



UNIVERSIDADE FEDERAL DO ACRE
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

UMA API RESTFUL PARA O GUIA DE SERVIÇOS DA OCA

RIO BRANCO

2018

ANTONIO GABRIEL ALMEIDA DE SOUZA
WALLISON SANTOS FERREIRA

UMA API RESTFUL PARA O GUIA DE SERVIÇOS DA OCA

Relatório de estágio apresentado como exigência parcial para obtenção do grau de bacharel em Sistemas de Informação da Universidade Federal do Acre.

Orientador: Prof. Dr. Daricélio Moreira Soares

RIO BRANCO

2018

TERMO DE APROVAÇÃO

ANTONIO GABRIEL ALMEIDA DE SOUZA
WALLISON SANTOS FERREIRA

TÍTULO DO TRABALHO

Este relatório de estágio supervisionado foi apresentado como trabalho de estágio supervisionado do Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Acre, sendo aprovado pela banca constituída pelo professor orientador e membros abaixo mencionados.

Compuseram a banca:

Prof. Daricélio Moreira Soares, D.Sc
Curso de Bacharelado em Sistemas de Informação

Prof. Jean Gonzaga de Oliveira, M.Sc.
Curso de Bacharelado em Sistemas de Informação

Rio Branco, 21 de Março de 2018.

*Dedicamos este trabalho aos membros da diretoria da
OCA por ter nos ensinado no passado a importância
da informação ao cidadão há alguns anos.*

AGRADECIMENTOS

Agradecemos aos colegas do Ministério Público do Estado do Acre pelo apoio, tempo e compreensão.

Ao professor Daricélio Moreira Soares, pela orientação no estágio, pelo seu tempo dedicado às sugestões e por nos ajudar a desenvolver este trabalho. Agradecemos também a professora Laura Sarkis por todo apoio prestado durante a disciplina de estágio supervisionado.

À nossa amiga e supervisora de TI Herlen Duarte Jorge por toda ajuda e acolhimento que tivemos na organização.

À Diretoria da Organização Central de Atendimento pelas informações e ajuda prestada com tanta confiança.

Aos nossos amigos da UFAC pela amizade colhida.

“Ideias e somente ideias podem iluminar a escuridão.”

(Ludwig Von Mises)

RESUMO

Informações sobre serviços oferecidos pelas organizações governamentais via websites é um importante meio para mostrar ao cidadão o passo-a-passo para o atendimento. Centrais de atendimento recorrem a esse meio para que o cidadão chegue ao local de serviço ciente em como proceder para iniciar o atendimento e evitar desperdícios de tempo com espera em filas para obter informações triviais. É imprescindível que estas informações estejam sempre atualizadas. O sistema utilizado e a maneira como o conteúdo dos serviços está estruturado dificulta o processo de manutenção das informações, havendo, portanto, sérios riscos de falha de comunicação entre a organização e o cidadão. A OCA (Organização Central de Atendimento) objetiva desenvolver uma solução que melhore as atividades de gerenciamento de conteúdo dos serviços e que forneça meios de compartilhar esses conteúdos com outras aplicações clientes. A solução proposta utiliza uma arquitetura web baseada em API para a gestão dos conteúdos, bem como um website que otimiza suas atividades de manutenção com consumo de api com princípios *rest* utilizando o framework Laravel e páginas reativas utilizando Vue.

Palavras-chave: Arquitetura web. API REST. Laravel. Vue.

ABSTRACT

Information on services offered by governmental organizations via websites is an important way to show citizens the step-by-step approach to service. Call centers use this means so that the citizen arrives at the place of service aware in how to proceed to start the service and avoid wasting time with waiting in queues to obtain trivial information. It is imperative that this information is always up to date. The system used and the way in which the content of the services is structured hinder the process of information maintenance, therefore, there are serious risks of communication failure between the organization and the citizen. OCA aims to develop a solution that enhances service content management activities and provides a means of sharing that content with other client applications. The proposed solution uses an API based web architecture for content management, as well as a website that optimizes its maintenance activities with api consumption with rest principles using the Laravel framework and reactive pages using Vue.

Key-words: Web architecture. REST API. Laravel. Vue.

LISTAS DE FIGURAS

| | |
|--|----|
| Figura 1. Edição de serviço no WordPress | 16 |
| Figura 2. Visualização de Serviço no Website | 17 |
| Figura 3. Cenário anterior do sistema | 18 |
| Figura 4. Processo de software | 23 |
| Figura 5. O modelo cascata | 25 |
| Figura 6. O modelo incremental | 25 |
| Figura 7. Requisitos de Usuário e de Sistema | 28 |
| Figura 8. Template e Script de um componente Vue.js | 31 |
| Figura 9. DETRAN na Praça amarela da OCA | 34 |
| Figura 10. Organograma da OCA | 35 |
| Figura 11. Diagrama de Entidade-Relacionamento | 38 |
| Figura 12. Métodos para requisição das informações do Laravel | 39 |
| Figura 13. Declaração das endpoints no Laravel | 40 |
| Figura 14. Requisição HTTP na endpoint de serviços usando Postman | 41 |
| Figura 15. Documentação da API Guia Oca | 42 |
| Figura 16. Protótipo da tela de preenchimento de conteúdo dos serviços | 43 |

LISTAS DE QUADROS

| | |
|---|----|
| Quadro 1. Testes de funcionalidades de interface de usuário | 44 |
| Quadro 2. Teste de requisitos não funcionais..... | 45 |

LISTAS DE TABELAS

| | |
|--|----|
| Tabela 1. Requisitos funcionais..... | 37 |
| Tabela 2. Requisitos não funcionais..... | 38 |
| Tabela 3. Requisitos funcionais..... | 61 |
| Tabela 4. Requisitos não funcionais..... | 61 |

SUMÁRIO

| | |
|---|----|
| LISTAS DE FIGURAS | 8 |
| LISTAS DE QUADROS | 9 |
| LISTAS DE TABELAS..... | 10 |
| 1 INTRODUÇÃO | 13 |
| 1.1 PROBLEMA DA PESQUISA | 15 |
| 1.2 OBJETIVOS DA PESQUISA | 19 |
| 1.2.1 OBJETIVO GERAL..... | 19 |
| 1.2.1 OBJETIVOS ESPECÍFICOS..... | 19 |
| 1.3 METODOLOGIA | 20 |
| 1.4 ORGANIZAÇÃO DO ESTUDO | 20 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 22 |
| 2.1 PROCESSO DE SOFTWARE..... | 22 |
| 2.2 ENGENHARIA DE REQUISITOS | 27 |
| 2.3 ARQUITETURA DE SOFTWARE..... | 29 |
| 2.4 ARQUITETURA WEB | 29 |
| 2.5 FRAMEWORK VUE..... | 30 |
| 3 ESTÁGIO SUPERVISIONADO..... | 32 |
| 3.1 ORGANIZAÇÃO | 32 |
| 3.2 ESTRUTURA ORGANIZACIONAL | 35 |
| 3.3 ATIVIDADES DESENVOLVIDAS | 36 |
| 3.3.1 IDENTIFICAÇÃO DO PROBLEMA | 36 |
| 3.3.2 LEVANTAMENTO DE REQUISITOS | 37 |
| 3.3.3 DIAGRAMA DE ENTIDADE-RELACIONAMENTO | 38 |
| 3.3.4 DESENVOLVIMENTO DA API | 39 |
| 3.3.5 PROTOTIPAÇÃO | 43 |
| 3.4 RESULTADOS OBTIDOS | 47 |

| | |
|---|-----------|
| 4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES | 51 |
| 4.1 CONSIDERAÇÕES FINAIS | 51 |
| 4.2 RECOMENDAÇÕES..... | 52 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 54 |
| APÊNDICE | 55 |

1 INTRODUÇÃO

Muitas organizações que realizam serviços de atendimento ao cidadão – as Centrais de Atendimento ao Cidadão (CAC) – se veem sob a necessidade de disponibilizar informações para a realização de cada serviço pelos cidadãos. Estas informações, tais como local, procedimentos para solicitação e documentos necessários são mostradas ao cidadão via website da organização. A gestão das informações dos serviços torna-se difícil quando a organização agrega muitos órgãos e cada órgão possuem muitos serviços e, além disso, as informações de cada serviço podem mudar frequentemente pelo órgão responsável. Sistemas de Gerenciamento de Conteúdo (CMS) como *Joomla* e *WordPress* tem atendido às necessidades dos CACs para a listagem de conteúdos de órgão e serviços. Porém, estes sistemas não são totalmente adequados à realidade dos CACs quando se refere à estrutura do conteúdo de cada serviço. Outro ponto diz respeito à necessidade de fornecer as informações dos serviços da organização para o consumo de outras aplicações clientes¹.

Este projeto de estágio objetivou desenvolver uma solução que forneça maior eficiência nos processos de manutenção de informações sobre os serviços da Organização Central de Atendimento (OCA)² e que permita também o consumo das informações destes serviços por aplicações clientes. O estágio foi desenvolvido junto à Diretoria da Organização Central de Atendimento.

A organização utilizava um sistema de gerenciamento de conteúdo *WordPress* para publicar páginas de informações de cada serviço de acordo com

¹ Não foi encontrado na literatura uma nomenclatura para aplicações que utilizam API para consumo de informações. Alguns autores de websites utilizam o termo “aplicações clientes” e outros apenas “clientes”. Mas para ficar mais claro neste trabalho, utiliza-se o termo “aplicações clientes”.

² A Organização Central de Atendimento é a principal Central de Atendimento ao Cidadão (CAC) do estado do Acre, e está presente também no município de Xapuri. Além disso, é responsável pela gestão dos serviços de CAC's presentes em outros municípios.

sua necessidade, e essas informações podiam ser visualizadas pelo cidadão em seu website Guia de Serviços Públicos³. Não havia meios para integrar as informações dos serviços com outras aplicações, tais como o Sistema de Gestão de Atendimento (SIGA)⁴, portais de informações do governo, portais de notícias e outras aplicações clientes. Além disso, a grande quantidade de páginas estáticas de serviços tornava o processo de manutenção cada vez mais difícil, pois haviam serviços semelhantes e repetitivos e muitas vezes confundindo o usuário do sistema durante suas operações de manutenção. Portanto, o estágio buscou otimizar o processo de manutenção das informações dos serviços e permitir também o compartilhamento destas informações com outras aplicações.

No estágio foram utilizadas técnicas de engenharia de software e atividades de engenharia de requisitos para especificação dos requisitos. Também foram utilizadas a linguagem de programação PHP com o framework *Laravel* para o desenvolvimento da API e a linguagem *Ecmascript* com o framework Vue para o desenvolvimento do website.

³ O Guia de Serviços Públicos é o website de informações sobre os serviços da Organização Central de Atendimento (OCA). As informações são organizadas em praças, órgãos e serviços. Cada serviço está contido em um órgão responsável e cada órgão está contido em uma praça, caracterizada pela cor.

⁴ O SIGA é um software proprietário da Visual para a gestão de atendimento ao cliente, realizando tarefas de agendamento online, gerenciamento de filas até a conclusão e avaliação do cliente.

1.1 PROBLEMA DA PESQUISA

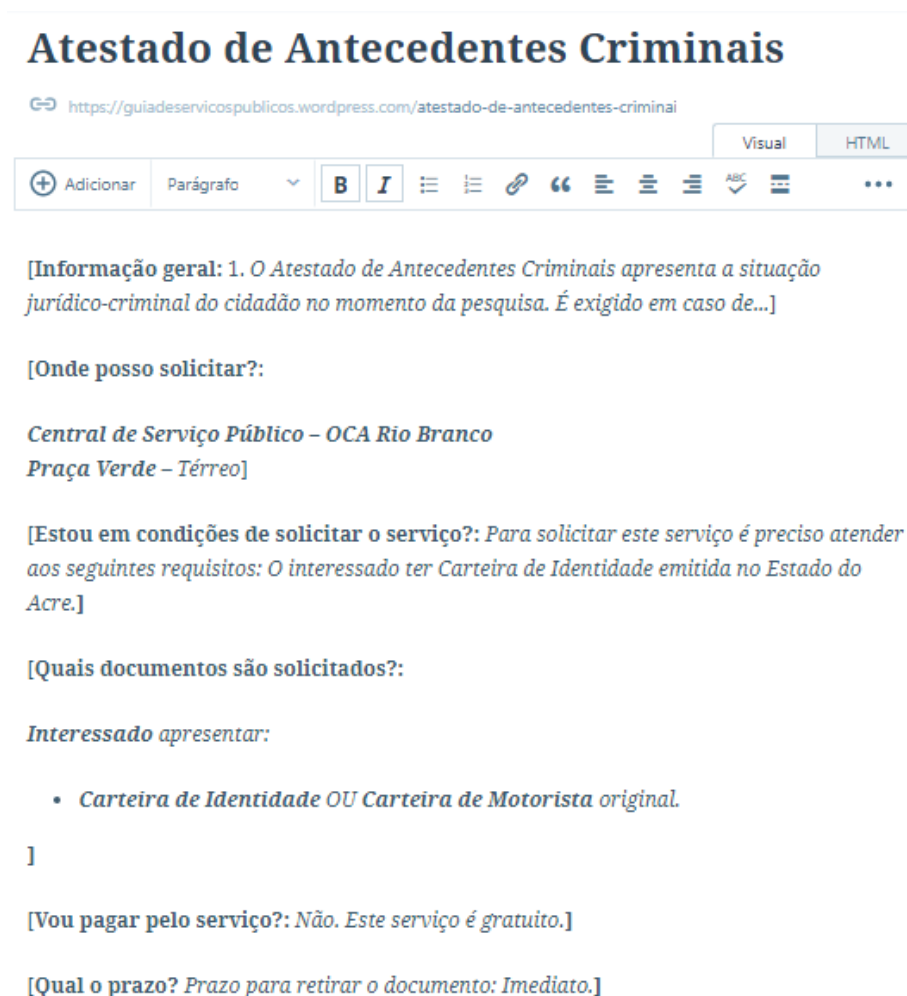
O processo de manutenção das informações dos serviços era corriqueiro. Podiam ocorrer dezenas de solicitações de modificação das informações pelos órgãos aos usuários do sistema diariamente. Não havia dependência entre os serviços no sistema, portanto cada página era inserida com os mesmos campos com informações diferentes, mas podendo ocorrer redundâncias.

