

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE
CAMPUS AVANÇADO LAJES

ADRIELE KARIN FELICIANO DA SILVA
LIANE CATARINO PEREIRA

SISTEMA DE GERENCIAMENTO DA EMPRESA W10 REVEST

LAJES – RN

2018

ADRIELE KARIN FELICIANO DA SILVA
LIANE CATARINO PEREIRA

SISTEMA DE GERENCIAMENTO DA EMPRESA W10 REVEST

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade Integrado, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus *Avançado* Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Orientador: Prof. Mario Andrade Vieira de Melo Neto.

ADRIELE KARIN FELICIANO DA SILVA

LIANE CATARINO PEREIRA

SISTEMA DE GERENCIAMENTO DA EMPRESA W10 REVEST

Relatório de Prática Profissional apresentado ao Curso Técnico de Nível Médio em Informática, na modalidade Integrado, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Campus *Avançado* Lajes, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Trabalho apresentado e aprovado em ___/___/___, pela seguinte Banca Examinadora:

Mario Andrade Vieira de Melo Neto

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Dannilo Martins Cunha

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Osailton da Silva Vieira

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

AGRADECIMENTOS

Primeiramente agradecemos ao nosso orientador, Mario Andrade Vieira de Melo Neto, pelo auxílio, compreensão e disponibilidade ao longo de todo o projeto. E também, a todos os professores que ajudaram e nos acompanharam durante toda a nossa formação estudantil, em especial, aos docentes da área técnica pelos conhecimentos que nos passaram, que resultou na implementação do sistema proposto neste relatório.

Agradecemos também, aos nossos amigos e colegas de turma que nos apoiaram, incentivaram e ajudaram em todos os momentos para que pudéssemos chegar sãs até o final do curso.

E por fim, agradecemos aos nossos familiares, que com muito carinho e apoio, não mediram esforços para que nós chegássemos até esta etapa de nossas vidas.

RESUMO

Atualmente as empresas encontram-se em um mercado bastante competitivo e estão sempre buscando formas de crescer e inovar. Nessa situação, surgem cada vez mais opções tecnológicas que podem desempenhar um papel fundamental dentro da organização. O presente trabalho apresenta um sistema web que tem como objetivo gerenciar e divulgar os serviços oferecidos pela W10 Revest, uma empresa de envelopamentos e películas, localizada na cidade de Lajes/RN. O sistema irá automatizar tais processos e facilitar a comunicação entre o cliente e a empresa. É composto por um controle de agendamento que vai permitir que o próprio cliente agende seus serviços online e solicite orçamentos.

Palavras-chave: Tecnologia; Sistema Web; Agendamento; Comunicação; divulgação.

ABSTRACT

Today companies are in a very competitive market and are always looking for ways to grow and innovate. In this situation, there are more and more technological options that can play a fundamental role within the organization. The present work presents a web system that aims to manage and divulge the services offered by W10 Revest, a company of enveloping and films, located in the city of Lajes/RN. The system will automate such processes and facilitate communication between the customer and the company. It consists of a scheduling control that will allow the customer to schedule their services online and request budgets.

Keywords: Technology; Web System; Scheduling; Communication; disclosure.

LISTA DE ILUSTRAÇÕES

Figura 1. Ativando o servidor pelo node.js	19
Figura 2. Arquitetura MVC	22
Figura 3. Diagrama de caso de uso do Cliente	28
Figura 4. Diagrama de caso de uso do Administrador	28
Figura 5. Diagrama de classe	34
Figura 6. Modelo relacional	35
Figura 7. Tela inicial.	37
Figura 8. Tela sobre a empresa.	38
Figura 9. Tela de serviços ofertados – Automóveis.	39
Figura 10. Tela de serviços ofertados – Motos.	39
Figura 11. Tela de serviços ofertados – Eletrodomésticos, móveis e paredes.	40
Figura 12. Funcionamento do solicitar orçamento.	40
Figura 13. Funcionamento do solicitar agendamento.	41
Figura 14. Funcionamento do login para o acesso negado.	42
Figura 15. Funcionamento do login para acesso autorizado.	43
Figura 16. Tela da área administrativa.	44
Figura 17. Funcionamento da tela de analisar agendamentos – Sem registros.	45
Figura 18. Funcionamento da tela analisar agendamentos - Com registrados.	45
Figura 19. Funcionamento do aprovar agendamento.	46
Figura 20. Funcionamento do negar agendamento.	46
Figura 21. Funcionamento da tela de analisar orçamento.	47

LISTA DE QUADROS

Quadro 1. Sprints do projeto.	13
Quadro 2. Lista de Requisitos Funcionais.	24
Quadro 3. Lista de Requisitos Não Funcionais.	27
Quadro 4. Expansão do caso de uso: Solicitar agendamento.	29
Quadro 5. Expansão do caso de uso: Analisar agendamento.	30
Quadro 6. Expansão do caso de uso: Solicitar orçamento.	31
Quadro 7. Expansão do caso de uso: Analisar orçamento.	32

LISTA DE SIGLAS

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
CPU	<i>Central Process Unit</i>
CSS	Cascading Style Sheets
CSJS	<i>Client-Side JavaScript</i>
DOM	Document Object Model
GNU/GLP	<i>General Public Licence</i>
HTML	HyperText Markup Language
HTTP	HiperText Transfer Protocol
LAMP	<i>Linux, Apache, MySQL e Perl/PHP/Python</i>
MVC	Model, View, Controller
SQL	Structured Query Language
SSJS	<i>Server-Side JavaScript</i>
RF	<i>Requisitos Funcionais</i>
RNF	<i>Requisitos Não Funcionais</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	OBJETIVO GERAL.....	13
1.2	OBJETIVOS ESPECÍFICOS.....	13
1.3		
	METODOLOGIA.....	13
1.4	ESTRUTURA DO TRABALHO.....	14
2	FUNDAMENTAÇÃO TEÓRICA.....	15
2.1	TECNOLOGIAS WEB.....	15
2.1.1	HTML.....	15
2.1.2	CSS.....	15
2.1.3		
	JAVASCRIPT.....	16
2.1.4	JQUERY.....	16
2.1.5	JSON.....	17
2.1.6	NODE.JS.....	17
2.2	TECNOLOGIAS DE MODELAGEM.....	18
2.2.1	ASTAH	
	COMMUNITY.....	18
2.2.2	UNIFIED MODELING LANGUAGE (UML).....	19
2.2.3	DIAGRAMA DE CASOS DE USO.....	19
2.2.4	DIAGRAMA DE CLASSE.....	20
2.3	TECNOLOGIA DE GERENCIAMENTO DE BANCO DE DADOS.....	20
2.3.1	MYSQL.....	20
2.4	TECNOLOGIA DE ARQUITETURA DE APLICAÇÃO.....	20
2.4.1	MODEL-VIEW-CONTROLLER (MVC).....	20

3	MODELAGEM.....	21
3.1	VISÃO GERAL DO SISTEMA.....	22
3.2	ATORES DO SISTEMA.....	22
3.3	LEVANTAMENTO DE REQUISITOS.....	22
3.3.1	REQUISITOS FUNCIONAIS.....	23
3.3.2	REQUISITOS NÃO FUNCIONAIS.....	25
3.4	DIAGRAMA DE CASOS DE USO.....	26
3.5	EXPANSÃO DE CASOS DE USO MAIOR RISCO.....	29
3.6	DIAGRAMA DE CLASSE.....	33
3.7	MODELO RELACIONAL.....	34
4	RESULTADOS.....	36
4.1	PÁGINA INICIAL (HOME).....	36
4.2	SOBRE A EMPRESA.....	37
4.3	SERVIÇOS OFERTADOS.....	38
4.4	SOLICITAR ORÇAMENTO.....	40
4.5	SOLICITAR AGENDAMENTO.....	40
4.6	REALIZAR LOGIN.....	41
4.7	ANALISAR AGENDAMENTO.....	43
4.8	ANALISAR ORÇAMENTO.....	46
5	CONCLUSÃO.....	48
5.1	TRABALHOS FUTUROS.....	48
	REFERÊNCIAS.....	50

1 INTRODUÇÃO

Conforme Sommerville (2003, p.458), “atingir um alto nível de qualidade de produto ou serviço é o objetivo da maioria das organizações. Atualmente, não é mais aceitável entregar produtos com baixa qualidade e reparar os problemas e as deficiências depois que os produtos foram entregues ao cliente”.

Para Cortês e Chiossi (2001), a “preocupação com a qualidade deixou de ser um diferencial competitivo e passou a ser um pré-requisito básico para participação no mercado”. O cenário atual das empresas é de alta competitividade e isso tem feito com que elas mudem sua forma de atuação em relação a seus clientes e, até mesmo a sociedade, buscando por novas alternativas para expansão do mercado.

As tecnologias possibilitam aos empresários ferramentas para alavancar seus negócios, tendo em vista que o seu desenvolvimento crescente traz a necessidade de realizar tarefas de maneira cada vez mais prática, simples e relevante, de modo que seja exigido o menor esforço possível por parte das pessoas interessadas.

Um exemplo disso, são as aplicações do tipo web, já que diversas tarefas podem ser realizadas de maneira online, utilizando apenas um dispositivo com acesso à internet. Em razão de ter se tornado uma ferramenta frequentemente presente na vida da maioria das pessoas, elas ainda possuem a habilidade de receber e enviar dados para o usuário e atender as reais necessidades das empresas.

Entretanto, observa-se que ainda existem muitas empresas que passam por determinados problemas, uma delas é a W10 Revest que oferece serviços de envelopamentos e películas em automóveis e eletrodomésticos.

Como maior parte do serviço da empresa é feito manualmente, sem o uso de tecnologias, o mecanismo de divulgação e comunicação vinha demonstrando-se ineficaz. A empresa não possui controle sobre a quantidade de serviços de envelopamentos e películas que faz por dia e, por consequência, há uma sobrecarga

de trabalho, não conseguindo atender a demanda de serviços. De acordo com os problemas levantados anteriormente, percebe-se que a empresa necessita de um sistema para gerenciamento e controle sobre os serviços a serem executados e planejados. Diante disso, é proposto um sistema web com intuito de facilitar e agilizar os processos de agendamento e na execução dos serviços.

1.1 OBJETIVO GERAL

Este projeto tem como objetivo principal o desenvolvimento de um sistema web para divulgação e gerenciamento dos serviços oferecidos pela empresa W10 Revest, de modo que venha melhorar e facilitar a comunicação entre o cliente e a empresa.

