

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LEONARDO DA ROCHA FERRARI

**PROTÓTIPO DE SISTEMA WEB PARA AUXILIAR NO GERENCIAMENTO DE
PROCESSOS NA CONSTRUÇÃO CIVIL**

**RIO DO SUL
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LEONARDO DA ROCHA FERRARI

**PROTÓTIPO DE SISTEMA WEB PARA AUXILIAR NO GERENCIAMENTO DE
PROCESSOS NA CONSTRUÇÃO CIVIL**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí, como condição parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Orientador: M.e Jullian Hermann Creutzberg

**RIO DO SUL
2022**

**CENTRO UNIVERSITÁRIO PARA O DESENVOLVIMENTO DO ALTO VALE DO
ITAJAÍ - UNIDAVI**

LEONARDO DA ROCHA FERRARI

**PROTÓTIPO DE SISTEMA WEB PARA AUXILIAR NO GERENCIAMENTO DE
PROCESSOS NA CONSTRUÇÃO CIVIL**

Trabalho de Conclusão de Curso a ser apresentado ao curso de Sistemas da Informação, da Área das Ciências Naturais, da Computação e das Engenharias, do Centro Universitário para o Desenvolvimento do Alto Vale do Itajaí- UNIDAVI, a ser apreciado pela Banca Examinadora, formada por:

Professor Orientador: M.e Jullian Hermann Creutzberg

Banca Examinadora:

Prof. M.e Fernando Andrade Bastos

Prof. Cleber Nardelli

Rio do Sul, 29 de novembro de 2022.

RESUMO

Mesmo com os amplos avanços tecnológicos e a presença constante da tecnologia em nosso cotidiano, na área da construção civil percebe-se a existência de poucos sistemas especializados que contribuem na execução das tarefas dos arquitetos, em especial o levantamento arquitetônico. Esta atividade consiste em levantar medidas de uma área, sendo que para facilitar e agilizar esse processo recomenda-se que esta etapa de levantamento seja feita através de ferramentas web, facilitando o entendimento das medições na hora visualizá-las. Esse trabalho tem como objetivo principal desenvolver um protótipo de aplicação web responsiva para auxiliar o gerenciamento de processos na área da construção civil, tendo como foco principal o levantamento arquitetônico, tarefa realizada para obter medidas de um determinado espaço. Quanto a metodologia, se caracteriza como uma pesquisa aplicada e descritiva. Para que os objetivos fossem atendidos foram realizadas pesquisas em relação a tecnologias e ferramentas que poderiam ser utilizadas, as quais foram descritas no capítulo da revisão da literatura. Foram também identificadas ferramentas com propósitos similares a protótipo, sendo estas apresentadas na seção do estado da arte. Antes do desenvolvimento do protótipo foi realizada uma análise do que seria necessário, contendo o levantamento de requisitos e a elaboração dos diagramas. Na seção de implementação são apresentados todos os detalhes sobre o desenvolvimento e funcionamento do protótipo. Com a utilização do protótipo, os custos de um determinado projeto podem ser reduzidos, pois será possível que parte das tarefas do projeto sejam realizadas por outro arquiteto, fazendo com que o mesmo fique responsável pelo levantamento arquitetônico. Desta forma, o profissional não precisará se deslocar até o local para fazer o levantamento, diminuindo o custo de deslocamento. Outro benefício é que o arquiteto poderá realizar outras atividades enquanto o profissional terceirizado realiza o levantamento arquitetônico.

Palavras-Chave: levantamento arquitetônico, desenvolvimento web, sistemas de informação.

ABSTRACT

Even with the extensive technological advances and the constant presence of technology in our daily lives, in the area of civil construction it is noticeable that there are few specialized systems that contribute to the execution of the architect's tasks, especially the architectural survey. This activity consists of surveying the measurements of an area, and to facilitate and speed up this process it is recommended that this survey stage be done through web tools, making it easier to understand the measurements when viewing them. The main objective of this work is to develop a prototype of a responsive web application to assist in the management of processes in the construction industry, with the main focus being the architectural survey, a task performed to obtain measurements of a given space. As for the methodology, it is characterized as an applied and descriptive research. For the objectives to be met, research was conducted regarding the technologies and tools that could be used, which were described in the literature review chapter. Tools with similar purposes to prototyping were also identified, and these are presented in the state-of-the-art section. Before the development of the prototype, an analysis of what would be needed was performed, containing the requirements survey and the elaboration of the diagrams. In the implementation section all the details about the development and operation of the prototype are presented. With the use of the prototype, the costs of a certain project can be reduced, because it will be possible for part of the project tasks to be performed by another architect, making him/her responsible for the architectural survey. This way, the professional will not need to travel to the site to do the survey, reducing the cost of travel. Another benefit is that the architect can perform other activities while the outsourced professional does the architectural survey.

Keywords: architectural survey, web development, information systems.

LISTA DE FIGURAS

Figura 1 – Modelo MVC	17
Figura 2 – Modelo MVC baseado no navegador.....	17
Figura 3 - Estrutura básica HTML	24
Figura 4 – Regras CSS	25
Figura 5 – CSS <i>in-line</i>	26
Figura 6 - CSS incorporado.....	26
Figura 7 - CSS em arquivo externo	27
Figura 8 – Link do arquivo CSS externo com HTML.....	28
Figura 9 - Software Asana - WEB.....	35
Figura 10 – Aplicativo Asana - iOS.	36
Figura 11 – Software FlowUp – Web.....	37
Figura 12 – Aplicativo FlowUp - iOS.	38
Figura 13 – Listagem de Projetos – Vobi.....	39
Figura 14 - Cadastro de projetos com checklist - Vobi.....	39
Figura 15 – Diagrama de Caso de Uso	48
Figura 16 – Diagrama de Atividade – Configuração do Perfil.....	49
Figura 17 – Diagrama de Atividade - Aceitação e Execução de Projetos	50
Figura 18 – Modelagem Física - Banco de Dados.....	52
Figura 19 – Diagrama de Classe.....	53
Figura 20 – Cadastro de Usuário (RF 01)	57
Figura 21 - Login (RF 02)	58
Figura 22 – Tela Inicial - Dashboard (RF 05)	59
Figura 23 – Alteração do Cadastro de Usuário (RF 01)	60
Figura 24 – Cadastro área de Atuação (RF 06)	61
Figura 25 – Consulta área de atuação (RF 06)	62
Figura 26 – Cadastro de Checklist (RF 07)	63
Figura 27 – Consulta de Checklist (RF 07)	63
Figura 28 – Consulta de Atividades da Checklist (RF 08)	64
Figura 29 – Cadastro de Atividades da Checklist (RF 08).....	64
Figura 30 – Cadastro de Projeto (RF 09).....	65
Figura 31 – Consulta de Projeto - Aguardando minha aceitação (RF 10).....	66
Figura 32 – Consulta de Projeto - Recusado (RF 10).....	67

Figura 33 – Consulta de Projeto - Em andamento (RF 10)	67
Figura 34 – Consulta de Cômodos (RF 13).....	68
Figura 35 – Cadastro de Cômodos (RF 12).....	68
Figura 36 – Consulta Área de Medição (RF 19)	69
Figura 37 – Cadastro Área de Medição (RF 19)	69
Figura 38 – Cadastro de Medidas (RF 15)	70
Figura 39 – Consulta de Medidas (RF 16)	71
Figura 40 – Tela de Conclusão das Atividades (RF 11).....	71
Figura 41 – Tela de Conclusão das Atividades (RF 11).....	72
Figura 42 – Informações do Profissional (RF 17)	72
Figura 43 – Informações do Cadastro de Projeto (RF 09).....	73
Figura 44 – Consulta de Arquivos do Projeto (RF 18).....	74
Figura 45 – Cadastro de Arquivos do Projeto (RF 18).....	74
Figura 46 – Consulta de Atuação por Região (RF 14)	75
Figura 47 – Tela iPhone 12 Pro – Tela Responsiva	76

LISTA DE QUADROS

Quadro 1 – Padrão modelo-visão-controlado (MVC)	16
Quadro 2 – Propriedades Implícita dos Bancos de Dados	20
Quadro 3 – Modelagem de Dados	22
Quadro 4 – Limites de Valores (PostgreSQL)	23
Quadro 5 – Propriedades de Configuração - Ajax.....	33
Quadro 6 – Recursos dos sistemas apresentados.....	42
Quadro 7 – Requisitos Funcionais.....	43
Quadro 8 – Requisitos Funcionais Opcionais	45
Quadro 9 – Requisitos Não Funcionais	46
Quadro 10 – Regras de Negócio.....	46

SUMÁRIO

1. INTRODUÇÃO	10
1.1 PROBLEMA DE PESQUISA	11
1.2 OBJETIVOS	11
1.2.1 Geral	11
1.2.2 Específicos	11
1.3 JUSTIFICATIVA	12
2. REVISÃO DA LITERATURA	13
2.1 ENGENHARIA DE SOFTWARE	13
2.1.1 Levantamento de Requisitos.....	13
2.1.2 Projeto de Arquitetura.....	15
2.1.3 Modelagem de Sistemas	18
2.1.4 Prototipação	19
2.2 BANCO DE DADOS	20
2.2.1 Sistema de Gerenciamento de Banco de Dados (SGBD).....	20
2.2.2 Modelagem de Dados	22
2.2.3 PostgreSQL	22
2.3 HTML.....	23
2.4 CSS	24
2.4.1 Bootstrap	28
2.5 PHP.....	28
2.5.1 Laravel.....	30
2.6 JAVASCRIPT	31
2.6.1 jQuery	32
2.6.2 Ajax.....	32
2.6.3 SweetAlert	33
3. METODOLOGIA DA PESQUISA	34
3.1 ESTADO DA ARTE	34
3.1.1 Asana	35
3.1.2 FlowUp.....	37
3.1.3 Vobi.....	38
4. PROTÓTIPO DE SISTEMA WEB PARA AUXILIAR NO GERENCIAMENTO DE PROCESSOS NA CONSTRUÇÃO CIVIL.....	41
4.1 ANÁLISE	41

4.1.1 Visão Geral do Protótipo	41
4.1.2 Comparação do Protótipo com o Estado da Arte.....	42
4.1.3 Requisitos	42
4.1.4 Diagramas	47
4.2 IMPLEMENTAÇÃO	54
4.2.1 Ferramentas e Técnicas Utilizadas	54
4.2.2 Utilização e Funcionamento.....	56
5. CONSIDERAÇÕES FINAIS.....	77
5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS	78
REFERÊNCIAS	80

1. INTRODUÇÃO

A tecnologia da informação está cada vez mais presente em nosso o dia a dia. Os sistemas criados contribuem para que processos fiquem menos burocráticos e complexos. Na área da arquitetura encontra-se poucos sistemas que possam contribuir com as atividades dos arquitetos. A maioria dos sistemas criados são para gerenciar os projetos, focando totalmente na agenda do arquiteto. Contudo, há uma deficiência em relação a automatização de tarefas que precisam ser executadas.

Dentre as tarefas a serem concluídas, o levantamento arquitetônico destaca-se como a principal tarefa. Para executar esta tarefa o arquiteto precisa se deslocar até a obra e realizar anotações das medidas de todos os cômodos da construção. Às vezes esse deslocamento do arquiteto torna-se inviável, devido o distanciamento, encarecendo o projeto de seus clientes.

Hoje em dia é muito comum nos depararmos com aplicações que colaboram para uma melhor gestão de um determinado negócio. Essas aplicações facilitam muito a vida de um profissional, permitindo que processos sejam otimizados e menos suscetíveis a erros. Outra grande vantagem é o controle das atividades, podendo ser verificado o andamento de uma determinada tarefa. Diante disso, surgiu a ideia de propor o desenvolvimento de um protótipo para auxiliar nos processos da construção civil, especificamente no levantamento arquitetônico.

O protótipo da aplicação web tem como público-alvo os arquitetos, sendo um dos seus objetivos centralizá-los através da ferramenta, possibilitando que os mesmos incluam seus projetos, ou até mesmo que participem de algum projeto, através da terceirização. A terceirização será opcional, o arquiteto poderá optar pela utilização deste recurso. Para optar por este recurso, o arquiteto deverá configurar sua área de atuação, podendo escolher diversos municípios no qual irá atuar.

A utilização de tecnologias da informação é uma alternativa para reduzir custos de um determinado trabalho. Desta forma, ao terceirizar o projeto faz com que o arquiteto não precise se deslocar até a obra, diminuindo o valor gasto com deslocamentos. Outra vantagem da terceirização, é que o arquiteto poderá utilizar este tempo livre realizando outra atividade, enquanto outro projeto está sendo executado por outro profissional.

Para elaboração deste trabalho, além desta seção introdutória, é apresentado um segundo capítulo com a revisão da literatura, onde são abordadas as tecnologias e ferramentas utilizadas para a construção do protótipo. Além disso, no terceiro capítulo são apresentadas informações em relação a metodologia da pesquisa, onde é explicado os detalhes e processos metodológicos

deste protótipo, e também, aplicações similares ao que foi proposto, esclarecido no estado da arte. No quarto capítulo é realizada a análise do protótipo, onde foram listados os requisitos levantados, bem como onde são apresentados os diagramas construídos para facilitar a compreensão e a construção do protótipo. Este quarto capítulo também dispõe de um detalhamento das tecnologias utilizadas e uma breve explicação sobre a utilização e funcionamento do protótipo, apresentando cada rotina implementada no protótipo. No último capítulo são apresentadas as considerações finais e também sugestões de desenvolvimento futuro.

1.1 PROBLEMA DE PESQUISA

Como facilitar para os arquitetos a gestão dos processos de engenharia civil, em especial o levantamento arquitetônico, utilizando tecnologias da informação?

1.2 OBJETIVOS

1.2.1 Geral

- Desenvolver protótipo de aplicação web responsiva para auxiliar no gerenciamento de processos na área da construção civil, tendo como foco principal o levantamento arquitetônico.

1.2.2 Específicos

- Descrever as tecnologias que serão utilizadas no desenvolvimento do protótipo;
- Detalhar a análise do protótipo por meio do levantamento de requisitos, da elaboração de diagramas, e da modelagem do banco de dados;
- Demonstrar ferramentas que possuem intuito similar ao protótipo proposto;
- Construir o protótipo através da aplicação das tecnologias selecionadas para o projeto;

1.3 JUSTIFICATIVA

A procura e a necessidade de obras de construção civil ocorrem constantemente em nosso dia a dia. É comum nos depararmos com pessoas que necessitam desses profissionais, sejam eles: engenheiros, arquitetos e designer de interiores, onde os mesmos possuem reconhecimento profissional e entidades de representatividade, tornando os mesmos responsáveis pelas atividades que serão exercidas.

Ao fechar o projeto com algum cliente surge a necessidade de realizar as medições ou até mesmo calcular a quantidade de materiais que serão necessários para concluir a obra. Na área da arquitetura é muito comum os arquitetos fecharem projetos com clientes que residem em outros municípios, o que dificulta um pouco o processo e o acompanhamento do andamento da obra, além disso, o deslocamento até outro município encarece o projeto. Dependendo da distância e do número de viagens, às vezes torna-se inviável o cliente arcar com esses gastos, e até mesmo para o arquiteto, o qual poderá ocupar o tempo para execução de outra tarefa. Segundo a Cronoshare (2022) o gasto em relação a terceirização pode variar de acordo com o tipo de imóvel. O custo é baseado na quantidade de metro quadrado (m²), sendo que o m² pode variar em média de R\$ 5,00 à R\$ 15,00.

Diante disso, o protótipo aqui proposto tem como objetivo permitir a relação entre os arquitetos, sendo eles de quaisquer municípios ou estados. Com esse relacionamento, pode-se tornar mais atrativo que o arquiteto atenda clientes em outras cidades ou estados, desta forma, o profissional poderá localizar outros profissionais que residem na mesma cidade deste cliente.

Sendo assim, o arquiteto pode terceirizar esse serviço com outro profissional capacitado, reduzindo os valores que seriam acrescentados no projeto e até mesmo ocupar o tempo para a execução de outras tarefas. Por fim, o protótipo terá o intuito de otimizar processos realizados pelo profissional com foco no levantamento arquitetônico, que envolve especialmente obter as medidas de um determinado espaço.

2. REVISÃO DA LITERATURA

Neste capítulo são apresentados os conceitos, ferramentas e linguagens utilizados para o desenvolvimento do protótipo.

2.1 ENGENHARIA DE SOFTWARE

A engenharia de software envolve em especial o planejamento, no que diz respeito às pessoas interessadas no software e ao cronograma de trabalho. Por meio da aplicação das técnicas e ferramentas de engenharia de software, pode-se atribuir tarefas à estas pessoas e também mensurar o tempo necessário para o desenvolvimento do projeto, facilitando a vida do gestor. Além disso, nesta temática também abordados os critérios de qualidade, buscando atender aquilo que o cliente solicitou, de acordo com o que foi definido em reuniões com o profissional que será responsável pelo desenvolvimento. (MORAIS, 2020).

Segundo Morais (2020), a engenharia de software está em todos os aspectos ao realizar o desenvolvimento do software, desde o estágio inicial (especificação do sistema), até sua manutenção. Seguindo esse contexto, a engenharia de software se preocupa em estudar e monitorar o andamento das atividades relacionadas ao projeto.

2.1.1 Levantamento de Requisitos

“O levantamento de requisitos (também chamado de elicitação de requisitos) combina elementos de solução de problemas, elaboração, negociação e especificação.”. (PRESSMAN, 2016, p.110). Nesta fase da Engenharia de Software os envolvidos no projeto trabalham juntos para encontrar problemas, discutir sobre as soluções encontradas para o problema e até mesmo negociar. (PRESSMAN, 2016).

2.1.1.1 Requisitos Não Funcionais

Segundo Paula Filho (2019) os requisitos não funcionais incluem requisitos de desempenhos e outros pontos de qualidade relacionados ao produto. É incluído também os requisitos lógicos sobre os dados e os requisitos de natureza técnica. Estes requisitos devem ser descritos de forma precisa e quantitativa. A maioria dos requisitos não funcionais são globais, sendo aplicados no projeto como um todo, não descartando a possibilidade do mesmo ser exclusivo para uma própria rotina.

Um requisito não funcional (NFR, *nonfunctional requirement*) pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como uma restrição geral em um sistema. Frequentemente, os envolvidos têm dificuldade de articulá-los. (PRESSMAN, 2016, p. 141).

A elaboração correta de um requisito não funcional é essencial para o sucesso de um software, pois costumam ter grande impacto, e capazes de serem perceptíveis aos olhos do usuário. Portanto, é um requisito que garante a qualidade e utilidade de um determinado produto. (PAULA FILHO, 2019).

Sommerville (2011, p. 59) complementa ainda que “São restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas.”.

2.1.1.2 Requisitos Funcionais

O requisito funcional é uma funcionalidade solicitada pelo *stakeholders* (aqueles que irão usufruir da aplicação) para atender o objetivo do negócio. O requisito funcional a ser implementado irá permitir que os usuários possam executar as atividades necessárias. Normalmente, os requisitos funcionais são descritos utilizando o trecho “o sistema deve”. Por exemplo: O sistema deve permitir que usuário pague com cartão de débito ou crédito. (REINEHR, 2020).

Requisitos funcionais (RF ou, em inglês, FR, de *functional requirement*): “São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns

casos, também podem explicitar o que o sistema não deve fazer” (SOMMERVILLE, 2011, p. 59).

2.1.1.3 Regras de Negócio

As regras de negócio definem ou restringem aspectos dos processos de negócio. Essas regras devem ser levantadas durante a modelagem de negócio. Durante a etapa de levantamento dos requisitos, a regra de negócio merece uma atenção especial, pois estão muito ligadas ao trabalho dos usuários, onde os mesmos esquecem de enunciá-los explicitamente. Muitas vezes, às regras de negócio são lembradas depois que os usuários recebem feedback através de um protótipo ou até mesmo da liberação de uma parte do sistema. (PAULA FILHO, 2019).

Uma regra de negócio pode ser definida como “Uma política, um guia, um padrão, uma regulamentação que define ou restringe algum aspecto de negócio. Não é um requisito do Software em si, mas a origem de diversos tipos de requisito de software.”. (REINEHR, 2020, pg. 36).

2.1.2 Projeto de Arquitetura

Segundo Pressman (2016) é um processo em várias etapas onde as características da interface, estruturação do programa, representação de dados e detalhes procedurais são combinados com base no requisito da informação.

O projeto de arquitetura tem como objetivo representar toda uma estruturação de componentes e dados que são necessários para construir qualquer sistema computacional. A partir desse projeto é considerado o estilo da arquitetura que será utilizada no sistema, bem como as relações internas que ocorrem os componentes da arquitetura de um sistema. (PRESSMAN, 2016).

Você pode pensar em um padrão de arquitetura como uma descrição abstrata, estilizada, de boas práticas experimentadas e testadas em diferentes sistemas e ambientes. Assim, um padrão de arquitetura deve descrever uma organização de sistema bem-sucedida em sistemas anteriores. (SOMMERVILLE, 2011, p. 108).

2.1.2.1 Arquitetura MVC

A arquitetura MVC (modelo-visão-controlador) é bastante utilizada em aplicações web. Nessa arquitetura, o comando do usuário é enviado da janela do navegador para o responsável que irá processar os comandos, neste caso o controlador, que gerencia o acesso ao conteúdo (modelo) e instrui o modelo de renderização das informações (visão) que será totalmente transformado para que as informações sejam exibidas através do software do navegador. (PRESSMAN, 2016).

De acordo com Sommerville (2011) o padrão MVC (Model-View-Controller), é a base do gerenciamento de muitos sistemas web. No Quadro 1 é possível verificar as descrições, vantagens e desvantagens deste padrão.