Essas redundâncias apareciam normalmente pois haviam muitos serviços contendo campos com informações comuns. Cada serviço possui campos como mostrados a seguir:

1. Onde posso solicitar?
2. Estou em condições de solicitar o serviço?
3. Quais documentos são solicitados?
4. Vou pagar pelo serviço?
5. Qual o prazo?

Incluindo campos como título e informações sobre o serviço. Estes campos são inseridos com suas informações correspondentes em uma página através do *WordPress*, como mostrado na figura 1. A edição de um serviço era completamente textual e esses campos eram repetidos para todas as páginas de serviços e nem todos os serviços contém todos os campos mencionados anteriormente, tornando ainda mais difícil o processo de manutenção.

Figura 1. Edição de serviço no WordPress



Fonte: Elaboração Própria

Essa forma de edição gerava gargalos quando haviam muitos serviços a serem editados com campos comuns e conteúdos semelhantes. O conteúdo do campo “Onde solicitar” era muitas vezes repetido para outros serviços, geralmente quando pertenciam ao mesmo órgão. Havia também a possibilidade de todos os campos terem conteúdos iguais em vários serviços exceto o campo de local. As possibilidades de redundância nos conteúdos aumentavam à medida que surgiam adições e alterações dos serviços, pois os campos são preenchidos em um corpo de texto estático como visto na figura 1. A edição acima gerava uma página sobre o serviço para a visualização pelo cidadão, representada na figura 2.

Figura 2. Visualização de Serviço no Website

Atestado de Antecedentes Criminais

Clique no endereço desejado:

Central de Serviço Público – OCA Rio Branco

1. O Atestado de Antecedentes Criminais apresenta a situação jurídico-criminal do cidadão no momento da pesquisa. É exigido em caso de:

- a) porte de arma;
- b) admissão de pessoas nas empresas ou para exercício em cargos públicos, etc.

2. O Atestado de Antecedentes Criminais tem validade de 03 meses, a contar da data de emissão;

3. O cidadão que tiver Carteira de Identidade emitida em outro estado deverá solicitar o Nada Consta.

↓ ONDE POSSO SOLICITAR?

Central de Serviço Público – OCA Rio Branco
Praça Verde – Térreo
Endereço: Rua Quintino Bocaiuva, 299 – Centro
Município: Rio Branco
Dias e horário de atendimento: de segunda a sexta-feira, das 8h às 16h.

↓ ESTOU EM CONDIÇÕES DE SOLICITAR O SERVIÇO?

Para solicitar este serviço é preciso atender aos seguintes requisitos:

O Interessado ter Carteira de Identidade emitida no Estado do Acre.

↓ QUAIS DOCUMENTOS SÃO SOLICITADOS?

Interessado apresentar:

- Carteira de Identidade OU Carteira de Motorista original.

↓ VOU PAGAR PELO SERVIÇO?

Não. Este serviço é gratuito.

↓ QUAL O PRAZO?

Prazo para retirar o documento: Imediato.

Responsável pelas informações: Instituto de Identificação.

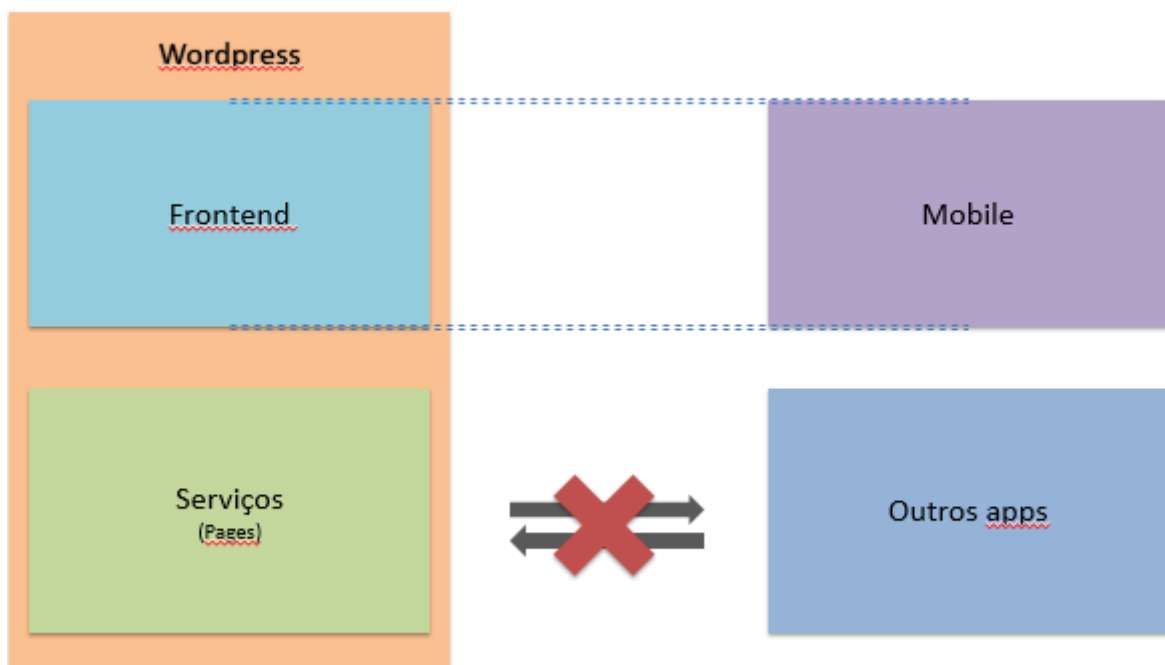
Imprimir

Fonte: Guia de Serviços Públicos⁵ (2018)

⁵ Guia de Serviços Públicos é o website de informações dos serviços da OCA. Disponível no site gsp.ac.gov.br.

O cenário do sistema utilizado, na figura 3, limitava-se à uma conexão cliente-servidor simples, não permitindo o compartilhamento das informações com outras aplicações. Além disso as páginas eram preenchidas com campos e conteúdo de maneira textual, dificultando o controle das informações e a contabilização estatística dos serviços. Não era possível compartilhar as páginas dos serviços com outras aplicações clientes, sendo o *frontend* o único meio de gestão e visualização do conteúdo dos serviços.

Figura 3. Cenário anterior do sistema



Fonte: Elaboração própria

Considerando o problema descrito, surgem as seguintes questões do trabalho: Como tornar o processo de publicação do conteúdo dos serviços mais eficiente, considerando a quantidade de serviços? Como tornar o sistema escalável para maximizar a normalização dos serviços e permitir sua contabilização estatística? Como compartilhar informações dos serviços com outras aplicações clientes?

1.2 OBJETIVOS DA PESQUISA

O objetivo geral e os objetivos específicos do projeto de estágio são apresentados nesta seção.

1.2.1 OBJETIVO GERAL

Desenvolver uma solução para uma gestão eficiente das informações dos serviços da Organização Central de Atendimento.

1.2.1 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral, tem-se os seguintes objetivos específicos:

- a) Realizar o processo de Engenharia de Requisitos;
- b) Mapear a estrutura dos serviços, órgãos e praças da OCA;
- c) Implementar a API REST e o website para a publicação e visualização dos serviços;
- d) Executar o processo de verificação e validação da solução desenvolvida.

1.3 METODOLOGIA

A metodologia proposta consistiu em executar algumas atividades distribuídas nas seguintes etapas:

1. A primeira etapa consiste na identificação das necessidades do usuário e estabelecimento de requisitos, sendo posteriormente elaborado um documento de requisitos;
2. A segunda etapa refere-se ao estudo da estrutura do conteúdo dos serviços da organização, com o intuito de saber quais eram as regras de apresentação, de modificação das informações e as dependências entre cada órgão responsável e seus serviços correspondentes;
3. A terceira etapa diz respeito à implementação da solução;
4. A quarta etapa dedicou-se à verificação e validação dos requisitos.

Para este trabalho foram utilizadas ferramentas CASE tanto para o desenvolvimento quanto para o apoio no desenvolvimento. Foram utilizadas ferramentas de baixo nível, tais como a Linguagem de programação *PHP*, o *framework Laravel*, o *Framework Vue* e o editor *Visual Studio Code*. E, também, ferramentas de alto nível, como o Word, para a elaboração do documento de requisitos e *mockups* para prototipação.

1.4 ORGANIZAÇÃO DO ESTUDO

O relatório de estágio, além da introdução, o problema, os objetivos bem como a metodologia utilizada, é composto por mais 3 capítulos.

No capítulo 2 são encontrados os conceitos que fundamentam este trabalho, abordando processo de software, engenharia de requisitos, Frameworks de

desenvolvimento como Vue.js e Laravel, e conceitos sobre arquitetura de software e arquitetura web.

No capítulo 3 apresenta as atividades desenvolvidas durante o estágio, as informações da organização e resultados obtidos.

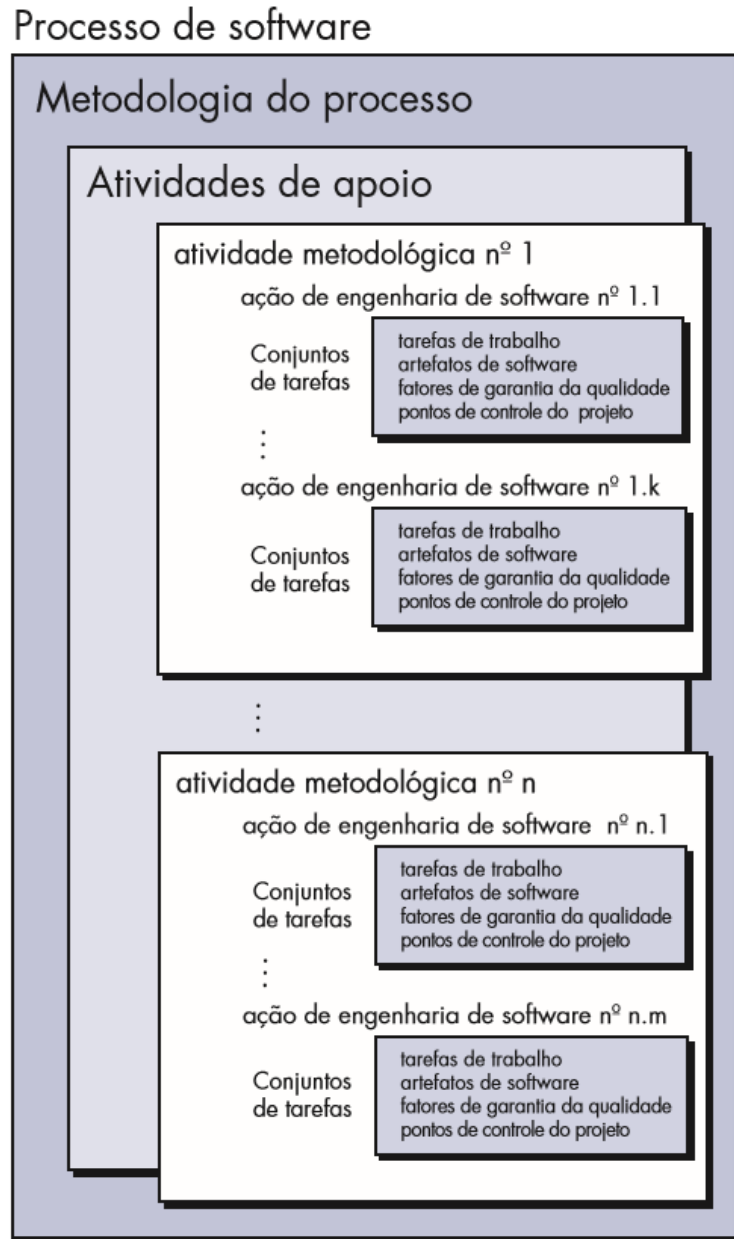
Por último são apresentadas as considerações finais, mostrando o que foi feito e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção estão os conceitos que fundamentam o trabalho realizado nas atividades de estágio. Serão abordados os conceitos de processo de software, engenharia de requisitos, Arquitetura Web e Plataforma web utilizando Vue.

2.1 PROCESSO DE SOFTWARE

Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação como Java ou C (SOMMERVILLE, 2011). Cada uma dessas atividades, ações e tarefas alocam-se dentro de uma metodologia ou modelo que determina seu relacionamento com o processo e seu relacionamento umas com as outras (PRESSMAN, 2011). O processo de software é representado esquematicamente na figura 4.

Figura 4. Processo de software

Fonte: Pressman (2011)

Sommerville (2011) considera a existência de muitos processos diferentes, mas destaca que em todos os processos deve-se incluir quatro atividades fundamentais para a engenharia de software: especificação, projeto e implementação, validação e evolução.

De alguma forma, essas atividades fazem parte de todos os processos de software. Na prática, são atividades complexas, que incluem subatividades como

validação de requisitos, projeto de arquitetura, testes unitários etc. Existem também as atividades que dão apoio ao processo, como documentação e gerenciamento de configuração de software (SOMMERVILLE, 2011).

Um modelo de processo de software é uma representação simplificada de um processo de software. Cada modelo representa uma perspectiva particular de um processo e, portanto, fornece informações parciais sobre ele. Por exemplo, um modelo de atividade do processo pode mostrar as atividades e sua sequência, mas não mostrar os papéis das pessoas envolvidas (SOMMERVILLE, 2011).