1.2 OBJETIVOS ESPECÍFICOS

- a) Compreender os anseios e as dificuldades enfrentadas pelos funcionários da empresa W10 Revest;
- b) Realizar pesquisas bibliográficas relacionadas ao objeto de estudo deste projeto e das tecnologias utilizadas para o desenvolvimento;
- c) Conceber e projetar um sistema web para divulgação e gerenciamento dos serviços da empresa W10 Revest;
- d) Implementar o sistema projetado;
- e) Testar o sistema implementado.

1.3 METODOLOGIA

O método para o desenvolvimento deste projeto foi baseado na *Scrum*, uma metodologia ágil para organizar e gerenciar trabalhos complexos, como projetos de desenvolvimento de software. Essa metodologia foi utilizada como apoio para a organização e planejamento das atividades deste projeto. Sendo dividido em cinco

etapas de desenvolvimento, que na metodologia *Scrum* também são chamados de *sprints*, conforme ilustrado no quadro 1.

Quadro 1. Sprints do projeto.

Sprint	Atividade	Equipe
1	Levantamento e Elaboração de Requisitos	Adriele e Liane
2	Análise e projeto do Sistema	Adriele e Liane
3	Implementação dos casos de uso de maior risco	Adriele e Liane
4	Elaboração do relatório parcial do projeto	Adriele e Liane
5	Elaboração do relatório final do projeto.	Adriele e Liane

Fonte: Elaboração própria (2018)

Cada *sprint*, mostrado anteriormente, teve duração de 15 (quinze) a 20 (vinte) dias consecutivos de execução e, no final de cada um deles era feito uma reunião no qual as conclusões e os conhecimentos sobre aquela etapa eram compartilhados para a definição de metas.

Posteriormente, eram selecionadas as atividades a serem feitas no próximo ciclo de execução.

Ao total, 5 *sprints* foram realizadas para atender as atividades de todo o desenvolvimento deste trabalho.

Para a construção do sistema web foram pesquisadas várias tecnologias para o seu desenvolvimento, entre as quais serão abordadas no capítulo 2 deste projeto e, em paralelo a elas também foram utilizadas o HTML e CSS para a implementação do sistema projetado.

1.4 ESTRUTURA DO TRABALHO

Este documento está organizado em cinco capítulos. No primeiro capítulo, está presente a introdução e seus objetivos. No segundo capítulo, são apresentados os conceitos das tecnologias e ferramentas que serviram de apoio para a implementação do sistema. No terceiro capítulo, é realizada a descrição do software proposto e sua modelagem. No quarto capítulo, é feita a apresentação das funcionalidades implementadas (interfaces) do sistema desenvolvido. Já o quinto capítulo apresenta as conclusões acerca do trabalho desenvolvido e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os aspectos teóricos deste trabalho, tais como: as linguagens de programação, softwares e tecnologias que serviram de apoio para elaboração do projeto. Entre elas, o *HTML*, *CSS*, *JavaScript*, *jQuery*, *JSON*, *Node.js*, *Astah Community*, *UML (Unified Modeling language)*, *MySQL* e o padrão *MVC (Model-view-controller)*.

2.1 TECNOLOGIAS WEB

2.1.1 HTML

Segundo Silva (2014, p. 22), a linguagem *HTML (HyperText Markup Language)*, significa, em português, linguagem para marcação de hipertexto. Hipertexto é como todo o conteúdo é inserido em um documento para web, e que tem como principal característica a possibilidade de se interligar a outros documentos do mesmo tipo. O que torna possível a construção de hipertextos são os links, presentes nas páginas dos sites que estamos acostumados a visitar quando entramos na internet. A Web está estruturada em dois princípios básicos: *HTTP (HyperText Transfer Protocol)* e *HTML (HyperText Markup Language)*. O *HTTP* é o

protocolo de transferência de hipertexto, ou seja, é o que permite a navegação na Web, com o simples clicar do mouse sobre o texto ou imagem que está associada a outro link.

2.1.2 CSS

O CSS (*Cascading Style Sheets*) se refere a uma linguagem de estilos que define o layout de documentos escritos, geralmente, em linguagem HTML. O CSS define e controla, por exemplo, os parâmetros de texto e figuras, além de posicionamento dos elementos, imagens de fundo e etc. O HTML estrutura o conteúdo da página e o CSS formata esse conteúdo. A principal vantagem do uso do CSS é o fato de separarmos a apresentação da *webpage* do seu conteúdo, o que facilita a edição tanto do texto como do layout. (BARROS; SANTOS, 2008, p.3).

2.1.3 JAVASCRIPT

JavaScript é uma linguagem de *script* criada pela Netscape em 1995, tendo sido lançada junto com o navegador Netscape Navigator 2.0. Com seu lançamento, as páginas na internet começaram a ganhar vida, implementando um mínimo de dinamicidade devido ao modo como a linguagem acessa e manipula os componentes do seu ambiente hospedeiro (nesse caso, o Netscape Navigator) (FRANK; SEIBT, 2001).

Segundo Damiani (2006, p.9), o JavaScript é uma linguagem orientada a objetos que define um paradigma de programação onde os dados e as funções são agrupados (ou encapsulados) numa entidade chamada objeto. A orientação a objetos tem dois objetivos principais: Isolar os dados, diminuindo o risco de alterações indevidas e isolar as áreas do código que sofrem constantes modificações, tendo em vista a manutenção e a reusabilidade.

De acordo com Morrison (2008, p.6), “O JavaScript coloca o HTML e o CSS como uma das três peças da construção da página Web moderna. O HTML fornece

a estrutura, o CSS adiciona o estilo e o JavaScript inicia e faz as coisas acontecerem”.

“Quase todo *script* precisa lidar com dados de uma forma ou de outra, e isso geralmente significa armazenar dados na memória. O interpretador de JavaScript que reside nos navegadores Web é o responsável por separar pequenas áreas de armazenamento para os dados JavaScript” (MORRISON, 2008, p.34).

2.1.4 JQUERY

O jQuery é uma biblioteca em JavaScript desenvolvida por John Resig e disponibilizada como software livre e aberto, com o objetivo de simplificar a criação de efeitos visuais e de interatividade em sites da Web (SANTOS; FREITAS, 2016, p.37).

Segundo Silva (2008, p.25-26), simplicidade é a palavra-chave que resume e norteia o desenvolvimento com jQuery. Linhas e mais linhas de programação JavaScript escritas para obter um simples efeito em um objeto são substituídas por apenas algumas, escritas com sintaxe jQuery, que foi justamente desenvolvida para tornar mais fácil e simples a navegação de documentos HTML, criação de animações, manipulação de eventos, a seleção de elementos DOM e o desenvolvimento de aplicações AJAX. O jQuery consegue criar extraordinários efeitos com uma grande simplicidade.

2.1.5 JSON

Segundo Fonseca e Simões (2005, p.8), o JSON (*JavaScript Object Notation*) é um formato de texto para a serialização de dados estruturados. Ele pode representar quatro tipos primários (*strings*, números, booleanos e nulos) e dois tipos estruturados (objetos e vetores). Um objeto é uma coleção não ordenada de zero ou mais pares nome/valor, onde o nome é uma string e o valor é uma string, número, booleano, nulo, objeto ou vetor. Um vetor é uma sequência ordenada de zero ou mais valores.

De acordo com Santos e Freitas (2016, p.37), com este formato é possível armazenar e interpretar os dados de modo simples e rápido. Desta forma, pode se dizer que *JSON* é o formato ideal para a comunicação entre o *back-end* e *front-end* quando se trata de transferência de um número considerável de dados.

2.1.6 NODE.JS

De acordo com Aguiar (2015, p.37-39), Ryan Dahl ao criar o *node.js* se inspirou depois de ver barra de progresso de *upload* de arquivos no *Flickr*, o mesmo compreendeu que o navegador não sabia o quanto do arquivo era carregado e tinha que consultar o servidor web. Dahl buscava uma linguagem de programação em que fosse simples realizar operações assíncronas, para que a CPU não ficasse ociosa em cada operação de I/O demorada, como leitura e escrita de arquivos grandes.

Segundo Flanagan (2011, p.288), o Node é um interpretador JavaScript rápido com o código aberto, baseado em C++, com vínculos para as *APIs Unix* de baixo nível para trabalhar com processos, arquivos, soquetes de rede, etc., e também para cliente HTTP e APIs de servidor. A não ser por alguns métodos síncronos com nomes especiais, os vínculos do Node são todos assíncronos e, por padrão, os programas node nunca são bloqueados, isso quer dizer que normalmente mudam bem de escala e lidam com cargas grandes de forma eficiente. Como as APIs são assíncronas, o Node conta com rotinas de tratamento de evento, as quais são frequentemente implementadas com funções aninhadas e *closures*. Ou seja, o Node.js é uma plataforma assíncrona, orientadas a eventos, para desenvolvimento de rede escaláveis, surgindo não só para solucionar, mas também para mudar o paradigma de desenvolvimento Web, integrando a mesma linguagem de programação no cliente e no servidor.

De acordo com Aguiar (2015, p.40), *Client-Side JavaScript* (CSJS) é nome que se dá ao código dessa linguagem que executa no lado do cliente para acessar elementos da *DOM*, fazer chamadas assíncronas ao servidor, renderizar elementos dinâmicos, calcular valores, entre outras coisas. *Server-Side JavaScript* (SSJS) é o código JavaScript que executa no lado do servidor. É responsável por acessar o

banco de dados, atender a requisições, autenticar o usuário e fazer tudo o que um servidor Web faz. Node.js é uma plataforma SSJS.

Node.js por si só já traz um servidor Web embutido. Consegue dar suporte a um sistema de porte razoável sozinho, lidando bem com centenas de milhares de conexões concorrentes. O servidor nativo do Node.js traz um benefício a pessoas que desejam aprender desenvolvimento Web, pois funciona com poucas linhas de código e não requer configuração de software externo (AGUIAR, 2015, p.45).