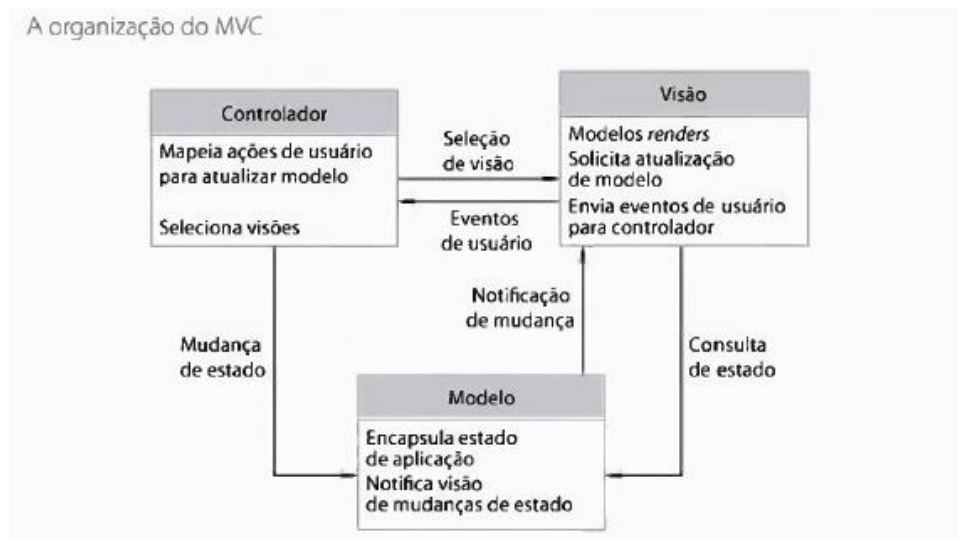
Quadro 1 – Padrão modelo-visão-controlado (MVC)

Descrição	Separa a apresentação e a interação dos dados do sistema. O sistema é estruturado em três componentes lógicos que interagem entre si. O componente Modelo gerencia o sistema de dados e as operações associadas a esses dados.
Quando é usado	É usado quando existem várias maneiras de se visualizar e interagir com dados. Também quando são desconhecidos os futuros requisitos de interação e apresentação de dados.
Vantagens	Permite que os dados sejam alterados de forma independente de sua representação, e vice-versa. Apoia a apresentação dos mesmos dados de maneiras diferentes, com as alterações feitas em uma representação aparecendo em todas elas.
Desvantagens	Quando o modelo de dados e as interações são simples, pode envolver código adicional e complexidade de código.

Fonte: Elaborado a partir de Sommerville (2011, p. 109).

A Figura 1 representa a estruturação do modelo MVC. O componente Visão define a gerência como os dados são apresentados ao usuário. O componente Controlador gerencia a interação do usuário (por exemplo, teclas, cliques do mouse e etc) e passa interações para a Visão e o Modelo. (SOMMERVILLE, 2011, p. 109).

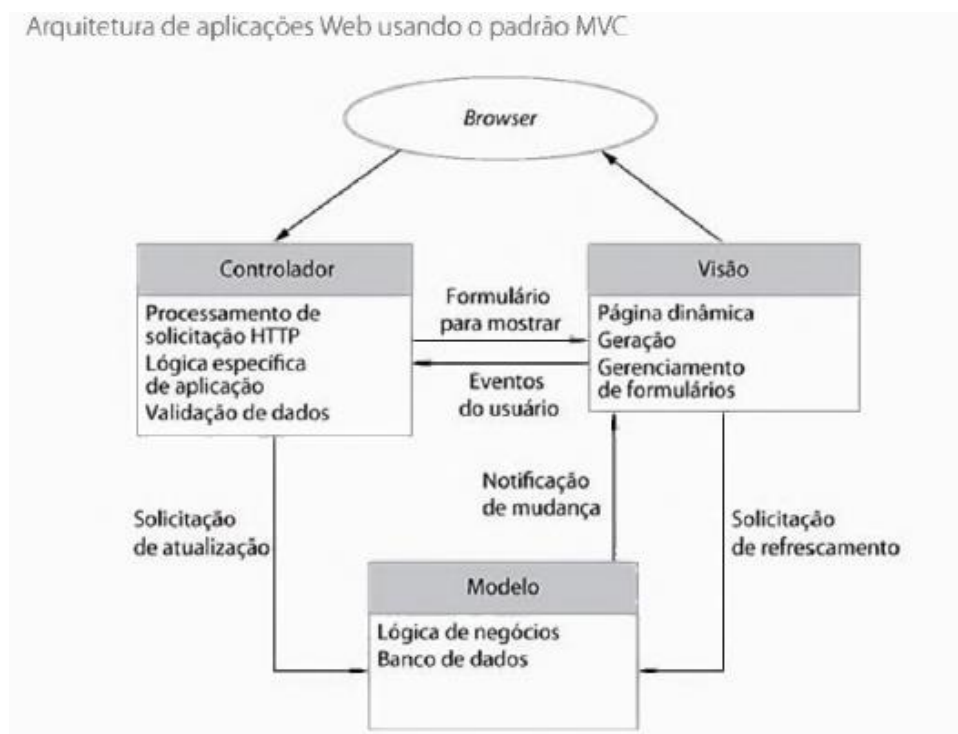
Figura 1 – Modelo MVC



Fonte: Elaborado a partir de Sommerville (2011, p. 109).

A Figura 2 mostra a arquitetura de um sistema aplicativo baseado na Internet, organizada pelo uso do padrão MVC.

Figura 2 – Modelo MVC baseado no navegador



Fonte: Sommerville (2011, p. 109).

2.1.3 Modelagem de Sistemas

De acordo com Vazquez e Simões (2016) os requisitos avulsos ou isolados não costumam ser complexos. Porém, a complexidade se dará devido a interdependência entre os requisitos. A modelagem é uma representação visual e descritiva para repassar informações, facilitando o entendimento e a comunicação. Desta forma, o modelo tem como objetivo representar informações de maneira mais organizada e estruturada, permitindo transferir informações de uma forma mais amigável.

Com a utilização da modelagem fica mais fácil representar ou entender a realidade dentro do negócio. Permite que a informação seja refinada, sendo vista de diversas formas e perspectivas. A modelagem permite deferir o conhecimento desenvolvido, identificar a falta de informação e eliminar aquelas informações que são redundantes. (VAZQUEZ; SIMÕES 2016).

Segundo Sommerville (2018), a modelagem de sistemas é um processo de desenvolvimento que representa um modelo abstrato de um sistema. Essa representação ocorre através da notação gráfica, baseadas em diagramas feitos em UML (*Unified Modeling Language*). O modelo não é uma representação completa de um sistema, sendo o propósito dele facilitar a compreensão do negócio. Essa representação deve manter as informações a respeito da entidade que está sendo representada.

Vazquez e Simões (2016) comentam ainda que há diversas formas de representar o negócio através de modelos. O tipo de modelagem a ser escolhido irá variar de acordo com a informação que precisa ser modelada e o objetivo do negócio.

Quanto ao tipo de modelagem podemos classificar as Matrizes e Diagramas. A matriz é considerada um tipo de modelagem, sendo utilizada para especificar um conjunto de requisito, oferecendo uma estrutura mais complexa, priorizando os requisitos a serem desenvolvidos. (VAZQUEZ, 2016).

Já os diagramas oferecem uma representação visual, contendo componentes e figuras que representam os requisitos do negócio. Permite que um requisito seja descrito através de imagem, diminuindo a complexidade. Os diagramas podem ser utilizados para delimitar fronteiras, criar heranças e classificar itens em categorias. Além disto, permite apresentar a estrutura do negócio e seus relacionamentos. (VAZQUEZ, 2016).

Um diagrama retrata os principais elementos de um fluxo de processo, mas omite detalhes menores de entendimento dos fluxos de trabalho. Uma analogia pode ser feita

com um diagrama simples que pode ser utilizado para demonstrar a rota até um local de armazenagem; ele pode retratar coisas como marcos geográficos e distâncias de forma simplificada ou exagerada, mas ainda assim serve para ajudar a encontrar o armazém. De maneira similar, um diagrama de processo ajuda rapidamente a identificar e entender as principais atividades do processo. (VAZQUEZ, 2016, p. 317).

Os diagramas de fluxo de dados, também conhecido como modelo dirigido por dados representam uma sequência de ações envolvidas em um processo, desde a entrada do processamento até a saída, representando o processo de ponta a ponta. Esse modelo foi um dos primeiros a serem utilizados ao representar modelos de software. Esse modelo de fluxo de dados é útil para que os analistas e projetistas possam acompanhar e compreender o que está acontecendo no sistema, permitindo assim, realizar as documentações necessárias. (SOMMERVILLE, 2018).

2.1.4 Prototipação

De acordo com Sommerville (2018), o protótipo é uma versão inicial de um sistema computacional. Pode ser utilizado para demonstrar conceitos, compartilhar experiências e descobrir mais sobre os problemas e conseqüentemente as soluções. O rápido desenvolvimento de forma iterativa de um protótipo é essencial para o controle de custos e também para que os stakeholders fiquem familiarizados com o software.

A prototipação é uma técnica que permite simular e mostrar ao usuário o funcionamento dos requisitos antes do produto final ser finalizado. Pode ser considerado um processo iterativo, onde irá ocorrer diversas versões. Através da prototipação é possível avaliar se o que foi definido na etapa de requisitos está sendo atendido, até mesmo descobrir novos requisitos. Além disso, permite melhorar a experiência do usuário, dando oportunidade para que o mesmo avalie o design. (VAZQUEZ; SIMÕES, 2016).

Os protótipos de um sistema permitem que *stakeholders* vejam se o sistema dá suporte ao seu trabalho. Através do protótipo podem surgir novas ideias e encontrar pontos fracos e fortes do sistema, além de propor novas funcionalidades. Enquanto um sistema está em projeto, o protótipo pode ser utilizado como experimento, verificando a viabilidade proposta. Devido à natureza dinâmica, a descrição de requisitos e diagramas não são suficientes para expressar os requisitos definidos. Portanto, a prototipação envolvendo o usuário final da aplicação é uma das

formas mais efetivas para desenvolver interfaces gráficas de forma rápida. (SOMMERVILLE, 2018).

2.2 BANCO DE DADOS

O banco de dados e sua tecnologia possui um impacto importante na área da computação. Os bancos de dados desempenham um papel crítico em todas as áreas que são utilizados os computadores, sendo no comércio tradicional, eletrônicos, mídias sociais, engenharia, medicina, direito e educação. (ELMASRI; NAVATHE, 2018).

O banco de dados é um conjunto de dados relacionados, representando uma parte do mundo real, contendo uma estrutura lógica com significados existente e comumente possui uma finalidade e usuários autorizados. (CAIUT, 2015).

De acordo com Elmasri e Navathe (2018) o banco de dados possui três propriedades implícitas, como apresentadas no Quadro 2.

Quadro 2 – Propriedades Implícita dos Bancos de Dados

Primeira	A representatividade do mundo real, conhecido como minimundo ou de universo de discurso, onde tudo que acontece reflete dentro do banco de dados;
Segunda	Possui uma coleção lógica de dados, contendo significados inerentes.
Terceira	Um banco de dados é projetado, montado e preenchido com informações de uma finalidade específica. O mesmo pertence a alguns usuários e pertence a uma aplicação no qual esses usuários estão interessados.

Fonte: Elaborado a partir de Elmasri e Navathe (2018, p.4 e 5).

Segundo Elmasri e Navathe (2018) um banco de dados pode ter qualquer tamanho e complexidade. Como exemplo podemos citar o Facebook, que possui mais de um bilhão de usuários. Neste exemplo citado o banco de dados é responsável por manter as informações sobre quais usuários estão relacionados a outros como amigos, as postagens de cada usuário, quais usuários terão acessos a estas postagens e outros inúmeros tipos de informações.

2.2.1 Sistema de Gerenciamento de Banco de Dados (SGBD)

Um sistema de gerenciamento de banco de dados (SGBD) é um sistema computadorizado que permite que usuários criem e manipulem um banco de dados. Pode ser

considerado um software de uso geral que facilita o processo de definição, manipulação e compartilhamento entre os usuários e aplicações. Ao criar um banco de dados é necessário especificar os tipos de dados e verificar as restrições de dados a serem armazenados. A manipulação de um banco de dados inclui funções como consulta, onde são recuperados dados específicos, realização de alterações e geração de relatórios com os dados que foram recuperados. (ELMASRI; NAVATHE, 2018).

De acordo com Elmasri e Navathe (2018) um software acessa o banco de dados enviando requisições para o SGBD. Uma consulta faz com que os dados solicitados sejam recuperados e uma transação faz com que os dados sejam lidos e até mesmo inseridos no banco de dados.

2.2.1.1 Linguagens do SGBD

Quando o projeto do banco de dados é concluído e um SGBD é escolhido, o primeiro passo é especificar esquemas conceituais e internos, até mesmo mapeamentos internos entre os bancos de dados. Os SGBDs possuem uma linguagem conhecida como DDL (Data Definition Language) Linguagem para definição de dados, que é utilizada pelo DBA (administrador de banco de dados) ou pelo projetista do banco de dados para a definição dos esquemas. O SGBD terá um compilador para processar essas instruções que serão armazenadas no catálogo do SGBD. (ELMASRI; NAVATHE, 2018).

Nos SGBDs que mantêm uma separação clara entre os níveis conceitual e interno, a DDL é usada para especificar apenas o esquema conceitual. Outra linguagem, a linguagem de definição de armazenamento (SDL - Storage Definition Language), é utilizada para especificar o esquema interno. Os mapeamentos entre os dois esquemas podem ser especificados em qualquer uma dessas linguagens. (ELMASRI; NAVATHE, 2018, p. 36).

De acordo com Elmasri e Navathe (2018) assim que os esquemas são compilados o banco de dados é alimentado com dados, isso ocorre através dos usuários que estarão utilizando a aplicação, sendo possível que eles façam manipulações. Dentre essas manipulações podemos citar a recuperação, inserção, exclusão e alteração dos dados. Esse conjunto de operações é

oferecido pelo SGBD e é chamada de linguagem de manipulação de dados (DML – *Data Manipulation Language*).

Para Caiut (2015) o SGBD é um pacote de software cuja finalidade é gerenciar o banco de dados, facilitando o processo de definição, construção e manipulação dos dados pertinentes ao banco.

2.2.2 Modelagem de Dados

De acordo com Elmasri e Navathe (2018) a modelagem é considerada uma fase muito importante no projeto de banco de dados, garantindo um projeto bem-sucedido. Nesta fase, existem alguns modelos que podem ser utilizados, todos eles seguindo o conceito de modelagem do modelo Entidade-Relacionamento (ER), que é popularmente conhecido. O conceito de modelagem de dados pode ser visto no Quadro 3 a seguir.

Quadro 3 – Modelagem de Dados

Modelo Conceitual	O modelo conceitual é criado após os requisitos terem sido levantados e analisados. O esquema conceitual é uma descrição resumida dos requisitos de dados relacionados ao usuário e inclui detalhes da entidade, relacionamentos e restrições. Esse modelo não possui descrições detalhadas e pode ser utilizada para comunicação com usuários não técnicos. Essa modelagem permite que os projetistas especifiquem as propriedades dos dados, sem se preocupar com o armazenamento e implementação.
Modelo Lógico	Nesta fase ocorre a implementação real do banco de dados, utilizando qualquer SGBD comercial. A maioria desses SGBDs utilizam modelos de dados de implantação - como o modelo relacional (SQL). Essa etapa pode ser chamada de projeto lógico. O resultado é um esquema de banco de dados no modelo de dados da implementação do SGBD.
Modelo Físico	O modelo físico é considerado a última etapa. Nessa etapa ocorre a especificação de índices, organizações de arquivos, caminhos de acesso e parâmetros físicos do projeto são especificados.

Fonte: Elaborado a partir de Elmasri e Navathe (2018).

2.2.3 PostgreSQL

O PostgreSQL é um Sistema Gerenciador de Banco de Dados Objeto-Relacional (SGBD-OR). É uma ferramenta *open source* (possui código aberto para que programadores possam realizar alterações ou até mesmo adicionar novas funcionalidades) e está no mercado há mais de 20 anos. É considerado um SGBD robusto, confiável e capaz de manter a integridade dos dados. Possui um suporte completo a recursos tais como: consultas completas, chaves

estrangeiras, joins, visões e triggers. É um SGBG que está disponível para a maioria dos Sistemas Operacionais, incluindo Linux, Mac OS e Windows. (CAIUT, 2015).

O PostgreSQL foi baseado no Postgres 4.2, da Universidade da Califórnia/Berkeley, projeto encerrado em 1994. Ele é dito Objeto-Relacional por ter suporte a funcionalidades compatíveis com o conceito de orientação de objetos, como herança de tabelas, sobrecarga de operadores e funções e uso de métodos e construtores através de funções. (CAIUT, 2015, p. 3).

Segundo Shaw (2013) o PostgreSQL tem uma grande vantagem que é ser *open source*, ou seja, uma ferramenta de código aberto que pode ser alterada pelos programadores inclusos na comunidade, bem como permite fazer praticamente tudo que é desejado com a aplicação e seu código. Incluindo personalizá-la para própria aplicação interna, objetivos e necessidades específicas. O PostgreSQL ao longo dos anos tem desenvolvido algumas funcionalidades muito únicas que outros SGBDs não oferecem. É uma ferramenta de fácil instalação em todas as plataformas, tão simples como instalar um único pacote. É considerada também um SGBD muito escalável e possui limites superiores e impressionantes sobre o que pode e não pode ser alcançado, conforme apresentado no Quadro 4.

Quadro 4 – Limites de Valores (PostgreSQL)

Máximo do Tamanho Database	Ilimitado
Máximo do Tamanho de Tabelas	32 Terabytes
Máximo do Tamanho da Linha	1.6 Terabytes
Máximo do Tamanho de Campos/Colunas	1 Gigabyte
Máximo de Linhas por Tabela	Ilimitado
Máximo de Colunas por Tabela	250 para 1600 dependendo dos tipos da coluna
Máximo de Índices por Tabela	Ilimitado

Fonte: Elaborado a partir de Shaw (2013, p.11).

2.3 HTML

Hyper Text Markup Language (HTML), em português Linguagem de Marcação de Hipertexto) é considerada uma linguagem de marcação padrão para criar websites. O HTML é composto por diferentes *tags* que poderão ser utilizadas para descrever a estrutura de uma

página web, onde o navegador será responsável por fazer a leitura da estrutura e realizar a exibição do conteúdo na página web. (WARNER, 2021).

Segundo Alves (2021) além das linguagens de programação também é necessária uma linguagem para a criação de páginas de sites. Essa linguagem é denominada HTML. Ao invés de digitar comandos para execução de cálculos ou processamento de dados, a linguagem possui marcadores que também são conhecidos como *tags*, utilizadas para formatar um texto, imagem ou qualquer outro objetivo que faz parte da estruturação de um documento HTML.

Os documentos HTML são formados por textos puros, que podem ser criados até por aplicativos de edição de texto, como o Bloco de Notas do Windows. Eles são formados por um grupo de seções, sendo que cada uma delas é destinada a abrigar um tipo de conteúdo específico. As tags <html> e </html> definem um documento no formato HTML válido. Já <head> e </head> são utilizadas para identificar a seção de cabeçalho do documento, a qual pode conter informações, como texto a ser exibido na barra de títulos do navegador, metadados para mecanismos de busca etc. Por fim, as tags <body> e </body> são utilizadas na definição do corpo do documento HTML, que é o local em que se encontra o conteúdo a ser exibido no monitor. (ALVES, 2021, p. 7).

De acordo com Alves (2021) a estrutura básica do HTML é semelhante à Figura 3.

Figura 3 - Estrutura básica HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" lang="pt-br"/>
    <title> Olá Mundo </title>
  </head>
  <body>
    <h1> Olá Mundo! </h1>
  </body>
</html>
```

Fonte: Alves (2021).

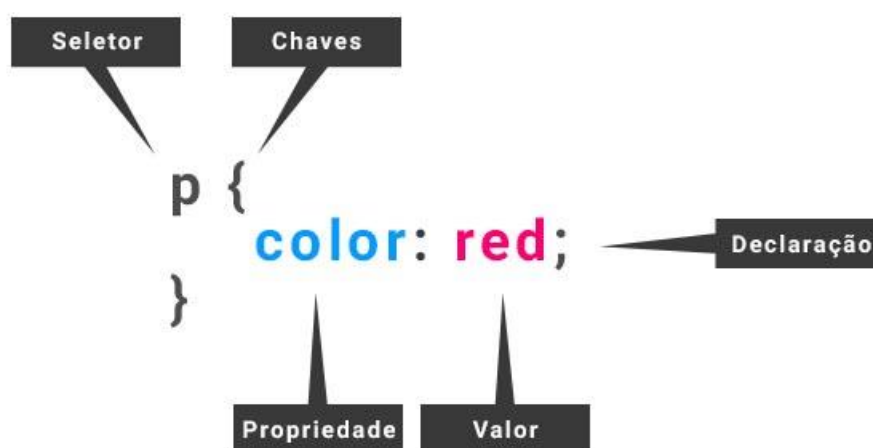
2.4 CSS

As formatações a serem aplicadas em componentes de uma página podem ser feitos através de inserção de comandos diretamente no código HTML. Porém a utilização dessa prática pode ser um problema devido a necessidade de repetir o mesmo trecho de código de formatação em todas as páginas que definem um site. Para evitar essa redundância de trabalho,

foram introduzidas as folhas de estilo CSS (*Cascading Style Sheets*, em português Folhas de Estilo em Cascata) onde permite formatações mais avançadas e também que o código de formatação seja reutilizado em diversos documentos HTML. (ALVES, 2021).

Segundo Alves (2021) as folhas de estilo CSS irão definir como a página HTML deve ser exibida ao usuário. O CSS não possui comandos de execução de operações, apenas regras de formatação que são definidas e associadas aos elementos que compõem uma página HTML. Cada regra é definida através de uma sintaxe, conforme exemplo na Figura 4.

Figura 4 – Regras CSS



Fonte: FRANCISCO (2018)

Conforme observado na Figura 4, o “seletor” representa uma tag HTML qualquer onde a regra CSS está definida pela declaração. A “declaração” é formada por uma “propriedade”, que consiste em um atributo válido da tag HTML que está sendo utilizada, e por fim, um “valor” a ser atribuído.

De acordo com Alves (2021) existem diversas técnicas para aplicação de formatação CSS em códigos HTML. Uma das primeiras técnicas seria o método *in-line* (em linha), como demonstrado na Figura 5.

Figura 5 – CSS *in-line*

```

<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-br" lang="pt-br">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Exemplo 4-1</title>
  </head>
  <body>
    <h1>Exemplo de folha de estilo CSS</h1>
    <p><font style="color:blue; font-family: Helvetica; font-style: italic;
        font-size=4">Método de definição in-line</font></p>
  </body>
</html>

```

Fonte: Elaborado a partir de Alves (2021).

De acordo com Alves (2021) a segunda técnica a regra de formatação é definida separadamente, permitindo que a utilização da mesma ocorra em diversos pontos dentro de um documento HTML, como no exemplo da Figura 6.

Figura 6 - CSS incorporado

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-br" lang="pt-br">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Exemplo 4-2</title>
    <style type="text/css">
      p {font-family: Arial,Helvetica,sans-serif;
        color: blue;
        font-style: italic}
    </style>
  </head>
  <body>
    <h1>Exemplo de folha de estilo CSS</h1>
    <p>Método de definição interno</p>
  </body>
</html>

```

Fonte: Alves (2021)

Segundo Alves (2021) a terceira técnica seria a folha de estilo externa, que difere das duas primeiras técnicas justamente pelo fato das regras estarem separadas em um arquivo à parte, independente do arquivo HTML. A partir daí esse arquivo CSS pode ser utilizado em diversos documentos HTML. O arquivo CSS deve ser gravado com a extensão *.css* e

referenciados pelos arquivos HTML dentro da *tag* específica, como exemplo demonstrado na Figura 8, no qual é referenciado o arquivo .css representado na Figura 7.

