dificuldades no início do aleitamento materno mediante aplicação de protocolo. *Jornal de Pediatria*, SciELO Brasil, p. 13–20, 2003. Citado na página 10.
- CAVALCANTI, D. C. et al. *Desenvolvimento de aplicativo na plataforma móvel referente ao programa Pró-Mamã da prefeitura de Osório - RS*. [S.l.]: Anais da MoExp, 2018. ISSN: 2526-3250. Citado 3 vezes nas páginas 33, 34 e 36.
- COELHO, M. A. P. et al. O uso do cms joomla e suas ferramentas hipertextuais na produção de sites educativos e de material didático online. *Texto Livre: Linguagem e Tecnologia*, v. 4, n. 2, p. 38–46, 2011. Citado na página 24.
- CORLOSQUET, S. et al. Produce and consume linked data with drupal! In: SPRINGER. *International Semantic Web Conference*. [S.l.], 2009. p. 763–778. Citado na página 23.
- DAS, R.; SAIKIA, L. P. Comparison of procedural php with codeigniter and laravel framework. *International Journal of Current Trends in Engineering & Research*, v. 2, n. 6, p. 42–8, 2016. Citado 2 vezes nas páginas 18 e 19.
- DRUBSCKY, L. *Sistema de gestão de conteúdos (CMS): por que implementar na sua empresa?* 2018. Disponível em: <<https://marketingdeconteudo.com/cms/>>. Citado na página 14.
- DRUPAL.ORG. *Drupal: About*. 2018. Disponível em: <<https://www.drupal.org/about>>. Citado na página 23.
- FARIAS, F. *Google Trends: o que é a ferramenta e como usá-la na sua estratégia*. 2017. Disponível em: <<https://resultadosdigitais.com.br/blog/google-trends/>>. Citado na página 18.

FAYAD, M. E.; SCHMIDT, D. C.; JOHNSON, R. E. *Building application frameworks: object-oriented foundations of framework design*. [S.l.]: John Wiley & Sons, Inc., 1999. Citado na página 16.

FERNANDES, A. *Application Program Interface*. 2018. Disponível em: <<https://vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples/>>. Citado na página 20.

FILHO, A. R. P. Um sistema de gerenciamento de cursos. *Brasília, DF: UNB*, 2005. Citado 2 vezes nas páginas 24 e 25.

FOLDOC.ORG. *Application Program Interface*. 1995. Disponível em: <<http://foldoc.org/Application+Program+Interface>>. Citado na página 20.

GAEA.COM.BR. *Entenda o que é Framework*. 2018. Disponível em: <<https://gaea.com.br/entenda-o-que-e-framework/>>. Citado na página 16.

HAVIV, Y. A.; TRIPP, O.; WEISMAN, O. *Static analysis of computer software applications having a model-view-controller architecture*. [S.l.]: Google Patents, 2014. US Patent 8,726,245. Citado na página 15.

HEUSER, C. A. *Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS*. [S.l.]: Bookman Editora, 2009. Citado na página 41.

JOOMLA.ORG. *About Joomla!* 2018. Disponível em: <<https://www.joomla.org/about-joomla.html>>. Citado na página 24.

KRASNER, G. E.; POPE, S. T. et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, v. 1, n. 3, p. 26–49, 1988. Citado na página 15.

LARAVEL.COM. *Laravel: The PHP Framework for Web Artisans*. 2018. Disponível em: <<https://laravel.com/>>. Citado na página 20.

MACHADO, A. R. M. et al. O lugar da mãe na prática da amamentação de sua filha nutriz: o estar junto. *Rev Bras Enferm*, SciELO Brasil, v. 57, n. 2, p. 183–7, 2004. Citado na página 12.

NATIONS, D. Web applications. *About. com*, 2014. Citado na página 13.

OSORIO.RS.GOV.BR. *Pró-Mamã Osório inova e lança Canal no YouTube*. 2017. Disponível em: <<http://www.osorio.rs.gov.br/site/noticia/visualizar/idDep/63/id/5792/?Pro-Mama-Osorio-inova-e-lanca-Canal-no-YouTube.html>>. Citado na página 10.

PAIVA, V. M. d. O. Ambientes virtuais de aprendizagem: implicações epistemológicas. *Educação em Revista*, SciELO Brasil, v. 26, n. 3, p. 353–370, 2010. Citado na página 25.

PATEL, S. K.; RATHOD, V.; PRAJAPATI, J. B. Performance analysis of content management systems-joomla, drupal and wordpress. *International Journal of Computer Applications*, Citeseer, v. 21, n. 4, p. 39–43, 2011. Citado 3 vezes nas páginas 14, 24 e 28.

PORTALMSSAUDE.GOV.BR. *OuvidorSUS*. 2017. Disponível em: <<http://portalms.saude.gov.br/participacao-e-controle-social/ouvidoria-do-sus/sistema-ouvidorsus>>. Citado na página 25.

PORTALMS.SAUDE.GOV.BR. *Apoio à Amamentação*. 2018. Disponível em: <<http://portalms.saude.gov.br/saude-para-voce/saude-da-crianca/aleitamento-materno/apoio-a-amamentacao>>. Citado na página 11.

PRESSMAN, R. S. *Engenharia de software*. [S.l.]: Makron books São Paulo, 1995. v. 6. Citado na página 63.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005. Citado na página 30.

RAHMEL, D. *Dominando Joomla: do Iniciante ao Profissional. Trad.* [S.l.]: Rio de Janeiro: Alta Books Editora, 2010. Citado na página 24.

RISHABHISOFT.COM. *Zend Vs Laravel – Comparison of the Two Best PHP Frameworks*. 2015. Disponível em: <<https://www.rishabhsoft.com/blog/laravel-or-zend>>. Citado 2 vezes nas páginas 19 e 20.

SITELEADS.NET. *The advantages of a (Web) Content Management System (WCMS)*. 2018. Disponível em: <<http://www.siteleads.net/bm/blog/the-advantages-of-a-webcontent-management-system-.shtml>>. Citado na página 14.

SKVORC, B. *The Best PHP Framework for 2015: SitePoint Survey Results*. 2015. Disponível em: <<https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>>. Citado na página 16.

SOMMERVILLE, I.; ARAKAKI, R.; MELNIKOFF, S. S. S. *Engenharia de software*. [S.l.]: Pearson Prentice Hall, 2008. Citado na página 43.

TELES, V. M. *Extreme programming*. São Paulo: Novatec, 2004. Citado na página 31.

THINKMOBILES.COM. *Symfony vs Laravel in 2018, analysis of top PHP frameworks*. 2018. Disponível em: <<https://thinkmobiles.com/blog/symfony-vs-laravel/>>. Citado na página 19.

W3TECHS.COM. *Usage of server-side programming languages for websites*. 2018. Disponível em: <https://w3techs.com/technologies/overview/programming_language/all>. Citado na página 13.

WIMBERLY, T. *WordPress publishes native Android application*. 2018. Disponível em: <<https://androidandme.com/2010/02/news/wordpress-publishes-native-android-application/>>. Citado na página 22.

WORDPRESS.ORG. *WordPress: About*. 2018. Disponível em: <<https://br.wordpress.org/about/>>. Citado na página 22.