Figura 1. Ativando o servidor pelo node.js

```
Microsoft Windows [versão 6.2.9200]
(c) 2012 Microsoft Corporation. Todos os direitos reservados.

C:\Users\adrielle karen>cd Downloads
C:\Users\adrielle karen\Downloads>cd W10 Revest/js
C:\Users\adrielle karen\Downloads\W10 Revest\js>node servidor.js
Servidor está ativo!
```

Fonte: Elaboração própria (2018)

2.2 TECNOLOGIAS DE MODELAGEM

2.2.1 ASTAH COMMUNITY

Segundo Brondani e Arend (2013), o *Astah Community* é uma ferramenta gratuita voltada para a modelagem de diagramas UML (*Unified Modeling Language*). Além do *Astah Community*, existem outras três versões: *Astah UML*, *Astah Professional* e *Astah Share* que disponibilizam outras funcionalidades além da modelagem UML, porém, sua licença é comercial. A ferramenta *Astah Community* é conhecida por sua praticidade e simplicidade em elaborar diagramas, como por exemplo: diagramas de classe, caso de uso, sequência, atividade, comunicação, máquina de estado, componentes, implantação, estrutura decomposição, objetos e pacotes.

2.2.2 UNIFIED MODELING LANGUAGE (UML)

Segundo Booch, Rumbaugh e Jacobson (2005, p.13), a Linguagem Unificada de Modelagem (UML) é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis.

Na UML possui vários tipos de diagramas entre os quais foram utilizados para o desenvolvimento deste projeto: Diagrama de classe e Diagrama de caso de uso.

2.2.3 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso especifica um conjunto de funcionalidades, através do elemento sintático “casos de uso”, e os elementos externos que interagem com o sistema, através do elemento sintático “ator” (SILVA, 2007).

Segundo Vargas (2007), além de casos uso e atores, o diagrama contém relacionamentos de dependência, generalização e associação e são basicamente usados para fazer a modelagem de visão estática do caso de uso do sistema. Essa visão proporciona suporte principalmente para o comportamento de um sistema, ou seja, os serviços externamente visíveis que o sistema fornece no contexto de seu ambiente. Neste caso, os diagramas de caso de uso são usados para fazer a modelagem do contexto de um sistema e fazer a modelagem dos requisitos de um sistema.

2.2.4 DIAGRAMA DE CLASSE

De acordo com Silva (2007), um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. Classes

e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes.

2.3 TECNOLOGIA DE GERENCIAMENTO DE BANCO DE DADOS

2.3.1 MYSQL

O MySQL é um sistema de gerenciamento de banco de dados relacional com código aberto. Ele é usado na maioria das aplicações gratuitas para gerir bases de dados. O MySQL usa a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar os dados armazenado em um banco de dados.

Na criação de sistemas web abertas e gratuitas, o conjunto de aplicações mais usado é o LAMP, que se refere as letras iniciais de: Linux (sistema operacional), Apache (servidor web), MySQL (gerenciador de banco de dados) e Perl/PHP/Python (linguagem de programação). Assim, o MySQL é um dos componentes centrais da maioria das aplicações públicas da internet. (PISA, 2012)

O MySQL é desenvolvido e distribuído sob as licenças GNU/GLP (*General Public Licence* – Licença Pública Geral). (TEIXEIRA, 2013)

2.4 TECNOLOGIA DE ARQUITETURA DE APLICAÇÃO

2.4.1 MODEL-VIEW-CONTROLLER (MVC)

O MVC (*Model-View-Controller*) é um padrão de arquitetura muito utilizado no desenvolvimento de projetos web, e também pode ser aplicado em qualquer outro tipo de projeto de software que exija a interação com o usuário. (SOUZA, 2018)

Souza (2016), descreve que na década de 70, foi criado por um engenheiro civil, Christopher Alexander, o que se considera o primeiro padrão de projeto. E por meio do trabalho de Alexander, profissionais da área de desenvolvimento de software utilizaram seus conceitos para iniciarem os primeiros padrões de

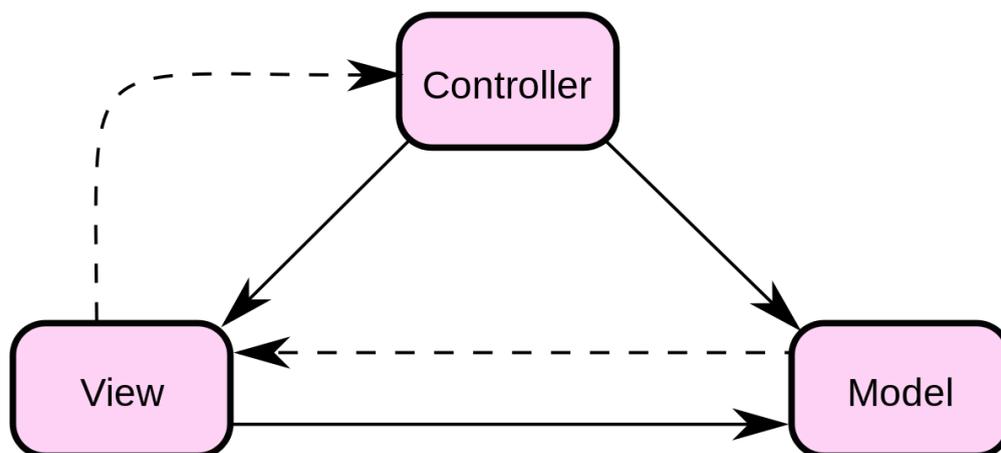
documentação de projetos. E que em 1979, Trygve Reenskaug, um então funcionário de uma corporação de pesquisa, viria a ser o responsável pelo nascimento do padrão de arquitetura MVC.

Segundo Luckow e Melo (2016, p. 185)

O principal objetivo do MVC não é definir como separar em camadas, mas, sim, definir como as camadas devem interagir. Em outras palavras, a separação de responsabilidades é conceito benéfico na implementação do MVC.

Luciano e Alves (2011) explicam que a dinâmica da arquitetura MVC é simples, todos os pedidos feitos na camada de Visão (*View*), são encaminhadas para a camada de controle (*Controller*), que se direciona para a camada de Modelo (*Model*), que processa a requisição solicitada, e por fim, a requisição retorna é exibida na camada de Visão (*View*). Para eles, a separação das camadas de Visão, Modelo e Controle aumentam a capacidade de reutilização e manutenção do projeto.

Figura 2. Arquitetura MVC



Fonte: Wikipédia (2018)

3 MODELAGEM

Este projeto prevê a criação de um sistema web para divulgação e gerenciamento dos serviços oferecidos pela empresa W10 Revest.

Esta seção especifica os resultados da modelagem do sistema proposto, com informações cruciais do projeto, implementação, testes e manutenção do sistema.

3.1 VISÃO GERAL DO SISTEMA

W10 Revest é um software empresarial que tem como objetivo gerenciar os serviços da empresa, disponibilizando que os clientes (público) agendem os serviços online, solicite o orçamento de um determinado serviço e possa se comunicar com a empresa, ou seja, que facilite a comunicação entre eles. Através desse sistema, o usuário administrador terá acesso a todos os dados dos clientes, incluindo agendamentos e orçamentos, posteriormente, analisá-los e respondê-los aos clientes. Este é dividido em 2 módulos, conforme descrito a seguir:

- Módulo Cliente (público): É utilizado pelos usuários que irão acessar o sistema para ver as informações da empresa, fazer agendamentos e solicitações de orçamentos. Além disso, eles poderão visualizar a galeria de serviços ofertados pela empresa.
- Módulo administrativo: É utilizado pelo usuário com registro no sistema e acesso de administrador. Será responsável por gerenciar os agendamentos e as solicitações de orçamento dos serviços.

3.2 ATORES DO SISTEMA

Ao total o sistema possui 2 (dois) atores, conforme descrito a seguir:

- Cliente (público): diz respeito aos usuários sem registro no sistema. Podem visualizar os serviços ofertados pela empresa, realizar agendamentos, solicitar orçamentos e se comunicar com a empresa.
- Administrador: diz respeito aos usuários com registro no sistema e acesso de dono da empresa. Podem gerenciar os serviços da empresa.

3.3 LEVANTAMENTO DE REQUISITOS

Na elicitação de requisitos foi utilizado a técnica de entrevista. Foram feitas entrevistas com o dono da empresa com objetivo de especificar os requisitos do

sistema proposto. Foram feitas duas entrevistas durante uma semana e, foram abordadas as seguintes perguntas:

- Quais as dificuldades tanto em relação aos clientes quanto a empresa para entrar em contato?
- Quais as dificuldades para divulgar a empresa? Quais serviços (redes sociais) usam?
- Há dificuldades no controle de serviços feitos por dia? (Se ficam sobrecarregados)
- Tendo base do tempo por serviço, poderiam dizer em média quantos serviços fazem por dia?
- Quais os tipos de serviços que a empresa oferece? (o cliente terá disponível uma lista pré-definidas de serviços ofertados? ou será livre ao cliente sua escolha?)
- Ajudaria o auxílio de uma ferramenta em que o cliente possa solicitar o orçamento e agendar o serviço?
- O que acha de colocar uma ferramenta que possibilite ao cliente anexar arquivos para que ajude na troca de informações? Ajudaria?

Essa estratégia foi aplicada pelos autores deste projeto, facilitando a coleta de boa parte das informações necessárias para o desenvolvimento dos requisitos do sistema.

3.3.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais (RF) representam as principais funcionalidades que o sistema deve executar em benefício dos usuários. Eles são representados pelo quadro 2.

Quadro 2. Lista de Requisitos funcionais da W10 Revest

Código	Nome	Descrição	Categoria
---------------	-------------	------------------	------------------

RF001	Realizar login	O sistema deve permitir ao administrador realizar o login para acessar a área restrita do sistema. Deverá ser informado e-mail e senha para realizar a autenticação. Caso os dados estejam corretos, o sistema encaminhará para a tela de área administrativa. Caso contrário, será exibido uma mensagem de erro.	Média
RF002	Recuperar Senha	O sistema deve permitir ao administrador que eles recuperem sua senha informando o e-mail de cadastro. Caso o e-mail esteja cadastrado, a nova senha será enviada para o e-mail informado.	Baixa
RF003	Gerenciar usuário	O sistema permitirá o cadastro do usuário administrador, que por sua vez terá acesso as solicitações de orçamentos e agendamentos. Para realizar este cadastro terá que ser preenchido os campos de nome e senha. Ou seja, poderá cadastrar um usuário, listar eles, remover e atualizar.	Média
RF004	Gerenciar serviços	O sistema permitirá que o administrador possa listar o catálogo de serviços da empresa, possibilitando o gerenciamento de cadastro, alteração, remoção, etc. No catálogo terá as informações dos tipos de serviços, o nome, a descrição, entre outros.	Alta
RF005	Solicitar agendamento	O sistema deve permitir aos clientes a realizar o agendamento de um determinado serviço. Para isso, deverá ser informado o nome, e-mail, telefone, data e horário desejado.	Alta

RF006	Apresentar serviços	Deverá apresentar os serviços disponíveis pela empresa, eles serão mostrados na página pública e qualquer tipo de usuário poderá acessar.	Média
RF007	Solicitar orçamento	O sistema deve permitir aos clientes a solicitar o orçamento de um serviço. Para realizar a solicitação de orçamento deverá ser informado o nome, e-mail, telefone, anexo e uma mensagem.	Alta
RF008	Analisar orçamento	O sistema deve permitir ao administrador a análise das solicitações de orçamentos. Deve ser exibido todos os orçamentos cadastrados permitindo que o administrador os analise, mostrando em detalhes o que todos eles contêm, tais detalhes podem ser definidos como: os dados pessoais do cliente (nome, email e telefone), o que foi solicitado, o anexo em imagem, etc.	Alta
RF009	Analisar agendamento	O sistema deve permitir ao administrador a análise dos agendamentos dos clientes. Deverá ser exibido todos os agendamentos cadastrados permitindo que o administrador os analise, mostrando em detalhes o que todos eles contêm tais detalhes podem ser definidos como: os dados pessoais do cliente (nome, email e telefone), o tipo de serviço, a data e o horário do agendamento.	Alta

Fonte: Autoria própria (2018).