Figura 7 - CSS em arquivo externo

```
h1 {  
    font-family:times new roman;  
    font-size:40px;  
    font-weight:bold;  
    font-style:italic;  
    color:blue  
}  
  
h2 {  
    font-family:helvetica;  
    font-size:25px;  
    font-style:italic;  
    color:red  
}  
  
p {  
    font-family:arial;  
    font-size:16px;  
    color:#990000  
}
```

Fonte: Alves (2021)

De acordo com Alves (2021) para que seja utilizada a formatação CSS na página HTML é necessário que seja inserida uma tag <link> no documento HTML, na seção do cabeçalho, referenciando o arquivo, como demonstrado na Figura 8.

Figura 8 – Link do arquivo CSS externo com HTML

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt-br" lang="pt-br">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Exemplo 5-3</title>
    <link rel="stylesheet" type="text/css" href="estilo05-03.css">
  </head>
  <body>
    <h1>Exemplo de folha de estilo CSS</h1>
    <h2>Método de definição em arquivo externo</h2>
    <p>Aqui está um texto formatado com folha de estilo externa</p>
  </body>
</html>
```

Fonte: Alves (2021).

2.4.1 Bootstrap

O Bootstrap é considerado um *framework front-end* de código aberto, com o intuito de criar uma interface web de forma mais rápida. Através dessa biblioteca é possível obter acesso a modelos existentes, que facilitam a criação de sites, permitindo que seja inserido designs responsivos, ou seja, páginas que se ajustam automaticamente ao rodar a aplicação em diferentes dispositivos, como celulares, *desktops* ou *tablets*, utilizando a técnica de *mobile-first*. (ZABOT; MATOS, 2020).

De acordo com Zabot e Matos (2020) é um framework que possui vários modelos de design que utilizam HTML e CSS, que contém componentes prontos e que estão disponíveis para serem utilizados, sem que haja necessidade de realizar grande esforço para estilização. É possível utilizar botões, tabelas, barras de navegação e diversos outros elementos.

2.5 PHP

De acordo com Bento (2014) o PHP (*Hypertext Preprocessor*) é uma linguagem que permite o pré-processamento de páginas HTML. Diante disso, o PHP consegue alterar o

conteúdo de uma página, antes de enviar a mesma para o navegador. Com o PHP torna-se possível também a coleta de entrada de dados vindos do usuário, através de formulários ou até mesmo outras formas de interação.

Para Bento (2014) o PHP nasceu para a web e sua integração com os servidores é simples, possui também uma curva de aprendizado agradável, compara com outras linguagens. Além do mais, é fácil encontrar serviços de hospedagem que oferecem PHP, esses serviços costumam ser baratos comparando com outras tecnologias, sendo que é realizado o mesmo serviço.

Segundo Prettyman (2020) o PHP é atualmente, uma das linguagens mais populares utilizadas para o desenvolvimento de aplicações web. A linguagem evoluiu rapidamente, permitindo que os programas sejam desenvolvidos rapidamente, permitindo também a programação orientada à objetos. Além disso, o PHP fornece vários recursos, possuindo bibliotecas pré-existentes, facilitando o desenvolvimento de uma determinada tarefa.

O PHP é uma linguagem de código aberto, também conhecido como *open source*. É uma comunidade composta por diversos programadores, que estão dispostos a criar novas funcionalidades, impulsionando a utilização da linguagem. O PHP possui diversas versões, a primeira foi lançada em 1995, por Rasmus Lerdorf, disponibilizadas na internet com fóruns para proporcionar aos usuários a possibilidade de dar sugestões e até mesmo realizar alterações no código-fonte. (PRETTYMAN, 2020).

O site oficial do PHP fornece informações relacionadas aos últimos lançamentos da linguagem. Fornece também informações sobre atualizações que serão lançadas futuramente, com datas de liberações previstas. O site permite que os programadores explorem por funcionalidades disponíveis pela própria linguagem, podendo até mesmo realizar o *download* da documentação completa. (PRETTYMAN, 2020).

De acordo com Macintyre e Tatroe (2020) o PHP funciona com todos os sistemas operacionais principais (incluindo Linux, FreeBSD, Ubuntu e Solaris). Pode ser utilizado também todos os principais servidores web, incluindo Apache e OpenBSD. O PHP possui um ótimo apoio relacionado a base dados, suportando as principais bases, tais como: PostgreSQL, Oracle, DB2 e ODBC.

2.5.1 Laravel

No final de 2014 ocorreu uma fase importante para a história do Laravel. Até então havia a versão 4.3, a partir daí surgiram muitos princípios fundamentais, onde a comunidade decidiu partir para a versão 5. A chegada do Laravel 5 fez com que houvesse muitas mudanças na hora de construir o software. A partir dessa versão foi adicionada a arquitetura MVC. A parte mais importante da história do Laravel foi a chegada da versão 5.1, contendo apoio ao longo prazo, desta forma, o Laravel ficou mais sólido dentro das empresas. Além disso, os requisitos mínimos do PHP foram alterados para a versão 5.5, ou até mesmo 5.6. (PECORARO, 2015).

Laravel é conhecido como um *framework* web PHP, onde possui sintaxe expressiva e elegante. Com o Laravel é possível realizar um ponto de partida para a criação do aplicativo ou aplicação desejada. O *framework* está sempre evoluindo, buscando fornecer uma experiência amigável para os desenvolvedores. Para isso, o Laravel conta com alguns recursos poderosos, sendo eles: injeção de dependência completa, camada de abstração de banco de dados, tarefas agendadas, filas, testes de unidade e integração. (LARAVEL, 2022)

Segundo Turini (2015) a utilização de *frameworks* facilita o processo de desenvolvimento, pois desta forma o desenvolvedor não precisa se preocupar com infraestrutura. Sendo assim o *framework* colabora no processo, deixando o projeto mais organizado, evitando códigos repetidos. Desta forma, a principal ideia é oferecer uma estrutura padrão entre os projetos, mantendo a organização de forma que seja fácil manter o código, contendo as melhores práticas do mercado. Com essa reutilização entre os projetos irá poupar muito tempo e trabalho.

Há diversas ferramentas e estruturas disponíveis para o desenvolvimento de uma aplicação web. Acredita-se que o Laravel é a melhor escolha para desenvolver aplicações web modernas e *full-stack* (programadores que trabalham com *front-end*, parte visual da aplicação, e *back-end*, código fonte da aplicação). É considerada uma estrutura progressiva, no qual tem objetivo de crescer junto ao desenvolvedor. Além disso, é uma estrutura escalável, isso devido à natureza amigável de dimensionamento do PHP e ao suporte rápido. O Laravel conta com milhares de programadores em todo o mundo, contribuindo com a estruturação de um *framework* decente, havendo toda uma estruturação comunitária por trás. (LARAVEL, 2022).

2.6 JAVASCRIPT

O JavaScript foi criado por Brendan Eich na Netscape em 1995 e foi originalmente nomeado como “LiveScript”, porém o Java estava em alta nesse momento, e por razões de marketing o “LiveScript” tornou-se “JavaScript”. Apesar do nome a linguagem não tem nada a ver com o Java. Por muito tempo o JS teve uma má reputação devido as janelas pop-up que eram odiadas e também devido a uma série de “vulnerabilidades de segurança”. (ROBBINS, 2018).

De acordo com Robbins (2018), o JavaScript é considerado uma linguagem de programação que acrescenta interatividade e customizações, além de dar ênfase à comportamentos para os sites. É uma linguagem de programação que funciona ao lado do cliente, ou seja, no navegador e não no servidor. É uma linguagem de script leve, mas incrivelmente poderosa. O JavaScript era frequentemente visto e encontrado em sites, a partir dos navegadores, mas com o tempo foi ganhando forma e começou a surgir em aplicações nativas, arquivos PDF’s e livros eletrônicos. Até mesmo servidores web’s podem ser alimentados por JavaScript.

Para Robbins (2018), o JavaScript é uma linguagem de programação dinâmica e não precisa ser executado através de um compilador. O código escrito é totalmente interpretado pelo navegador no qual está sendo utilizado. É muito comum encontrarmos o JavaScript como forma de adicionar eventos e interatividade nas páginas. Considerando assim a camada “estrutural” de uma página que seria a marcação HTML, e a camada “apresentável” de uma página, que é composta por CSS, e a terceira camada seria a “comportamental” que é composta pelo JavaScript. Todos esses elementos e atributos citados podem ser aplicados utilizando o JavaScript, através da DOM (*Document Object Model*). Com o JavaScript podemos alterar o conteúdo e o estado de um determinado objeto conforme a entrada do usuário.

É muito comum nos depararmos com o JavaScript em ação, principalmente em validações de formulários, tais como e-mail, CPF e CEP. Ao inserirmos um e-mail já cadastrado normalmente recebemos um feedback imediato informando que este e-mail já foi cadastrado por outra pessoa. Quando acontece isso o campo costuma ficar em vermelho, no qual o JavaScript foi responsável por alterar o comportamento deste campo. É comum também realizar o bloqueio de formulários quando alguma informação está incorreta, todas esses comportamentos possuem feedback imediato, sem que haja o recarregamento da página. (ROBBINS, 2018).

2.6.1 jQuery

O jQuery é considerado como uma biblioteca dominante do JavaScript. Foi escrita em 2005 por John Resig, sendo utilizado por mais de dois terços de todos os sites web. É uma biblioteca gratuita e de código aberto, emprega uma sintaxe que facilita a utilização do CSS, JavaScript e o DOM. (ROBBINS, 2018).

Com o jQuery é obtido um nível altíssimo de compatibilidade com os navegadores (até mesmo o Internet Explorer), mas também têm acesso rápido e fácil quanto à manipulação de HTML e DOM, inclusive funções especiais para interagir diretamente com o CSS, sendo possível controlar eventos, no qual é uma funcionalidade poderosa para criar efeitos e animações, e também, realizar comunicações assíncronas com o servidor web. (NIXON, 2018).

De acordo com Flanagan (2014), a utilização do jQuery facilita o encontro dos elementos de um documento, tornando-se fácil a manipulação e a adição de conteúdo. Sendo assim, é possível editar os atributos HTML e propriedades CSS, definindo métodos e tratamentos de evento, até mesmo animações. Com o jQuery é possível realizar requisições HTTP através do Ajax.

O jQuery se concentra em consultas (*query*, em inglês). Uma consulta comum seria a utilização de um seletor CSS para identificar um determinado elemento dentro do documento HTML, retornando um objeto representando os elementos. Com o retorno do objeto pode ser utilizado muitos métodos úteis para operar como um grupo nos elementos condizentes. (FLANAGAN, 2014).

2.6.2 Ajax

Ajax (*Asynchronous JavaScript and XML*) é uma coleção de classes especiais baseadas em *scripts*. Essa classe permite que o cliente possa interagir com o servidor, possibilitando que uma parte da página seja recriada, reformatada, reenviada ao cliente e reenviada ao navegador do cliente. Para facilitar o entendimento do Ajax é necessário entender um pouco do conceito de postbacks síncronos e renderização de página inteira com postbacks assíncronos e renderização de página parcial. (BEASLEY, 2020).

Segundo Beasley (2020) um *postback* síncrono ocorre quando uma página é enviada para o servidor, e processada pelo lado do servidor, onde toda a página é recriada e reformatada. Após a página inteira ter sido recriada é enviada de volta para o cliente, e reenviado no *browser* do cliente, onde ocorre a reprodução da página completa. Às vezes o *postback* pode ser ineficiente pelo simples fato de realizar algum processamento desnecessário ao recriar uma página, sendo que apenas uma informação da página precisa ser atualizada.

As classes Ajax são implementadas em HTML, CSS e JavaScript. Portanto o Ajax não irá funcionar corretamente em navegadores que não suportam *scripts*, ou até mesmo aqueles navegadores que possuem o script do navegador desativado. (BEASLEY, 2020).

Segundo Patel (2014), o jQuery fornece o método Ajax \$.ajax() para realizar a chamada dos dados remotos, para que a aplicação possa ler o formato JSON. O Método AJAX faz um pedido de *XMLHttpRequest* ao *servlet*, para a obtenção de dados. Algumas propriedades podem ser definidas na hora de fazer a requisição, conforme apresentado no Quadro 5:

Quadro 5 – Propriedades de Configuração - Ajax

<i>accepts</i>	Este campo representa o método de pedido a utilizar, tal como <i>GET</i> ou <i>POST</i> . Por padrão, se o tipo de parâmetro não for especificado, assume-se que a chamada é um Pedido de método <i>GET</i> .
<i>beforeSend</i>	Este campo pode ser uma função ligada a ele e utiliza um pré-processamento dos pedidos antes de fazer a chamada final para o recurso do servidor.
<i>cache</i>	Este campo tem um valor booleano e representa o cache de recursos no navegador.
<i>crossDomain</i>	Este campo toma um valor booleano e indica se um pedido é uma chamada de domínio cruzado.
<i>type</i>	Este campo representa o método de pedido a utilizar, tal como <i>GET</i> ou <i>POST</i> . Por padrão, se o tipo não for especificado, assume-se que a chamada é um pedido de método <i>GET</i> .

Fonte: Elaborado a partir de Patel (2014).

2.6.3 SweetAlert