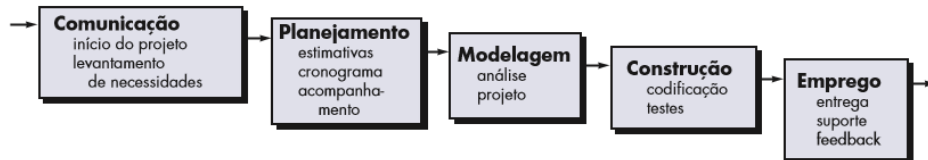
Sommerville (2011) destaca os modelos genéricos como abstrações que podem ser utilizadas para explicar diferentes abordagens de desenvolvimento de software: Modelo cascata, desenvolvimento incremental e engenharia de software orientado a reuso.

Pressman (2011) aponta sobre casos em que os requisitos de um problema são bem compreendidos, quando o processo de desenvolvimento flui da comunicação para o emprego de maneira linear. Nestes casos recorre-se ao modelo de processo tradicional, como o modelo cascata.

O modelo cascata, algumas vezes chamado ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com o levantamento de necessidades por parte do cliente, avançando pelas fases de planejamento, modelagem, construção, emprego e culminando no suporte contínuo do software concluído (PRESSMAN, 2011).

A figura 5 apresenta o modelo cascata:

Figura 5. O modelo cascata

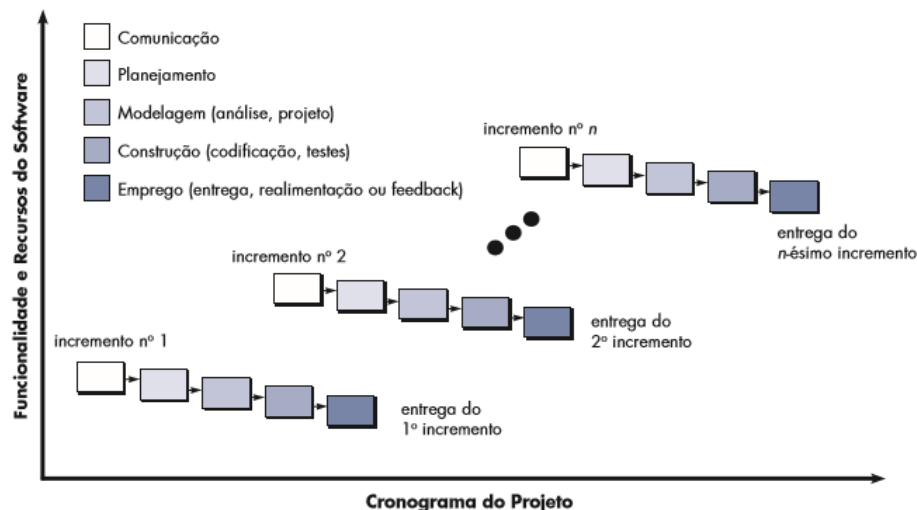


Fonte: Pressman (2011)

Em várias situações, os requisitos iniciais do software são razoavelmente bem definidos, entretanto, devido ao escopo geral do trabalho de desenvolvimento, o uso de um processo puramente linear não é utilizado. Pode ser necessário o rápido fornecimento de um determinado conjunto funcional aos usuários, para somente após esse fornecimento, refinar e expandir sua funcionalidade em versões de software posteriores (PRESSMAN, 2011).

A figura 6 apresenta o modelo incremental:

Figura 6. O modelo incremental



Fonte: Pressman (2011)

Atualmente, novos softwares de negócios são desenvolvidos por meio da extensão e modificação de sistemas existentes ou por meio da configuração e integração de prateleira ou componentes do sistema (SOMMERVILLE, 2011). O

reuso é um meio que facilita a construção de sistemas de maneira rápida através da reutilização de componentes de software. BASS et al. (2000) destaca que a engenharia de software baseada em componentes aborda especificamente problemas de produtividade que são enfrentadas por organizações comerciais e governamentais no emergente ambiente de TI.

Esses problemas de produtividade estão relacionados ao tempo e esforço necessário para o desenvolvimento de sistemas sem a utilização de componentes. O reuso de componentes aborda o uso artefatos, como padrões de projetos, componentes reutilizáveis, códigos preparados, *frameworks* ou biblioteca de componentes no desenvolvimento.

O argumento de que os componentes de software irão melhorar a produtividade do programador é antigo, com raízes no estudo da reutilização de software. O argumento, simplesmente, é que a produtividade e a qualidade aumentam se os programadores podem reutilizar o software. Existem muitas técnicas para reutilizar, incluindo a reutilização de padrões de design, o uso de linguagens de programação específicas de aplicativos, juntamente com geradores de programas, e assim por diante (BASS et al., 2000, p.13).

Segundo Pressman (2011), o modelo de desenvolvimento baseado em componentes desenvolve aplicações a partir de componentes de software pré-empacotados.

As atividades de modelagem e construção começam com a identificação de possíveis candidatos a componentes. Esses componentes podem ser projetados como módulos de software convencionais, como classes orientadas a objeto ou pacotes de classes (PRESSMAN, 2011).

O modelo de desenvolvimento baseado em componentes conduz ao reuso do software e a reusabilidade proporciona uma série de benefícios mensuráveis aos engenheiros de software (PRESSMAN, 2011). O reuso diminui os custos e o ciclo de desenvolvimento do projeto através da reutilização de componentes.

2.2 ENGENHARIA DE REQUISITOS

O levantamento de requisitos (também chamado elicitación de requisitos) combina elementos de resolução de problemas, elaboração, negociação e especificação (PRESSMAN, 2011). Os requisitos podem ser descobertos através da comunicação da comunicação entre os interessados – a equipe e o cliente – de maneira que se identifique o problema e haja discussões para encontrar uma solução.

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, cadastrar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (SOMMERVILLE, 2011).

Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos (SOMMERVILLE, 2011). Os requisitos dos sistema MHC-PMS são apresentados a seguir:

Requisitos funcionais:

1. Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas.
2. O sistema deve gerar a cada dia, para cada clínica, a lista dos pacientes para as consultas daquele dia.
3. Cada membro da equipe que usa o sistema deve ser identificado apenas por seu número de oito dígitos (SOMMERVILLE, 2011, p. 59).

Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários (SOMMERVILLE, 2011). Os requisitos não funcionais estão ligados às funcionalidades não associadas às operações dos usuários, mas estão “relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área [...]” (SOMMERVILLE, 2011).

Os requisitos precisam ser escritos em diferentes níveis de detalhamento para que diferentes leitores possam usá-los de diversas maneiras (SOMMERVILLE, 2011).

A figura 7 apresenta os diferentes níveis de requisitos de um Sistema de Gerenciamento da Saúde Mental de Pacientes (MHC-PMS, do inglês *Mental Health Care Patient Management System*).

Figura 7. Requisitos de Usuário e de Sistema

Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Fonte: Sommerville (2011, p. 58)

Sommerville (2011) destaca que os processos de engenharia de requisitos podem incluir quatro atividades de alto nível: Elas visam avaliar se o sistema é útil para a organização (estudo de viabilidade), descobrindo requisitos (elicitação e análise), convertendo-os em alguma forma-padrão (especificação), e verificar se os requisitos realmente definem o sistema que o cliente quer (validação). Para que fosse possível a implementação da solução no estágio foram realizadas as seguintes atividades de engenharia de requisitos:

1. Estudo de viabilidade, onde foi estudado a estrutura das páginas de serviços da organização, obtendo informações também relacionadas aos processos e atividades diárias realizadas pelos usuários do sistema e os recursos disponíveis;
2. Elicitação e análise de requisitos, onde foi feito reuniões junto ao cliente para buscar informações a respeito do problema;
3. Especificação dos requisitos, colocando-os em uma forma padronizada;
4. Validação dos requisitos junto ao usuário.

Para estas atividades utilizamos ferramentas de apoio, tais como documento de requisitos, para estabelecer e registrar os requisitos de usuário.

2.3 ARQUITETURA DE SOFTWARE

Segundo Shawn e Garlan (1996) apud (Sizo, Lino, & Favero, 2010) uma arquitetura de software define o que é sistema em termos computacionais e o relacionamento entre esses componentes.

Além disso, Krafzig, Banke e Slama (2004) destacam a arquitetura como o esqueleto do sistema, sendo, portanto, o plano de mais alto-nível para a construção de sistemas. A arquitetura visa diminuir dificuldades no desenvolvimento de sistemas complexos, buscando uma forma de padronizar e estruturar a utilização dos componentes, facilitando o reuso na construção do sistema. “À medida que se aumenta a complexidade e abrangência dos sistemas de informação, também aumenta a dificuldade em organizar e estruturar seus componentes” (Sordi, Marinho e Nagy, 2006).

2.4 ARQUITETURA WEB

No final de 1993, Roy Fielding, cofundador do Apache HTTP Server Project, ficou preocupado com o problema de escalabilidade da Web. Após a análise,

Fielding reconheceu que a escalabilidade da Web era governada por um conjunto de restrições de chave. Ele e outros se propuseram a melhorar a implementação da Web com uma abordagem pragmática: satisfaça uniformemente todas as restrições para que a Web possa continuar a expandir (MASSÉ, 2011, tradução nossa).

MASSÉ (2011, tradução nossa) destaca as restrições propostas por Fielding, que as agrupou em seis categorias e que são coletivamente referidas como o estilo arquitetônico da web: Cliente-servidor, interface uniforme, sistema em camadas, *cache*, *stateless* e código sob demanda.

Web services são servidores web de propósito construído que suportam as necessidades de um site ou qualquer outra aplicação. Os programas cliente usam interfaces de programação de aplicativos (APIs) para se comunicar com serviços web. De um modo geral, uma API expõe um conjunto de dados e funções para facilitar as interações entre programas de computador e permitir que eles troquem informação (MASSÉ, 2011, tradução nossa).

O estilo arquitetônico REST é comumente aplicado ao design de APIs para os modernos serviços web. Uma API da Web em conformidade com o estilo arquitetônico REST é uma API REST (MASSÉ, 2011, tradução nossa).

2.5 FRAMEWORK VUE



Vue.js é uma biblioteca em javascript utilizada no desenvolvimento de interfaces web. Ao utilizar o padrão *Model View ViewModel*, seu objetivo é prover os benefícios de uma estrutura de dados reativa e orientada à componentes com uma API (*Application Programming Interface*) simples (GOMES, 2016). É uma estrutura progressiva para a criação de interfaces de usuário. Ao contrário de outras estruturas monolíticas, o Vue é projetado desde o início para ser adotável de maneira incremental (Vue.js).

A principal proposta do Vue é transformar a camada visual da página em um sistema orientado a dados. Normalmente usa-se um paradigma imperativo para controlar elementos da página, o que torna o código repetitivo, difícil e propenso a erros. Vue recorre à reatividade para montar uma estrutura em que os elementos da

página estão diretamente conectados a uma camada de dados que controla todo seu comportamento. Isso permite que o programador se concentre na lógica da aplicação ao invés de ter que se preocupar em modificar determinado elemento. (GOMES, 2016).

Uma característica do Vue é a abstração dos componentes. Cada componente pode possuir dados, métodos e eventos agregados dentro de seu próprio escopo. Neste caso, os componentes não influenciam nos dados externos de seu escopo, e isso permite que o componente seja facilmente reutilizável. Na figura 8 apresenta um componente utilizando o *framework* Vue.

Figura 8. Template e Script de um componente Vue.js

```
<template>   
</template>  
  
<script>  
  import Servico from '@models/Servico'  
  import { mapGetters } from 'vuex'  
  import vSelect from 'vue-select'  
  
  export default {   
  }  
</script>
```

Fonte: Elaboração própria

Os componentes Vue são altamente reutilizáveis. A partir de um mesmo componente é possível declarar quantos componentes forem necessários. Cada componente possui a declaração de sua classe *Script* utilizando a linguagem de programação EcmaScript 6.

3 ESTÁGIO SUPERVISIONADO

O estágio supervisionado foi realizado junto à Organização Central de Atendimento, na Diretoria da Organização Central de Atendimento, sob supervisão da coordenadora responsável, Herlen Duarte Jorge.

Para atingir o objetivo de desenvolver uma API REST que permita a escalabilidade e interoperabilidade com outras aplicações para o compartilhamento de informações dos serviços prestados na empresa, além de um website que melhore a gestão, a estrutura de apresentação, e a normalização das informações dos serviços, foram realizadas atividades de engenharia de requisitos, verificação e validação, modelagem, elaboração do diagrama de Entidade-Relacionamento, dentre outras.

A solução está atualmente disponível para inserção de informações dos serviços pela organização e a disponibilização de seu acesso em seu website e o consumo da API por outras aplicações.

3.1 ORGANIZAÇÃO

A Organização Central de Atendimento é uma organização privada de direito público e vinculada ao Governo do Estado do Acre. Fornece um modelo centralizado de serviços públicos e agrega um conjunto de órgãos e serviços para que possam ser oferecidos aos cidadãos.

Tem como finalidade centralizar os órgãos e serviços de responsabilidade dos municípios, estado e governo federal em um local físico e facilitar a locomoção dos cidadãos de forma que possam utilizar os serviços fornecidos pelos órgãos públicos em um único local e prover qualidade de atendimento.

O atendimento ao cidadão na organização funciona sob o regime da Política de Atendimento ao Cidadão, que estabelece o relacionamento entre a organização prestadora de serviços ao cidadão e o Estado, bem como os procedimentos que deverão ser adotados para o fornecimento do atendimento pela organização aos cidadãos.

Para melhorar o relacionamento entre o Estado e a população e para que todos tenham o direito a receber dos órgãos públicos as informações de interesse. Uma maneira democrática na prestação de serviços públicos e um processo constante de formas inovadoras do atendimento aos cidadãos em todo o Estado (Governo do Estado do Acre, 2008).

A Política de excelência no atendimento ao cidadão foi criada por meio do Decreto estadual N°3.357, de 20 de agosto de 2008, para garantir: a universalização do acesso às informações e aos serviços públicos e de natureza pública; promoção da qualidade e celeridade dos processos de atendimento; o resgate da credibilidade do cidadão no serviço público e a dignidade do agente público (Guia de Serviços Públicos, entre 2015 e 2017).

