PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS - INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA - UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE

Bacharelado em Engenharia de Software

Disciplinas: Fundamentos de Engenharia de Software e Algoritmos e Estrutura de dados.

Alunos: Juliana Serra, Matheus Pessoa, Sophia Rocha

Projeto: Locadora Locamais

- **APRESENTAÇÃO:**

Criamos o sistema de uma locadora de veículos, em que o objetivo era atender às diversas demandas de uma empresa de aluguel de carros, como exemplos, temos o cadastro de clientes, veículos e locações, a listagem de todos os clientes e de todos os veículos cadastrados na plataforma, além disso, implementamos uma função extra que lista apenas os veículos que estão disponíveis para serem alugados, acreditamos que possa ser útil para os funcionários da empresa no momento de cadastrar uma locação.

- **BACKLOG DO PRODUTO E SUA EVOLUÇÃO:**

Para a organização do nosso projeto, separamos nossas tarefas em 5 sprints com duração de 2 a 3 dias. Na primeira sprint, realizamos a divisão de papéis, ou seja, dividimos pelo quê cada integrante do grupo seria responsável no decorrer do projeto e, além disso, distribuímos essas funções entre as próximas sprints.

Em seguida, na segunda sprint, criamos a estrutura do nosso programa, organizando não só a main e suas funções como também o funcionamento do loop e a estrutura geral da interface do projeto. Nas sprints 3 e 4, cada integrante do grupo ficou responsável por desenvolver suas respectivas funções - que estão detalhadas neste documento- levando os erros e bugs encontrados pelos testes para serem solucionadas na sprint seguinte.

Por fim, na quinta sprint, realizamos mais testes e fizemos o aperfeiçoamento de nosso programa, promovendo melhorias na interface e solucionando alguns erros que ainda estavam presentes, de modo a preparar nosso software para a entrega final. Além disso, na quinta sprint, elaboramos a documentação de nosso projeto, para expor como nosso grupo se organizou, para detalhar nossas funções e também para explor nossos planos de testes.

- **LISTA DE ASSINATURA DAS FUNÇÕES E PARÂMETROS**

Cadastrar clientes - Sophia

Método de assinatura: void cadastraCliente()

Função para cadastrar um novo cliente no banco de dados da aplicação. Não recebe nenhum parâmetro. Adiciona um novo elemento à estrutura sClientes.

Cadastrar veículos - Matheus

Método de assinatura: void cadastraVeiculo()

Função para cadastrar um novo veículo no banco de dados da aplicação. Não recebe nenhum parâmetro. Adiciona um novo elemento à estrutura sVeiculos.

Mostrar veículos livres - Matheus

Método de assinatura: void printVeiculosLivres()

Não recebe nenhum parâmetro. Função para listar os veículos que possuem o status diferente de “Alugado”.

Cadastrar locação - Matheus

Método de assinatura: void cadastraLoc()

Não recebe nenhum parâmetro. Função para cadastrar uma nova locação no banco de dados da aplicação. Modifica o status do veículo para “Alugado” no vetor vetVeiculos e adiciona o veículo ao vetor vetLocacao.

Carregar banco de dados clientes - Juliana

Método de assinatura: void carregaClientes();

Função para carregar e ler um arquivo .txt com o banco de dados de clientes. A função não recebe nenhum parâmetro e salva os dados obtidos no banco de dados no vetor vetClientes. O objetivo da função é facilitar a manipulação dos dados do arquivo sem precisar acessá-lo sucessivas vezes.

Carregar banco de dados veículos - Juliana

Método de assinatura: void carregaVeiculos();

Função para carregar e ler um arquivo .txt com o banco de dados de veículos. A função não recebe nenhum parâmetro e salva os dados obtidos no banco de dados no vetor vetVeiculos. O objetivo da função é facilitar a manipulação dos dados do arquivo sem precisar acessá-lo sucessivas vezes.

Carregar banco de dados locações - Juliana

Método de assinatura: void carregaVeiculos();

Função para carregar e ler um arquivo .txt com o banco de dados de locações. A função não recebe nenhum parâmetro e salva os dados obtidos no banco de dados no vetor vetLocacoes. O objetivo da função é facilitar a manipulação dos dados do arquivo sem precisar acessá-lo sucessivas vezes.

Listar veículos - Juliana

Método de assinatura: void printVeiculos();

Não recebe nenhum parâmetro. Função para listar os veículos do vetor vetVeiculos.

Listar clientes - Juliana

Método de assinatura: void printClientes();

Não recebe nenhum parâmetro. Função para listar os clientes do vetor vetClientes.