O *SweetAlert* é considerado uma biblioteca JavaScript que tem como objetivo facilitar a criação de alertas em aplicações web. Todo o código-fonte está aberto, sendo disponibilizado no GitHub, sendo assim qualquer programador possui acesso. Um dos grandes benefícios é que a biblioteca é compatível com todos os navegadores atuais. Desta forma, essa biblioteca permite a criação de alertas mais agradáveis, responsivos e customizados. (PINHO, 2018).

3. METODOLOGIA DA PESQUISA

O presente trabalho de conclusão de curso é caracterizado como uma pesquisa aplicada e descritiva, que tem como objetivo desenvolver uma aplicação web responsiva para auxiliar no gerenciamento de processos na área da construção civil, tendo como foco principal o levantamento arquitetônico, tarefa realizada para obter medidas de um determinado espaço.

Este trabalho busca resolver os seguintes problemas: Como reduzir valores em um projeto arquitetônico? Como buscar por profissionais que atuam em um determinado município? Como otimizar o tempo do arquiteto em relação as atividades de um projeto? Como evitar o deslocamento nos casos de longas distâncias?

Desta forma, o projeto é destinado àqueles profissionais que precisam terceirizar seus projetos, buscando arquitetos credenciados mais próximos de seu cliente. Sendo assim, seu tempo será otimizado e o valor do projeto será menor para o cliente final, pois os custos do profissional contratado pelo cliente, serão reduzidos.

Para a elaboração deste trabalho foi realizada a revisão da literatura, capítulo no qual foi utilizado para demonstrar conceitos e aprofundar um pouco sobre as tecnologias e ferramentas que serão utilizadas para desenvolver a aplicação. Também foram pesquisadas ferramentas com propósitos similares em relação ao protótipo proposto neste trabalho. As ferramentas encontradas serão apresentadas no capítulo a seguir, no Estado da Arte.

3.1 ESTADO DA ARTE

Nesta seção são demonstradas três aplicações disponíveis no mercado que possuem características e objetivos similares comparado com o protótipo proposto neste trabalho. São abordadas características gerais e as principais funcionalidades destas aplicações para que haja um complemento na análise do levantamento de requisitos para o protótipo.

Dentre as três ferramentas pesquisadas, as aplicações Asana e FlowUp são utilizados para gestão de projetos, atendendo diversas áreas profissionais e não somente a área da arquitetura. Já a aplicação da Vobi é direcionada apenas para a área da arquitetura e designer. Até o momento não foram encontradas ferramentas quem possuem a funcionalidade específica

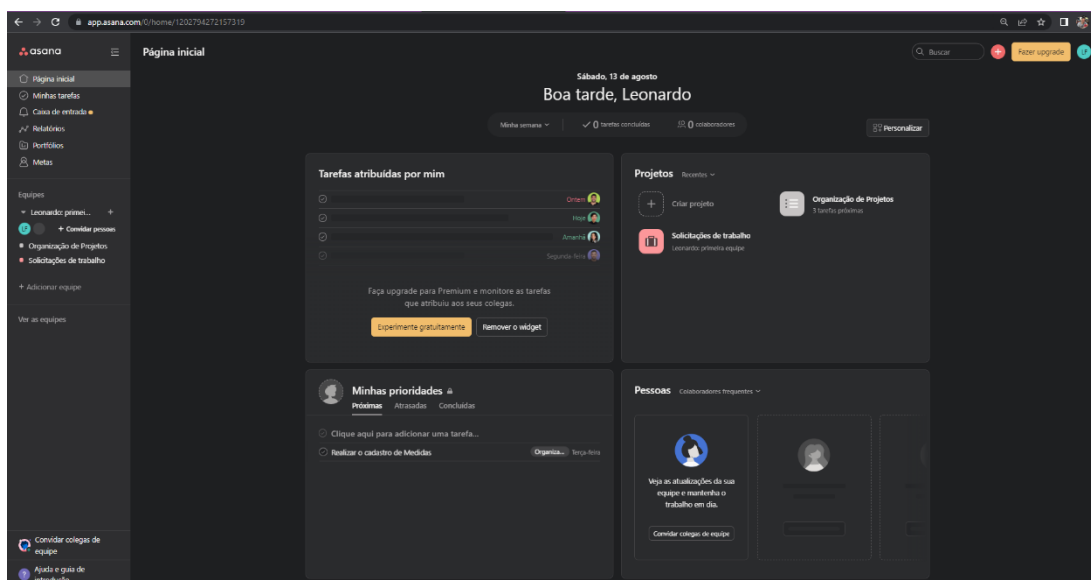
para realizar o levantamento arquitetônico, que é o principal objetivo proposto por este trabalho, se apresentando como um diferencial.

3.1.1 Asana

O software Asana, representado na Figura 9 é uma ferramenta de gestão de projetos e tarefas, desenvolvido para diversas áreas. A aplicação permite que o trabalho seja realizado em equipe, disponibilizando gráficos e relatórios para que haja um melhor acompanhamento do trabalho que está sendo realizado. O sistema permite que sejam criados diversos projetos, e permite também atribuir tarefas a outros usuários, fazendo com que o usuário possa consultar essas tarefas a qualquer momento.

Com esta aplicação é possível obter *insights* em tempo real sobre o andamento do trabalho de sua equipe, avaliando os projetos, solucionando problemas de forma instantânea, mantendo o trabalho organizado. O software permite que diversos recursos possam ser integrados, como e-mail, Google Drive, Slack, entre outras integrações, que atualmente são mais de 100 integrações disponíveis. (ASANA, 2022).

Figura 9 - Software Asana - WEB.

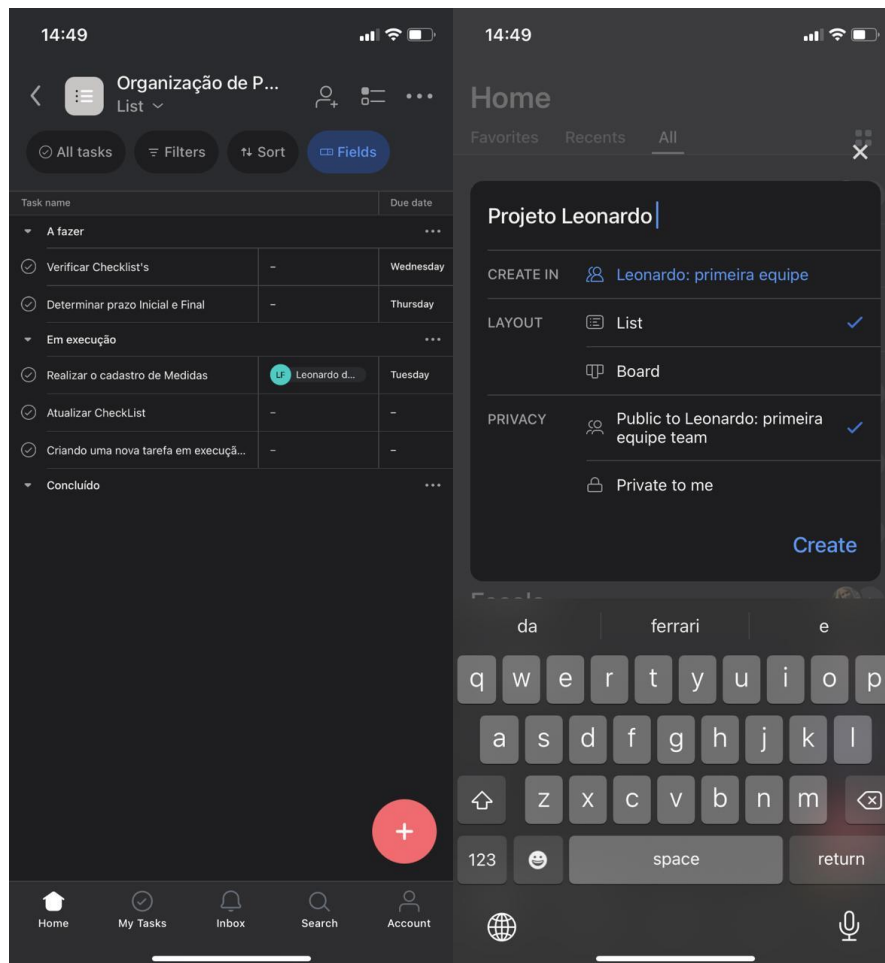


Fonte: Asana (2022).

A aplicação possui quatro planos disponíveis para contratação, sendo eles: *Basic*, *Premium*, *Business* e *Enterprise*. Além disso a aplicação está disponível para plataformas

móveis, podendo ser utilizado em iOS ou Android (conforme Figura 10), e até mesmo disponível para ser utilizado no Windows, bastando apenas realizar o *download*. (ASANA, 2022)

Figura 10 – Aplicativo Asana - iOS.



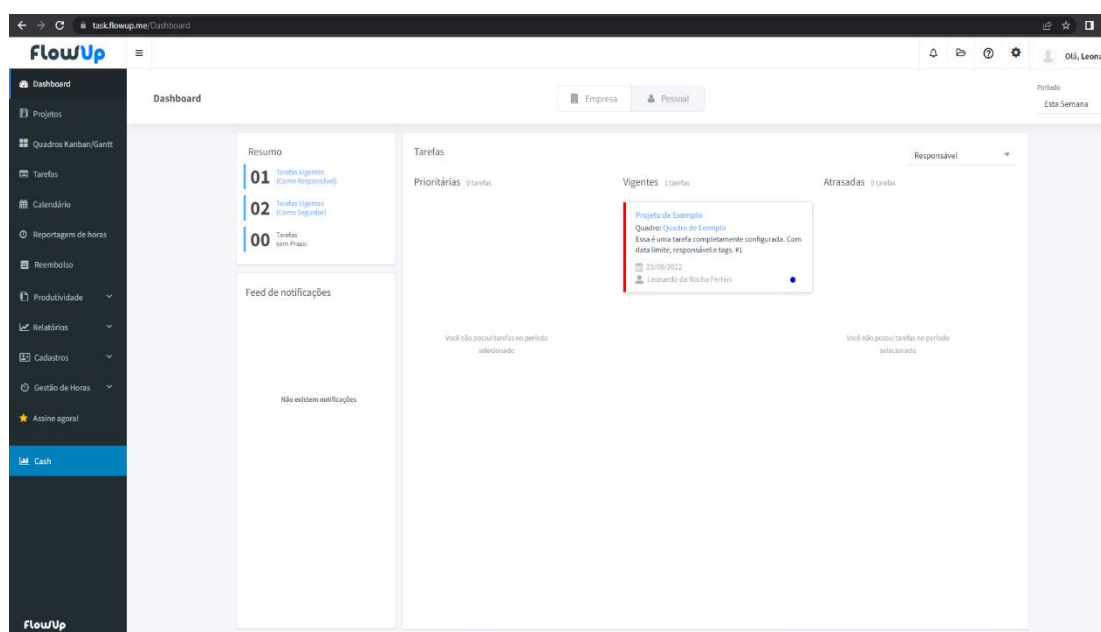
Fonte: Asana (2022).

O aplicativo possui diversos recursos para a gestão dos projetos e de equipes. Além disso, conta com uma interface moderna e intuitiva, contribuindo com a usabilidade, sendo prática e rápida. Em um projeto pode haver diversas seções, ou estados, e essas tarefas podem ser atribuídas em algum desses estados. Caso uma tarefa seja concluída basta arrastar a tarefa para o estado desejado, podendo ser definido prioridade para a mesma, sendo: alta, média ou baixa.

3.1.2 FlowUp

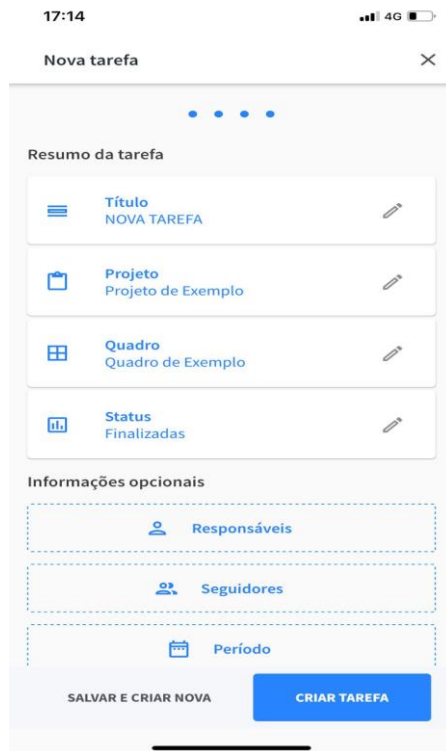
O software FlowUp (Figura 11) tem como intuito controlar tarefas em vários projetos, de forma integrada e totalmente *on-line*. Utilizando esse software é possível fazer uma gestão em relação aos projetos, permitindo que o líder direcione tarefas/trabalhos para cada membro da equipe. A otimização e gestão de tarefas torna-se simples ao utilizar essa aplicação. Os prazos de entregas podem ser gerenciados através do quadro Kanban ou Diagrama de Gantt. (FLOWUP, 2022).

Figura 11 – Software FlowUp – Web.



Fonte: FlowUp (2022).

Uma grande vantagem dessa aplicação é poder organizar o *dashboard* da aplicação, conforme a necessidade do usuário. Através do *dashboard* é possível visualizar o andamento das tarefas e da equipe em uma única tela, contendo atalhos e visualizações rápidas das principais informações relacionadas ao projeto. Outro recurso disponível é o compartilhamento de arquivos, permitindo que o cliente comente se aprova o que foi realizado, ou até mesmo que solicite alterações. O FlowUp disponibiliza aplicativo para agilizar o processo das atividades, o aplicativo está disponível para iOS e Android (Figura 12). (FLOWUP, 2022).

Figura 12 – Aplicativo FlowUp - iOS.

Fonte: FlowUp (2022).

O aplicativo permite criar novas tarefas e vinculá-las ao projeto que for necessário. Além disso é possível atribuir um título ao projeto e definir o seu status. Pode ser definido o período e também é permitido relacionar responsáveis que realizarão essas atividades.

3.1.3 Vobi

O software da Vobi (Figura 13) é totalmente direcionado para escritórios de arquitetura e para designers. O principal objetivo é simplificar a rotina diária, ganhando mais tempo e possibilitando o crescimento do negócio. A ferramenta permite que o usuário possa acompanhar o andamento de cada projeto, adicionar tarefas, visualizar status e compartilhar informações e arquivos com clientes e até mesmo com membros da própria equipe. (VOBI, 2022).

Figura 13 – Listagem de Projetos – Vobi.

#	Projeto	Etapa	Cliente	Status	Criado em
55179	Cobertura Mário	-	Mario MCK	Em andamento	30 de mai 2022
51252	Cabana FC Gregorio	Anteprojeto	Francesco Di Gregorio	Em andamento	09 de mai 2022
50614	Ap. Leonardo	3D	Leonardo Espindola - Dézzou Decor	Em andamento	04 de mai 2022
47166	Interiores Dócio e Ivone	-	Dócio Stolf	Em andamento	13 de abr 2022
47089	Abner - geminadas	-	Abner Santos	Em andamento	13 de abr 2022
47085	Krissy	-	Krissy Sfredo	Em andamento	13 de abr 2022
47084	Mari - casa	-	Mariene Fabricio	Em andamento	13 de abr 2022
47081	Carlos C - Quarto filha	Projeto Executivo	Carlos Cunha	Em andamento	13 de abr 2022
46549	Fernando Budag - casa	Anteprojeto	Fernando Budag	Em andamento	11 de abr 2022
46078	Quarto Lillan	3D	Toni Buatim	Em andamento	08 de abr 2022

Fonte: Vobi_(2022).

A aplicação permite que ao incluir o projeto seja adicionado *checklists's* (Figura 14). Portanto estas *checklist's* devem ser concluídas durante o andamento do projeto.

Figura 14 - Cadastro de projetos com checklist - Vobi.

Nova tarefa

Selecionar data Selecionar hora Selecionar data Selecionar hora

Prioridade Status

Projeto Etapa do projeto

Descrição

Checklist

+ Adicionar item

Cancelar

Fonte: Vobi_(2022).

A principal vantagem do sistema, além de permitir realizar a gestão de projetos, é que o mesmo possibilita gerenciar pagamentos, fornecedores e até mesmo orçamentos. Além disso possui uma lista de compras *on-line*, onde é possível verificar informações do produto, preços,

detalhes técnicos. Caso algum produto seja de interesse do arquiteto/designer o mesmo pode optar em salvar o produto na pasta do cliente. (VOBI, 2022).

4. PROTÓTIPO DE SISTEMA WEB PARA AUXILIAR NO GERENCIAMENTO DE PROCESSOS NA CONSTRUÇÃO CIVIL

Neste capítulo são apresentados todos os aspectos técnicos do processo de análise de implementação do protótipo.

4.1 ANÁLISE

No capítulo da revisão da literatura foram abordados assuntos relacionados a engenharia de software, que foram necessários aplicar nesta etapa, para que pudesse obter o entendimento necessário das necessidades do negócio do cliente, e quais os objetivos da aplicação. Diante disso, a análise torna-se fundamental para deixar transparente o que será desenvolvido. A análise também contribui na hora de prestar manutenibilidade na aplicação, pois todos os processos utilizados durante o desenvolvimento estão bem descritos e qualquer problema a ser resolvido torna-se mais simples de corrigi-los.

4.1.1 Visão Geral do Protótipo

O protótipo da aplicação tem como objetivo auxiliar a gestão de processo. O principal objetivo é disponibilizar recurso para o levantamento arquitetônico. A aplicação poderá ser utilizada para arquitetos e até mesmo *designers* de interiores. O foco principal da aplicação é a terceirização de projetos voltados ao levantamento de medidas, bem como o acompanhamento do projeto.

Ao utilizar essa aplicação o profissional poderá cadastrar seus projetos e até mesmo terceirizar eles, caso seja necessário. Ao optar pela terceirização do projeto, a aplicação permite verificar quais profissionais atuam no município no qual será necessário executar o levantamento de medidas, ou até mesmo outras tarefas. Portanto, ao realizar o cadastro do projeto será permitido optar pela terceirização do serviço, onde serão listados todos os profissionais que trabalham no município, e também, será permitido que o usuário inclua manualmente a pessoa no qual prestará o serviço.

Após realizar o cadastro de um projeto terceirizado, o projeto ficará com a situação “Pendente” até que o profissional terceirizado aceite realizar as tarefas deste projeto. Portanto,

a aplicação disponibiliza uma área onde o profissional possa gerenciar essas solicitações, aceitando ou recusando os trabalhos.

A principal vantagem da aplicação é evitar o deslocamento de longa distância nos casos em que for necessário. A terceirização fará com que o profissional no qual está terceirizando o projeto, reduza os valores que seriam acrescentados no projeto e até mesmo que poupe seu tempo podendo priorizar outras tarefas. Além disso o profissional poderá acompanhar o andamento da execução das atividades que estão sendo realizadas.

4.1.2 Comparação do Protótipo com o Estado da Arte

No Quadro 6 é possível verificar algumas comparações realizadas entre os recursos disponibilizados pelos sistemas pesquisados e listados no estado da arte com o protótipo proposto neste trabalho. O objetivo é mostrar quais recursos estão implementados nos sistemas avaliados.

Quadro 6 – Recursos dos sistemas apresentados

Recurso	Protótipo Proposto	Asana	FlowUp	Vobi
Listagem de Projetos	X	X	X	X
Atribuição de Tarefas	X	X	X	X
Andamento do Projeto	X	X	X	X
Levantamento Arquitetônico	X			
Aplicativo para Dispositivos Móveis		X	X	X
Aplicação Web Responsiva	X	X	X	X
Terceirização de Projeto	X			

Fonte: Acervo do autor (2022).

4.1.3 Requisitos

Nesta seção estão elencados os requisitos para o funcionamento da aplicação. Esses requisitos deverão ser desenvolvidos para que seja atendido a necessidade do cliente, e também, cumprindo aquilo que foi proposto. O Quadro 7, apresenta os requisitos funcionais necessários para o desenvolvimento do protótipo e as regras de negócio serão apresentadas no Quadro 10.

Quadro 7 – Requisitos Funcionais

Número	Nome	Descrição	Regras de Negócio
RF01	Cadastro de Usuário	O protótipo, em sua tela inicial, deverá obter uma ação onde o usuário possa efetuar o cadastro. Para a realização do cadastro deverá ser informado os seguintes campos: Nome (obrigatório), sobrenome (obrigatório), apelido (opcional), e-mail (obrigatório), data de nascimento (obrigatório), CPF (obrigatório), celular (obrigatório), telefone fixo (obrigatório) e senha (obrigatório). Além disso, deve ser informado o Conselho Regional de Engenharia e Agronomia (CREA), para que seja possível identificar o profissional, opção para que o usuário permita que outros profissionais incluam o mesmo como profissional terceirizado. Junto ao cadastro deve ser informado o endereço completo (Complemento, nº, bairro, CEP e município).	RN01 RN02 RN03
RF02	Login	O protótipo deverá permitir que o usuário realize o login para acessar determinados recursos da aplicação. Para realizar o acesso será necessário informar o e-mail e senha.	
RF03	Logout	O protótipo deverá permitir que o usuário finalize seu acesso.	
RF04	Menu	O protótipo deverá possuir um menu, possibilitando que o usuário acesse determinados recursos da aplicação. Deverá dispor das opções: Áreas de Atuação (Arquitetos), Novo Projeto, Meu Perfil e Logout.	
RF05	Tela Inicial - Dashboard	O protótipo deverá possuir um Dashboard contendo informações dos projetos que pertencem ao usuário logado. Esse dashboard deverá aparecer após ser realizado o login na aplicação. Deverá aparecer as seguintes opções para o usuário (em relação aos projetos): Pendente (aguardando aceitação de terceiros); Pendente (aguardando minha aceitação); em andamento; em alteração; Cancelado; Concluído; Recusado;	RN04