3.3.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais (RNF) estão relacionados às restrições, ou atributos de qualidade para o sistema. Eles são representados pelo quadro 3, mostrado a seguir:

Quadro 3: Lista de Requisitos não funcionais.

Código	Nome	Descrição	Categoria
RNF001	Desempenho	O sistema tem que responder a qualquer operação que o usuário faça em no máximo 5 segundos.	Média
RNF002	Segurança	O usuário do sistema deve ser autenticado com login e senha e deve permitir o acesso restrito somente para o administrador.	Alta
RNF003	Confiabilidade	O sistema não deve apresentar falhas, porém caso as apresente, os desenvolvedores devem ser notificados.	Média
RNF004	Usabilidade	O sistema deve ser fácil de ser utilizado.	Média

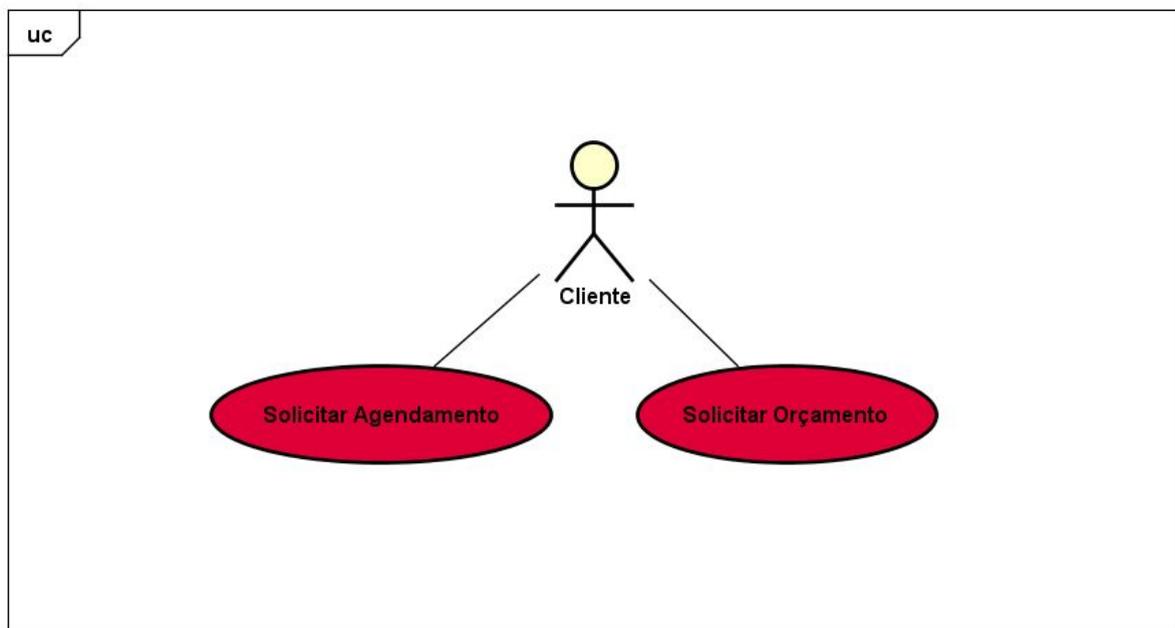
Fonte: Elaboração própria (2018)

3.4 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso descreve o escopo do sistema projetado. Ele é formado por um conjunto de atores que representam os papéis desempenhados pelos usuários e as funções. Um caso de uso representa a interação entre um ator e o sistema.

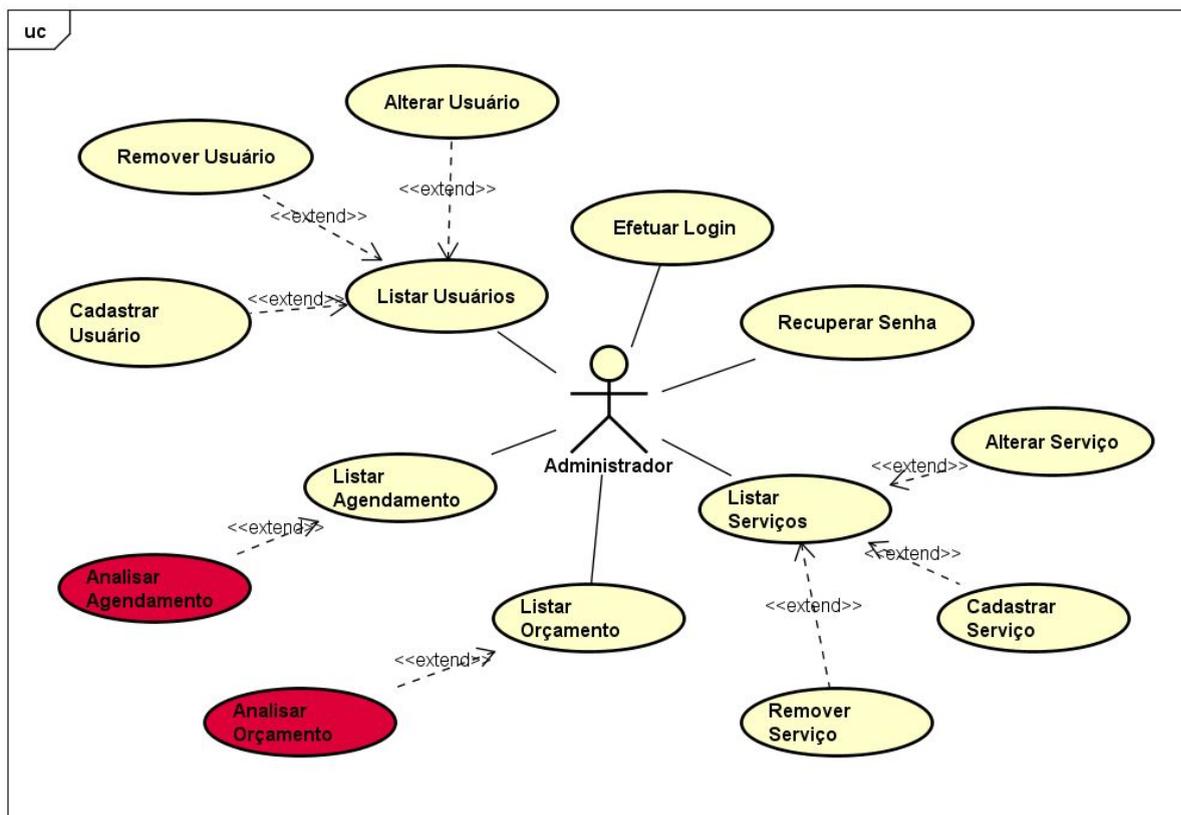
Dessa forma, levando em consideração os atores e os requisitos funcionais deste trabalho, foram definidos 2 atores, sendo eles o cliente e o administrador, e 16 (dezesseis) casos de uso, conforme ilustrado nas Figuras 3 e 4.

Figura 3: Diagrama de caso de uso do Cliente.



Fonte: Elaboração própria (2018).

Figura 4: Diagrama de caso de uso do Administrador.



Fonte: Elaboração própria (2018).

Ao total, 4 (quatro) casos de uso foram considerados como de maior risco. São eles: Solicitar agendamento, solicitar orçamento, analisar agendamento e analisar orçamento.

3.5 EXPANSÃO DE CASOS DE USO MAIOR RISCO

Nessa seção são apresentadas a expansão dos 4 (quatro) casos de uso de maior risco do sistema proposto.

Quadro 4. Expansão do caso de uso: Solicitar agendamento.

Solicitar agendamento:

Este caso de uso especifica a ação que o usuário cliente executa para realizar o agendamento de um determinado serviço da empresa.

Atores:

Cliente

Pré-condição:

O cliente deve acessar o sistema, entrar na página de agendar serviço e preencher o formulário. Ele deverá estar ciente dos serviços oferecidos pela empresa, qual o tipo de serviço que deseja realizar e os horários para que o agendamento seja realizado.

Pós-condição:

O sistema registra que o usuário realizou o agendamento.

Requisitos Funcionais:

RF005 - Solicitar agendamento

RF006 - Apresentar serviços

Requisitos Não Funcionais:

RNF001 - Desempenho

RNF004 - Usabilidade

Fluxo Básico:

1. Na tela inicial do sistema terá o menu com várias opções, entre elas, terá “Serviços ofertados” que será exibido na sua tela a lista de quais serviços são oferecidos pela empresa, e terá o “Agendar Serviços”, no qual o usuário deverá clicar.
2. Ao clicar na opção “Agendar Serviços” o sistema redireciona o usuário para a tela de Agendamento.
3. O sistema exibe um formulário para que o cliente informe todos os seus dados (Nome, e-mail, telefone, tipo de serviço, data e horário).
4. O usuário clica no botão “Agendar”.
5. O sistema informa que o agendamento foi realizado com sucesso.
6. O sistema registra todos os dados do cliente.

Quadro 5. Expansão do caso de uso: Analisar agendamento.

Analisar agendamento:

Este caso de uso especifica as ações que o usuário administrador executa no sistema com o objetivo de analisar os agendamentos solicitados pelos clientes, confirmando-os ou negando. Somente usuário administrador poderá executar essa ação.

Atores:

Administrador

Pré-condição:

O usuário deve estar logado.

Pós-condição:

Caso tenha algum agendamento disponível, o sistema exibe todas as informações sobre ele e o administrador deverá informar se o agendamento foi aprovado ou negado.