A atendimento ao cidadão apresenta uma estrutura física organizada em Praças e Órgãos. As praças são identificadas por cores e, em casa praça encontra-se um número variado de órgãos. Na Figura 9, O DETRAN, na praça amarela da organização.

Figura 9. DETRAN na Praça amarela da OCA

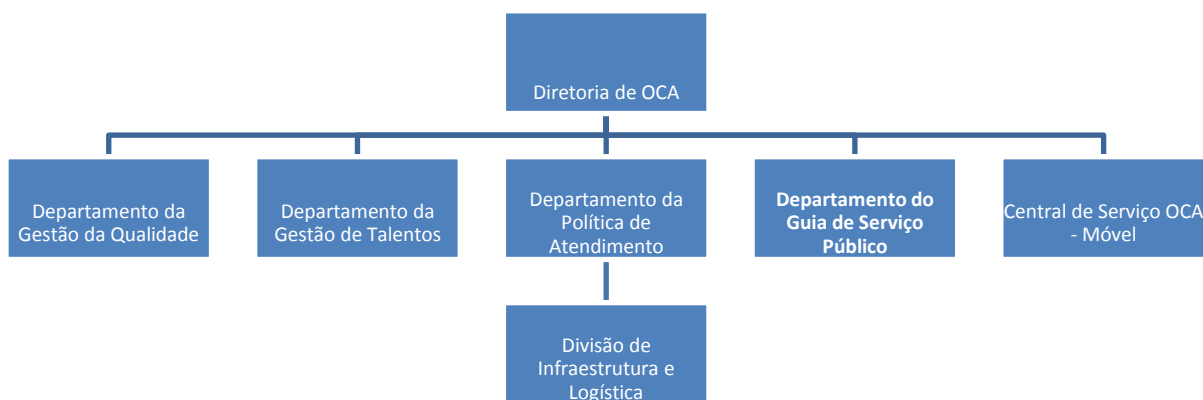


Fonte: Agência de Notícias do Acre.

3.2 ESTRUTURA ORGANIZACIONAL

A estrutura organizacional da Organização Central de Atendimento é mostrada na figura 10. Esta é uma versão reduzida do organograma da instituição, apresentando, em destaque, o setor onde foi realizado o estágio supervisionado. O Departamento de Guia de Serviços Públicos é o setor responsável pela gestão das informações dos serviços do Guia de Serviços Públicos.

Figura 10. Organograma da OCA



Fonte: Adaptado da Diretoria da OCA

3.3 ATIVIDADES DESENVOLVIDAS

Para atingir o objetivo do estágio, foram realizadas as seguintes atividades:

- a) Identificação do problema;
- b) Engenharia de requisitos;
- c) Modelagem de Banco de Dados;
- d) Prototipação;
- e) Implementação;
- f) Verificação & Validação,

3.3.1 IDENTIFICAÇÃO DO PROBLEMA

O problema que desencadeou o estágio surgiu a partir da dificuldade do setor do Guia de Serviços da OCA em utilizar seu atual sistema para disponibilizar as informações sobre seus serviços para que outras aplicações venham consumi-las, não havendo, portanto, interoperabilidade. Além disso, gerenciar as informações dos serviços públicos, realizando operações tais como: adicionar, alterar e excluir é uma atividade que gestores vinham passando por dificuldades. Atualmente a website da OCA possui um atendimento com milhares de serviços e dezenas de órgãos, e, em cada serviço, um conjunto estruturado de campos de informações tais como: onde retirar, quando retirar e quem pode retirar, contando ainda com a possibilidade de variações, onde os serviços podem ou não ter um determinado campo e, somado à estes problemas, a quantidade de locais para cada serviço e em diferentes municípios.

Com o intuito de estabelecer o problema a ser resolvido, foi realizado uma reunião com os gestores da organização para discutir à respeito dos problemas enfrentados pelos usuários na realização das atividades de manutenção do

conteúdo dos serviços e o que poderia ser melhorado. Na ocasião, foi demonstrado a estrutura dos serviços, a maneira como os serviços são cadastradas e foi descrito o fluxo necessário para se obter a informação de um determinado serviço.

3.3.2 LEVANTAMENTO DE REQUISITOS

Com os problemas identificados e a viabilidade do projeto avaliada na revisão bibliográfica, foi possível se obter uma versão de um documento de requisitos, descrevendo cada artefato do sistema de maneira que atendesse às necessidades do usuário. Os requisitos funcionais abordam às funcionalidades da API, descrevendo de maneira técnica as restrições das *endpoints* e dos recursos que seriam fornecidos por meio de requisições HTTP. O documento de requisitos do sistema pode ser visto no apêndice A.

O quadro 1 apresenta os requisitos funcionais do sistema.

Tabela 1. Requisitos funcionais

| ID | Funcionalidade | Necessidades | Prioridade |
|------|--|--|------------|
| RF01 | Login/ Autenticação de usuários do sistema | Autenticação para perfis de usuários autorizados a acessar informações gerenciais dos serviços | Essencial |
| RF02 | Gerenciamento de Informações dos Serviços | O Website permite a publicação, a edição e exclusão de serviços. | Essencial |
| RF03 | Visualização dos serviços | Permite a visualização dos serviços. | Essencial |
| RF04 | Atribuições de serviços aos locais | Permite a atribuição de um mesmo serviço para diversos locais | Importante |
| RF05 | Preenchimento de conteúdo de serviço | Permite a atribuição de campos customizados à um mesmo serviço. | Importante |
| RF06 | Estatísticas | Permite a visualização de informações de cunho estatístico dos serviços. | Importante |

Fonte: Elaboração própria

O quadro 2 apresenta os requisitos não funcionais do sistema, relacionados às funcionalidades emergentes do sistema.

Tabela 2. Requisitos não funcionais

| ID | Funcionalidade | Necessidades | Prioridade |
|-------|----------------------------|---|------------|
| RNF01 | <i>Endpoints</i> | Rotas para métodos e recursos | Importante |
| RNF02 | Autenticação e autorização | Autenticação de perfis de usuário com oauth | Importante |
| RNF03 | Boas práticas | Utilizar boas práticas de API | Importante |
| RNF04 | Confiabilidade | Requisições precisam ser retornadas com sucesso | Importante |

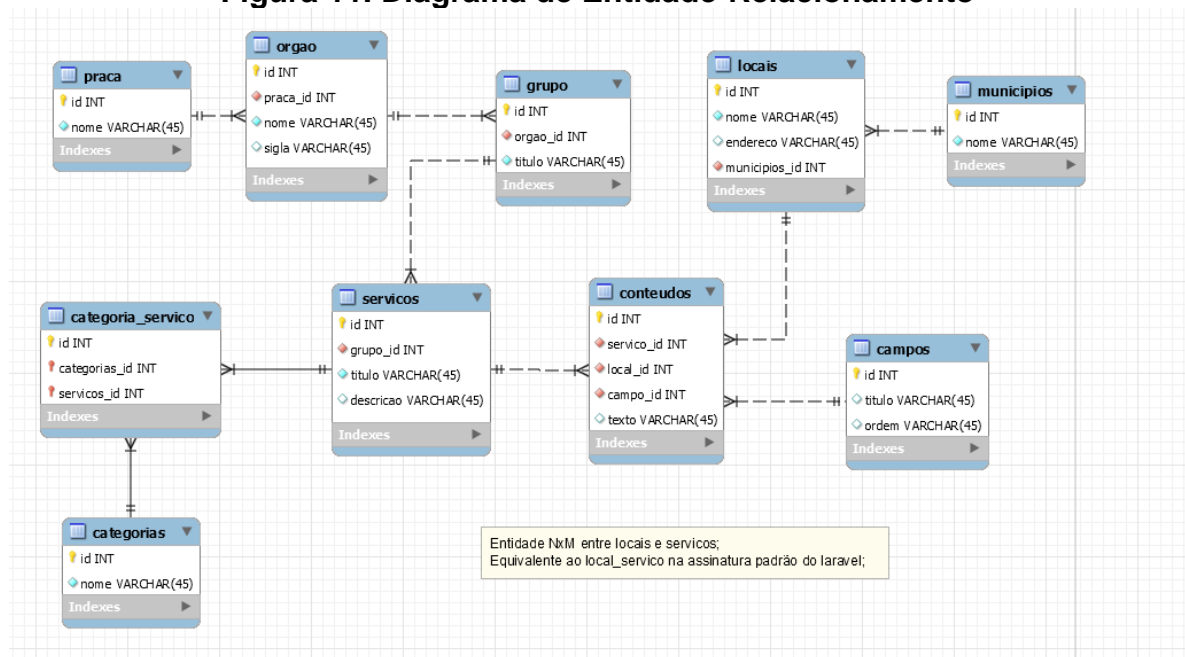
Fonte: Elaboração própria

3.3.3 DIAGRAMA DE ENTIDADE-RELACIONAMENTO

Após a elaboração do documento de requisitos, foi desenhado um modelo de Entidade-Relacionamento pensando nas entidades do sistema e de acordo com o documento de requisitos, pensando no relacionamento entre as entidades e uma estrutura que fosse escalável para a normalização das informações dos serviços.

E, também, foi levado em consideração a forma como as *endpoints* seriam organizadas para que se pensasse no modelo. Na figura 11, a entidade de conteúdos representa a informação de um serviço de um determinado campo e local, com uma informação textual.

Figura 11. Diagrama de Entidade-Relacionamento



Fonte: Elaboração própria

3.3.4 DESENVOLVIMENTO DA API

Com a validação do DER e os requisitos validados, foi desenvolvido uma API utilizando recursos para a programação do *Backend* com o framework *Laravel*. O *laravel* utiliza o paradigma orientado à objetos e um padrão de projetos MVC. Na figura 12, mostra a parte lógica do Backend.

Figura 12. Métodos para requisição das informações do Laravel

```

public function index(Request $request)
{
    $this->addAcesso('servicos');
    if ($request->search) {
        return $this->search(Servico::class, $request, null, 'titulo');
    }

    $servicos = Servico::orderBy($request->sort ?? 'titulo')->get();
    return $this->sendCollection($servicos, $request->include);
}

/** ...
public function store(Request $request)
{
    $this->validator($request);

    $orgao = Orgao::findOrFail($request->orgao_id);
    $grupo = Grupo::create($request->all());

    $servico = new Servico;
    $servico->fill($request->all());
    $servico->grupo()->associate($grupo);
    $servico->save();

    // Anexa as tags
    $servico->tags()->attach($request->tags);

    return $this->sendModel($servico, 'grupo', 201);
}

/** ...
public function show(Request $request, Servico $servico)
{
    $this->addAcesso('servicos', $servico->id);
    return $this->sendModel($servico, $request->include);
}

/** ...
public function update(Request $request, Servico $servico)
{
    $this->validator($request, $this->updateRules);

```

Fonte: Elaboração própria.

O *Laravel* utiliza o componente *routes* para declarar as *endpoints*, tais como as rotas, que funcionam como entrada para a requisição das informações via HTTP. Na figura 13, é mostrada a representação as *endpoints* nas rotas do *Laravel* para invocação de métodos dos controllers.

Figura 13. Declaração das endpoints no Laravel

```
#-----
# Orgaos
#-----
Route::resource('orgaos', 'OrgaoController', ['except' => ['create', 'edit']]);
Route::prefix('orgaos')->group(function () {

    # orgaos/grupos
    Route::get('/{orgao}/grupos', 'OrgaoController@listGrupos')
        ->name('orgaos.grupos.index');
    Route::post('/{orgao}/grupos', 'OrgaoController@storeGrupo')
        ->name('orgaos.grupos.index');
});

#-----
# Pracas
#-----
Route::resource('pracas', 'PracaController', ['except' => ['create', 'edit']]);
Route::prefix('pracas')->group(function () {

    # praca/orgaos
    Route::get('/{praca}/orgaos', 'PracaController@listOrgaos')
        ->name('pracas.orgaos.index');
});

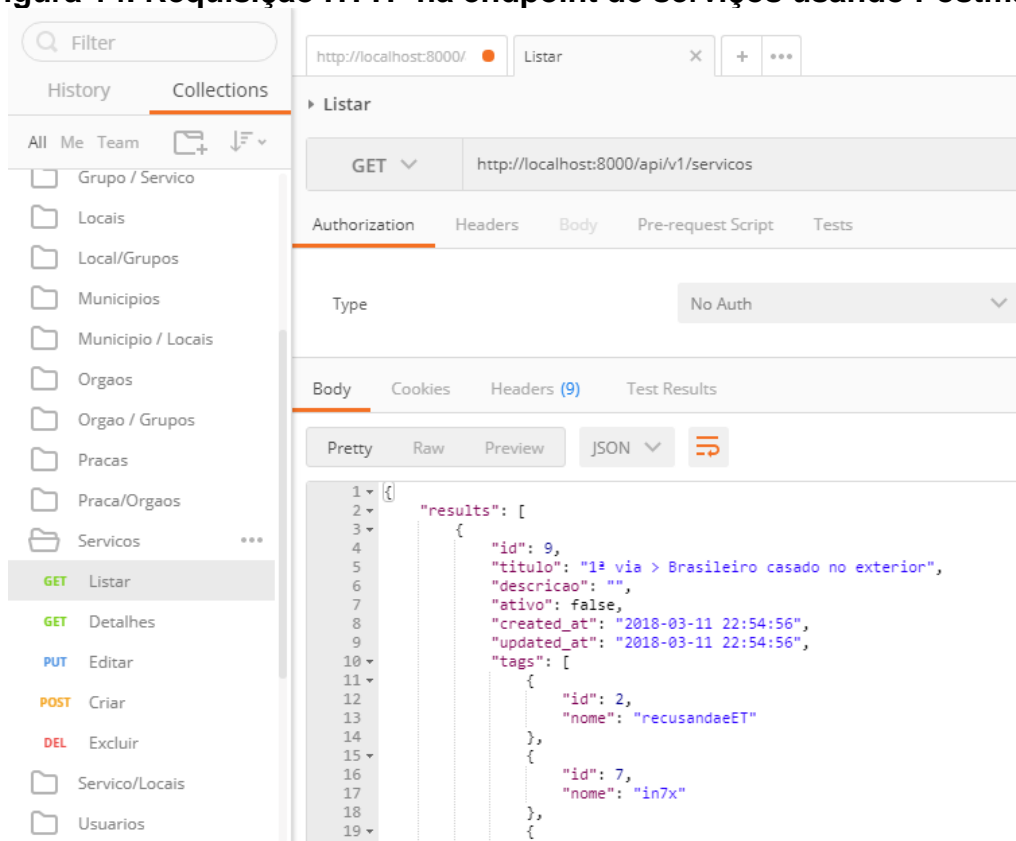
#-----
# Servicos
#-----
Route::resource('servicos', 'ServicoController', ['except' => ['create', 'edit']]);
Route::prefix('servicos')->group(function () {

    # servico/locais
    Route::get('/{servico}/locais', 'ServicoController@listLocais')
        ->name('servicos.locais.index');
    Route::post('/{servico}/locais', 'ServicoController@storeConteudo')
        ->name('servicos.locais.store');
    Route::get('/{servico}/locais/{local}', 'ServicoController@getConteudos')
        ->name('servicos.locais.show');
});
```

Fonte: Elaboração própria

Essas *endpoints* foram testadas utilizando a ferramenta *Postman*. Estas requisições retornam uma *response* da API em formato JSON (*Javascript Object Notation*), fornecendo uma visualização mais estruturada do conteúdo retornado, como na figura 14.