RF06	Cadastro área de Atuação	O protótipo deverá permitir que o usuário configure os municípios no qual o mesmo irá atuar. Apenas o Município deverá ser informado para que seja possível realizar o cadastro.	RN04
RF07	Cadastro CheckList	O protótipo deverá permitir o cadastro de checklist's, que estarão relacionados ao usuário. Os campos necessários para a realização do cadastro são: Nome (obrigatório) e descrição (opcional).	RN04
RF08	Cadastro CheckList x Atividade	O protótipo deve conter uma área para que o usuário possa cadastrar as atividades relacionadas a checklist. Os campos necessários para a realização do cadastro são: O código da CheckList (obrigatório) e a descrição da atividade (obrigatório).	RN04 RN06

RF09	Cadastro de Projeto	O protótipo deverá permitir que o usuário cadastre seus projetos. Para realizar o cadastro de um projeto serão necessários que alguns campos sejam preenchidos, tais como: Nome (obrigatório), descrição (obrigatório), nome do cliente (obrigatório), e-mail do cliente (obrigatório), número de telefone do cliente (obrigatório), situação (cadastrado automaticamente pelo protótipo), data de criação (cadastrado automaticamente pelo protótipo), data do atendimento (obrigatório), hora do atendimento (obrigatório), data do prazo final (obrigatório), data de conclusão (inicialmente esse campo deve estar desabilitado), código do profissional terceirizado (opcional) e por último, o código da checklist (opcional). Além disso, deverá ser informado o endereço completo (Complemento, nº, bairro, CEP e município) da onde será realizado o serviço.	RN02 RN04 RN05 RN06 RN07 RN11 RN12 RN13
RF10	Consulta de Projetos	O protótipo deverá oferecer uma tela onde o usuário possa consultar os projetos. Essa consulta deverá filtrar a situação, de acordo com aquilo que foi solicitado através do <i>dashboard</i> . Deverá ser filtrada uma das seguintes situações: Pendente (aguardando aceitação de terceiros); Pendente (aguardando minha aceitação); em andamento; em alteração; Cancelado; Concluído; Recusado; A consulta de projetos deverá dispor das seguintes ações: Alterar, Cômodos, Atividades, Info. Arquiteto, Info. Projeto, Concluir, Cancelar e Arquivos.	RN10
RF11	Tela de Conclusão das Atividades	O protótipo deverá possuir uma tela onde o usuário possa concluir as atividades que já foram realizadas, de acordo com a checklist vinculada ao projeto. O único campo que deverá ser preenchido é a situação.	RN08
RF12	Cadastro de Cômodos	O protótipo deverá conter uma tela para o cadastro de cômodo. Os campos necessários para realização do cadastro são: Nome (obrigatório) e descrição.	RN07 RN15
RF13	Consulta de Cômodos	O protótipo deverá conter uma tela para consultar os cômodos relacionados ao projeto. A consulta de cômodos deverá ser acessada ao consultar o projeto.	RN15
RF14	Consulta de Atuação por Região	O protótipo deverá possuir uma consulta que permite verificar profissionais que atuam em determinada região. A consulta deverá mostrar todos os profissionais cadastrados. Nesta consulta não deverá aparecer as regiões do usuário que está olhando a consulta.	RN04
RF15	Cadastro de Medidas	O protótipo deverá possuir uma área para que seja possível realizar o levantamento arquitetônico. Ao realizar o cadastro da medida o código do projeto e do cômodo deve ser preenchido automaticamente. Para o preenchimento do formulário deverão ser preenchidos os campos: Descrição da medição (obrigatório), tipo de unidade de medida	RN07 RN09

		(obrigatório), tipo de medida (obrigatório), medição (obrigatório), tipo de ponto (opcional), medida de referência (opcional).	
RF16	Consulta de Medidas	O protótipo deverá possuir uma consulta para verificar as medidas de um determinado cômodo. As medidas estarão relacionadas à algum cômodo do projeto, portanto, ao consultar o cômodo de um projeto será possível consultar as medidas deste cômodo.	RN09
RF17	Informações do Profissional	O protótipo deverá permitir verificar o perfil de outros profissionais. Ou até mesmo permitir verificar mais informações sobre o profissional, permitindo que os profissionais possam sem comunicar através do telefone.	
RF18	Cadastro de Arquivos do Projeto	O protótipo deverá oferecer uma tela para que seja possível o usuário adicionar URL's relacionados ao projeto.	RN14
RF19	Cadastro Área de Medição	O protótipo deverá possuir uma área para que seja possível realizar o cadastro da área de medição de um determinado cômodo. A tela de inclusão deverá ser acessada pela consulta de cômodos.	

Fonte: Acervo do autor (2022).

O Quadro 8, apresenta os requisitos funcionais opcionais para o desenvolvimento do protótipo. Estes requisitos poderão ser implementados em versões futuras da aplicação. Todos os requisitos listados neste Quadro não fazem parte da versão inicial.

Quadro 8 – Requisitos Funcionais Opcionais

Número	Nome	Descrição
RF20	Log's	O protótipo deverá permitir realizar a inclusão de Log's, principalmente em relação a tabela de projetos. Toda ação realizada na rotina de projeto deve ser armazenada no banco de dados, contendo o dia e hora no qual o registro sofreu alteração, gravando o registro anterior e o atual. Deverá possuir log's de login também.
RF21	Envio de e-mail	Qualquer alteração na situação de projetos o protótipo deverá disparar um e-mail para o usuário responsável pelo projeto, devendo também enviar um e-mail para o terceirizado quando um novo projeto for atribuído a ele.
RF22	Expiração de Senha	O protótipo deverá permitir que a senha expire a cada 6 meses. Esse aviso de expiração deve acontecer com antecedência para que usuário troque a senha caso ache necessário.
RF23	Cadastro Múltiplo de Atendimento	O protótipo deverá permitir que o cadastro de municípios onde o usuário irá atuar, seja feito de forma múltipla, selecionando municípios de vários estados.
RF24	Chat	O protótipo deverá oferecer um chat onde seja possível realizar a conversação entre o arquiteto que abriu o projeto e o arquiteto terceirizado. Essa conversação deverá ser realizada para cada projeto, possibilitando manter o histórico do que foi definido no decorrer do projeto.

RF25	Pagamento	O protótipo deverá dispor de uma funcionalidade para realizar o pagamento para o profissional terceirizado, assim que o projeto for concluído.
RF26	Avaliação	O protótipo deverá permitir que seja possível avaliar determinado profissional pelo trabalho realizado. Deverá possuir o número total de avaliações, mostrando a média das avaliações em seu perfil.
RF27	Validação de Data/Hora	O protótipo deverá permitir que seja validado data e hora ao cadastrar um projeto. O projeto a ser cadastrado não deve ter conflito com outro (no mesmo horário de uma determinada data). Ou seja, não pode haver projetos concomitantes.
RF28	Mapa da área de atuação	O protótipo deverá permitir que o usuário possa ter acesso ao mapa. No mapa deve ser mostrado os profissionais que atuam em determinada área do mapa.
RF29	Orçamento do Projeto	O protótipo deverá possuir uma área para que seja realizado orçamentos. Caso o orçamento seja aprovado o mesmo deverá estar relacionado ao projeto na hora de cadastrá-lo.

Fonte: Acervo do autor (2022).

O Quadro 9, tem como finalidade apresentar os requisitos não funcionais, que são características gerais da aplicação, sendo requisitos também que envolvem desempenho, usabilidade e segurança.

Quadro 9 – Requisitos Não Funcionais

Número	Descrição
RNF01	O protótipo deverá ser desenvolvido com PHP, com a versão 7 ou superior.
RNF02	Toda tela de cadastro deverá possuir validação no <i>front-end</i> , não permitindo inserir valores nulos no banco de dados.
RNF03	A autenticação deverá expirar em uma hora, essa contagem deverá partir do momento em que não está sendo utilizado o sistema.
RNF04	Para o armazenamento dos dados deverá ser utilizado o banco de dados PostgreSQL.
RNF05	O protótipo deverá ser totalmente responsivo, possibilitando que a aplicação seja aberta em telas diferentes, de diversos tamanhos.
RNF06	Toda rotina que possui relacionamento com usuário a aplicação deverá mostrar registros que são relacionados ao mesmo.
RNF07	As requisições não deverão demorar mais que 20 segundos.

Fonte: Acervo do autor (2022).

O Quadro 10 descreve as regras de negócio do protótipo, ou seja, são regras e comportamentos necessários para o funcionamento adequado da aplicação.

Quadro 10 – Regras de Negócio

Número	Descrição
RN01	Ao realizar o cadastro de usuário o preenchimento da senha deverá ser requerido no mínimo 6 caracteres.
RN02	Para o preenchimento de cidades e estados deve ser utilizada a API disponível pelo IBGE.
RN03	Ao realizar o cadastro deve ser obrigatório informar o número do CREA, para que seja possível identificar o profissional.

RN04	A aplicação deve garantir que toda informação disponibilizada seja relacionada ao usuário logado. Garantindo a confidencialidade da informação.
RN05	Somente o arquiteto terceirizado poderá concluir o projeto. Nos casos de projetos que não são terceirizados, o arquiteto que incluiu o projeto poderá concluir o projeto.
RN06	O projeto não pode ser concluído enquanto todas as atividades da checklist não forem concluídas.
RN07	O projeto não pode ser alterado quando o mesmo estiver com a situação “Concluído”. Deverá permitir cadastrar ou alterar medidas apenas quando a situação for “Em andamento”.
RN08	O profissional terceirizado deverá concluir as atividades realizadas na checklist no decorrer do projeto.
RN09	As medidas deverão ser consultadas a partir da área de medição selecionada.
RN10	Todo projeto terceirizado deverá gerar uma pendência (com a situação "Pendente", no qual poderá ser consultada através do <i>dashboard</i> , através da opção "Pendente – Aguardando aceitação de terceiros"). Verificando se o profissional terceirizado aceita ou não realizar o projeto.
RN11	Para o tipo de unidade de medida a ser informado no projeto deverão ser disponibilizadas as opções: Metro, centímetro e milímetro.
RN12	Para o tipo de medida a ser informado no projeto deverão ser disponibilizadas as seguintes opções: Altura, largura, profundidade e inclinada.
RN13	Para o tipo de ponto a ser informado no projeto deverão ser disponibilizadas as seguintes opções: Hidráulico e Elétrico.
RN14	Todo projeto poderá ter imagens relacionadas ao mesmo. Essas imagens deverão estar disponibilizadas no Google Drive. Desta forma, deverá possuir uma tela para cadastrar esses endereços (URL).
RN15	Os cômodos deverão ser acessados pela consulta de projetos. Ao consultar os cômodos deverá aparecer os cômodos relacionados ao projeto. Ao cadastrar o cômodo deverá ser vinculado ao projeto selecionado.

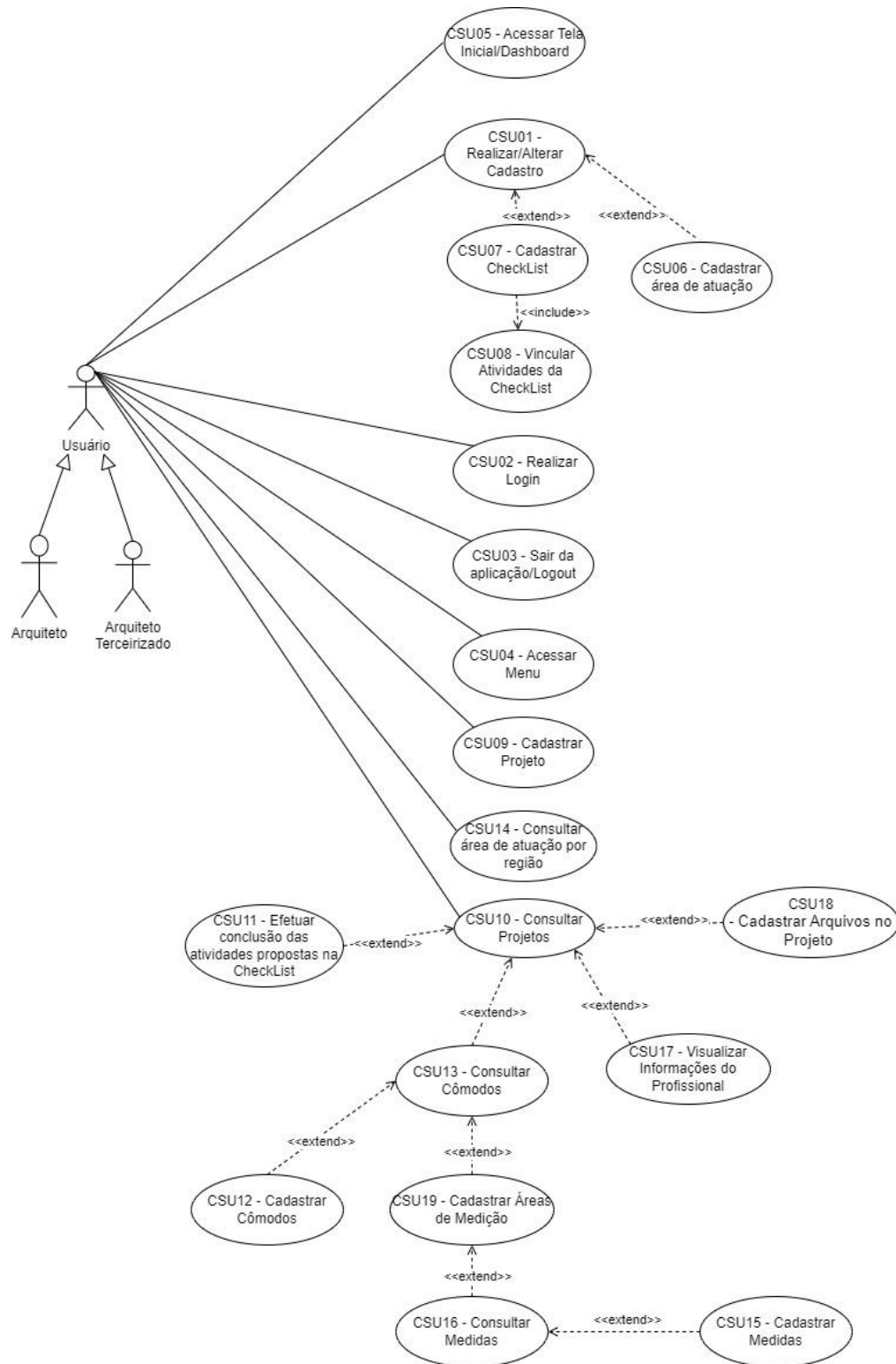
Fonte: Acervo do autor (2022).

4.1.4 Diagramas

Para facilitar a compreensão dos requisitos elencados e das funcionalidades do protótipo foram elaborados alguns diagramas. Os diagramas representam os recursos que estão disponíveis para os usuários, bem como o passo a passo para executar a atividade principal proposta por esse protótipo.

A Figura 15 representa um diagrama de caso de usos, onde apresenta as funcionalidades do protótipo, bem como demonstra sua disponibilidade aos atores do sistema, neste caso o Arquiteto e o Arquiteto Terceirizado.

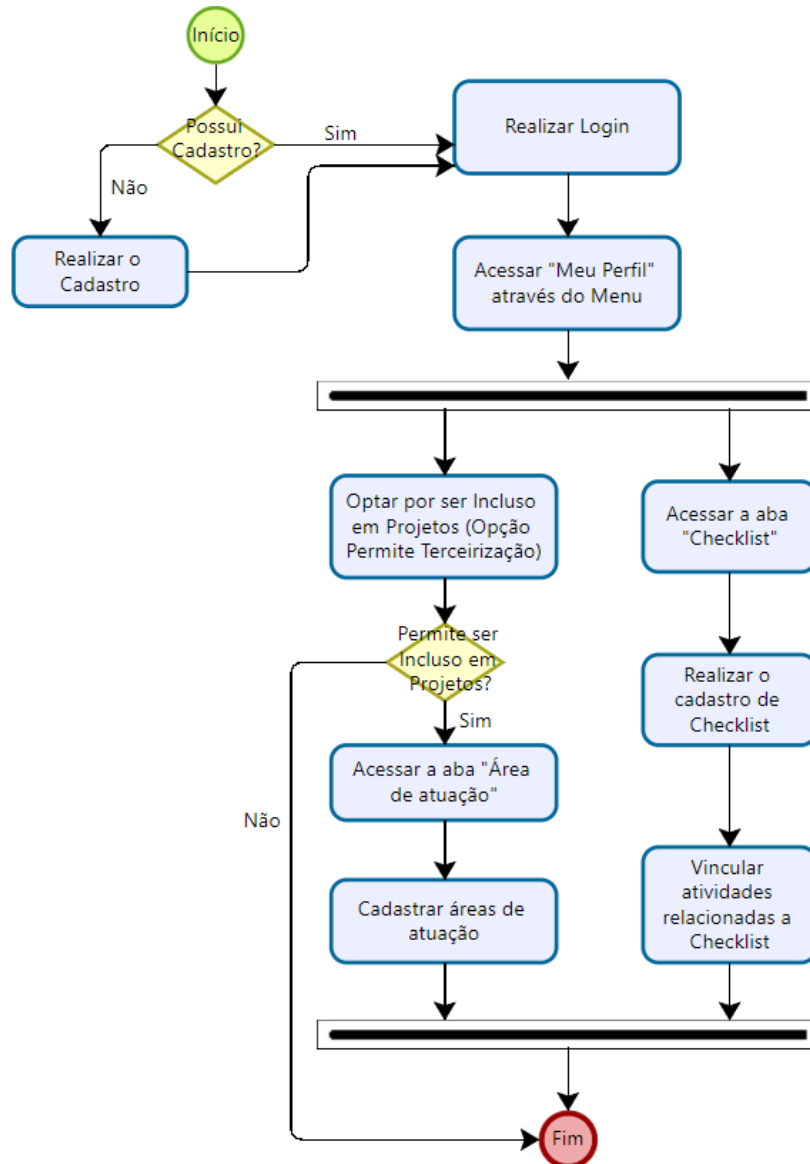
Figura 15 – Diagrama de Caso de Uso



Fonte: acervo do autor (2022).

Para que o profissional trabalhe como terceirizado é necessário realizar algumas configurações. Para exemplificar a configuração a ser realizada foi elaborado um diagrama de atividade, podendo ser observado na Figura 16.

Figura 16 – Diagrama de Atividade – Configuração do Perfil.

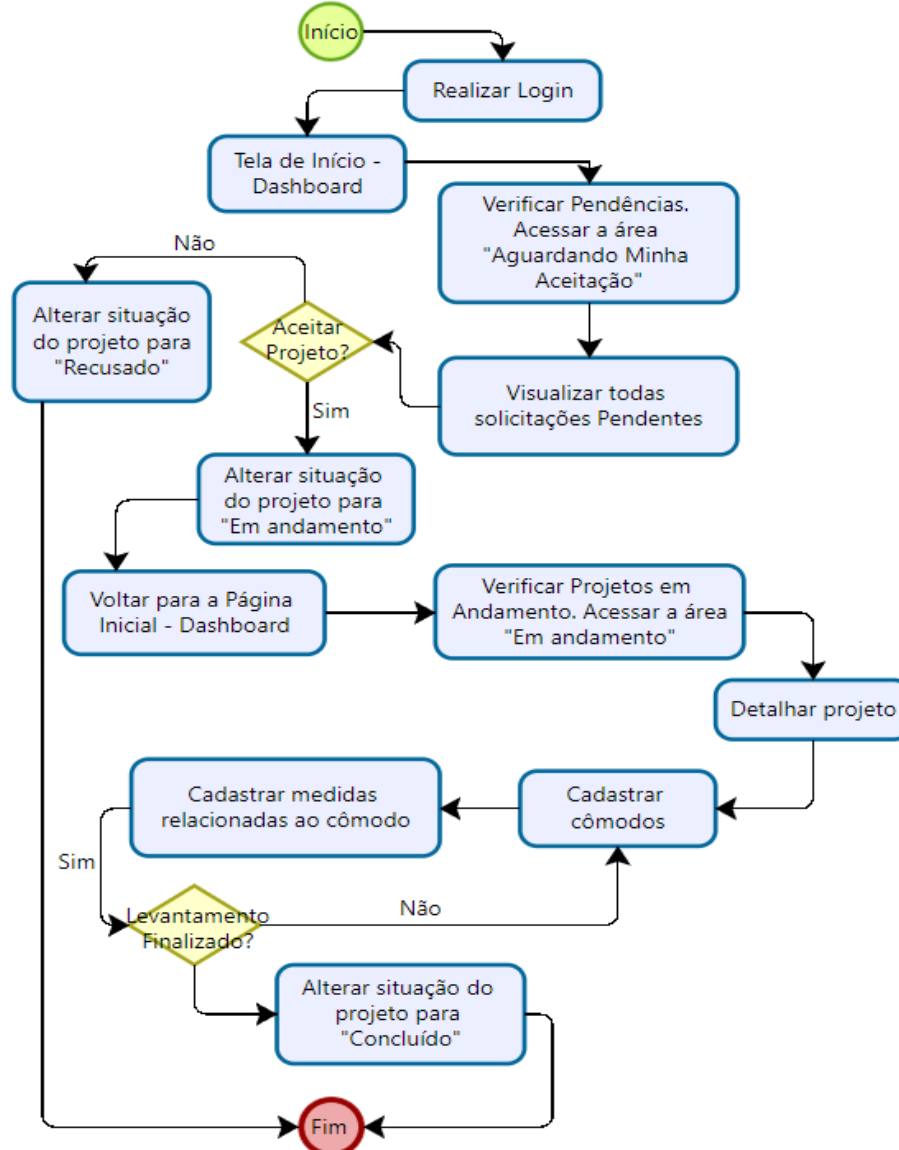


Fonte: acervo do autor (2022).

Conforme pode ser verificado, primeiramente, caso ainda não possua, é necessário que o usuário faça seu cadastro e, logo em seguida, realize seu login, onde o mesmo vai ser direcionado para a página inicial (*dashboard*). No Menu, deve acessar a opção “Meu Perfil”. Ao acessar o perfil o usuário poderá optar em trabalhar com projetos terceirizados. Caso ele opte por esse tipo de serviço, será conduzido ao cadastro da área de atuação. Ao acessar a opção “Meu Perfil”, o usuário também pode cadastrar *checklist* e as atividades relacionada a mesma.

O diagrama apresentado na Figura 17 representa o passo a passo em relação a funcionalidade de aceitação e execução de projetos.

Figura 17 – Diagrama de Atividade - Aceitação e Execução de Projetos



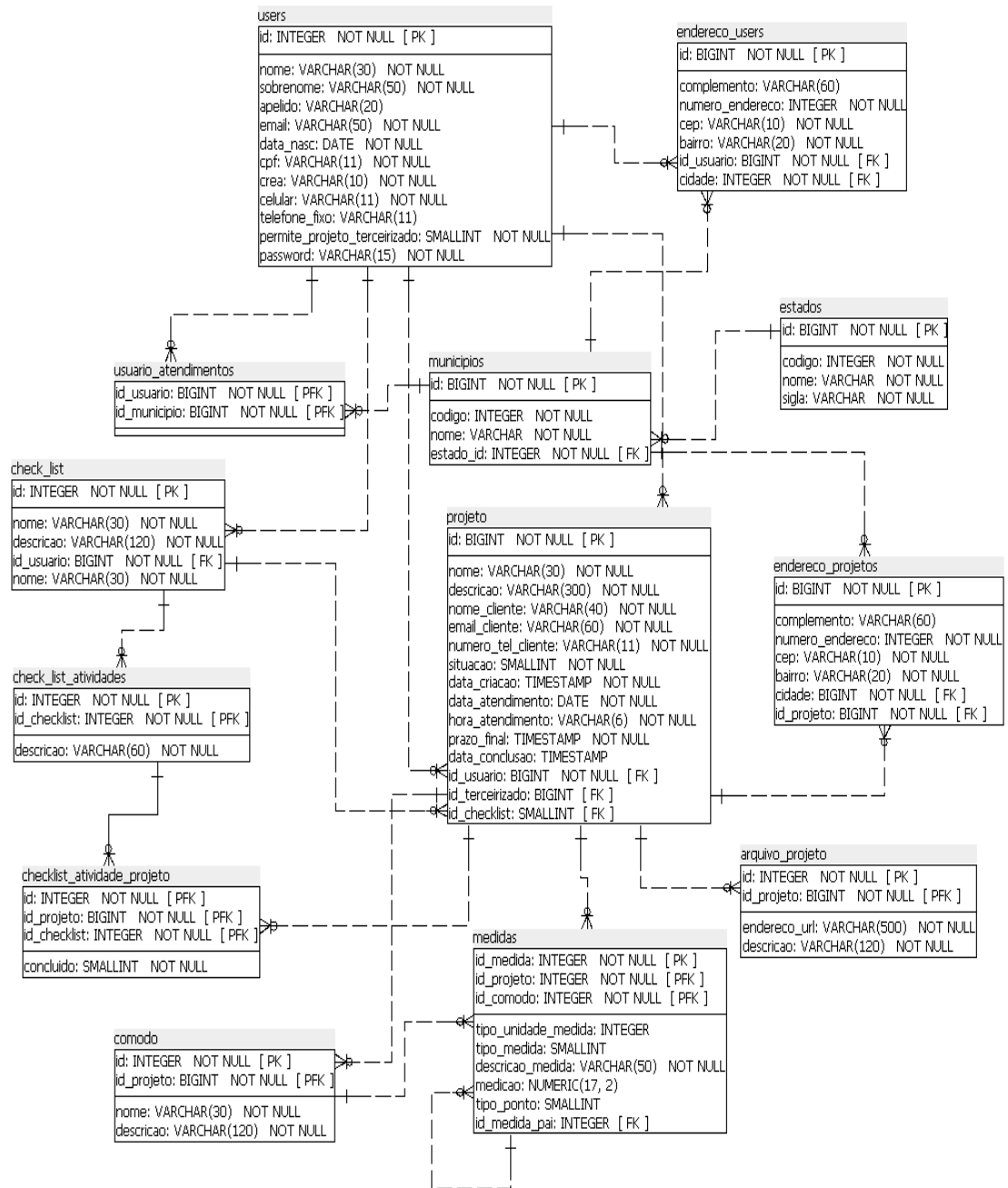
Fonte: acervo do autor (2022).

Para a execução das atividades é necessário primeiramente que o usuário realize o seu *login*. Após realizar a autenticação o usuário poderá visualizar os projetos que estão em andamento, e também poderá verificar os projetos que estão com a situação “Aguardando Minha Aceitação”. Essas opções podem ser acessadas através do *dashboard*. Ao verificar as solicitações pendentes o profissional poderá aceitar ou recusar participar do projeto. Ao recusar o sistema deverá alterar a situação do projeto para “Cancelado”. Ao aceitar o Projeto o sistema deverá alterar a situação do projeto para “Em andamento”.

Para realizar o cadastro de medidas é necessário que o usuário acesse o *dashboard*, selecionando a opção “Em andamento”. Neste caso, serão listados todos os projetos em