Requisitos Funcionais:

RF009 - Analisar agendamento

Requisitos Não Funcionais:

RNF001 - Desempenho

RNF002 - Segurança

RNF003 - Confiabilidade

Fluxo Básico:

1. Na tela inicial do administrador, o usuário clica em “Analisar agendamentos”.
2. O sistema redireciona o usuário para a tela de Análise de agendamento.
3. O sistema apresenta todos os agendamentos realizados.
4. Em um determinado pedido de agendamento, o sistema mostra todas as informações sobre ele, incluindo os dados pessoais do cliente (nome, e-mail e telefone), o tipo de serviço escolhido, a data e o horário.
5. Para o usuário analisar o pedido de agendamento, o sistema apresenta dois botões (aprovar e negar) abaixo de cada pedido, ao clicar em um deles é redirecionado para um campo de texto no qual o usuário irá informar a justificativa de resposta de acordo com o botão selecionado ou colocar possíveis orientações sobre agendamento

6. O usuário clica no botão “Enviar” para que o cliente receba a resposta e saiba se o agendamento foi aprovado ou negado.
7. O sistema informa que a análise foi feita com sucesso.

Quadro 6. Expansão do caso de uso: Solicitar orçamento.

Solicitar orçamento:

Este caso de uso especifica as ações que o usuário cliente executa no sistema, com o objetivo de fazer a solicitação do orçamento de um determinado serviço da empresa.

Atores:

Cliente

Pré-condição:

Qualquer tipo de usuário independentemente de ser cadastrado ou não, podem solicitar o orçamento do serviço. O cliente deve acessar o sistema, entrar na página de Contato e preencher o formulário. O usuário deverá especificar o tipo de serviço que deseja, para que o administrador possa definir um valor exato para o orçamento analisado.

Pós-condição:

O sistema registra que o usuário realizou a solicitação de orçamento.

Requisitos Funcionais:

RF007 - Solicitar orçamento

Requisitos Não Funcionais:

RNF001 - Desempenho

RNF003 - Confiabilidade

RNF004 - Usabilidade

Fluxo Básico:

1. Na tela inicial do sistema terá o menu com várias opções, entre elas, terá o “Contato” no qual o cliente deverá clicar.

2. Ao clicar na opção “Contato” o sistema redireciona o usuário cliente para a tela de solicitar orçamento.
3. O sistema exibe um formulário para que o cliente informe todos os seus dados (Nome, e-mail, telefone, um tipo de anexo do serviço e uma mensagem), além disso, mostrará todas as formas de contato com a empresa.
4. O usuário clica no botão “Enviar”.
5. O sistema informa que a solicitação de orçamento foi realizada com sucesso.
6. O sistema registra todos os dados do cliente.

Quadro 7. Expansão do caso de uso: Analisar orçamento.

Analisar orçamento:

Este caso de uso especifica as ações que o usuário administrador executa no sistema com o objetivo de analisar as solicitações de orçamento realizadas pelos clientes para que depois possa enviar a resposta. Somente o administrador poderá executar essa ação.

Atores:

Administrador

Pré-condição:

O usuário deve estar logado.

Pós-condição:

Caso tenha alguma solicitação de orçamento disponível, o sistema exibe todas as informações presentes nas solicitações (os dados pessoais dos clientes, um anexo sobre o tipo de serviço e a mensagem).

Requisitos Funcionais:

RF008 - Analisar orçamento

Requisitos Não Funcionais:

RNF001 - Desempenho

RNF002 - Segurança

RNF003 - Confiabilidade

Fluxo Básico:

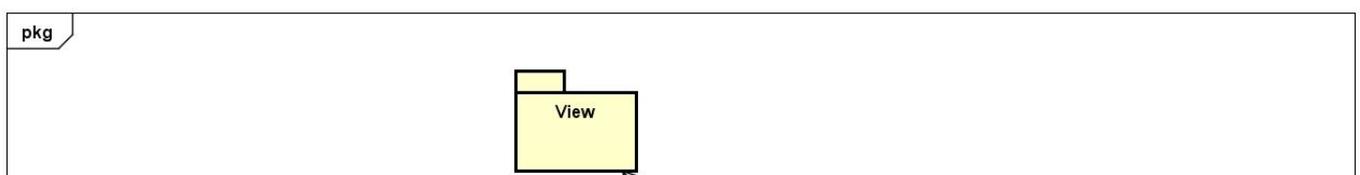
1. Na tela inicial do administrador, o usuário clica em “Analisar orçamentos”.
2. O sistema redireciona o usuário administrador para a tela de Análise de orçamento.
3. O sistema apresenta todas as solicitações de orçamento.
4. O sistema mostra todas as informações de uma determinada solicitação de orçamento, incluindo os dados pessoais do cliente (nome, e-mail e telefone), o anexo do tipo de serviço e a mensagem.
5. Para o administrador analisar o pedido de orçamento, o sistema apresenta o botão “responder” abaixo de cada pedido, ao clicar nele é redirecionado para um campo de texto no qual o usuário irá informar o valor do orçamento desejado pelo cliente e colocar possíveis informações.
6. O usuário clica no botão “Enviar”.
7. O sistema informa que a análise foi feita com sucesso.

Fonte: Elaboração própria (2018).

3.6 DIAGRAMA DE CLASSE

Seguindo o padrão de arquitetura MVC, a organização do sistema foi determinada em 3 camadas: o pacote View, que contém todos os componentes da interface utilizados pelos usuários, que são basicamente, arquivos HTML compostos com Tecnologia Web (HTML, CSS, JavaScript, jQuery, JSON e Node.js). O pacote Controller que contém 2 (duas) classes de controle que dispõe de componentes que cuidam e manipulam os dados enviados pela visão. Ele basicamente funciona como intermediador entre visão e negócio do sistema (Model). E o pacote Model, que contém 5 (cinco) classes com os dados e itens de negócio do sistema, neles são definidas as entidades e regras de negócio da aplicação.

Figura 5. Diagrama de classe

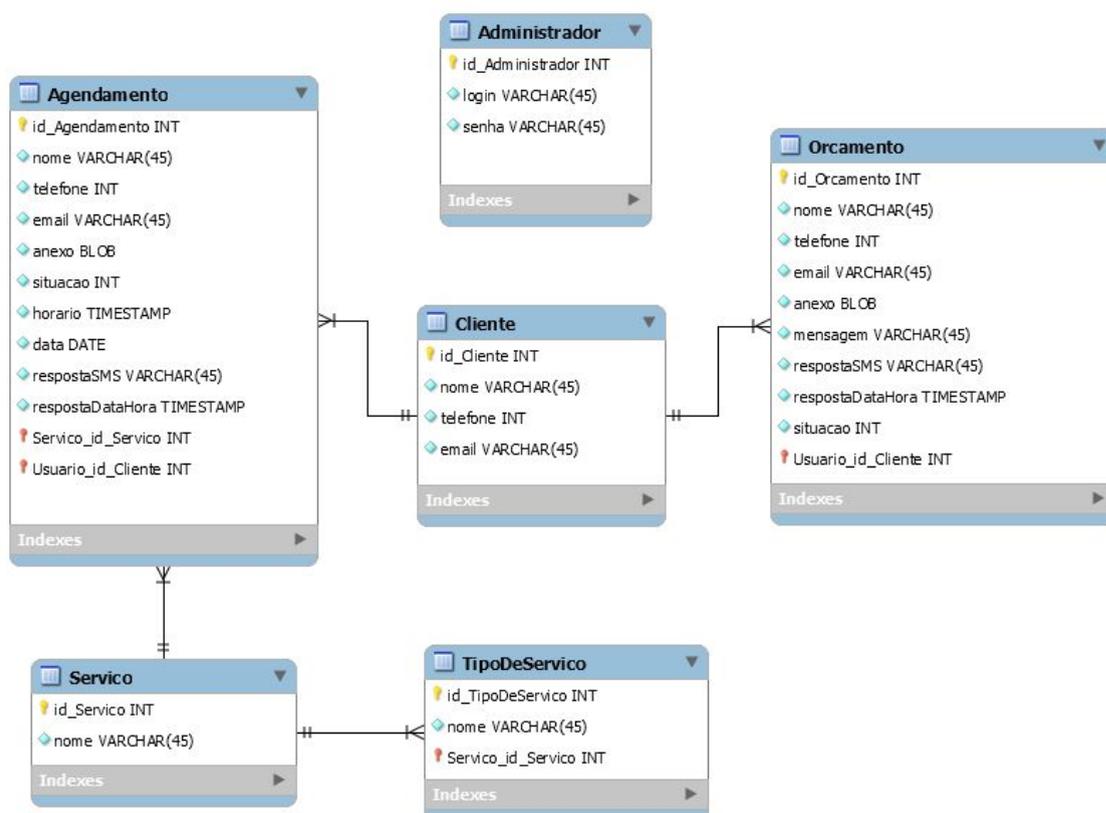


Fonte: Elaboração própria (2018).

3.7 MODELO RELACIONAL

O modelo relacional representa os dados como uma coleção de relações, onde cada uma delas é representada por uma tabela. Cada tabela é formada por um nome, que é único, e por um conjunto de atributos com seus referentes nomes e domínios, como é mostrado na figura 4.

Figura 6. Modelo relacional.



Fonte: Elaboração própria (2018).

A **tabela Administrador** temos como atributos: id_Administrador (chave primária), login (e-mail para acessar o sistema) e senha. Essa tabela serve para o administrador logar no sistema.

A **tabela Cliente** tem como tem como atributos: id_Cliente (chave primária), nome (nome do cliente), telefone (contato do cliente) e email (e-mail do cliente). Essa tabela se relaciona com a tabela agendamento, pois um cliente pode fazer um

ou vários agendamentos, e também com a tabela orçamento, pois um cliente pode fazer um ou vários pedidos de orçamentos.

A **tabela Agendamento** tem como atributos: id_Agendamento (chave primária), nome (nome do cliente), telefone (contato do cliente) e email (e-mail do cliente). Anexo (foto do objeto (carro, moto, etc.) para ser feito o serviço), horário (hora desejada para o serviço), data (dia desejado para o serviço), situação (*status* do agendamento: aprovado ou negado), respostaSMS (resposta da aprovação ou negação do agendamento), respostaDataHora (hora a mensagem foi respondida).

Essa tabela está se relacionando com a tabela cliente e também com a tabela Serviço, pois para um ou vários agendamentos se tem um serviço.