Figura 14. Requisição HTTP na endpoint de serviços usando Postman



Além disso do desenvolvimento da API, é necessário que as informações de uso das *endpoints* estejam estruturadas em uma documentação para que os desenvolvedores de aplicações clientes saibam como realizar as requisições. Na figura 15, a documentação da API do guia desenvolvida utilizando a ferramenta *Swagger*.

Figura 15. Documentação da API Guia Oca



Fonte: Elaboração própria

3.3.5 PROTOTIPAÇÃO

A prototipação da interface web consistiu em elaborar um design para a solução dos problemas de navegação dos usuários. Como forma de diminuir evitar redundâncias, como os órgãos, serviços e campos foram categorizados em entidades, como mostrado na Figura 8. A prototipação das telas do sistema buscou seguir a mesma estrutura para que cada entidade mantivesse um escopo e uma estrutura de navegação no padrão REST de API.

A figura 16 apresenta o protótipo da tela de grupo de serviços:

Figura 16. Protótipo da tela de preenchimento de conteúdo dos serviços

Guia Oca

http://

Início Usuários Estatística Ir para o site Sair

Praças
Orgãos
Grupos
Serviços
Locais
Municípios

1ª via > Brasileiro Nato > A partir de 16 anos

Salvar Cancelar Visualizar

Onde Solicitar?

Delegacia Geral de Policia
Agência: Posto de Identificação – 15
Endereço: Rua José Pereira Gurgel, 291 – Centro
Município: Bujari
Dias e horário de atendimento: de segunda a sexta-feira, das 7h às 13h.

Quem pode solicitar?

Somente o Pai OU Mãe OU Responsável.

Posso solicitar?

Para solicitar este serviço é preciso atender aos seguintes requisitos:
A criança ou adolescente deve estar acompanhado do pai OU mãe OU responsável.

Locais...

Adicionar um local...

Oca Rio Branco ☒
Oca Xapuri ☒
Delegacia Geral ☒
Posto de Identificação ☒
Other ☒
Another ☒

Campos...

Adicionar um campo...

Onde solicitar? ☒
Quem pode solicitar? ☒
Posso solicitar? ☒

Fonte: Elaboração própria

Os demais protótipos podem ser encontrados no apêndice B deste trabalho.

3.3.5 VERIFICAÇÃO E VALIDAÇÃO

A verificação foi realizada analisando se cada requisito e componente do sistema estavam de acordo com as necessidades da cliente. Foi feita uma análise de cada funcionalidade implementada e se ele estava de acordo com as necessidades.

Com esses requisitos, foi elaborado um documento de testes (Apêndice C), com questões referindo-se à cada funcionalidade do sistema, e através desta atividade pôde-se validar cada funcionalidade do sistema. O quadro 1 apresenta os testes de funcionalidade de usuário. O documento de requisitos pode ser visto no apêndice A.

Quadro 1. Testes de funcionalidades de interface de usuário

Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.1

| | |
|--|---|
| Casos de Testes: RF03 - Navegue através das páginas de serviços, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado: Critério atendido, todas as páginas estão facilmente compreensíveis | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.2 | |
| Casos de Testes: RF01 - Verifique que o usuário não autorizado não pode acessar a tela de alteração dos conteúdos. Clique em um grupo de serviços e veja se há botões disponíveis para a alteração. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Não há botões para alteração de conteúdos sem prévia autenticação. | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.3 | |
| Casos de Testes: RF01 – Verifique que o usuário autorizado pode realizar o login no sistema. Utilize a o login “ <i>admin</i> ” e senha “ <i>admin</i> ”. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Atendido, o usuário pode realizar o login. | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.4 | |
| Casos de Testes: RF01, RF04 e RF05 - Verifique que o usuário pode alterar o conteúdo das páginas de serviços. Selecione um grupo, depois selecione um serviço e veja se a página de alteração aparece ao lado do conteúdo. Clique nos botões de alteração de conteúdo, altere o conteúdo e salve. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado O usuário pode realizar alterações no conteúdo dos serviços. | |

| | |
|--|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.5 | |
| Casos de Testes: RF02 – Verifique que ao adicionar, alterar ou excluir um conteúdo de serviço há o feedback apropriado. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Operações de adicionar, alterar e excluir retornam um feedback na tela indicando que foi realizada com sucesso. | |

Além disso, foi verificado também os requisitos não funcionais, que estão associados às funcionalidades emergentes do sistema. O quadro 2 apresenta o teste funcional para requisitos não funcionais do sistema.

Quadro 2. Teste de requisitos não funcionais

| | |
|--|---|
| Requisito de Teste: TESTE FUNCIONAL 1.1 | |
| Casos de Testes: RNF01 - Verifique que as <i>endpoints</i> estão retornando <i>Status Code</i> corretamente para os casos de sucesso ou falha nas requisições. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |

| | |
|---|---|
| Observações/Resultado Dados para os casos de sucesso tiveram retorno de cabeçalho com <i>status code</i> 200 e/ou 201 para operações com sucesso e 404 e/ou 500 para os casos de falha. | |
| Requisito de Teste: TESTE FUNCIONAL 1.2 | |
| Casos de Testes: RNF01 e RNF04 - Verifique que as <i>endpoints</i> estão retornando os dados corretamente para as entidades e suas dependências. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Os dados são retornados em Json para a aplicação, com todas as suas dependências quando requisitadas. | |
| Requisito de Teste: TESTE FUNCIONAL 1.3 | |
| Casos de Testes: RNF01 e RNF04 - Verifique que o usuário não pode acessar as rotas de inclusão, edição e exclusão das entidades sem autorização. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Alterações no banco de dados são possíveis somente através da autenticação com login de API através do Oauth. Não foi possível o acesso a rotas não autorizadas ao usuário comum. | |
| Requisito de Teste: TESTE FUNCIONAL 1.4 | |
| Casos de Testes: RNF02 - Verifique que o usuário autenticado tem permissões para realizar inserções, alterações e exclusões de dados, verificando também se a requisição retorna os dados de acordo com as alterações realizadas. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Todas as <i>endpoints</i> retornaram seus dados de acordo com as alterações, não havendo, portanto, problemas de integridade. | |

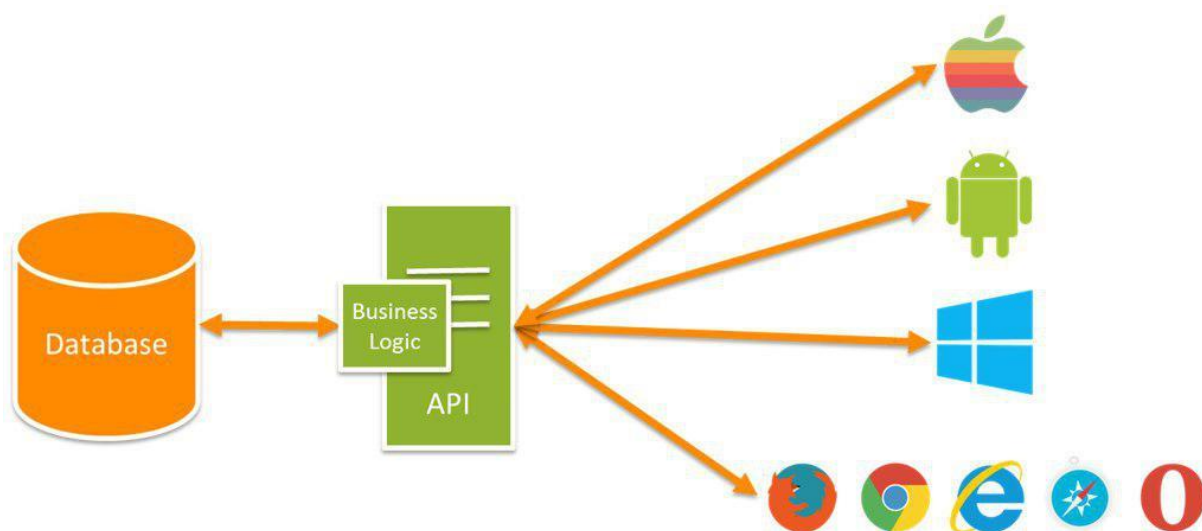
Utilizando esta abordagem, pôde-se verificar se o sistema atendia aos requisitos funcionais e, através dos testes foi possível garantir que o sistema atende às necessidades do usuário.

3.4 RESULTADOS OBTIDOS

Os resultados obtidos com este trabalho mostram que a solução proposta cumpriu seu objetivo. A solução permite a interoperabilidade através da API com outras aplicações clientes. A API, desenvolvida com o framework *Laravel* está de acordo com os padrões arquiteturais de uma aplicação REST, tornando-a uma API RESTful.

Esta arquitetura permite uma escalabilidade maior no desenvolvimento de aplicações clientes, além de permitir o seu consumo por uma quantidade ilimitada de aplicações.

Figura 17. Solução desenvolvida



Também foi elaborada uma documentação que facilita o entendimento da utilização das endpoints da API. Na documentação pode ser encontrada informações à respeito de cada *endpoints* e os recursos que são retornados em suas requisições. A figura 18 apresenta a documentação da API.

Figura 18. Documentação da API

Documentação da API Guia Oca ^{1.0.0}

[Base URL: guiaoca-api.gov.br/api/v1]

Esta é uma API Restful para consumo de informações dos serviços do Guia de Serviços Públicos da Organização Central de Atendimento - OCA. Para mais informações, visite o site gsp.ac.gov.br.

[Find out more about Swagger](#)

Schemes
HTTP

Authorize

Autenticação

Informações para autenticação

POST /auth/token Autenticação de cliente

Campos

consumo de informações dos campos

GET /campos Listar campos

POST /campos Adicionar campo

PUT /campos/{id} Atualizar campo

DELETE /campos/{id} Excluir um campo

GET /campos/{id} Buscar um campo

Cada uma das *endpoints* listadas na figura 16 apresentam informações que devem ser seguidas pelo programador para o seu devido consumo. A figura 17 mostra as informações da *endpoints* de autenticação. Estas informações referem-se aos argumentos que devem ser submetidos ao servidor para que o conteúdo requisitado seja retornado para esta *endpoint*.

Figura 19. Informações da endpoint de autenticação

Autenticação Informações para autenticação ▼

POST /auth/token Autenticação de cliente

Autenticação para o cliente/aplicação que requisita acesso ao consumo da api

Parameters Try it out

| Name | Description |
|---|---|
| grant_type * required string (formData) | Ex: password |
| client_id * required integer (formData) | ID do cliente/aplicação |
| client_secret * required string (formData) | Chave de autorização do cliente/aplicação |
| username * required string (formData) | Nome do cliente/aplicação |
| password * required string (formData) | Senha de autorização do cliente/aplicação |

Responses Response content type: application/json ▼

| Code | Description |
|------|-------------|
|------|-------------|

Além disso, foi desenvolvido o website de acordo com os requisitos funcionais. Os requisitos foram verificados e validados, como mostrado no tópico anterior. O website desenvolvido permite mais eficiência nas atividades de gerenciamento de conteúdo dos serviços. A figura 20 apresenta a tela de inserção de serviços. No cenário anterior, os campos e conteúdos eram inseridos de maneira textual, gerando redundância não controlada no sistema. Neste cenário é possível normalizar o serviço ao máximo e de maneira dinâmica. Para cada serviço, pode-se associar quantos locais forem necessários à um mesmo, além do usuário poder também adicionar campos e conteúdos de acordo com a estrutura de cada serviço.

Figura 20. Tela de inserção de conteúdo dos serviços

The screenshot displays the 'Guia OCA' web application interface. The top navigation bar includes links for 'Início', 'Órgãos', 'Praças', 'Serviços Online', and 'Blog da OCA'. The left sidebar contains a menu with 'Dashboard', 'Grupos', 'Serviços', 'Órgãos', 'Municípios', 'Locais', 'Campos', 'Praças', 'Categorias', 'Tags', and 'Arquivos'. The main content area is titled 'Central de Serviço Público - OCA Rio Branco > Achados e Perdidos > 1 via'. It features a form for 'Preencha o conteúdo do serviço' with three main sections: 'Achados e Perdidos' (containing '1 via' and 'Sem descrição'), 'COMO FAÇO PARA RETIRAR?' (with a rich text editor and a text input field containing 'Ir ao local de solicitação'), and 'QUAL O PRAZO?' (with a rich text editor and a text input field containing '12 dias'). On the right, there is a blue button 'Atribuir conteúdo aos locais selecionados'. Below it, the 'Selecionar Campos' section shows two dropdown menus: 'COMO FAÇO PARA RETIRAR?' and 'QUAL O PRAZO?'. The 'Selecionar Locais' section shows a dropdown menu with 'CAC - Sobral' selected. A blue box below this section states 'Este serviço já está atribuído aos locais abaixo.' and lists three locations: 'Central de Serviço Público - OCA Rio Branco', 'Central de Serviço Público - OCA Xapuri', and 'Delegacia de Exemplo'. The bottom footer contains three columns: 'Serviços e Informações' (with 'Diretoria OCA'), 'Utilidades' (with 'Telefones Úteis'), and 'Portais Governamentais' (with 'Portal do Governo do Acre').