andamento. Para que seja realizado o cadastro de medidas é necessário detalhar o projeto, ao detalhar o projeto o usuário poderá cadastrar cômodos e ao detalhar o cômodo poderá realizar o cadastro de medidas. Após ser finalizado o cadastro de medidas o usuário deverá concluir o projeto. Desta forma, o sistema deverá trocar a situação do projeto para “Concluído”. Como parte do processo de desenvolvimento do protótipo foi elaborado a modelagem do banco de dados. Optou-se por desenvolver a modelagem física, contendo todos os relacionamentos e tabelas, sendo que esta modelagem é representada na Figura 18.

Figura 18 – Modelagem Física - Banco de Dados

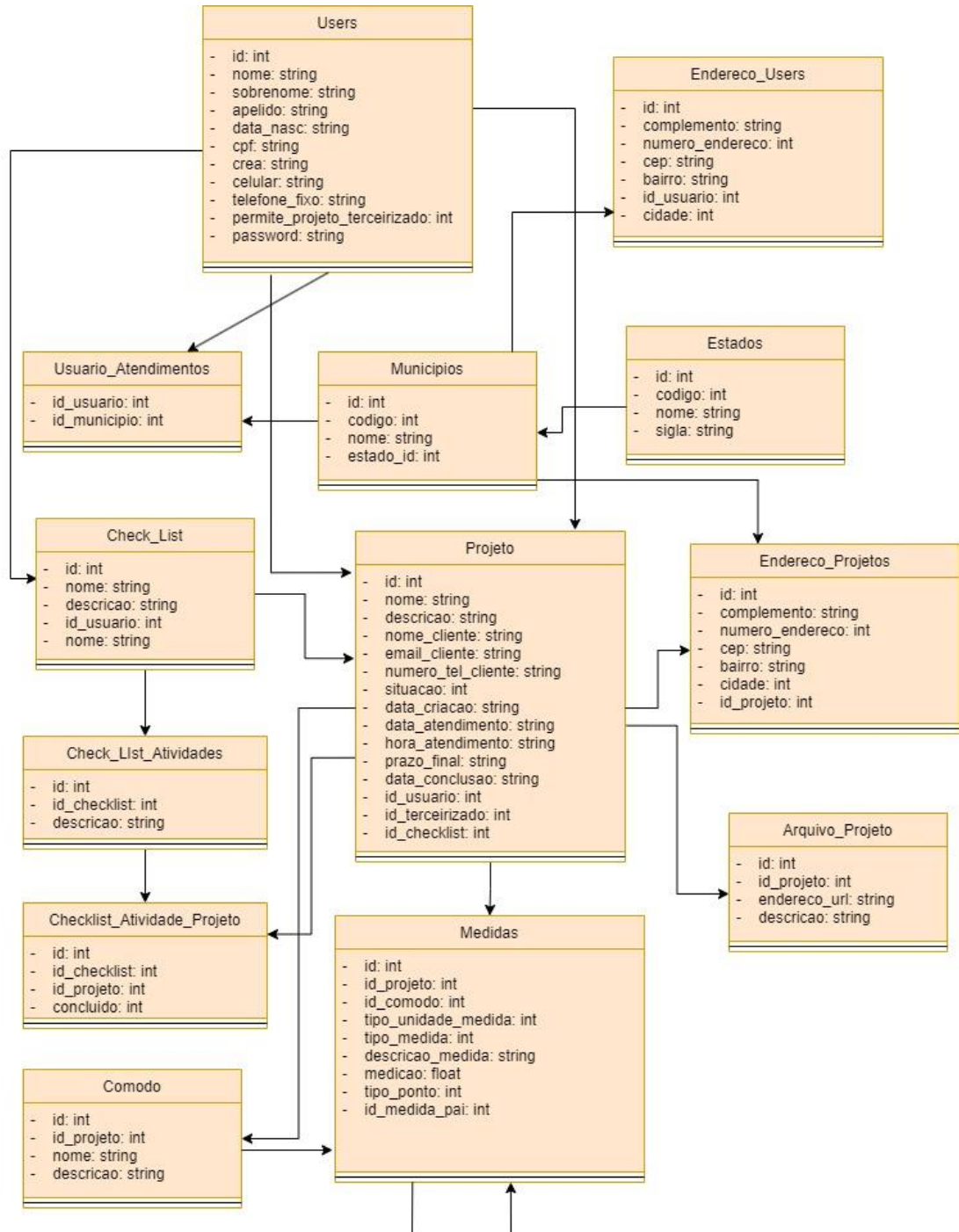


Fonte: acervo do autor (2022).

A modelagem criada para atender a necessidade do projeto possui treze tabelas, sendo elas: usuário, endereço do usuário, estados, municípios, checklist, atividades da checklist, cômodos, medidas, projeto, arquivos do projeto e endereço do projeto. Todas as tabelas possuem chaves primárias, e tem-se duas tabelas com relacionamento N para N, sendo tabelas associativas.

Para o desenvolvimento do projeto foi utilizado o padrão MVC, portanto utiliza-se a programação orientada a objetos. Devido a isso, foi realizado o desenvolvimento do diagrama de classe, conforme apresentado na Figura 19.

Figura 19 – Diagrama de Classe



Fonte: acervo do autor (2022).

Na próxima seção serão apresentadas todas as informações em relação a implementação do protótipo, também serão demonstrados e detalhados os recursos desenvolvidos.

4.2 IMPLEMENTAÇÃO

Nesta seção serão listadas as ferramentas e tecnologias utilizadas para o desenvolvimento do protótipo, e também, serão apresentadas as rotinas implementadas, atendendo os requisitos funcionais do projeto.

4.2.1 Ferramentas e Técnicas Utilizadas

Para a etapa de análise (apresentada anteriormente na seção 4.1) foram utilizadas algumas ferramentas. Uma delas foi o *Draw.io*, que possibilitou a criação do diagrama de caso de uso e do diagrama de classes. Outra ferramenta foi o *SQL Power Architect*, utilizado para a construção da entidade-relacionamento, facilitando a construção do banco de dados. Por fim, para a construção dos diagramas de atividade foi utilizado o *Bizagi Modeler*.

No que se refere ao desenvolvimento das funcionalidades, estas foram criadas a partir de tecnologias web, sendo elas: HTML, CSS, Bootstrap, JavaScript e PHP.

Para a construção básica das páginas e de formulários foi utilizado a versão 5 do HTML. É também conhecido como uma linguagem de marcação de texto *front-end*, a base da parte visual do protótipo. O HTML5 é compatível com a maioria dos navegadores, a implementação da marcação é relativamente simples.

Para deixar a página com um melhor aspecto visual foi realizada a estilização utilizando a versão 3 do CSS. Essa tecnologia permite que elementos HTML sejam personalizados, deixando as telas mais agradáveis para o usuário, contribuindo com a usabilidade do mesmo. Para facilitar a construção das telas foi utilizado o Bootstrap, um *framework* que possui estilizações prontas para serem utilizadas. Além disso, com a utilização desta biblioteca deixa a aplicação responsiva.

Para realizar a criação de elementos ou até mesmo disparar tal comportamento, foi utilizado o *JavaScript*, no qual é executado no navegador do cliente. Sendo assim, quando o usuário executar uma ação desejada elementos podem ser manipulados em tempo real, fazendo

com que campos sejam desbloqueados, ou até mesmo desbloqueados, dependendo da ação do usuário e da regra de negócio existente na rotina que está sendo acessada.

Junto com o *JavaScript* foi implementada a biblioteca jQuery, que é voltada para a manipulação de dados do *JavaScript*. Em algumas rotinas foi utilizado o jQuery para fazer requisições Ajax, permitindo a consulta, inserção, alteração, e remoção de informações do banco de dados, sem a necessidade de atualizar a tela. Com a utilização do jQuery torna-se mais simples a implementação do *JavaScript*, permitindo que o código fonte fique menor.

Em alguns cadastros é necessário realizar requisições para buscar um determinado dado do banco de dados. Essas requisições foram feitas utilizando AJAX. Todas as requisições do sistema não ultrapassam o tempo de 20 (vinte) segundos.

Para realizar a construção do protótipo, partindo para a programação no lado do servidor, foi utilizado o PHP, na versão 7.3.20. No decorrer do projeto foi utilizada essa linguagem juntamente com o *framework* Laravel, facilitando o processo de desenvolvimento.

O framework Laravel permite a criação de rotas, sendo possível definir a rota desejada da melhor maneira possível. A partir do momento que é definida a rota, é especificado o *Controller* que será responsável pelos processamentos necessários.

Além disso, a manipulação do banco de dados torna-se mais simples. Tabelas podem ser criadas através de *migrations*. Sendo que nessas *migrations* é definido o nome da tabela e suas colunas, com sua respectiva tipagem de dado. Além disso, permite a inserção de vários registros através das *Seeders*. Desta forma, com apenas um comando todas as tabelas são criadas, e com outro comando as tabelas são manipuladas a partir da *Seeders*. Ao criar uma rota é possível proteger a mesma, para que ela seja acessada apenas por usuários que estejam logados.

Para o armazenamento dos dados foi utilizado o PostgreSQL, sendo um banco de dados relacional, muito utilizado em grandes organizações e por diversos desenvolvedores. É um banco de dados que disponibiliza inúmeros recursos, permitindo atender a necessidade de qualquer negócio. Além disso é totalmente gratuito.

Para o desenvolvimento do protótipo foi utilizado a IDE do *Visual Studio Code*, que oferece diversas extensões que facilitam o desenvolvimento e a organização do código-fonte. Além disso permite realizar o *commit* para o GitHub de forma simples.

Em algumas telas de inclusão foi realizada a validação ao confirmar o cadastro. Portanto, caso um campo obrigatório não tenha sido preenchido, será disparada uma mensagem

ao usuário, informando que tal campo não foi preenchido. Essa validação foi realizada utilizando o *SweetAlert*.

Ao realizar o login no protótipo toda a informação disponibilizada deverá permanecer ao arquiteto logado. Quando o arquiteto permanecer por mais de uma hora sem alterar nada no sistema o mesmo deverá ser desconectado, sendo assim o arquiteto terá que fazer o login novamente.

4.2.2 Utilização e Funcionamento

Conforme citado anteriormente, o protótipo foi desenvolvido para realizar o levantamento arquitetônico de uma determinada obra. No entanto, possui dois atores principais, porém, ambos usuários possuem o mesmo nível de acesso, o que difere são algumas funcionalidades e recursos ao trabalhar em algum projeto.

A diferença entre as funcionalidades e recursos do protótipo podem ser observadas através dos projetos terceirizados, onde o profissional terceirizado possui acesso à funcionalidades diferentes das funcionalidades disponíveis para o arquiteto que criou o projeto. Para ter acesso às funcionalidades do protótipo é necessário realizar um cadastro (conforme a Figura 20), através da opção "Registre-se" na área de Login.

Figura 20 – Cadastro de Usuário (RF 01)

Registre-se

Primeiro nome Sobrenome

Apelido E-mail Data de Nascimento

CPF Senha Confirmação de Senha

CREA Celular Telefone Fixo

Estado Cidade

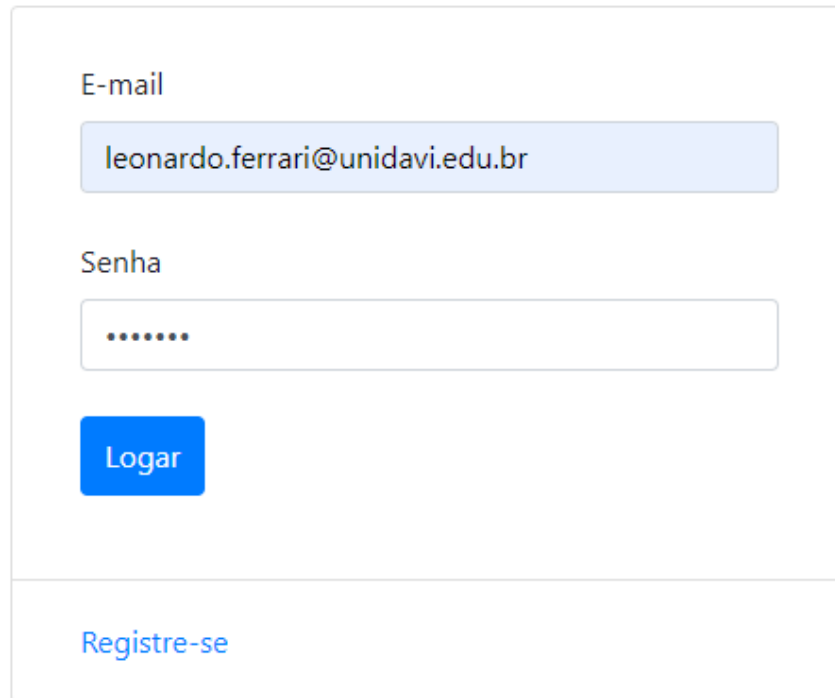
Bairro CEP Número

Complemento

Fonte: acervo do autor (2022).

Para a realização do cadastro é necessário informar os campos que requerem o nome do usuário, seu apelido (opcional), e-mail, data de nascimento, CPF, número do registro no CREA, celular, telefone fixo (opcional), Estado, Cidade, Bairro, CEP, número e Complemento (opcional). O campo e-mail exige uma informação válida. Para o cadastramento da senha, o protótipo exige que seja informado no mínimo 6 (seis) caracteres. Por padrão todo arquiteto que realizará o cadastro irá poder receber solicitações para trabalhar em um projeto.

Após ser efetuado o cadastro o arquiteto pode realizar o login, tendo acesso às funcionalidades disponíveis no protótipo. O Login é realizado através da página inicial do sistema, informando apenas o e-mail e senha, conforme demonstrado na Figura 21.

Figura 21 - Login (RF 02)

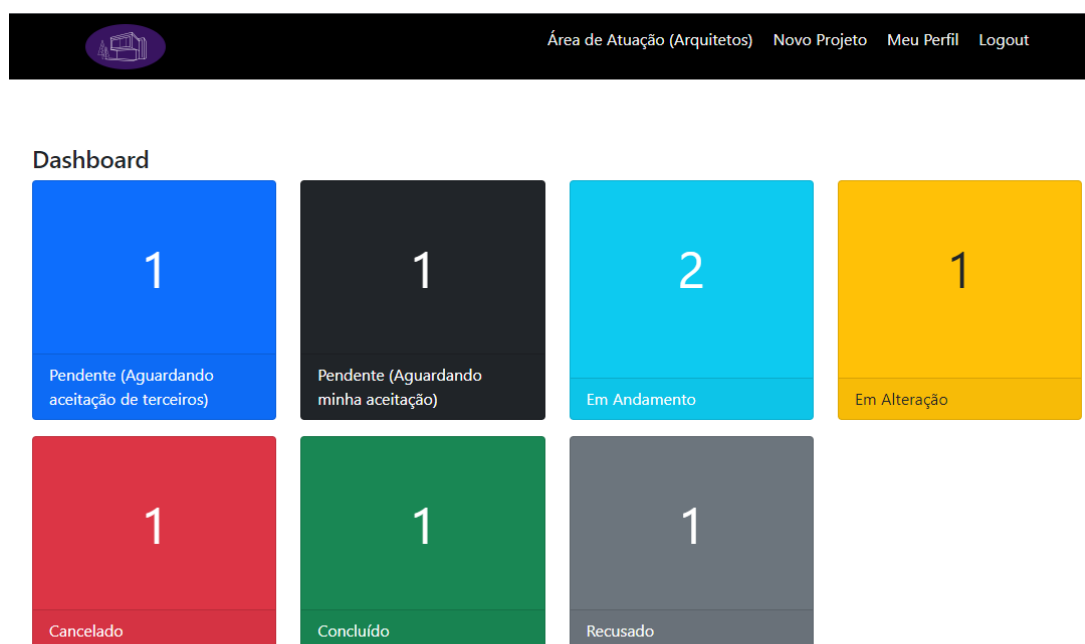
The image shows a login form with the following elements:

- E-mail:** A text input field containing the email address `leonardo.ferrari@unidavi.edu.br`.
- Senha:** A password input field with seven dots representing masked characters.
- Logar:** A blue button with the text "Logar" in white.
- Registre-se:** A blue text link located below the login button.

Fonte: acervo do autor (2022).

As informações preenchidas nos campos da área de Login são validadas no banco de dados. Desta forma, é permitido o acesso apenas para usuários que possuem cadastro com as informações preenchidas. Caso as informações inseridas sejam válidas o arquiteto será direcionado para o *dashboard* do protótipo, onde terá as informações do número de projetos para cada situação, conforme Figura 22.

Figura 22 – Tela Inicial - Dashboard (RF 05)



Fonte: acervo do autor (2022).

Nessa tela inicial é possível visualizar na parte superior o Menu do protótipo, que será o mesmo durante toda ação executada pelo usuário. O Menu disponibiliza acesso aos seguintes recursos: Área de Atuação (Arquitetos), Novo Projeto, Meu Perfil e *Logout*.

Através do *dashboard* é possível acessar os projetos de acordo com a situação atual, verificando também a quantidade de projetos em cada situação. O arquiteto pode visualizar os projetos que ele cadastrou e que está pendente, aguardando aceitação de terceiro, e também, que estão pendentes aguardando a aceitação do mesmo (essa opção só ficará disponível de acordo com a configuração do usuário, através da opção "Terceirizar Projeto", que pode ser acessada pela opção "Meu Perfil" conforme Figura 23). Além disso, pode visualizar os projetos com as seguintes situações: Em Andamento, Em Alteração, Cancelado, Concluído e Recusado.

Figura 23 – Alteração do Cadastro de Usuário (RF 01)

The screenshot shows a web application interface for changing user registration. At the top, there is a navigation bar with a logo on the left and links for 'Área de Atuação (Arquitetos)', 'Novo Projeto', 'Meu Perfil', and 'Logout'. Below this, a secondary navigation bar contains 'CheckList's' and 'Área de Atuação'. The main content area is titled 'Altere seu Cadastro' and contains the following form fields:

- Primeiro nome:** Leonardo
- Sobrenome:** Ferrari
- Apelido:** Léo
- E-mail:** leonardo.ferrari@unidavi.edu.br
- Data de Nascimento:** 05/04/1999
- CPF:** 999.999.999-99
- CREA:** 250.678
- Celular:** 47 - 99999-9999
- Telefone Fixo:** 47 - 99999-9999
- Estado:** Santa Catarina
- Cidade:** Rio do Sul
- Bairro:** Pamplona
- CEP:** 89164-322
- Número:** 210
- Complemento:** Em frente a casa X
- Terceirizar projeto:**

At the bottom of the form, there is a blue button labeled 'Alterar'.

Fonte: acervo do autor (2022).

Através da funcionalidade "Meu Perfil" o arquiteto pode definir os locais em que o profissional estará disponível para exercer o trabalho. Para definir os locais de trabalho o usuário deverá acessar a opção "Área de Atuação", que irá direcionar o arquiteto para a área de consulta e cadastro. A tela de cadastro possui uma lista de Estados. Ao selecionar um estado serão listados todos os municípios relacionados ao Estado selecionado. Ao listar os municípios o usuário pode escolher um ou mais municípios, permitindo que o mesmo selecione todos os municípios através da opção "Selecionar Todos", conforme demonstrado na Figura 24.

Figura 24 – Cadastro área de Atuação (RF 06)

Área de Atuação

Estado

Acre

Selecionar Todos

Acrelândia

Assis Brasil

Brasília

Bujari

Capixaba

Cruzeiro do Sul

Epitaciolândia

Feijó

Jordão

Mâncio Lima

Manoel Urbano

Marechal Thaumaturgo

Plácido de Castro

Porto Walter

Rio Branco

Rodrigues Alves

Santa Rosa do Purus

Senador Guiomard

Sena Madureira

Tarauacá

Xapuri

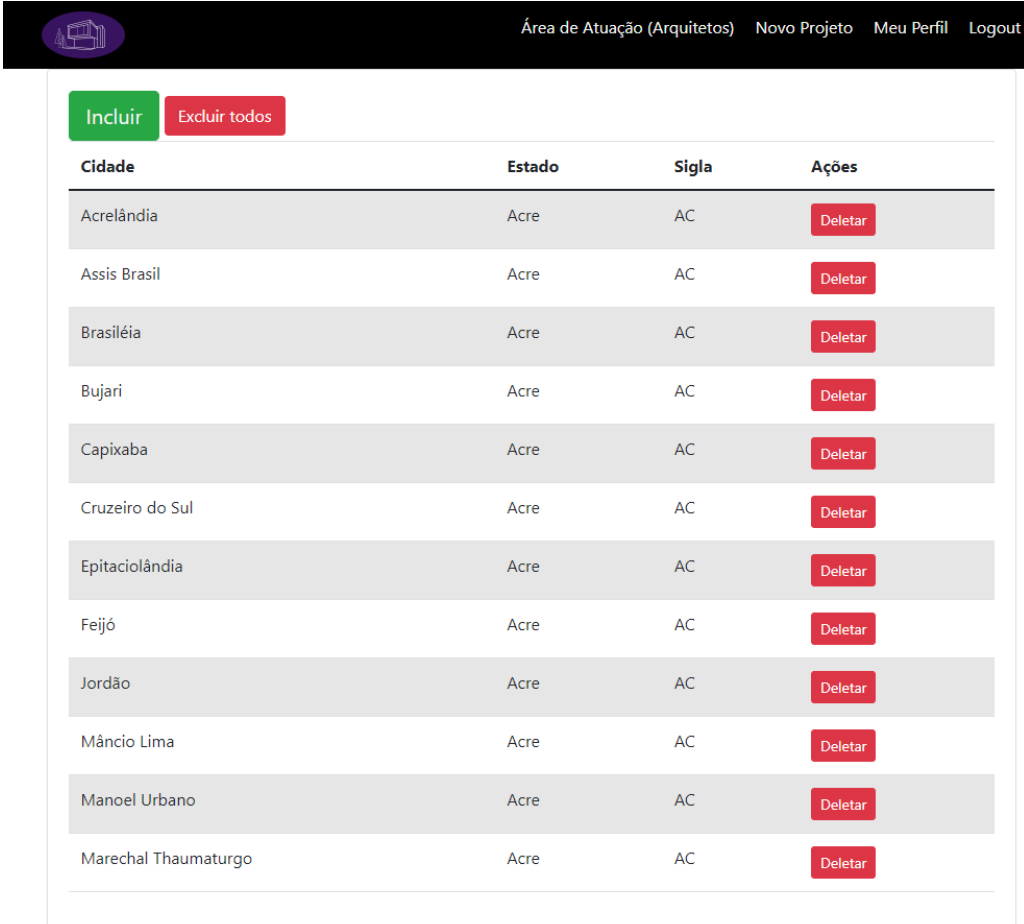
Porto Acre

Cadastrar

Fonte: acervo do autor (2022).

Após realização do cadastro da área de atuação o arquiteto pode consultar os Municípios e Estados que o mesmo atende. A consulta possui somente a ação para deletar o registro. Permite também que o arquiteto exclua todos os municípios. Conforme demonstrado na Figura 25.

Figura 25 – Consulta área de atuação (RF 06)



Cidade	Estado	Sigla	Ações
Acrelândia	Acre	AC	Deletar
Assis Brasil	Acre	AC	Deletar
Brasiléia	Acre	AC	Deletar
Bujari	Acre	AC	Deletar
Capixaba	Acre	AC	Deletar
Cruzeiro do Sul	Acre	AC	Deletar
Epitaciolândia	Acre	AC	Deletar
Feijó	Acre	AC	Deletar
Jordão	Acre	AC	Deletar
Mâncio Lima	Acre	AC	Deletar
Manoel Urbano	Acre	AC	Deletar
Marechal Thaumaturgo	Acre	AC	Deletar

Fonte: acervo do autor (2022).

Através da funcionalidade "Meu Perfil" o arquiteto pode cadastrar suas *Checklist's*, para que possam ser relacionadas a um projeto. O arquiteto poderá cadastrar mais de uma *checklist*, conforme Figura 26.

Figura 26 – Cadastro de Checklist (RF 07)

Fonte: acervo do autor (2022).

Para realizar o cadastro da *checklist* o arquiteto deve informar o Nome e uma Descrição relacionada a *checklist*. Durante o cadastro o arquiteto pode vincular várias atividades a mesma. Após ser efetuado o cadastro, será exibida uma mensagem ao usuário, informando que o cadastro foi realizado com sucesso. Ao clicar no botão "Ok" o arquiteto será direcionado à consulta de *checklist's*. Conforme demonstrado na Figura 27.