Na **tabela Serviço** temos como atributos: id_Servico (chave primária) e nome (se o serviço é envelopamento ou película). Essa tabela se relaciona com a tabela Agendamento e a tabela Tipo de Serviço, pois para cada um serviço temos um ou vários tipos de serviços.

Na **tabela Tipo de Serviço** temos como atributos: id_TipoDeServico (chave primária) e nome (se o tipo de serviço é em carro, moto, móveis ou eletrodomésticos). Essa tabela se relaciona com a tabela Serviço, pois para um ou vários tipos de serviços temos um serviço, como descrito anteriormente.

Na **tabela orçamento** temos como atributos: id_Orcamento (chave primária), nome (nome do cliente), telefone (contato do cliente), email (e-mail do cliente), anexo (foto do objeto (carro, moto, etc.) para ser feito o orçamento), mensagem (refere-se ao texto que o cliente fará sua solicitação de orçamento), respostaSMS (resposta para o cliente sobre a solicitação orçamento), respostaDataHora (hora da resposta sobre a solicitação de orçamento) e situação (*status* do orçamento: avaliado ou em espera).

Essa tabela se relaciona com a tabela Cliente, pois um cliente pode fazer um ou vários pedidos de orçamentos.

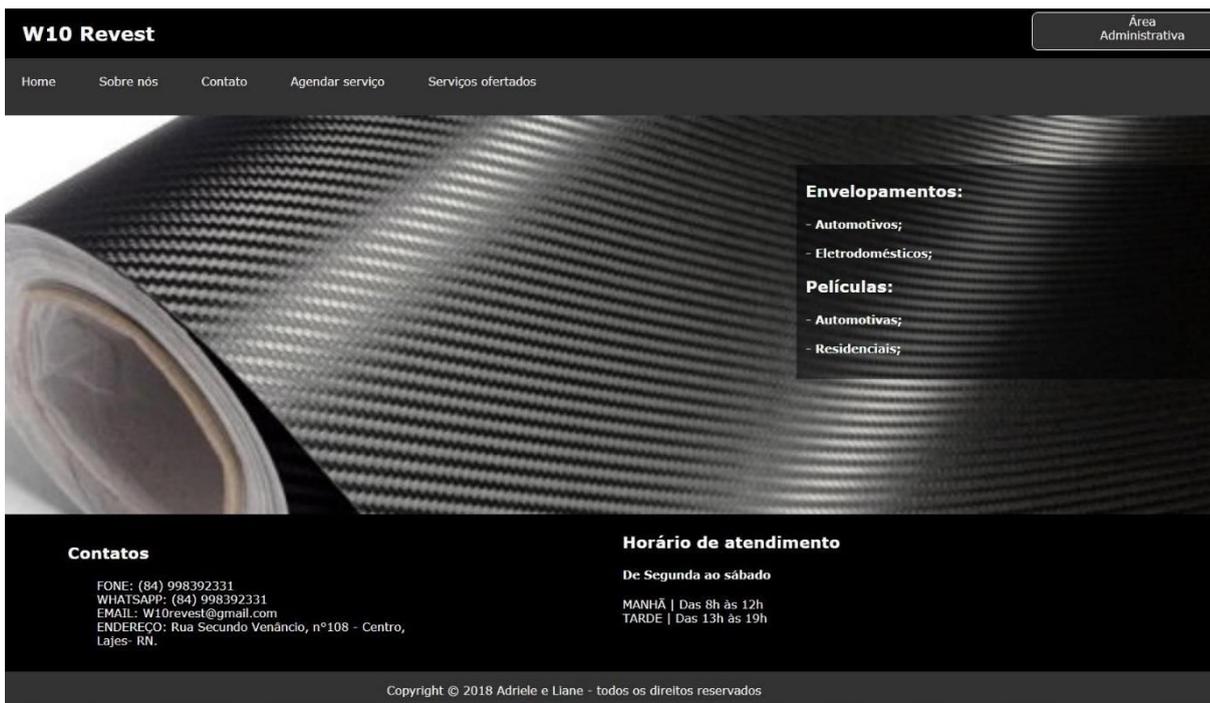
4 RESULTADOS

Essa seção terá a apresentação das telas e funcionalidades implementadas no sistema proposto neste projeto.

4.1 PÁGINA INICIAL (HOME)

Na Figura 7, a página é representada como pública, e pode-se observar que temos a opção de fazer login para entrar na Área Administrativa do sistema, de saber um pouco mais sobre a empresa (Sobre nós), a opção de Contato que basicamente serve para mandar alguma mensagem ou solicitar o orçamento de um determinado serviço, o Agendar serviço e a opção de serviços ofertados pela W10 Revest.

Figura 7: Tela inicial.



Fonte: Elaboração própria (2018).

4.2 SOBRE A EMPRESA

Na figura 8, é apresentado a tela onde terá as informações sobre o que é a empresa W10 Revest.

Figura 8: Tela sobre a empresa.

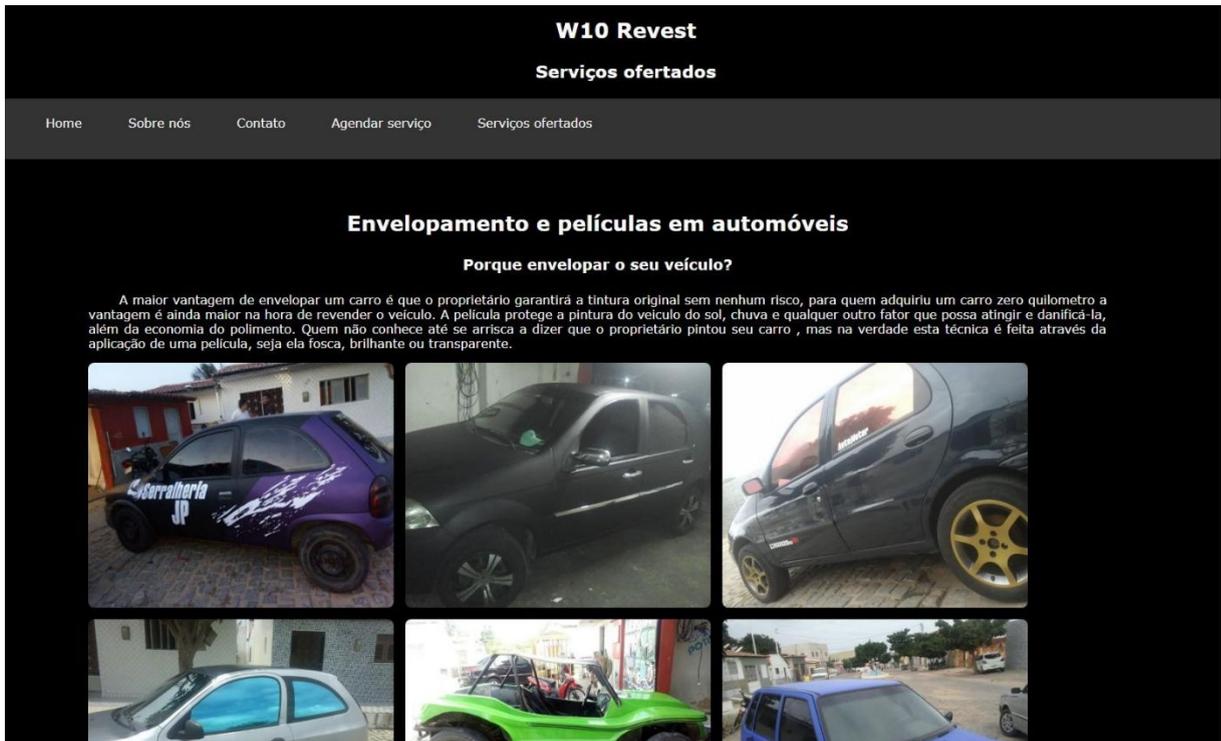


Fonte: Elaboração própria (2018).

4.3 SERVIÇOS OFERTADOS

Nas figuras 9, 10 e 11 são apresentadas as telas em que os clientes (público) poderão ver a lista de serviços que já foram feitos pela empresa W10 Revest, com o objetivo de auxiliar na solicitação de agendamento e orçamento.

Figura 9: Tela de serviços ofertados – Automóveis.



Fonte: Elaboração própria (2018).

Figura 10: Tela de serviços ofertados – Motos.



Fonte: Elaboração própria (2018).

Figura 11: Tela de serviços ofertados – Eletrodomésticos, móveis e paredes.

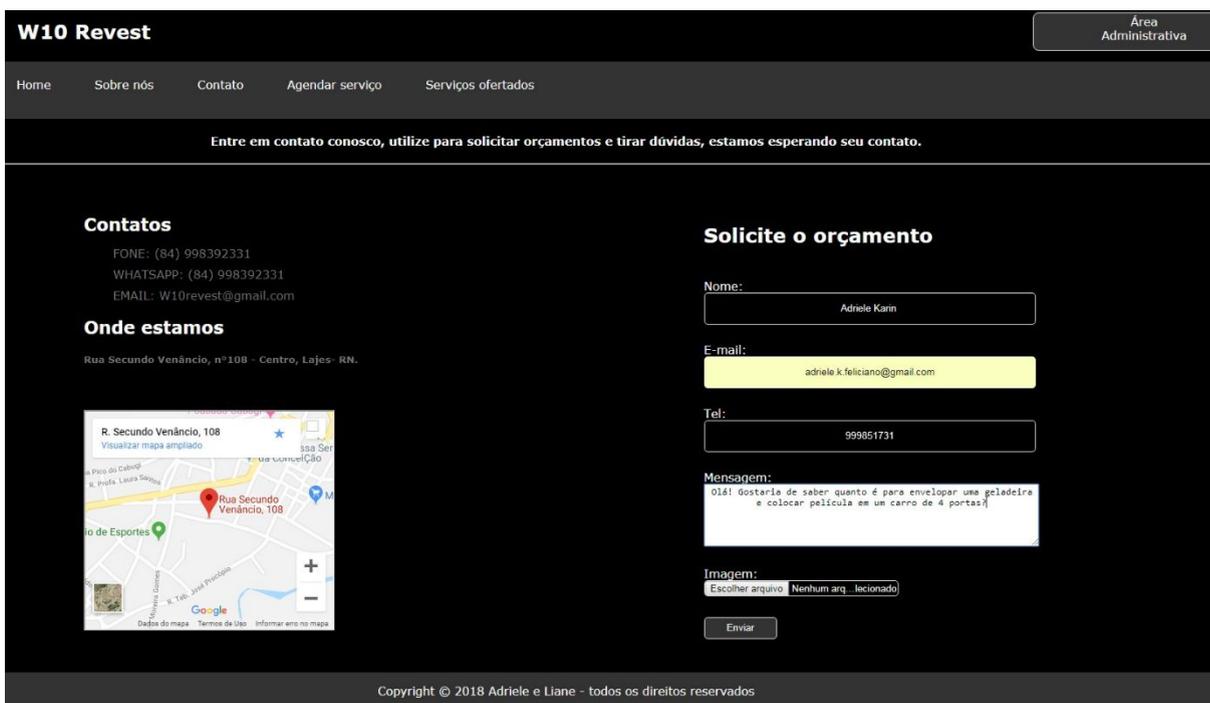


Fonte: Elaboração própria (2018).

