Imagem em preto e branco

Descrição gerada automaticamente

**UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO**

**BACHARELADO EM CIENCIA DA COMPUTAÇÃO**

**GABRIEL SEVERO DA SILVA**

**LUCAS ALBERTO**

**RELATÓRIO DO PROJETO**

**LOCADORA DE CARROS**

**GARANHUNS-PE**

**2023**

**GABRIEL SEVERO DA SILVA**

**LUCAS ALBERTO**

Relatório do projeto de Locadora de carros apresentado ao curso de Bacharelado em Ciência da Computação, como requisito parcial para obtenção da aprovação na disciplina Programação Orientada a Objetos.

Orientador: Igor Medeiros Vanderlei

**GARANHUNS-PE**

**2023**

**RESUMO**

A locadora é um sistema de gerenciamento de aluguel de carros, que permite aos clientes realizar reservas e aluguel de veículos online, além de gerenciar todo o processo de locação de carros. O sistema é composto por um frontend em Vue.js, um backend em Java com Spring Framework e um banco de dados PostgreSQL. O sistema possui funcionalidades como cadastro de clientes e funcionários, registro de carros e suas informações, além do gerenciamento de reservas e aluguel de carros. O projeto foi desenvolvido com base em requisitos e testes unitários foram implementados para garantir o funcionamento correto das funcionalidades.

**Palavras-chave**: aluguel; carros; java

1. **INTRODUÇÃO**

Este documento tem como objetivo apresentar o sistema desenvolvido para gerenciamento de uma locadora de veículos, contemplando tanto o seu backend, desenvolvido em Java, quanto o frontend, desenvolvido em Vue.js.

O sistema possibilita a gestão de clientes, funcionários, carros, reservas e aluguéis, oferecendo uma interface intuitiva e amigável para o usuário. Além disso, foram implementados testes unitários, de integração e de API para garantir a qualidade do código e a segurança das informações gerenciadas pelo sistema.

Através deste documento, é possível ter uma visão geral do projeto, desde a arquitetura até as funcionalidades oferecidas. Além disso, serão apresentados os requisitos do sistema, as tecnologias utilizadas, os testes implementados e as considerações finais.

* 1. **VISÃO GERAL DO DOCUMENTO**

A visão geral do projeto é apresentar uma solução de software para uma locadora de carros. A aplicação é composta por um sistema web, com uma interface para usuário e outra para funcionário, além de um banco de dados relacional para armazenamento de informações. O objetivo principal é permitir que os usuários possam alugar carros de forma rápida e fácil, enquanto a locadora consegue gerenciar e monitorar as suas operações de aluguel de maneira eficiente.

O sistema foi desenvolvido utilizando a arquitetura de três camadas, com separação clara entre a interface do usuário, a lógica de negócios e o acesso ao banco de dados. As principais funcionalidades incluem a busca e reserva de carros, a criação de novos clientes e funcionários, a gestão de aluguéis e a geração de relatórios para monitoramento e análise das operações da locadora.

Para garantir a qualidade do software, foram realizados testes unitários, de integração e de interface, além de implementação de tratamento de exceções e validação de entrada de dados.

1. **DESCRIÇÃO GERAL DO SISTEMA**

O sistema em questão é uma locadora de carros que permite a seus clientes alugarem veículos de acordo com suas necessidades. O sistema possui funcionalidades para cadastro de clientes, cadastro de carros, realização de reservas e aluguel de veículos. Além disso, o sistema permite que os funcionários gerenciem as informações dos clientes, dos carros e das reservas, bem como gerem relatórios sobre o desempenho da locadora. O objetivo é proporcionar uma experiência agradável e eficiente para os clientes, ao mesmo tempo em que se mantém a organização e o controle da locadora. O sistema é desenvolvido utilizando tecnologias web, incluindo o framework Vue.js para o frontend e o framework Spring Boot para o backend.

1. **REQUISITOS DO SISTEMA**

A seguir estão os requisitos funcionais e não-funcionais do sistema de locação de veículos:

Requisitos Funcionais:

* O sistema deve permitir que um usuário se cadastre como cliente;
* O sistema deve permitir que um usuário se cadastre como funcionário;
* O sistema deve permitir que um cliente alugue um veículo;
* O sistema deve permitir que um funcionário cadastre um novo veículo;
* O sistema deve permitir que um funcionário cadastre novos funcionários;
* O sistema deve permitir que um funcionário consulte um relatório de locações;
* O sistema deve permitir que um funcionário consulte um relatório de clientes;
* O sistema deve permitir que um funcionário consulte um relatório de veículos;
* O sistema deve permitir que um funcionário consulte um relatório de funcionários;