Figura 27 – Consulta de Checklist (RF 07)

Código	Nome	Descrição	Ações
1	Checklist 1	Teste	Deletar Alterar Atividades

Fonte: acervo do autor (2022).

A consulta dispõe das ações: Deletar, Alterar e Atividades. Ao clicar no botão "Atividades" o arquiteto é direcionado para consulta de atividades. Essa consulta mostra as atividades relacionadas a *checklist*, como demonstrado na Figura 28.

Figura 28 – Consulta de Atividades da Checklist (RF 08)

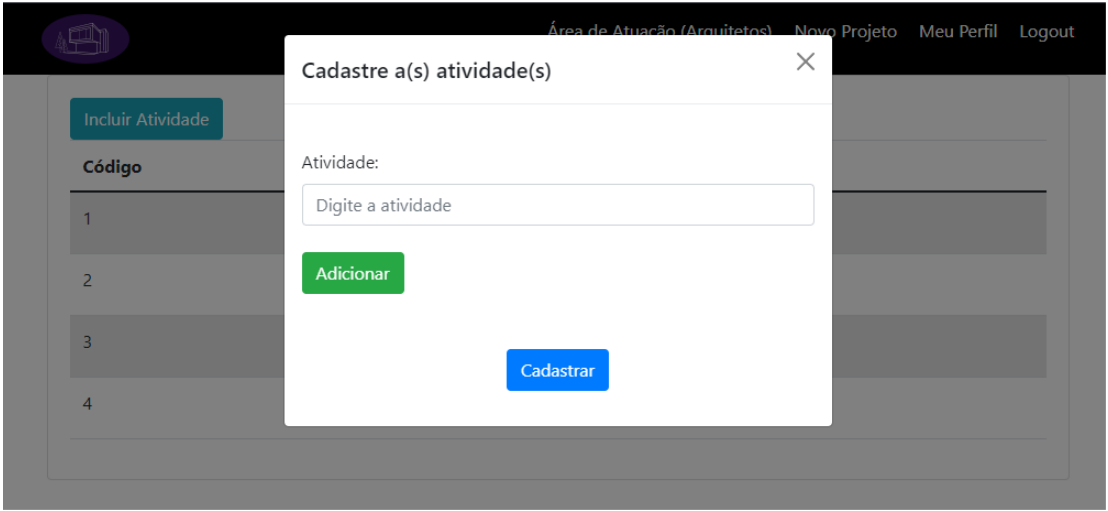


Código	Descrição	Ações
1	Atividade 1	Alterar Deletar
2	Atividade 2	Alterar Deletar
3	Atividade 4	Alterar Deletar
4	Atividade 5	Alterar Deletar

Fonte: acervo do autor (2022).

Ao ser exibida a consulta são filtradas as atividades relacionadas a *checklist* desejada. A consulta dispõe das ações para Alterar e Deletar a atividade. Ao incluir a Atividade deve ser informada a descrição. O arquiteto pode ainda cadastrar diversas atividades, sendo que as atividades cadastradas devem estar relacionadas a *checklist* desejada. O cadastro das atividades é apresentado na Figura 29.

Figura 29 – Cadastro de Atividades da Checklist (RF 08)



Fonte: acervo do autor (2022).

Com as *checklist's* cadastradas e suas atividades relacionadas, o arquiteto pode selecionar a mesma ao cadastrar um projeto. Conforme Figura 30.

Figura 30 – Cadastro de Projeto (RF 09)

O formulário, intitulado "Cadastre seu Projeto", contém os seguintes campos:

- Nome:** Campo de texto com o placeholder "Digite o Nome do Projeto".
- Descrição:** Campo de texto grande com o placeholder "Descrição".
- Nome Cliente:** Campo de texto com o placeholder "Digite o Nome do Cliente".
- E-mail Cliente:** Campo de texto com o placeholder "Digite o e-mail do Cliente".
- Número Celular:** Campo de texto com o placeholder "Digite seu número".
- Data Atendimento:** Campo de data com o formato "dd/mm/aaaa" e ícone de calendário.
- Hora:** Campo de hora com o formato "--:--" e ícone de relógio.
- Prazo Final:** Campo de data com o formato "dd/mm/aaaa" e ícone de calendário.
- CheckList:** Campo de seleção com o placeholder "Selecione...".
- Estado:** Campo de seleção com o placeholder "Selecione um Estado" e ícone de seta para baixo.
- Cidade:** Campo de seleção com o placeholder "Selecione um Município" e ícone de seta para baixo.
- Bairro:** Campo de texto com o placeholder "Bairro".
- CEP:** Campo de texto com o placeholder "CEP".
- Número:** Campo de texto com o placeholder "Número".
- Complemento:** Campo de texto com o placeholder "Complemento".
- Projeto Terceirizado:** Campo de seleção com o placeholder "Projeto Terceirizado:" e uma caixa de seleção vazia.

Um botão azul "Cadastrar" está localizado na base do formulário.

Fonte: acervo do autor (2022).

Ao cadastrar um projeto deve ser atribuído um nome ao mesmo. O arquiteto deve descrever um pouco sobre o projeto, informar o nome, e-mail, e número de celular do cliente. Além disso, deve ser informada a data e hora do atendimento, bem como o prazo final. O arquiteto pode escolher uma *checklist* para ser relacionada ao projeto, e assim garantir que todas as atividades sejam executadas. Deve ainda ser informado o endereço de onde será realizado o

levantamento arquitetônico. Assim que preenchido o Estado e o Município a opção "Projeto Terceirizado" fica habilitada. Quando marcada, é possível escolher um arquiteto dentre os que atuam no município informado, sendo este vinculado como terceirizado.

Quando cadastrado um projeto terceirizado o projeto fica com situação Pendente. Sendo assim, quando o arquiteto terceirizado acessar o sistema, o mesmo poderá consultar os projetos pendentes disponíveis no *dashboard* através da opção "Pendente – Aguardando minha aceitação". Ao acessar a consulta o arquiteto terceirizado poderá visualizar as informações do projeto, podendo aceitar ou recusar. Além disso, ele poderá visualizar mais informações sobre o arquiteto que cadastrou o projeto, pela opção "Info. Arquiteto". Também irá possuir a ação "Info. Projeto", para que o arquiteto possa verificar mais informações sobre o projeto, conforme pode ser verificado pela Figura 31.

Figura 31 – Consulta de Projeto - Aguardando minha aceitação (RF 10)



Nome	Cliente	E-mail	Data	Hora	Prazo Final	Ações
teste	Leo	teste@teste	15/09/2022	15:00	20/09/2022	Aceitar Recusar Info. Arquiteto Info. Projeto

Fonte: acervo do autor (2022).

Ao aceitar o projeto a situação do projeto é alterada para "Em andamento", podendo ser acessado pelo *dashboard*.

O primeiro projeto que aparece na listagem é um projeto no qual foi aberto pelo arquiteto que está logado. Já o segundo é um projeto no qual o mesmo está terceirizado. Ao recusar o projeto a situação será alterada para "Recusado". Ao consultar os projetos recusados e cancelados existe uma ação para reabrir o projeto. Neste caso, o profissional terceirizado deve ser desvinculado do projeto, de forma automática, retornando o projeto para a situação de: "Em andamento", como demonstrado na Figura 32.

Figura 32 – Consulta de Projeto - Recusado (RF 10)



Nome	Cliente	E-mail	Data	Hora	Ações
teste	Leo	teste@teste	15/09/2022	15:00	Reabrir

Fonte: acervo do autor (2022).

Para os projetos "Em andamento" o protótipo disponibiliza as seguintes ações: Cômodos, Atividades, Informações do Arquiteto, Informações do projeto, Arquivos, Concluir, Alterar e Cancelar, conforme demonstrado na Figura 33.

Figura 33 – Consulta de Projeto - Em andamento (RF 10)



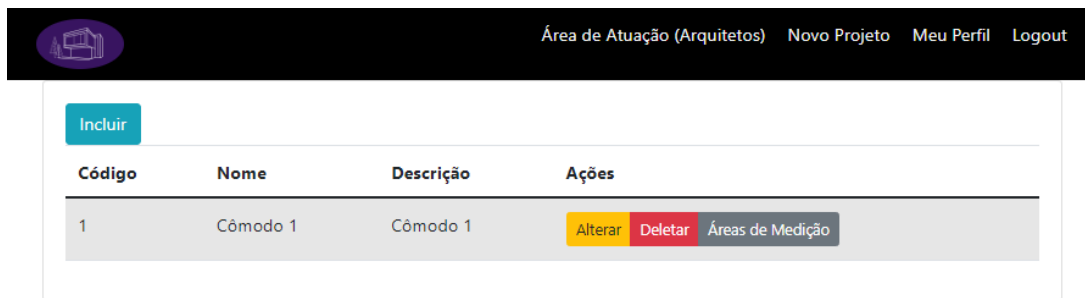
Nome	Cliente	E-mail	Data	Hora	Ações
teste	Leo	teste@teste	15/09/2022	15:00	Cômodos Atividades Info. Arquiteto Info. Projeto Concluir Arquivos
teste	Leo	teste@teste	15/09/2022	15:00	Alterar Cômodos Atividades Info. Arquiteto Info. Projeto Cancelar Arquivos

Fonte: acervo do autor (2022).

Quando o projeto for terceirizado, somente o arquiteto terceirizado pode concluir o projeto, através da ação "Concluir". O arquiteto que incluiu o projeto pode cancelar o projeto através da ação "Cancelar". Quando o projeto não for terceirizado o arquiteto pode concluir e cancelar em qualquer momento.

A ação de cômodos será disponibilizada tanto para o arquiteto que abriu o projeto, quanto para o arquiteto terceirizado. A ação de alterar aparece somente para o profissional que incluiu o projeto. Portanto, o arquiteto terceirizado não consegue alterar informações pertinentes ao projeto. Ao acessar a ação de cômodos é apresentada a consulta conforme Figura 34.

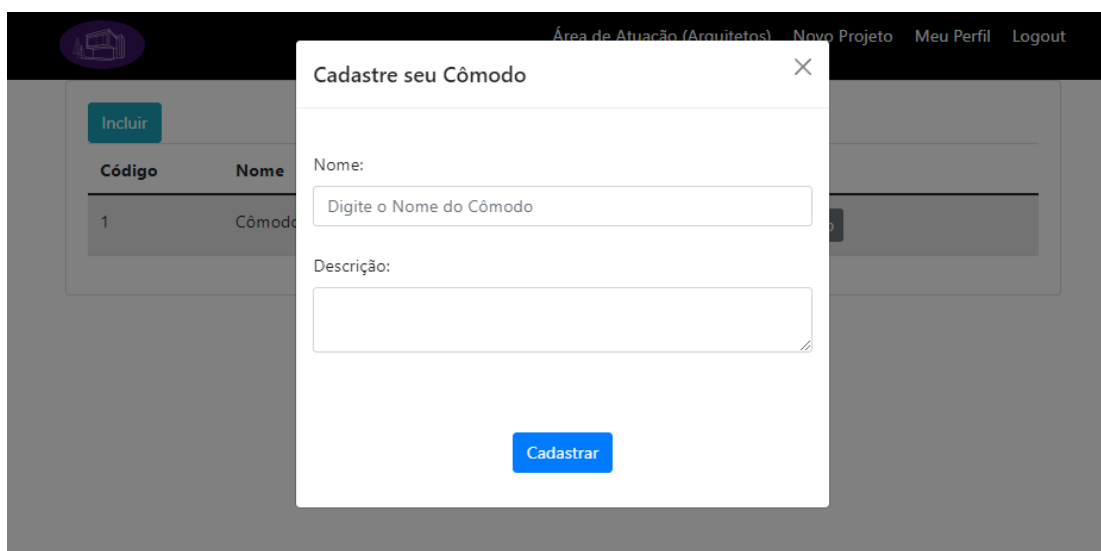
Figura 34 – Consulta de Cômodos (RF 13)



Fonte: acervo do autor (2022).

Ao consultar os cômodos são exibidos todos os cômodos relacionados ao projeto selecionado. Para cada cômodo é possível alterar a informação, deletar e consultar áreas de medições. Na própria consulta é disponibilizada a ação para incluir um novo cômodo. Ao incluir um cômodo é disponibilizada a tela conforme Figura 35.

Figura 35 – Cadastro de Cômodos (RF 12)



Fonte: acervo do autor (2022).

Para o cadastro de cômodo deve ser informado o nome do cômodo e uma breve descrição. Assim que incluído um cômodo o arquiteto será direcionado para a consulta, na qual são apresentados todos os cômodos relacionados ao projeto.

Assim que realizado o cadastro do cômodo o arquiteto consegue incluir as Áreas de Medição referente ao mesmo. Ao clicar na ação "Áreas de Medição" é disponibilizada a consulta listando todas as áreas de medição relacionada ao cômodo, conforme demonstrado na Figura 36.

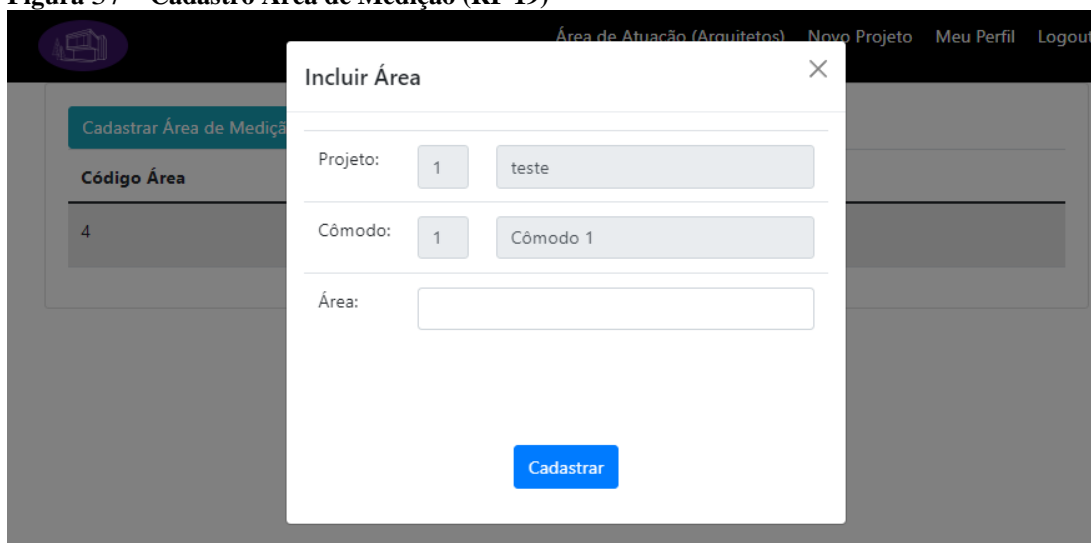
Figura 36 – Consulta Área de Medição (RF 19)



Fonte: acervo do autor (2022).

Ao abrir a consulta é possível cadastrar a área medição através da ação "Cadastrar Área de Medição". Ao acessar a tela de cadastro o campo de projeto e cômodo será carregado automaticamente, além disso, é necessário informar um nome para a área que está sendo cadastrada, conforme demonstrado na Figura 37.

Figura 37 – Cadastro Área de Medição (RF 19)



Fonte: acervo do autor (2022).

Ao incluir a área de medição o arquiteto será direcionado para a consulta novamente. Na consulta o arquiteto pode alterar a informação de uma área de medição e até mesmo deletá-la. Além disso, possui a ação de "Incluir Medida", que será relacionada a área de medição desejada, onde as mesmas poderão ser consultadas pela ação de "Medidas". Ao clicar na ação de "Incluir Medida" aparecer é apresentada a tela demonstrada na Figura 38.

Figura 38 – Cadastro de Medidas (RF 15)

The image shows a web application interface for recording measurements. A modal window titled "Incluir Medida" is open, allowing a user to add a new measurement. The form includes the following fields:

- Medida Pai:** A dropdown menu with the value "4" and a text input field containing "Teste 2".
- Projeto:** A dropdown menu with the value "1" and a text input field containing "teste".
- Cômodo:** A dropdown menu with the value "1" and a text input field containing "Cômodo 1".
- Tipo Unidade Medida:** A dropdown menu with the value "Metro (m)".
- Tipo Medida:** A dropdown menu with the value "Altura (Vertical)".
- Tipo Ponto:** A dropdown menu with the value "Selecione...".
- Descrição:** An empty text input field.
- Medição:** An empty text input field.

A blue button labeled "Cadastrar" is positioned at the bottom center of the modal.

Fonte: acervo do autor (2022).

Ao incluir a medida o projeto e cômodo são carregados automaticamente, referenciando a medida ao projeto e cômodo desejado. Além disso, é relacionada a área de medição escolhida. Para o cadastramento da medida deve ser informado o tipo de unidade de medida desejado, o tipo de medida, o tipo de ponto (elétrico ou hidráulico, se houver), a descrição e a sua medição. Ao concluir o cadastro o arquiteto é direcionado para a consulta novamente, conforme demonstrado na Figura 39.

Figura 39 – Consulta de Medidas (RF 16)



Código	Descrição	Und. Medida	Tipo Medida	Tipo Ponto	Descrição	Medição	Ações
7	Teste	Centímetro (cm)	Altura (Vertical)	Não informado	Teste	3.5	Incluir Medida Medidas Alterar Deletar

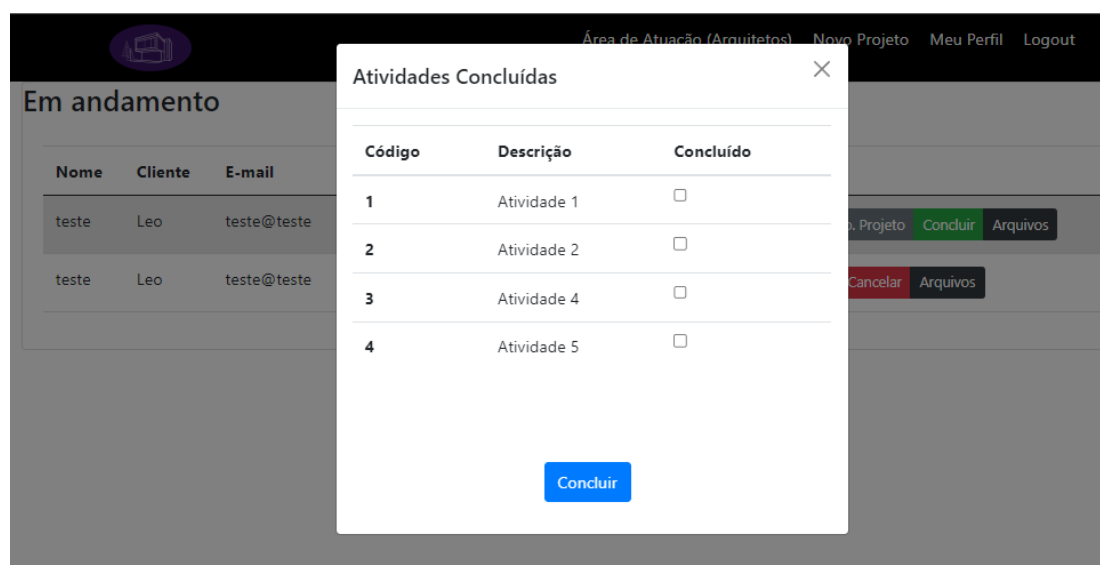
Fonte: acervo do autor (2022).

A consulta de Medidas possui as ações para Incluir Medida, Alterar, Deletar e consultar as medidas relacionadas.

No decorrer do projeto o arquiteto terceirizado ou até mesmo o arquiteto que está trabalhando no próprio projeto pode concluir as atividades relacionadas a *checklist*. Esta ação está disponível na consulta de projetos.

A ação de Atividades aparece para àqueles projetos que possuem *checklist* relacionada. Para os projetos que não são terceirizados, ao abrir a tela de atividades, o arquiteto pode concluir às atividades. Nos casos em que o projeto é terceirizado, somente o arquiteto terceirizado pode concluir as atividades. Conforme demonstrado na Figura 40.

Figura 40 – Tela de Conclusão das Atividades (RF 11)

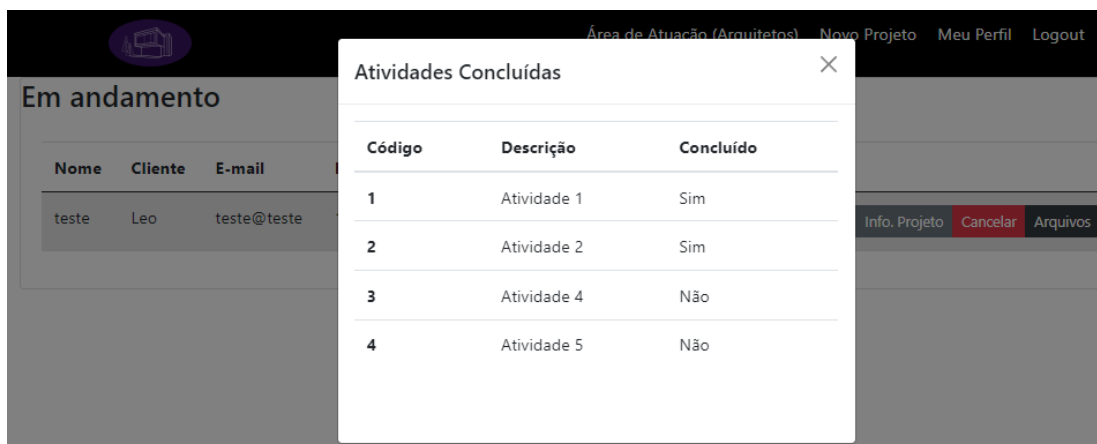


Código	Descrição	Concluído
1	Atividade 1	<input type="checkbox"/>
2	Atividade 2	<input type="checkbox"/>
3	Atividade 4	<input type="checkbox"/>
4	Atividade 5	<input type="checkbox"/>

Fonte: acervo do autor (2022).

Assim que concluída uma atividade ou mais, o arquiteto que incluiu o projeto pode visualizar quais atividades foram concluídas, conforme demonstrado na Figura 41.

Figura 41 – Tela de Conclusão das Atividades (RF 11)

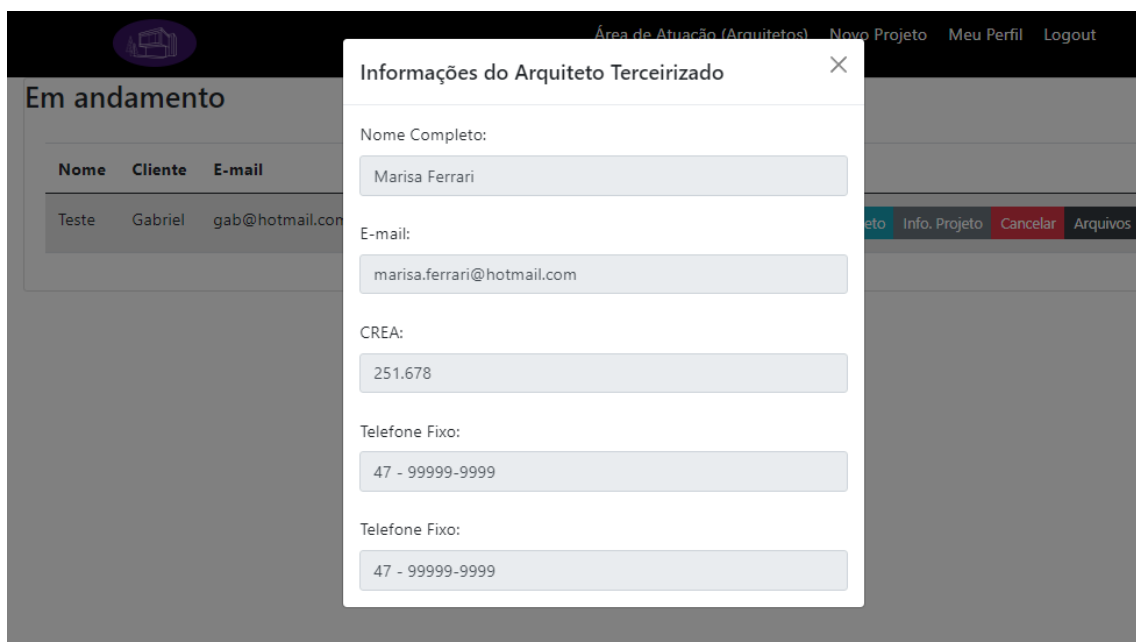


Fonte: acervo do autor (2022).

Quando não houver arquiteto terceirizado e houver *checklist* relacionada, o arquiteto que incluiu o projeto deve concluir as atividades.

A ação "Info. Arquiteto" aparece somente para projetos que são terceirizados. Ao abrir a tela são exibidas as informações do arquiteto terceirizado, conforme demonstrado na Figura 42.

Figura 42 – Informações do Profissional (RF 17)



Fonte: acervo do autor (2022).

A ação "Info. Projeto" é exibida para todos os projetos, permitindo que o arquiteto possa verificar mais informações sobre o projeto, como pode ser visto na Figura 43.

Figura 43 – Informações do Cadastro de Projeto (RF 09)

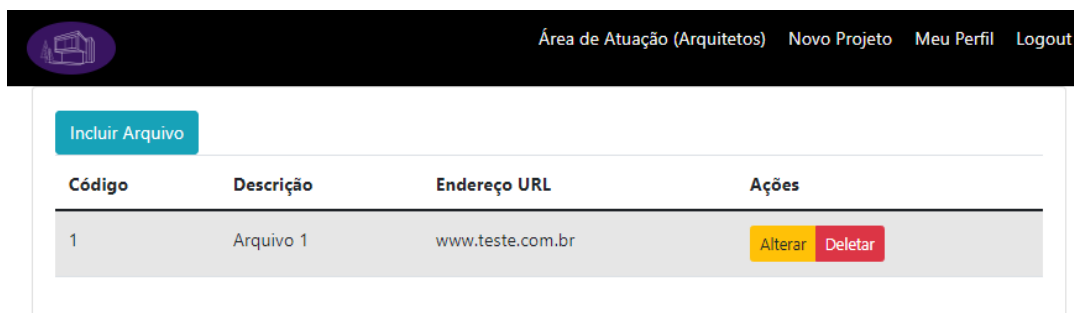
A imagem mostra a interface de usuário para o cadastro de um projeto. No topo, há uma barra de navegação com o nome do usuário 'Área de Atuação (Arquitetos)' e links para 'Novo Projeto', 'Meu Perfil' e 'Logout'. O formulário principal, intitulado 'Informações do Projeto', contém os seguintes campos:

- Nome: Teste
- Descrição: teste
- Nome Cliente: Gabriel
- E-mail Cliente: gab@hotmail.com
- Número Celular: 323
- Data Atendimento: 09/12/2022
- Hora: 22:51
- Prazo Final: 28/11/2022
- Data Criação: 02/11/2022
- Data Conclusão: 05/11/2022
- CheckList: Teste
- Estado: Piauí
- Cidade: Barra D'Alcântara
- Bairro: fdsf
- CEP: 23
- Número: 3
- Complemento: vcx

Fonte: acervo do autor (2022).

Além dessas ações citadas, ao consultar os projetos é possível consultar arquivos relacionados ao projeto, que está disponível tanto para o arquiteto terceirizado quanto para o arquiteto que incluiu o projeto. Ao clicar na ação é exibida a consulta de arquivos, filtrando todos os arquivos relacionado ao projeto, conforme demonstrado na Figura 44.

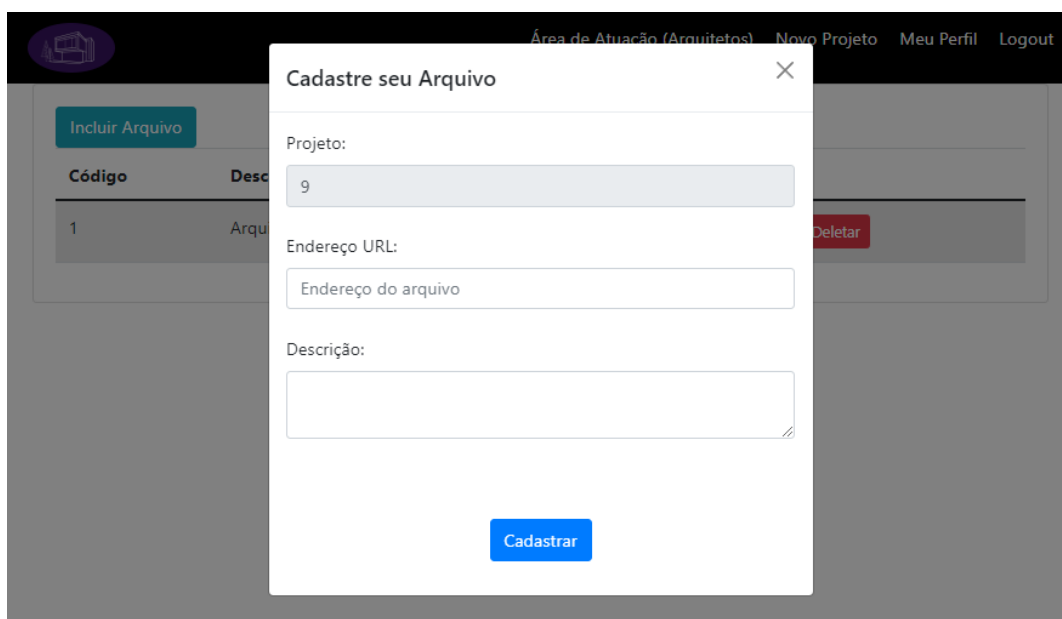
Figura 44 – Consulta de Arquivos do Projeto (RF 18)



Fonte: acervo do autor (2022).