4.4 SOLICITAR ORÇAMENTO

Na figura 12, é apresentado a tela de contato que terá todas as formas de comunicação com a empresa e a sua localização. Sendo um usuário cliente (público), ele poderá mandar qualquer tipo de mensagem ou solicitar o orçamento de um determinado serviço a um usuário administrador (a empresa). Para solicitar o orçamento o cliente deverá informar o nome, e-mail, telefone, uma mensagem e, se quiser especificar o tipo de serviço poderá enviar um anexo (imagem).

Figura 12: Funcionamento do solicitar orçamento.



Fonte: Elaboração própria (2018).

4.5 SOLICITAR AGENDAMENTO

A figura 13 mostra a tela de agendar serviço. O cliente (público) poderá solicitar o agendamento a um usuário administrador informando o nome, e-mail, telefone, os tipos de serviço, a data e a preferência de horário.

Figura 13: Funcionamento do solicitar agendamento.

Realizar agendamento

Nome:

E-mail:

Telefone:

Tipo de Serviço:

Data:

Preferência de Horário:

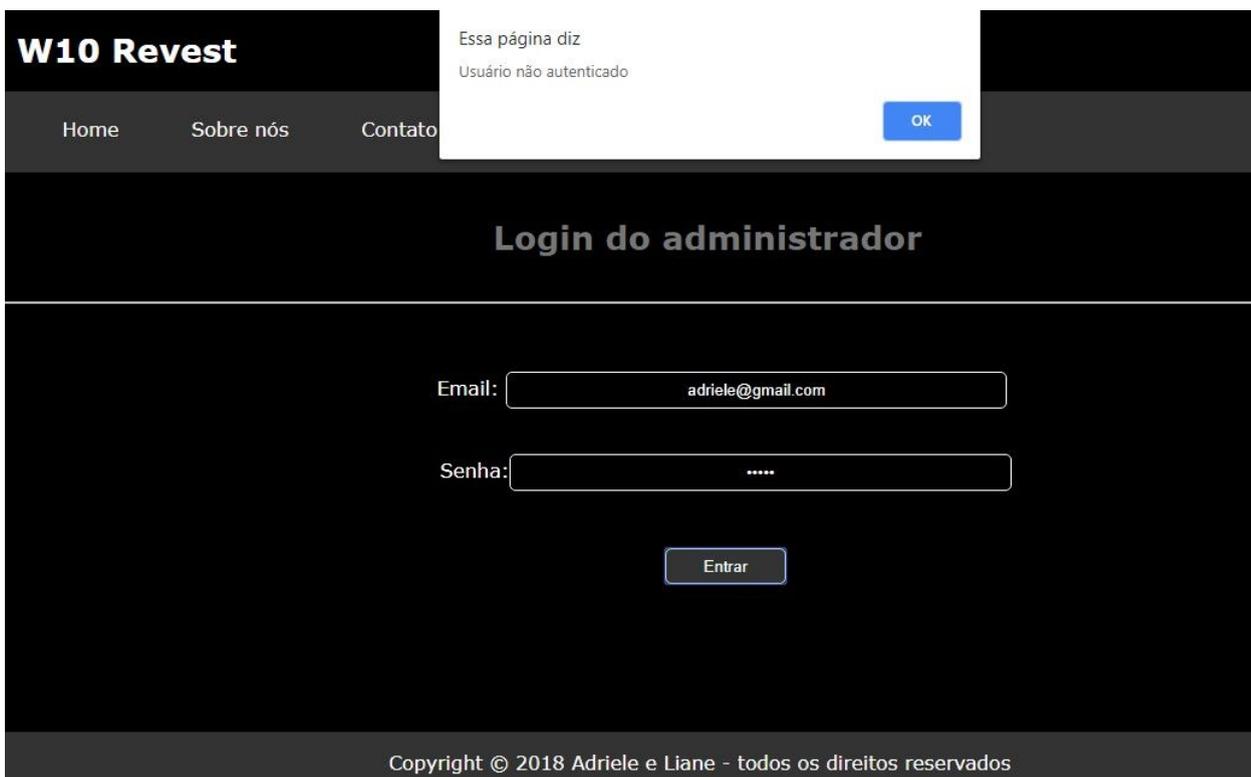
Copyright © 2018 Adriele e Liane - todos os direitos reservados

Fonte: Elaboração própria (2018).

4.6 REALIZAR LOGIN

Nas figuras 14 e 15, é apresentado a tela de autenticação do administrador, ele deverá informar seu e-mail e senha para ter acesso a área administrativa do sistema. Caso um dos itens citados for informado errado, o sistema apresentará uma mensagem de erro e negará o acesso do usuário ao sistema.

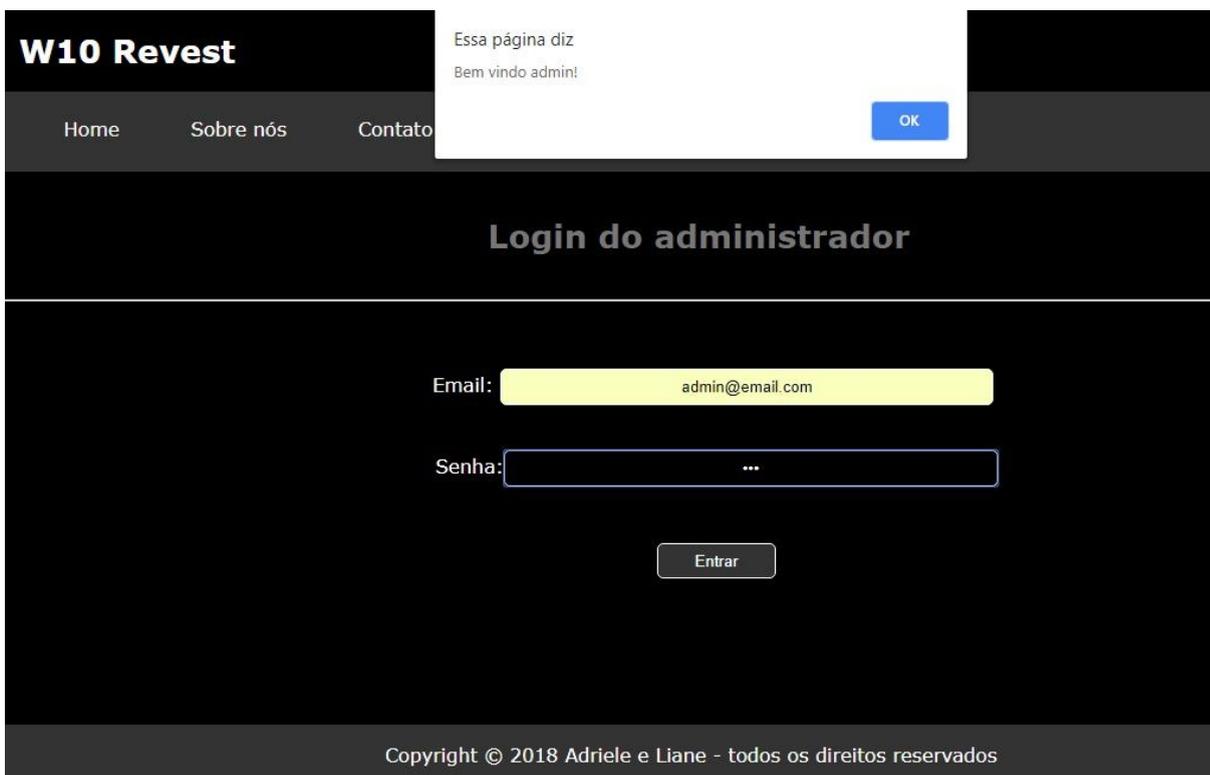
Figura 14: Funcionamento do login para o acesso negado.



Fonte: Elaboração própria (2018).

Caso o e-mail e a senha sejam informados corretamente, o sistema encaminhará o administrador para a tela inicial da área administrativa (Figura 16) e exibirá todas as funções que o usuário administrador pode executar, como gerenciar usuário, gerenciar serviços, analisar agendamento, analisar orçamento e exibir os relatórios.

Figura 15. Funcionamento do login para acesso autorizado.



Fonte: Elaboração própria (2018).

Figura 16: Tela da área administrativa.



Fonte: Elaboração própria (2018).

4.7 ANALISAR AGENDAMENTO

Nas figuras 17 e 18, são mostradas as telas de análise de agendamentos e para ter acesso a essa tela o usuário precisa ser um administrador. O sistema mostrará ao administrador todos os agendamentos registrados, e também uma mensagem caso não tenha nenhum pedido de agendamento.

Figura 17: Funcionamento da tela de analisar agendamentos – Sem registros.



Fonte: Elaboração própria (2018).

Depois do cliente ter realizado o agendamento, como é representado na figura 13, percebe-se que um agendamento foi registrado no sistema, como é mostrado na figura a seguir:

Figura 18. Funcionamento da tela analisar agendamentos - Com registrados.



Fonte: Elaboração própria (2018).

O administrador poderá analisar todos os agendamentos e respondê-los através dos botões aprovar e negar.

Caso o usuário clique em “Aprovar” o sistema o redirecionará para um campo de texto no qual deverá escrever algumas informações (ex.: o horário exato) sobre o agendamento analisado.

Figura 19. Funcionamento do aprovar agendamento.