Além disso, para otimizar as atividades do usuário, as telas funcionam de maneira reativa, através dos recursos da linguagem EcmaScript 6. Dessa maneira, as páginas não são totalmente carregadas quando há uma nova requisição. Como o Vue é um framework orientado à componentes, cada componente possui suas próprias propriedades, funcionando como se fossem subpáginas dentro do escopo de uma página.

Isso significa que cada componente de uma página será imediatamente atualizada sempre que houver uma modificação, permitindo mais segurança e confiabilidade aos usuários do sistema. Quando houver muitos usuários trabalhando na edição de serviços, os riscos de haver redundâncias será menor, pois os dados na página são atualizados imediatamente sem a necessidade de o usuário recarregá-la. Esse sistema encaixa-se em uma categoria de *Single Page Application*.

4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES

Neste capítulo são apresentadas as considerações finais acerca das atividades do estágio e as recomendações para trabalhos futuros.

4.1 CONSIDERAÇÕES FINAIS

Este trabalho permitiu uma visão mais abrangente a respeito das necessidades de organizações que utilizam sistemas de gerenciamento de conteúdo. Foi possível ver problemas atuais quando as atividades envolvem a gestão de grandes quantidades de informações. Este trabalho mostrou as atividades de gestão de informações dos serviços da Organização Central de Atendimento, percorrendo sobre os problemas e a apresentação da solução.

Este trabalho também proporcionou um período de grande aprendizado, pois durante o seu desenvolvimento pode-se ter inúmeros *insights* à respeito da solução do problema e uma experiência maior para outros trabalhos neste ramo.

Durante a realização deste trabalho tivemos todo o apoio possível pelos membros da organização, fornecendo informações importantes e com confiança. Houve bastante disposição das pessoas em colaborar com sugestões durante as reuniões. A supervisora de TI Herlen Duarte Jorge acompanhou todos os passos

deste trabalho, fornecendo feedback e apoio na comunicação com outros profissionais interessados no trabalho.

Haviam várias dificuldades para a especificação dos requisitos do sistema no início deste trabalho, pois à medida que conhecíamos diferentes estruturas de serviços que apareciam, surgia a necessidade de alterar o documento de requisitos, adaptar o modelo de banco de dados, os protótipos e toda a estrutura da aplicação.

Através deste trabalho, explorou-se problemas e, neste sentido, novas soluções foram apresentadas. Neste caso, desenvolveu-se uma solução bem específica, envolvendo a gestão de vários conteúdos de serviços sem a estrutura de um CMS.

Conclui-se que o problema deste trabalho foi resolvido e todas as necessidades do cliente foram atendidas. A solução permite tanto a interoperabilidade com aplicações clientes quanto a eficiência nas atividades de gestão de conteúdos de serviços.

4.2 RECOMENDAÇÕES

Apesar de a solução proposta atender às necessidades do usuário, é possível também implementar outras funcionalidades importantes à solução. É importante a integração do sistema atual com o sistema SIGA da organização. O SIGA é um sistema utilizado no gerenciamento de atendimento, filas e balcões, sendo, portanto, o sistema mais utilizado pelos usuários da organização. Neste caso, é interessante que haja um estudo de viabilidade para o SIGA consumir as informações da API para a visualização do conteúdo dos serviços. Este estudo permitiria seria importante para o desenvolvimento de um módulo de visualização de serviços, permitindo que os atendentes de balcões acessem as informações dos serviços com mais rapidez, para que não seja necessário o acesso via website.

Outra recomendação é o desenvolvimento de um aplicativo para plataforma móvel do guia de serviços da OCA, para que o usuário possa visualizar as

informações e solicitar mais informações através do atendimento on-line da organização.

REFERÊNCIAS BIBLIOGRÁFICAS

BASS et al. **Market Assessment of Component-Based Software Engineering**. Carnegie Mellon: Software Engineering Institute, 2000.

GARLAN, David; SHAW, Mary. **An Introduction to Software Architecture: Advances in Software Engineering and Knowledge Engineering**. Pittsburgh: Carnegie Mellon University, 1994.

GOVERNO DO ESTADO DO ACRE. (2008). **Política de Atendimento ao Cidadão**. disponível em: <<http://www.ac.gov.br/wps/portal/acre/Acre/cidadao>>. Acesso em 12 de Março de 2018.

KRAFZIG, D; BANKE, K. e SLAMA, D. **Enterprise SOA: Service-Oriented Architecture Best Practices**. Cidade: Indianapolis: Prentice Hall, 2004.

MASSÉ, Mark. **Rest API design rulebook: Designing consistent restful webservice interfaces**. Sebastopol: O'reilly, 2011.

PRESSMAN, R. S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011.

REINEHR, S. S. **Reuso sistematizado de software e linhas de produto de software no setor financeiro: Estudos de caso no Brasil**. São Paulo, 2008.

SIZO, A. M., LINO, A. D., & FAVERO, E. L. **Uma proposta de arquitetura de software para Construção e Integração de Ambientes virtuais de Aprendizagem**. Belém: risti, 2010.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

VUE.JS: Guide. disponível em: <<https://vuejs.org/v2/guide/>>. Acesso em 15 de Março de 2018.

APÊNDICES

APENDICE A – DOCUMENTO DE REQUISITOS

Documento de Requisitos da API Restful

GuiaOca

VERSÃO 2.0

Equipe:

Antônio Gabriel Almeida de Souza

Wallison Santos Ferreira

Histórico das Revisões

| Data | Versão | Descrição | Autor |
|-------------|---------------|---|--------------------------------------|
| 03/07/2017 | 1.0.0 | Especificação dos requisitos do sistema | Antonio Gabriel Wallison Ferreira |
| 01/01/2018 | 2.0.0 | Mudanças nos requisitos funcionais | Wallison Ferreira |

1. Escopo do produto

Uma API que fornece uma interface para que outras aplicações a utilizem com finalidade de se obter um recurso ou informação utilizando chamadas HTTP, de forma padronizada com os princípios de REST.

2. Missão do produto

Permitir a interoperabilidade e o uso compartilhado de recursos do servidor com aplicações Web e Mobile. Escalabilidade e manutenibilidade para que clientes internos e externos tenham a capacidade de utilizar os dados do servidor em suas aplicações clientes.

- ✓ Cadastro e Gestão de serviços dos órgãos públicos sob a perspectiva do usuário administrador;
- ✓ Cadastro de serviços dos órgãos públicos sob a perspectiva do usuário comum;
- ✓ Visualização de serviços dos órgãos sob a perspectiva do usuário cidadão;
- ✓ Website que permite a visualização dos serviços dos órgãos do estado utilizando os dados do servidor através da API;
- ✓ Permitir o cadastro e o gerenciamento de serviços pelo administrador como se fosse um gerenciador de conteúdo customizado e a visualização dos serviços pelos usuários cidadãos.

3. IDENTIFICAÇÃO DOS REQUISITOS

4.1 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações: essencial, importante e desejável. Abaixo temos a descrição de significado de cada uma dessas denominações:

| | |
|-------------------|---|
| Essencial | É o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente. |
| Importante | É o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim. |
| Desejável | É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada. |

4.2 Requisitos funcionais

Tabela 3. Requisitos funcionais

| ID | Funcionalidade | Necessidades | Prioridade |
|------|--|--|------------|
| RF01 | Login/ Autenticação de usuários do sistema | Autenticação para perfis de usuários autorizados a acessar informações gerenciais dos serviços | Essencial |
| RF02 | Gerenciamento de Informações dos Serviços | O Website permite a publicação, a edição e exclusão de serviços. | Essencial |
| RF03 | Visualização dos serviços | Permite a visualização dos serviços. | Essencial |
| RF04 | Atribuições de serviços aos locais | Permite a atribuição de um mesmo serviço para diversos locais | Importante |
| RF05 | Preenchimento de conteúdo de serviço | Permite a atribuição de campos customizados à um mesmo serviço. | Importante |
| RF06 | Estatísticas | Permite a visualização de informações de cunho estatístico dos serviços. | Importante |

4.3 Requisitos não funcionais

Tabela 4. Requisitos não funcionais

| ID | Funcionalidade | Necessidades | Prioridade |
|-------|----------------------------|--|------------|
| RNF01 | <i>Endpoints</i> | Rotas para métodos e recursos | Importante |
| RNF02 | Autenticação e autorização | Autenticação de perfis de usuário com <i>Oauth</i> | Importante |
| RNF03 | Boas práticas | Utilizar boas práticas de API | Importante |
| RNF04 | Confiabilidade | Requisições precisam ser retornadas com sucesso ou falha | Importante |

APENDICE B – PROTÓTIPOS

1. ESCOPO DE ÓRGÃOS

The screenshot shows a web browser window with the address bar displaying `http://guiacoca/api/v1/grupos/orgaos`. The page title is "Guia Oca". The navigation bar includes links for "Início", "Usuários", "Estatística", and "Ir para o site", along with a "Sair" button. The main content area is divided into a left sidebar and a central panel. The sidebar contains buttons for "Praças", "Órgãos", "Grupos", "Serviços", "Locais", and "Municípios". The central panel, titled "Órgãos", features a search bar labeled "Digite o nome do órgão...", a dropdown menu for "Praça", and an "Adicionar Órgão" button. Below these, a table lists several organizations, each with a pencil icon for editing and a cross icon for deletion.

| Órgão | Editar | Excluir |
|----------------------------|--------|---------|
| DETRAN | | |
| Instituto de Identificação | | |
| SESACRE | | |
| SINDCOL | | |
| SINE | | |
| Other | | |
| Another | | |

2. ESCOPO DE UM ÓRGÃO

The screenshot shows a web browser window with the title 'Guia Oca'. The address bar contains 'http://'. The navigation bar includes links: 'Início', 'Usuários', 'Estatística', 'Ir para o site', and a 'Sair' button with a power icon. The main content area is divided into a left sidebar and a main panel. The sidebar contains a vertical list of buttons: 'Praças', 'Orgãos', 'Grupos', 'Serviços', 'Locais', and 'Municípios'. The main panel is titled 'Instituto de Identificação' and contains a search bar with the placeholder 'Digite o nome do Grupo...' and an 'Adicionar Grupo' button. Below this is a table with five rows, each representing a document type. Each row has a text input field and two action buttons (edit and delete).

| Instituto de Identificação | | |
|------------------------------------|--|--|
| Digite o nome do Grupo... | | |
| Adicionar Grupo | | |
| Atestado de Antecedentes Criminais | | |
| Boletim de Ocorrência | | |
| Carteira de Identidade | | |
| Certidão de Dados | | |
| Nada Consta | | |

3. ESCOPO DE GRUPOS

The screenshot shows a web browser window with the title "Guia Oca". The address bar contains "http://". The navigation menu includes "Início", "Usuários", "Estatística", and "Ir para o site". A "Sair" button is located in the top right corner. The main content area is divided into two sections. On the left is a sidebar with buttons for "Praças", "Órgãos", "Grupos", "Serviços", "Locais", and "Municípios". The "Grupos" button is highlighted. The right section, titled "Grupos", contains a form for adding a new group. It has a text input field labeled "Digite o nome do grupo...", a dropdown menu labeled "Órgão", and an "Adicionar Grupo" button. Below this is a table listing existing groups with edit and delete icons for each.

| Grupos | |
|------------------------------------|-----------------|
| Digite o nome do grupo... | |
| Órgão | Adicionar Grupo |
| Atestado de Antecedentes Criminais | |
| Boletim de Ocorrência | |
| Carteira de Identidade | |
| Certidão de Dados | |
| Nada Consta | |
| Other | |
| Another | |

4. ESCOPO DE UM GRUPO

Guia Oca

http://guiadoca/api/v1/grupos/1/servicos

Início Usuários Estatística Ir para o site Sair

Praças

Orgãos

Grupos

Serviços

Locais

Municípios

Carteira de Identidade

Digite o nome do serviço..

Descrição do serviço..

Adicionar Serviço

| | | |
|---|--|--|
| 1º via > Brasileiro Nato > Até 15 anos | | |
| 1º via > Brasileiro Nato > A partir de 16 anos | | |
| 1º via > Brasileiro Naturalizado | | |
| 1º via > Brasileiro Nascido no exterior > Até 15 anos | | |

5. EDIÇÃO DE UM SERVIÇO

← → × 🏠

http://

🔍

Guia Oca

Início

Usuários

Estatística

Ir para o site

Sair 🏠

Praças

Orgãos

Grupos

Serviços

Locais

Municípios

1ª via > Brasileiro Noto > A partir de 16 anos

Salvar

Cancelar

Visualizar

Onde Solicitar?

Delegacia Geral de Policia
Agência Posto de Identificação – 15
Endereço: Rua José Pereira Gurgel, 291 – Centro
Município: Bujari
Dias e horário de atendimento: de segunda a sexta-feira, das 7h às 13h.

Quem pode solicitar?

Somente o Pai OU Mãe OU Responsável.

Posso solicitar?

Para solicitar este serviço é preciso atender aos seguintes requisitos:
A criança ou adolescente deve estar acompanhado do pai OU mãe OU responsável.