A consulta disponibiliza uma ação para incluir um arquivo relacionado ao projeto. Para cada arquivo cadastrado o usuário pode alterar as informações, e até mesmo deletar o arquivo. Ao incluir o arquivo através da ação "Incluir Arquivo" deve abrir a tela demonstrada na Figura 45.

Figura 45 – Cadastro de Arquivos do Projeto (RF 18)



Fonte: acervo do autor (2022).

Para cadastrar o arquivo deve ser informado o endereço da imagem ou arquivo, ou seja, deverá ser colado o link do endereço. Além disso, o arquiteto deverá informar uma breve descrição do que se trata o arquivo. Assim que o arquivo for cadastrado, o arquiteto será direcionado para a consulta de arquivos.

Por fim, o protótipo dispõe de uma rotina para consultar a área de atuação de todos os arquitetos cadastrados. A consulta tem como objetivo facilitar a busca por arquitetos em uma determinada região, conforme demonstrado na Figura 46.

Figura 46 – Consulta de Atuação por Região (RF 14)

The screenshot shows a web application interface for searching architects by region. At the top, there is a navigation bar with a logo on the left and links for 'Área de Atuação (Arquitetos)', 'Novo Projeto', 'Meu Perfil', and 'Logout'. Below the navigation bar, there is a search form with the following fields:

- Nome:** A text input field with the placeholder text 'Digite o Nome do Arquiteto'.
- Estado:** A dropdown menu with the placeholder text 'Selecione um Estado'.
- Cidade:** A dropdown menu with the placeholder text 'Selecione um Município'.
- Filtrar:** A blue button to execute the search.

Below the search form, there is a table displaying the search results. The table has the following columns: Nome, Celular, E-mail, Município, Estado, Sigla, and Ações. The results show a list of architects named Leonardo Ferrari, each associated with a specific municipality in the state of Acre (AC). Each row includes a '+ Informações' button.

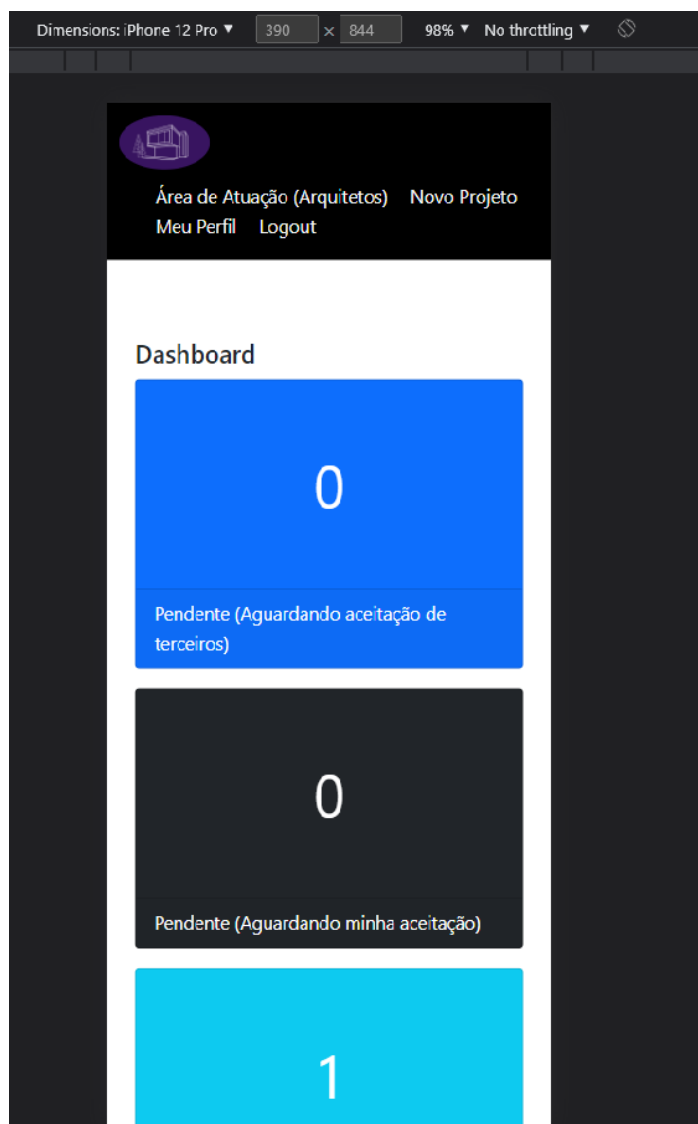
Nome	Celular	E-mail	Município	Estado	Sigla	Ações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Acrelândia	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Assis Brasil	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Brasiléia	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Bujari	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Capixaba	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Cruzeiro do Sul	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Epitaciolândia	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Feijó	Acre	AC	+ Informações
Leonardo Ferrari	47 - 99999-9999	leonardo.ferrari@unidavi.edu.br	Jordão	Acre	AC	+ Informações

Fonte: acervo do autor (2022).

A consulta permite filtrar o arquiteto desejado ou até mesmo arquitetos que trabalham em outras cidades. Ao clicar no botão "+ Informações" deverá ser mostrado o nome completo, e-mail, CREA, telefone fixo e celular.

Conforme citado, o protótipo é totalmente responsivo. Desta forma ao utilizar a aplicação em telas menores ou até mesmo em dispositivos móveis, como demonstrado na Figura 47, acessando o *dashboard* utilizando a tela de um iPhone 12 Pro como exemplo.

Figura 47 – Tela iPhone 12 Pro – Tela Responsiva



Fonte: acervo do autor (2022).

Desta forma, como a aplicação é responsiva a tela será ajustada automaticamente, permitindo que o usuário possa visualizar da melhor forma possível.

5. CONSIDERAÇÕES FINAIS

O trabalho desenvolvido tem como principal característica facilitar a etapa de levantamento de medidas em um projeto arquitetônico. A ferramenta permite que arquitetos realizem o seu cadastro e incluir projetos. Esses projetos podem ser terceirizados, permitindo que outro arquiteto realize o trabalho desejado, facilitando o processo da etapa de levantamento arquitetônico.

As tecnologias adotadas e a linguagem de programação escolhida foram suficientes para atender o objetivo deste protótipo. Para facilitar o desenvolvimento do protótipo foi utilizado o *framework* Laravel na parte do *backend*. O Laravel possui uma estrutura pronta para facilitar o desenvolvimento em PHP. Já no *frontend* foi utilizada a biblioteca do Bootstrap, a qual possui componentes estilizados (com CSS) e classes que podem ser utilizadas para ajustar uma determinada tela. Assim, na criação de telas com HTML, foram aplicadas as classes disponíveis na biblioteca. Para o comportamento das telas, foi utilizado o JavaScript, o qual permite mudar o estado da tela em tempo real. Além do Bootstrap foi utilizada a biblioteca do *SweetAlert*, fazendo com que o usuário receba a notificação se determinada ação foi bem-sucedida. Por fim, foi utilizado o PostgreSQL para manter as informações armazenadas no banco de dados.

Em relação aos objetivos específicos apresentados neste trabalho, o primeiro objetivo proposto neste projeto visava descrever as tecnologias utilizadas no desenvolvimento do protótipo, este foi bem-sucedido através do conteúdo apresentado na revisão da literatura.

O segundo objetivo, consistia em detalhar a análise do protótipo por meio do levantamento de requisitos, da elaboração de diagramas, e da modelagem do banco de dados. Este objetivo foi cumprido com a seção de Análise, onde foram apresentados os requisitos funcionais, não funcionais e as regras de negócio, bem como, onde foi apresentado o diagrama de casos de uso, os diagramas de atividades e o modelo de dados.

O terceiro objetivo visava demonstrar ferramentas que possuem propósito similar ao protótipo proposto. Durante o desenvolvimento deste objetivo foram identificadas três ferramentas com funcionalidades similares, sendo estas demonstradas na seção do estado da arte. As aplicações demonstradas foram: Vobi, Asana e FlowUp.

O quarto e último objetivo visava a construção do protótipo utilizando as tecnologias escolhidas para concluir o projeto proposto. O objetivo foi cumprido e apresentado na seção de implementação, onde foram abordadas as tecnologias e técnicas utilizadas. Além disso foram

demonstradas todas as rotinas desenvolvidas no protótipo, contendo a explicação de cada uma delas.

Com a utilização do protótipo, os arquitetos poderão reduzir custos ao realizar um determinado projeto, pois será possível que parte das tarefas do projeto sejam terceirizadas. Este outro arquiteto irá realizar o levantamento arquitetônico. Desta forma, o arquiteto que irá terceirizar o projeto não precisará se deslocar até o local para fazer o levantamento, diminuindo o custo de deslocamento. Outro benefício é que o arquiteto poderá realizar outra atividade em relação ao projeto que está sendo executado, enquanto o arquiteto terceirizado realiza o levantamento arquitetônico.

Já para os arquitetos que trabalham com a terceirização é uma forma de expandir seus locais de trabalho, podendo atender a região que bem quiser, permitindo uma maior demanda de trabalho.

Por fim, pode-se observar que todos os objetivos específicos propostos foram bem-sucedidos, incluindo o objetivo geral, onde todos os requisitos funcionais definidos foram desenvolvidos. Contudo, durante o desenvolvimento novas ideias foram surgindo, porém não puderam ser executadas devido ao limite do escopo e tempo pré-definidos. Portanto, esses requisitos, tratados aqui como requisitos opcionais, são apresentados como sugestões para trabalhos futuros. Sendo que estas recomendações são descritas a seguir.

5.1 RECOMENDAÇÕES DE TRABALHOS FUTUROS

Para aperfeiçoamento do protótipo foram elencados requisitos funcionais opcionais, brevemente apresentados no Quadro 8 da seção de Análise. O primeiro requisito funcional opcional proposto seria a implementação de *Log's*, possibilitando verificar as ações realizadas pelos usuários com precisão. A segunda proposta seria o envio de e-mail, atuando como uma forma de notificar o arquiteto assim que ele recebe uma nova solicitação para trabalhar em algum projeto ou exista a mudança de alguma situação do mesmo. O terceiro requisito visa implementar um método para a expiração de senha, informando o arquiteto de que sua senha precisa ser alterada, para uma maior segurança. O quarto requisito sugere a implementação de um cadastro múltiplo em relação a área de atuação do arquiteto, permitindo que o mesmo possa selecionar diversos municípios e diversos estados em apenas um cadastro.

Para o quinto requisito opcional foi proposta a implementação de um chat, possibilitando a conversa entre os arquitetos diretamente na ferramenta. Essa funcionalidade

permitiria uma melhor negociação e a manutenção de registrados da conversa entre os arquitetos. O sexto requisito visa aplicar um método de pagamento, possibilitando realizar o pagamento acordado entre os arquitetos. O sétimo requisito sugere a implementação de uma funcionalidade para avaliar projetos assim que os mesmos forem concluídos, permitindo aumentar o nível de confiança em relação ao(s) trabalho(s) realizado(s) pelo arquiteto terceirizado.

A oitava funcionalidade sugere permitir realizar validações ao cadastrar projetos, não permitindo incluir projetos com datas e horas iguais. O nono requisito visa construir uma nova rotina onde seja possível visualizar a área de atuação dos arquitetos através de um mapa, agilizando o processo de busca por esses profissionais. O décimo e último requisito opcional possui como objetivo desenvolver uma rotina para realizar o orçamento de projetos, permitindo que haja uma negociação entre os arquitetos antes de darem início ao projeto.

Além disto, sugere-se que seja realizado um processo de testes e homologação do protótipo, podendo ser realizado com a sua utilização por arquitetos em projetos piloto reais ou em projetos simulados nas turmas de arquitetura da instituição.

REFERÊNCIAS

- ALVES, William Pereira. **HTML & CSS: aprenda como construir páginas web**. São Paulo: EXPRESSA, 2021. Ebook.
- ASANA. **Por que Asana?** 2022. Disponível em: <https://asana.com/pt..> Acesso em: 14 ago. 2022.
- BEASLEY, Robert E. **Essential ASP.NET Web Forms Development: full stack programming with c#, sql, ajax, and javascript**. Franklin: Apress, 2020. Ebook.
- BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. São Paulo: Casa do Código, 2014. Ebook.
- CAIUT, Fabio. **Administração de Banco de Dados**. Rio de Janeiro: RNP/ESR, 2015. Ebook.
- CRONOSHARE. **Quanto custa um levantamento Arquitetônico?: faixa de preços**. Faixa de preços. 2022. Disponível em: <https://www.cronoshare.com.br/quanto-custa/levantamento-arquitetonico#:~:text=Cada%20planta%20e%20cada%20projeto,a%20R%24%2015%20%2Fm2..> Acesso em: 30 nov. 2022.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de Banco de Dados**. São Paulo: PEARSON, 2018. Ebook.
- FLANAGAN, David. **JavaScript**. Porto Alegre: Bookman, 2014. Ebook.
- FLOWUP. **O Software de Gestão de Projetos, Equipes e Financeiro para toda a sua empresa**. 2022. Disponível em: <https://www.flowup.me/>. Acesso em: 14 ago. 2022.
- FRANCISCO, Ed. **O que é CSS?** 2018. Disponível em: <https://chiefofdesign.com.br/o-que-e-css/>. Acesso em: 07 jun. 2022.
- LARAVEL. **Why Laravel?** 2022. Disponível em: <https://laravel.com/docs/9.x#why-laravel>. Acesso em: 24 maio. 2022
- MACINTYRE, Peter; TATROE, Kevin. **Programming PHP**. Sebastopol: O'Reilly Media, 2020. Ebook.
- MORAIS, Izabelly Soares de. **Engenharia de Software**. Porto Alegre: SAGAH, 2020. Ebook.
- NIXON, Robin. **Learning PHP, MySQL, & JavaScript**. Sebastopol: O'REILLY MEDIA, 2018. Ebook.
- PAULA FILHO, Wilson de Pádua. **Engenharia de Software: projetos e processos**. Rio de Janeiro: LTC, 2019. Ebook.
- PATEL, Sandeep Kumar. **Developing Responsive Web Applications with AJAX and jQuery: design and develop your very own responsive web applications using Java, jQuery, and AJAX**. Birmingham – Inglaterra: Packt Publishing Ltd, 2015. Ebook.

PECORARO, Christopher John. **Mastering Laravel**: Develop robust modern web-based software applications and RESTful APIs with Laravel, one of the hottest PHP frameworks. Birmingham – Inglaterra: Packt Publishing Ltd, 2015. Ebook.

PINHO, Diego Martins de. **Alertas bonitos, responsivos e customizados com o SweetAlert2**. 2018. Disponível em: <https://medium.com/code-prestige/alertas-bonitos-responsivos-e-customizados-com-o-sweetalert2-8db930038137>. Acesso em: 11 nov. 2022.

PRESSMAN, Roger S. **Engenharia de Software**: uma abordagem profissional. Porto Alegre: AMGH, 2016. Ebook.

PRETTYMAN, Steve. **Learn PHP 8**: Using MySQL, JavaScript, CSS3, and HTML5. New York: APRESS, 2020. Ebook.

REINEHR, Sheila. **Engenharia de Requisitos**. Porto Alegre: SAGAH, 2020. Ebook.

ROBBINS, Jennifer Niederst. **Learning Web Design**. Sebastopol: O'REILLY MEDIA, 2018. Ebook.

SHAW, Peter. **Postgres Succinctly**. Morrisville: Syncfusion Inc, 2013. Ebook.

SIMÕES, Guilherme Siqueira; VAZQUEZ, Carlos Eduardo. **Engenharia de Requisitos**: software orientado ao negócio. Rio de Janeiro: Brasport, 2016. Ebook.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Prentice Hall, 2011. Ebook.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2018. Ebook.

TURINI, Rodrigo. **PHP e Laravel**: Crie aplicações web como um verdadeiro artesão. Rio de Janeiro: Casa do Código, 2015. Ebook.

VOBI. **Transforme o seu escritório de arquitetura e design**. 2022. Disponível em: <https://www.vobi.com.br/>. Acesso em: 14 ago. 2022.

WARNER, Andrew. **Coding for Absolute Beginners**: Master the Basics of Computer Programming with Python, Java, SQL, C, C++, C#, HTML, and CSS. United States of America, 2021. Ebook.

ZABOT, D.; MATOS, E.D.S. **APLICATIVOS COM BOOTSTRAP E ANGULAR – COMO DESENVOLVER APPS RESPONSIVOS**. Editora Saraiva, 2020. Ebook.