Requisitos Não-Funcionais:

* O sistema deve ser fácil de usar e intuitivo;
* O sistema deve ser responsivo e fornecer uma boa experiência ao usuário.
  1. **DIAGRAMA DE CASOS DE USO**

O diagrama incluiu dois atores: Cliente e Funcionário. Eles foram divididos nas respectivas funções:

Cliente

* Fazer reserva: permite que um cliente reserve um carro para aluguel em uma determinada data e hora.
* Alugar carro: permite que um cliente alugue um carro para uso em um determinado período.
* Devolver carro: permite que um cliente devolva um carro que ele alugou anteriormente.

Funcionário

* Gerenciar estoque: permite que um funcionário adicione, remova ou atualize as informações dos carros disponíveis para locação.
* Gerenciar clientes: permite que um funcionário adicione, remova ou atualize as informações dos clientes que utilizam os serviços da locadora.
* Gerenciar reservas: permite que um funcionário gerencie as reservas de carros feitas pelos clientes.
* Gerenciar aluguéis: permite que um funcionário gerencie os aluguéis de carros feitos pelos clientes.
* Relatórios: permite que um funcionário gere relatórios sobre as atividades da locadora, como número de carros alugados, clientes ativos, entre outros.

Diagrama

Descrição gerada automaticamente

**Fonte:** Os autores (2023)

**3.5 TESTES**

Foram realizados alguns testes no sistema. Os testes realizados no sistema incluíram:

**3.5.1 TESTES UNITÁRIOS**

Foram implementados testes unitários para as classes de domínio e para as classes que fazem a persistência dos dados no banco de dados. Esses testes verificam se as funcionalidades dessas classes estão sendo implementadas corretamente.

**3.5.2 TESTES DE INTEGRAÇÃO**

Foram implementados testes de integração para as classes que fazem a integração entre as classes de domínio e as classes que fazem a persistência dos dados no banco de dados. Esses testes verificam se a integração está sendo feita corretamente e se as funcionalidades estão sendo implementadas corretamente.

**3.5.3 TESTES DE API**

Foram implementados testes de API para as classes que fazem a comunicação entre o front-end e o back-end. Esses testes verificam se as APIs estão respondendo corretamente às requisições do front-end.

1. **ARQUITETURA**

A arquitetura da aplicação é baseada em uma abordagem em camadas, que separa a lógica de negócios, apresentação e acesso a dados em diferentes componentes. A aplicação segue o modelo de arquitetura em camadas padrão, onde cada camada é responsável por um conjunto específico de tarefas.

Abaixo está a descrição das camadas da aplicação:

* Camada de apresentação: Responsável por apresentar a interface gráfica do usuário. É implementado usando tecnologias como Vue.js e HTML/CSS.
* Camada de negócios: Responsável por implementar a lógica de negócios da aplicação. É implementado usando tecnologias Java e Spring.
* Camada de acesso a dados: Responsável por acessar os dados do sistema de armazenamento de dados. É implementado usando tecnologias Java e Spring.

Essa abordagem permite que cada camada seja desenvolvida e testada separadamente, aumentando a modularidade e a escalabilidade da aplicação.

**4.1 DIAGRAMA DE CLASSES**

Diagrama

Descrição gerada automaticamente

1. **RESULTADOS**

Aqui está um pequeno protótipo da página inicial do site, ela contém informações como: Contato, Sobre nós e Login (onde será direcionado para uma pagina de login tanto para cliente como para funcionário.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. **CONSIDERAÇÕES FINAIS**

As considerações finais apresentam uma reflexão sobre o processo de desenvolvimento do sistema, destacando seus pontos positivos e possíveis melhorias. É importante mencionar a importância de um processo de desenvolvimento estruturado e organizado, que permitiu a criação de um sistema funcional e eficiente. Além disso, pode-se destacar a importância dos testes realizados, que garantiram a qualidade do sistema e a detecção de possíveis problemas.

Entre as possíveis melhorias, pode-se destacar a implementação de mais funcionalidades, como um sistema de pagamento online e um sistema de avaliação dos clientes.

Em resumo, o desenvolvimento desse sistema de locação de carros foi uma experiência enriquecedora, que permitiu a aplicação de conhecimentos teóricos em um projeto prático. Através do trabalho em equipe e da aplicação de boas práticas de desenvolvimento, foi possível entregar um sistema funcional e de qualidade, que atende às necessidades dos usuários e pode ser utilizado como base para futuros desenvolvimentos e melhorias.