Locais... 🔍

Adicionar um local...

Oca Rio Branco

Oca Xapuri

Delegacia Geral

Posto de Identificação

Other

Another

Campos... 🔍

Adicionar um campo...

Onde solicitar?

Quem pode solicitar?

Posso solicitar?

APÊNDICE C – TELAS DO WEBSITE

1. TELA INICIAL DE USUÁRIO LOGADO NO SISTEMA

The screenshot displays the user interface of the OCA system. At the top, there is a navigation bar with the 'Guia OCA' logo on the left and a menu containing 'Início', 'Órgãos', 'Praças', 'Serviços Online', and 'Blog da OCA' in the center. On the far right of the navigation bar is an 'Admin' dropdown menu. Below the navigation bar, a left sidebar contains a list of menu items: 'Dashboard', 'Grupos', 'Serviços', 'Órgãos', 'Municípios', 'Locais', 'Campos', 'Praças', 'Categorias', 'Tags', and 'Arquivos'. The main content area is divided into two columns. The left column, titled 'Avisos da OCA', contains four identical news items. Each item is titled 'Suspensão Atendimento Defensoria Pública do Estado do Acre' with a date of '11 de Novembro de 2017'. The text of each item states: 'A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre'. Each item concludes with a 'Ver mais...' link. The right column, titled 'Órgãos e Entidades', features a list of various organizations, including AGEAC, ACISA, Central de Serviço Público - OCA Rio Branco, SEBRAE/AC, CEJUSC, CORINES, CBMAC, Correspondente CAIXA, Correspondente MAIS BB, Defensoria Pública do Estado do Acre, Depasa, PROCON, DETRAN, ELETROBRAS, ECT, and Fundeseg.

Guia OCA Início Órgãos Praças Serviços Online Blog da OCA Admin

Dashboard

Grupos

Serviços

Órgãos

Municípios

Locais

Campos

Praças

Categorias

Tags

Arquivos

Qual serviço você está procurando?

Avisos da OCA

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

[Ver mais...](#)

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

[Ver mais...](#)

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

[Ver mais...](#)

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

[Ver mais...](#)

Órgãos e Entidades

AGEAC - Agência Estadual Reguladora de Serviços Públicos

ACISA - Associação Comercial, Industrial de Serviço e Agrícola do Acre

Central de Serviço Público - OCA Rio Branco

SEBRAE/AC - Central do Empreendedor

CEJUSC - Centro Judiciário de Soluções de Conflito

CORINES - Coordenação de Registro e Inspeção Escolar

CBMAC - Corpo de Bombeiros Militar

Correspondente CAIXA

Correspondente MAIS BB

Defensoria Pública do Estado do Acre

Depasa - Departamento Estadual de Pavimentação e Saneamento

PROCON - Departamento Estadual de Proteção e Defesa do Consumidor - Rio Branco

DETRAN - Departamento Estadual de Trânsito

ELETROBRAS - Distribuição Acre

ECT - Empresa Brasileira de Correios e Telégrafos

Fundeseg - Fundo Estadual de Segurança Pública

2. TELA INICIAL DE USUÁRIO NÃO LOGADO

Guia OCA
Início
Órgãos
Praças
Serviços Online
Blog da OCA
Entrar

Qual serviço você está procurando?

Avisos da OCA

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

Ver mais...

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

Ver mais...

Suspensão Atendimento Defensoria Pública do Estado do Acre
11 de Novembro de 2017

A Defensoria Pública do Estado do Acre informa aos cidadãos que entrará de recesso no período de 11 de dezembro de 2017 a 08 de Janeiro 2018. O atendimento será normalizado no dia 09 de Janeiro de 2018. Atenciosamente, Defensoria Pública do Estado do Acre

Ver mais...

Órgãos e Entidades

AGEAC - Agência Estadual Reguladora de Serviços Públicos
ACISA - Associação Comercial, Industrial de Serviço e Agrícola do Acre
Central de Serviço Público - OCA Rio Branco
SEBRAE/AC - Central do Empreendedor
CEJUSC - Centro Judiciário de Soluções de Conflito
CORINES - Coordenação de Registro e Inspeção Escolar
CBMAC - Corpo de Bombeiros Militar
Correspondente CAIXA
Correspondente MAIS BB
Defensoria Pública do Estado do Acre
Depasa - Departamento Estadual de Pavimentação e Saneamento
PROCON - Departamento Estadual de Proteção e Defesa do Consumidor - Rio Branco
DETRAN - Departamento Estadual de Trânsito
ELETROBRAS - Distribuição Acre
ECT - Empresa Brasileira de Correios e Telégrafos
Fundeseg - Fundo Estadual de Segurança Pública

3. LISTA DE ÓRGÃOS

Guia OCA
Início
Órgãos
Praças
Serviços Online
Blog da OCA
Entrar

Órgãos e Entidades

| | |
|--|---|
| AGEAC - Agência Estadual Reguladora de Serviços Públicos | 4 |
| ACISA - Associação Comercial, Industrial de Serviço e Agrícola do Acre | 0 |
| Central de Serviço Público - OCA Rio Branco | 2 |
| SEBRAE/AC - Central do Empreendedor | 0 |
| CEJUSC - Centro Judiciário de Soluções de Conflito | 0 |
| CORINES - Coordenação de Registro e Inspeção Escolar | 0 |
| CBMAC - Corpo de Bombeiros Militar | 0 |
| Correspondente CAIXA | 0 |
| Correspondente MAIS BB | 0 |
| Defensoria Pública do Estado do Acre | 0 |
| Depasa - Departamento Estadual de Pavimentação e Saneamento | 0 |
| PROCON - Departamento Estadual de Proteção e Defesa do Consumidor - Rio Branco | 0 |
| DETRAN - Departamento Estadual de Trânsito | 0 |
| ELETROBRAS - Distribuição Acre | 0 |
| ECT - Empresa Brasileira de Correios e Telégrafos | 0 |
| Fundeseg - Fundo Estadual de Segurança Pública | 0 |
| IAPEN - Instituto de Administração Penitenciária | 0 |
| Instituto de Identificação | 5 |

Instituto de Identificação

Atestado de Antecedentes Criminais

Atestado de Antecedentes Criminais
Novo serviço

Boletim de Ocorrência
Carteira de Identidade
Certidão de Dados
Nada Consta

4. LISTA DE GRUPOS DE UM ÓRGÃO

The screenshot displays the 'Guia OCA' website interface. The top navigation bar includes links for 'Início', 'Órgãos', 'Praças', 'Serviços Online', and 'Blog da OCA'. A sidebar on the left contains a 'Dashboard' link and a menu with categories: 'Grupos', 'Serviços', 'Órgãos', 'Municípios', 'Locais', 'Campos', 'Praças', 'Categorias', 'Tags', and 'Arquivos'. The main content area features a search bar with the placeholder 'Digite sua busca...' and a '+ Novo' button. Below the search bar is a list of services, including 'Achados e Perdidos', 'Atestado de Antecedentes Criminais', 'Boletim de Ocorrência', 'Carteira de Identidade', 'Certidão de Dados', 'Certidão de Teste', 'Emissão/Reemissão de taxas', 'Grupo de Teste #1', 'Nada Consta', 'Taxas e Tributos', and 'Título de Teste'. The 'Carteira de Identidade' service is highlighted, showing a list of options: '1ª via > Brasileiro Nato > Até 15 anos', '1ª via > Brasileiro Nato > A partir de 16 anos', '1ª via > Brasileiro Naturalizado', '1ª via > Brasileiro Nascido no exterior > Até 15 anos', '1ª via > Brasileiro Nascido no exterior > A partir de 16 anos', and '1ª via > Brasileiro casado no exterior'. The interface also includes a date and time display ('8 de março de 2018 às 17:22') and a user profile ('Autor'). At the bottom, there are links for 'Serviços e Informações', 'Utilidades', and 'Portais Governamentais'.

5. LISTA DE SERVIÇOS

Guia OCA Início Órgãos Praças Serviços Online Blog da OCA Admin

Dashboard

Grupos

Serviços

Órgãos

Municípios

Locais

Campos

Praças

Categorias

Tags

Arquivos

1ª via
Achados e Perdidos

1ª via > Brasileiro casado no exterior
Carteira de Identidade

1ª via > Brasileiro Nascido no exterior > A partir de 16 anos
Carteira de Identidade

1ª via > Brasileiro Nascido no exterior > Até 15 anos
Carteira de Identidade

1ª via > Brasileiro Nato > A partir de 16 anos
Carteira de Identidade

1ª via > Brasileiro Nato > Até 15 anos
Carteira de Identidade

1ª via > Brasileiro Naturalizado
Carteira de Identidade

1ª via
Achados e Perdidos

Atestado de Antecedentes Criminais
Atestado de Antecedentes Criminais

Consulta ao Título de Eleitor e local de votação
Taxas e Tributos

Emissão/Remissão de de taxas
Emissão/Remissão de taxas

Novo serviço
Atestado de Antecedentes Criminais

Outro
Achados e Perdidos

Registro
Boletim de Ocorrência

Preencher Editar Excluir

Instituto de Identificação

Carteira de Identidade

1ª via > Brasileiro casado no exterior

Sem descrição

8 de março de 2018 às 17:22 Autor

#similiquehj #pariatuY4 #magniw3 #modiTy #libero9A

6. TELA DE VISUALIZAÇÃO DO CONTEÚDO DO SERVIÇO

The screenshot displays the 'Guia OCA' website interface. The top navigation bar is orange and includes links for 'Início', 'Órgãos', 'Praças', 'Serviços Online', and 'Blog da OCA', along with an 'Entrar' button. The breadcrumb trail indicates the user is in the 'Instituto de Identificação' > 'Carteira de Identidade' > '1ª via > Brasileiro Nato > Até 15 anos' section.

On the left, a sidebar lists three locations: 'Rio Branco' (Central de Serviço Público - OCA Rio Branco), 'Xapuri' (Central de Serviço Público - OCA Xapuri), and 'Senador Guiomard' (Delegacia Geral de Polícia Civil de Senador Guiomard). Each entry has a right-pointing arrow.

The main content area features a 'Carteira de Identidade' header with an 'Imprimir' button. Below this, the title '1ª via > Brasileiro Nato > Até 15 anos' is displayed. The service provider is identified as 'Central de Serviço Público - OCA Rio Branco' with the address 'Rua Quintino Bocaiuva, 299 Centro'. A section titled 'VOU PAGAR PELO SERVIÇO?' contains a placeholder text: 'Iure dignissimos qui et at nostrum. Aliquid corrupti odit vitae voluptatibus ullam. Natus rerum dolorum omnis facilis autem sunt.' The 'Entidade responsável' is listed as 'Instituto de Identificação'.

The footer is divided into three columns: 'Serviços e Informações' (linking to 'Diretoria OCA'), 'Utilidades' (linking to 'Telefones Úteis'), and 'Portais Governamentais' (linking to 'Portal do Governo do Acre').

7. TELA DE INSERÇÃO DE GRUPO

Guia OCA

[Início](#)
[Órgãos](#)
[Praças](#)
[Serviços Online](#)
[Blog da OCA](#)

Admin

Dashboard

Grupos

Serviços

Órgãos

Municípios

Locais

Campos

Praças

Categorias

Tags

Arquivos

Digite sua busca...

Q

+ Novo

Achados e Perdidos

Central de Serviço Público - OCA Rio Branco

Atestado de Antecedentes Criminais

Instituto de Identificação

Boletim de Ocorrência

Instituto de Identificação

Carteira de Identidade

Instituto de Identificação

Certidão de Dados

Instituto de Identificação

Certidão de Teste

Agência Estadual Reguladora de Serviços Públicos

Emissão/Reemissão de taxas

Agência Estadual Reguladora de Serviços Públicos

Grupo de Teste #1

Agência Estadual Reguladora de Serviços Públicos

Nada Consta

Instituto de Identificação

Taxas e Tributos

Central de Serviço Público - OCA Rio Branco

Título de Teste

Agência Estadual Reguladora de Serviços Públicos

Adicionar grupo

Título*

Grupo novo

Órgão*

Associação Comercial, Industrial de Serviço e Agrícola do Acre

Categorias*

Online

Salvar

Cancelar

Limpar

Serviços e Informações

Utilidades

Portais Governamentais

Diretoria OCA

Telefones Úteis

Portal do Governo do Acre

8. TELA DE EDIÇÃO DE SERVIÇO

9. TELA DE PREENCHIMENTO DE CONTEÚDO DO SERVIÇO

The screenshot displays the 'Guia OCA' web application interface. The top navigation bar includes links for 'Início', 'Órgãos', 'Praças', 'Serviços Online', and 'Blog da OCA'. The left sidebar contains a menu with 'Dashboard', 'Grupos', 'Serviços' (highlighted), 'Órgãos', 'Municípios', 'Locais', 'Campos', 'Praças', 'Categorias', 'Tags', and 'Arquivos'. The main content area is titled 'Central de Serviço Público - OCA Rio Branco > Achados e Perdidos > 1 via'. It features a form titled 'Preencha o conteúdo do serviço' with three sections: 'Achados e Perdidos' (containing '1 via' and 'Sem descrição'), 'COMO FAÇO PARA RETIRAR?' (containing a rich text editor with the text 'Ir ao local de solicitação'), and 'QUAL O PRAZO?' (containing a rich text editor with the text '12 dias'). To the right of the form is a blue button 'Atribuir conteúdo aos locais selecionados'. Below this button are two sections: 'Selecionar Campos' with dropdowns for 'COMO FAÇO PARA RETIRAR?' and 'QUAL O PRAZO?', and 'Selecionar Locais' with a dropdown for 'CAC - Sobral' and a list of other locations. A blue box at the bottom right states 'Este serviço já está atribuído aos locais abaixo.' followed by a list of locations. The footer contains three columns: 'Serviços e Informações' (Diretoria OCA), 'Utilidades' (Telefones Úteis), and 'Portais Governamentais' (Portal do Governo do Acre).