Principais entidades:

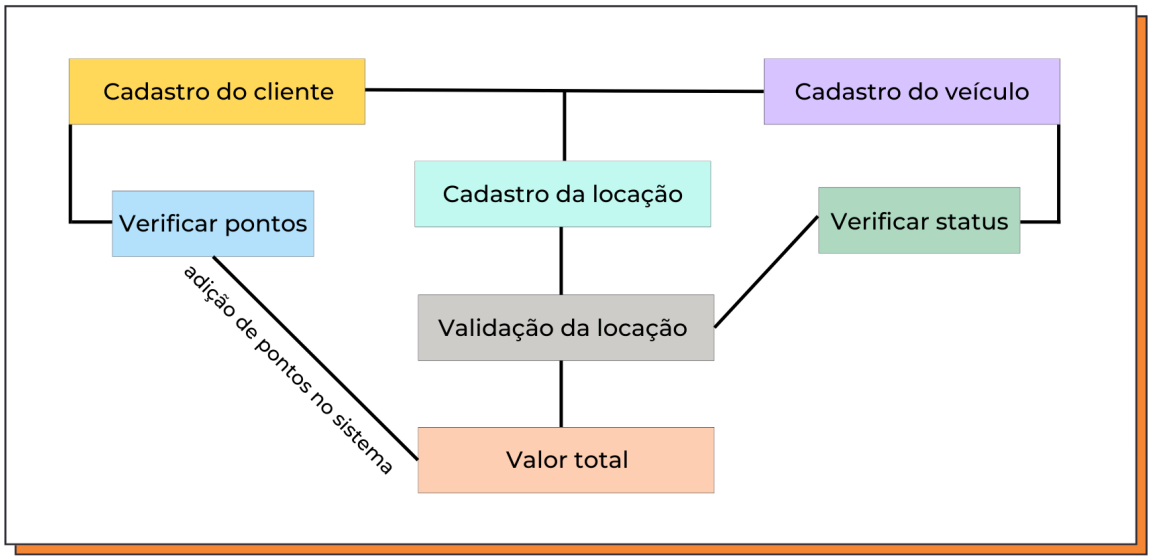
Cliente:
- Código
- Nome
- CNH
- Telefone
- Endereço

Veículo:
- Código
- Placa
- Marca
- Modelo
- Lugares
- Categoria
- Cor
- Valor diária
- Status
- Qtd pontos

Locação:
- Código
- Data retirada
- Data devolução
- Seguro
- Qtd dias
- Valor total

Regras:

- Atraso
- Valor do seguro
- Sistema de pontos



.

- **PLANO DE TESTES**

carregaClientes()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	Arquivo 'clientes.txt'	Salva os dados do arquivo no vetor vetClientes	*Não é necessária nenhuma entrada	

carregaVeiculos()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	Arquivo ‘veiculos.txt'	Salva os dados do arquivo no vetor vetVeiculos	*Não é necessária nenhuma entrada	

carregaLocacao()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	Arquivo ‘locacoes.txt'	Salva os dados do arquivo no vetor vetLocacoes	*Não é necessária nenhuma entrada	

printVeiculos()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	*Não é necessário nenhum parâmetro	Lista os veículos presentes no vetor VetVeiculos	*Não é necessária nenhuma entrada	

pesquisaCliente()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
código do cliente	Número inteiros e que estejam contidos no vetor.	Retorna o nome, cnh, telefone e endereço do cliente	código inexistente, números não inteiros, letras e outros caracteres.	Mensagem: "Código não encontrado"

pesquisaVeiculo()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
código do veículo	Número inteiros e que estejam contidos no vetor.	Retorna a placa, marca, modelo, ano, categoria e status do veículo	código inexistente, números não inteiros, letras e outros caracteres.	Mensagem: "Código não encontrado"

printVeiculosLivres()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	*Não é necessário nenhum parâmetro	Lista os veículos presentes no vetor VetVeiculos que não possuem o status "Alugado"	*Não é necessária nenhuma entrada	

printClientes()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
*Não é necessária nenhuma entrada	*Não é necessário nenhum parâmetro	Lista os clientes presentes no vetor vetClientes	*Não é necessária nenhuma entrada	

cadastraVeiculo()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
modelo: string cor: string placa: string status: string categora: string marca: string ano: string qtdOcupantes: string qtdAssentos: string	modelo: string tamanho 100 cor: string tamanho 100 placa: string tamanho 7 status: string tamanho 100 categora: string tamanho 100 marca: string tamanho 100 ano: string tamanho 5 qtdOcupantes: string tamanho 10 qtdAssentos: string tamanho 10	Insere o veículo no vetor vetVeiculos	strings maiores do que o esperado	Erro na compilação

cadastraCliente()

Entradas	Classes válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
nome: string telefone: string cnh: string endereço: string	nome: string tamanho 30 telefone: string tamanho 20 cnh: string tamanho 20 endereço: string tamanho 80	Insere o veículo no vetor vetCliente	strings maiores do que o esperado	Erro na compilação

- **RELATÓRIO DE EXECUÇÃO DE TESTES**

Entrada	Classes válidas	Resultado	Classes inválidas	Resultado Esperado
Opção: Numero inteiro	Opções escolhidas entre 0 e 9	Entrada na função correspondente ao número digitado pelo usuário	Números menores que 0 ou maiores que 9, letras e caracteres especiais	Mensagem: opção inválida.

Relatório de execução do teste		
Entradas	Resultado	Aprovado?
Valor: 1	Entra na função de cadastrar um veículo	Sim
Valor: 2	Entra na função de cadastrar um usuário	Sim
Valor: 3	Entra na função de cadastrar uma locação	Sim
Valor: 4	Entra na função de validar uma locação	Sim
Valor: 5	Entra na função de pesquisar um cliente	Sim
Valor: 6	Entra na função de listar todos os clientes	Sim
Valor: 7	Entra na função de pesquisar um veículo	Sim
Valor: 8	Entra na função de listar todos os veículos	Sim
Valor: 9	Entra na função de listar todos os veículos livres	Sim
Valor: 0	SAIR. Encerra o programa	Sim