Essa página diz

Digite mais informações sobre o agendamento: (ex: especificar o horário)

Home Contato Agendar s

Analisar agendamento

NOME: Liane Catarino

EMAIL: liane@gmail.com

TELEFONE: 84999851731

TIPO DE SERVIÇO: Envelopamento

OPÇÃO DE SERVIÇO: Carro

DATA: 2018-12-28

PREFERÊNCIA DE HORÁRIO: 8:00 às 12:00

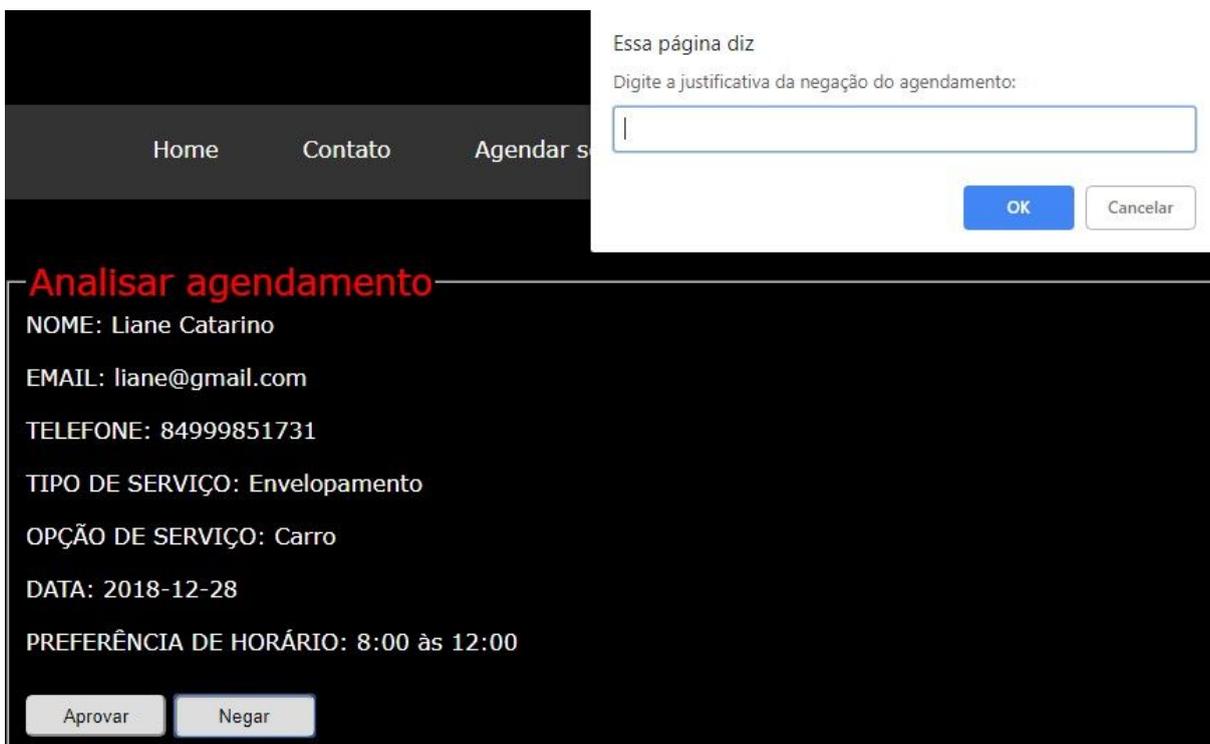
Aprovar Negar

OK Cancelar

Fonte: Elaboração própria (2018).

Ou se o usuário clicar em “negar” ele também será redirecionado para um campo de texto que irá justificar a negação do agendamento.

Figura 20. Funcionamento do negar agendamento.



Fonte: Elaboração própria (2018).

4.8 ANALISAR ORÇAMENTO

Na figura 21, é apresentado a tela de análise de orçamentos, no qual o administrador recebeu todos os dados da solicitação de orçamento que foi apresentada na figura 12. Para ter acesso a essa tela o usuário precisa ser um administrador. Ele poderá analisar todos os orçamentos e, posteriormente clicar no botão “Responder”, onde o sistema redirecionará o administrador para um campo de texto no qual irá informar a resposta do orçamento (ex.: valor do serviço).

Figura 21. Funcionamento da tela de analisar orçamento.

Home Contato Agendar serviço Serviços ofertados

Essa página diz
Digite a resposta da solicitação de orçamento: (ex: valor do serviço)

OK Cancelar

Analisar orçamento

NOME: Adriele Karin
EMAIL: adriele.k.feliciano@gmail.com
TELEFONE: 999851731
MENSAGEM: Olá! Gostaria de saber quanto é para envelopar uma geladeira e colocar película em um carro de 4 portas?
ANEXO: undefined

Responder

Copyright © 2018 Adriele e Liane - todos os direitos reservados

Fonte: Elaboração própria (2018).

5 CONCLUSÃO

A finalidade deste projeto foi propor um sistema que fosse implantado na empresa W10 Revest, no qual foi desenvolvido uma ferramenta de apoio, voltada a divulgação e gerenciamento dos serviços ofertados pela empresa. O sistema desenvolvido foi devidamente projetado para divulgar os serviços e que os clientes possam realizar o agendamento online e se comunicar com a W10 Revest.

O sistema foi elaborado, projetado, parcialmente implementado e validado. Em comum acordo com a empresa, foram implementados os casos de uso de maior importância para a empresa, sendo eles, o de Efetuar Login, Solicitar Agendamento e Solicitar Orçamento. Assim, a implementação total do sistema ficará para um trabalho futuro.

Esse trabalho foi importante para conhecer mais sobre a área da criação de sistemas webs, nos permitindo aplicar esse conhecimento como solução prática para os problemas de diversas empresas. Assim, percebe-se que a sociedade está cada vez mais dependente da tecnologia no mercado de trabalho. Além disso, realizar essa aplicação fez com que houvesse a percepção de como é a área de desenvolvimento de software, desde a ideia inicial de desenvolvimento até a conclusão da aplicação.

Portanto, uma experiência como essa servirá de base para a carreira profissional, embora a sua realização tenha sido feita com dificuldades, a sua conclusão nos fez compreender e pôr em prática o que aprendemos durante o curso.

5.1 TRABALHOS FUTUROS

Os trabalhos futuros deste projeto são terminar a implementação dos demais casos de uso do sistema, como analisar agendamento, analisar orçamento, gerenciar usuário, gerenciar serviços e ver relatórios, para que assim ele seja implantado na empresa W10 Revest. Além disso, criar novas funcionalidades, como o cadastro de clientes e controle de caixa.

REFERÊNCIAS

AGUIAR, G. S. **Node.js: Estudo tecnológico e desenvolvimento full-stack JavaScript de plataformas de competições em problemas algorítmicos**. 2015. p.37. Graduação em Ciência da Computação - Universidade Federal de Pernambuco, Recife, 2015.

BARROS, Isabelle M. O.; SANTOS, Carlos Felipe A. **Apostila de Introdução ao CSS**. 2008. 21f. Curso de Engenharia de Telecomunicações - Universidade Federal Fluminense, Rio de Janeiro, 2008.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. Rio de Janeiro: Elsevier, 2005.

BORGES, J. M. **Ambiente web de suporte ao processo de avaliação de qualidade de produtos de software**. 2006. 134f. Trabalho de Conclusão de Curso II do curso de Ciências da Computação - Bacharelado – Universidade Regional de Blumenau, Santa Catarina, 2006.

BRONDANI, C. H.; AREND, C. F. **Guia prático de utilização da ferramenta Astah Community 6.1**. 2013. Disponível em: <https://pt.scribd.com/doc/139768773/Astah-Community#user-util-view-profile>
Acesso em: 18 de novembro de 2018.

DAMIANI, E. B. **Guia de Consulta Rápida JavaScript**. São Paulo - SP: Novatec, 2001.

FLANAGAN, D. **JavaScript: O Guia Definitivo**. São Paulo: Bookman, 2011.

FONSECA, R.; SIMÕES, A. **Alternativas ao XML: YAML e JSON**. 2005. 14f. Departamento de Informática - Universidade do Minho, Braga, Portugal, 2005.

FRANK, Diego, R.; SEIBT, Leonardo. **JavaScript**. 2001. 5f. Fundação Educacional Encosta Inferior do Nordeste- Faculdades de Informática de Taquara, Taquara, 2001.

LUCIANO, Josué; ALVES, Wallison Joel Barberá. **PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER**. Epeq Fafibe, Bebedouro, v. 1, n. 3, p.102-107, jan. 2011. Disponível em: <<http://www.unifafibe.com.br/revistasonline/arquivos/revistaepqfafibe/sumario/20/16112011142249.pdf>>. Acesso em: 15 nov. 2018.

LUCIDCHART. **Tipo de modelos de banco de dados**. [2018]. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-um-modelo-de-banco-de-dados>>. Acesso em: 01 de dez. 2018.

LUCKOW, Décio Heinzemann; MELO, Alexandre Altair de. **Programação Java para a Web**. 2. ed. São Paulo: Novatec, 2016. 672 p.

MARÇAL, Ana Sofia Cysneiros; FREITAS, Bruno Celso Cunha de; BELCHIOR, Arnaldo Dias. **Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI**. 2004. 12 f. Dissertação (Mestrado) - Curso de Informática Aplicada, Mestrado em Informática Aplicada, Universidade de Fortaleza, Fortaleza, 2004. Cap. 1.

MORRISON, M. **Use a cabeça JavaScript**. Rio de Janeiro: Alta Books, 2008.

NOGUEIRA, M. **Gestão de configuração software: Uma visão estabelecida pelas normas e modelos de qualidade**. 2003. 12f. Universidade Paulista, São Paulo, 2003.

PISA, Pedro. **O que é e como usar o MySQL**. 2012. Disponível em: <<https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 26 nov. 2018.

SANTOS, C.H.C.; FREITAS, M.C. **Visdadosenem: Visualização de dados do enem com a biblioteca D3**. 2016. 75f. Grau de Bacharel em Tecnologias da Informação e Comunicação - Universidade Federal de Santa Catarina, Araranguá, 2016.

SILVA, M. S. **jQuery - A Biblioteca do Programador JavaScript**. São Paulo: Novatec, 2008.

SILVA, S. M. **HTML5 – 2ª Edição: A linguagem de marcação que revolucionou a web**. São Paulo: Novatec, 2014.

SOUZA, Marcio Ballem de. **Padrão MVC - Java Magazine**. Disponível em: <<https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>>. Acesso em: 15 out. 2018.

TEIXEIRA, José Ricardo. **Introdução ao MySQL**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-mysql/27799>>. Acesso em: 27 nov. 2018.

VARGAS, T. C. S. **A história de UML e seus diagramas**. 2007. 9f. Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2007.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. **MVC**. Disponível em:<<https://pt.wikipedia.org/wiki/MVC>>. Acesso em: 30 nov. 2018.