Guia OCA Início Órgãos Praças Serviços Online Blog da OCA Admin

Dashboard Central de Serviço Público - OCA Rio Branco > Achados e Perdidos > 1 via

Preencha o conteúdo do serviço

Achados e Perdidos
1 via
Sem descrição

COMO FAÇO PARA RETIRAR?

B I U G " " H₁ H₂ x₁ x₂ A A

Ir ao local de solicitação

QUAL O PRAZO?

B I U G " " H₁ H₂ x₁ x₂ A A

12 dias

Atribuir conteúdo aos locais selecionados

Selecionar Campos

COMO FAÇO PARA RETIRAR? QUAL O PRAZO?

Selecionar Locais

CAC - Sobral
Delegacia Geral de Polícia Civil de Senador
Gulomard

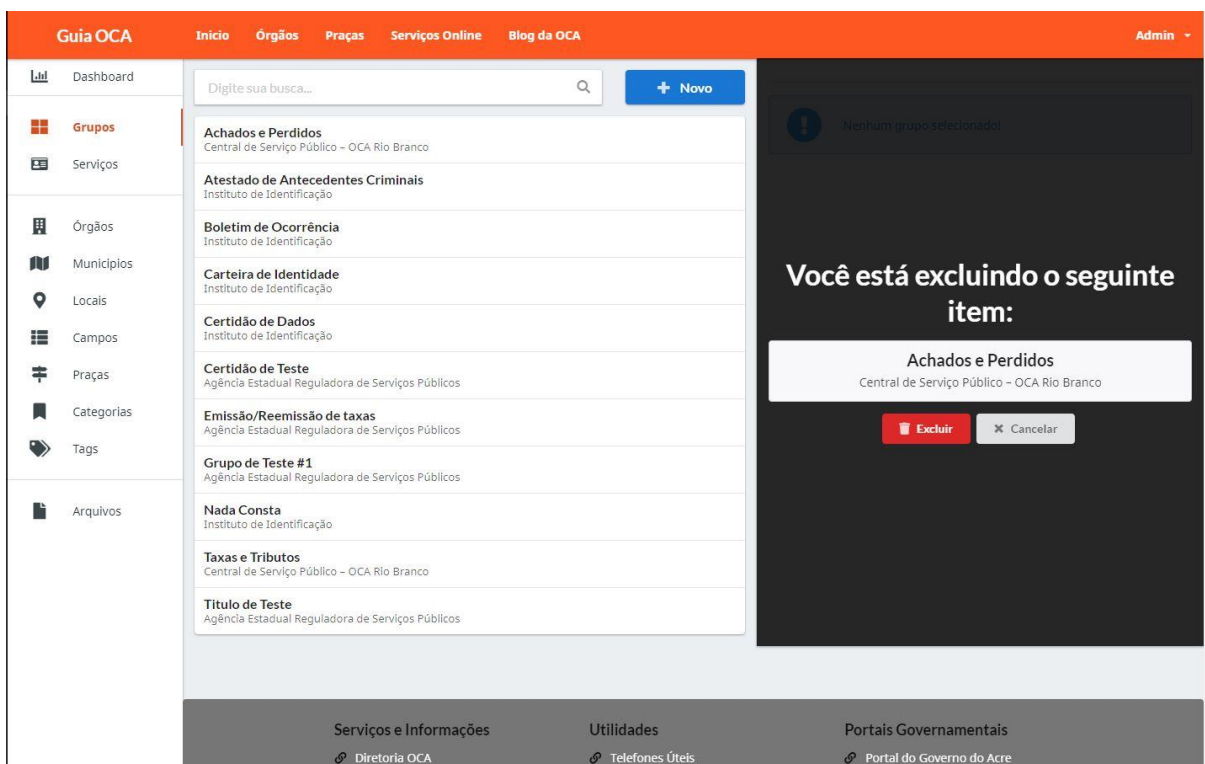
Este serviço já está atribuído aos locais abaixo.

- Central de Serviço Público - OCA Rio Branco
- Central de Serviço Público - OCA Xapuri
- Delegacia de Exemplo

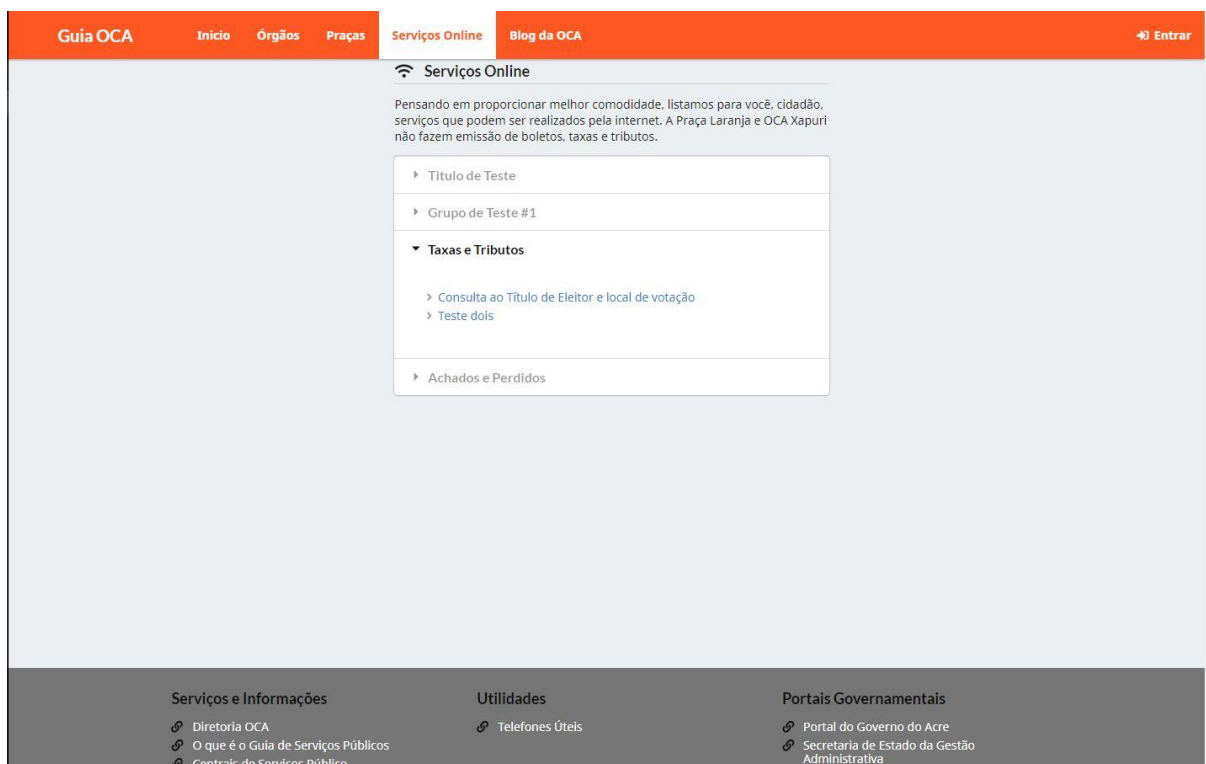
Serviços e Informações Utilidades Portais Governamentais

Diretoria OCA Telefones Úteis Portal do Governo do Acre

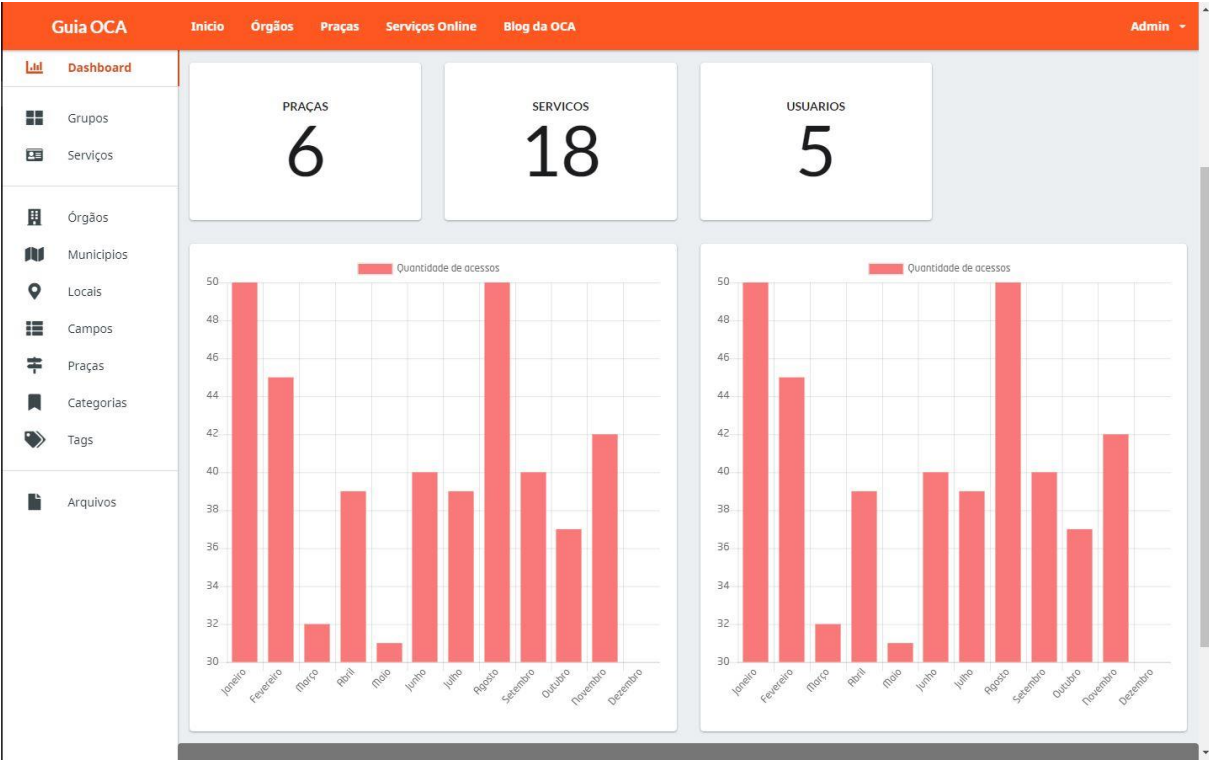
10. TELA DE EXCLUSÃO DE SERVIÇO



11. TELA DE SERVIÇOS ONLINE



12. TELA DE INFORMAÇÕES ESTATÍSTICAS DOS SERVIÇOS



APÊNDICE D – DOCUMENTO DE TESTES

Documentação de Testes

Testadores: Antônio Gabriel Almeida de Souza e Wallison Santos Ferreira

Software em teste: Guia OCA

| | |
|--|---|
| Requisito de Teste: TESTE FUNCIONAL 1.1 | |
| Casos de Testes: RNF01 - Verifique que as <i>endpoints</i> estão retornando <i>Status Code</i> corretamente para os casos de sucesso ou falha nas requisições. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Dados para os casos de sucesso tiveram retorno de cabeçalho com <i>status code</i> 200 e/ou 201 para operações com sucesso e 404 e/ou 500 para os casos de falha. | |

| | |
|--|---|
| Requisito de Teste: TESTE FUNCIONAL 1.2 | |
| Casos de Testes: RNF01 e RNF04 - Verifique que as <i>endpoints</i> estão retornando os dados corretamente para as entidades e suas dependências. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Os dados são retornados em Json para a aplicação, com todas as suas dependências quando requisitadas. | |

| | |
|--|---|
| Requisito de Teste: TESTE FUNCIONAL 1.3 | |
| Casos de Testes: RNF01 e RNF04 - Verifique que o usuário não pode acessar as rotas de inclusão, edição e exclusão das entidades sem autorização. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Alterações no banco de dados são possíveis somente através da autenticação com login de API através do Oauth. Não foi possível o acesso a rotas não autorizadas ao usuário comum. | |

| | |
|---|---|
| Requisito de Teste: TESTE FUNCIONAL 1.4 | |
| Casos de Testes: RNF02 - Verifique que o usuário autenticado tem permissões para realizar inserções, alterações e exclusões de dados, verificando também se a requisição retorna os dados de acordo com as alterações realizadas. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Todas as <i>endpoints</i> retornaram seus dados de acordo com as alterações, não havendo, portanto, problemas de integridade. | |

| | |
|--|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.1 | |
| Casos de Testes: RF03 - Navegue através das páginas de serviços, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado: Critério atendido, todas as páginas estão facilmente compreensíveis | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.2 | |
| Casos de Testes: RF01 - Verifique que o usuário não autorizado não pode acessar a tela de alteração dos conteúdos. Clique em um grupo de serviços e veja se há botões disponíveis para a alteração. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Não há botões para alteração de conteúdos sem prévia autenticação. | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.3 | |
| Casos de Testes: RF01 – Verifique que o usuário autorizado pode realizar o login no sistema. Utilize a o login “ <i>admin</i> ” e senha “ <i>admin</i> ”. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () |

| | |
|---|-----|
| | NÃO |
| Observações/Resultado Atendido, o usuário pode realizar o login. | |

| | |
|---|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.4 | |
| Casos de Testes: RF01, RF04 e RF05 - Verifique que o usuário pode alterar o conteúdo das páginas de serviços. Selecione um grupo, depois selecione um serviço e veja se a página de alteração aparece ao lado do conteúdo. Clique nos botões de alteração de conteúdo, altere o conteúdo e salve. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado O usuário pode realizar alterações no conteúdo dos serviços. | |

| | |
|--|---|
| Requisito de Teste: TESTE DE INTERFACE DE USUARIO 2.5 | |
| Casos de Testes: RF02 – Verifique que ao adicionar, alterar ou excluir um conteúdo de serviço há o feedback apropriado. | |
| Factível? (X) SIM () NÃO | Critério de finalização atendido? (X) SIM () NÃO |
| Observações/Resultado Operações de adicionar, alterar e excluir retornam um feedback na tela indicando que foi realizada com sucesso. | |