

Documentação de Software

**Documentação do Software
SLEM Logística**

Autores:

André Fortini de Mello
Larissa Monção de Oliveira
Lucas Maia Marques Pinheiro
Sofia Figueiredo de Oliveira

Belo
Horizonte
Junho de 2025.

Documentação do Software

Documentação do Software.....	2
Introdução.....	3
Escopo do software.....	3
Nome do sistema e de seus componentes principais	3
Missão ou objetivo do software.....	3
Descrição do domínio do Cliente (Regras de Negócio).....	3
Funções do produto / Backlog do produto com Histórias de usuário	3
Usuários e sistemas externos	4
Descrição	4
Documentação do código	5
Documentação da Estrutura de dados geral do software	5
Função restaurarDados()	5
Função menu()	3
Função cadastrarLocal()	5
Função listarLocal()	5
Função apagarLocal()	5
Função cadastrarLocal()	6
Função cadastrarVeiculo()	6
Função listarVeiculo()	6
Função editarVeiculo()	7
Função apagarVeiculo()	7
Função cadastrarPedido()	7
Função listarPedido()	7
Função editarPedido()	8
Função apagarPedido()	8
Função distancia()	8
Função menuLocal()	8
Função menuVeiculo()	8
Função menuPedido()	9
Função salvarDados()	9
Testes do software.....	10
Casos de teste do software: função restaurarDados().....	10
Casos de teste do software: função menu().....	10

Introdução

Escopo do software

O sistema SLEM foi desenvolvido com o objetivo de gerenciar e monitorar entregas, veículos, locais de destino e pedidos. A aplicação permite o cadastro de dados, consulta e manipulação de informações relacionadas ao processo logístico.

Nome do sistema e de seus componentes principais

- Nome do sistema: SLEM

Componentes principais:

- Cadastro e gerenciamento de locais
- Cadastro e gerenciamento de veículos
- Cadastro e gerenciamento de pedidos
- Interface de menu com funcionalidades integradas
- Funções de persistência de dados

Missão ou objetivo do software

A missão da SLEM é otimizar a logística de entregas por meio do controle eficiente de locais, veículos e pedidos, proporcionando organização e rastreabilidade

Descrição do domínio do Cliente (Regras de Negócio)

Número	Regra de Negócio	Descrição
1	Cadastro de Locais	O sistema deve permitir o cadastro de locais com nome e coordenada
2	Cadastro de Veículos	O sistema deve permitir o cadastro com placa, modelo e autonomia
3	Cadastro de Pedidos	O sistema deve associar pedidos a locais e veículos disponíveis
4	Persistência de Dados	Todos os dados cadastrados devem ser restaurados automaticamente
5	Interface de Menu	O sistema deve apresentar um menu responsivo com todas as funções
6	Distância Geográfica	O sistema deve calcular a distância com base nas coordenadas
7	Validação de Entrada	O sistema deve validar corretamente a entrada de dados

Funcionalidades do produto

Número	Funcionalidade do sistema
1	Cadastrar locais
2	Cadastrar veículos
3	Cadastrar pedidos
4	Restaurar dados salvos
5	Acessar menu principal (e de cada item) com todas as funcionalidades

Usuários e sistemas externos

Descrição

<i>Número</i>	<i>Usuários</i>	<i>Definição</i>
1	Administrador	Pessoa responsável por cadastrar locais, veículos e pedidos
2	Operador Logístico	Utiliza o sistema para acompanhar entregas e consultar dados
3	Sistema de Local	Utiliza arquivos locais para salvar e restaurar dados

Documentação do código***Documentação da Estrutura de dados geral do software***

Local: struct que representa um local de entrega, com nome e coordenadas (x,y)

Veículo: struct que representa um veículo com placa, modelo e autonomia

Pedido: struct que associa um pedido a um local e um veículo, contendo informações do cliente, ID do local e ID do veículo

Os dados são armazenados em vetores globais:

Locais - local[MAX]

Veículo - veiculos[MAX]

Pedido - pedidos[MAX]

Função restaurarDados()

- **Assinatura:** void restaurarDados(void);

- **Descrição:** Restaura os dados dos locais, veículos e pedidos a partir de arquivos binários salvos anteriormente

- **Parâmetros:** nenhum

- **Retorno:** void (sem retorno)

Função menu()

- **Assinatura:** void menu(void);

- **Descrição:** Exibe o menu principal e redireciona para as funcionalidades do sistema de acordo com as escolhas do usuário

- **Parâmetros:** nenhum

- **Retorno:** void (sem retorno)

Função cadastrarLocal()

- **Assinatura:** void cadastrarLocal(void)

- **Descrição:** Esta função cadastra um novo local no sistema de gestão logística, armazenando seu nome e coordenadas geográficas (X, Y).

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função listarLocal()

- **Assinatura:** void listarLocais(void)

- **Descrição:** Lista todos os locais cadastrados no sistema, exibindo índice, nome e coordenadas (X, Y) de cada local. Se nenhum local estiver cadastrado, exibe mensagem informativa.

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função editarLocal()

- **Assinatura:** void editarLocal(void);

- **Descrição:** Permite editar um local existente. Lista todos os locais cadastrados, solicita o índice do local a ser editado, valida o índice e permite atualizar o nome e coordenadas (X, Y) do local selecionado.

- **Parâmetros:** nenhum

- **Retorno:** void (sem retorno)

Função menu()

- **Assinatura:** void menu(void);

- **Descrição:** Exibe o menu principal e redireciona para as funcionalidades do sistema de acordo com as escolhas do usuário

- **Parâmetros:** nenhum

- **Retorno:** void (sem retorno)

Função cadastrarLocal()

- **Assinatura:** void cadastrarLocal(void)

- **Descrição:** Esta função cadastra um novo local no sistema de gestão logística, armazenando seu nome e coordenadas geográficas (X, Y). Parâmetros: nenhum

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função listarLocais()

- **Assinatura:** void listarLocais(void)

- **Descrição:** Lista todos os locais cadastrados no sistema, exibindo índice, nome e coordenadas (X, Y) de cada local. Se nenhum local estiver cadastrado, exibe mensagem informativa.

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função apagarLocal()

- **Assinatura:** void apagarLocal(void)

- **Descrição:** Remove um local existente do sistema. Lista todos os locais cadastrados, solicita o índice do local a ser removido, valida o índice e, se válido, remove o local deslocando os elementos subsequentes no array para preencher a posição removida. Atualiza o contador de locais.

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função cadastrarVeiculo()

- **Assinatura:** void cadastrarVeiculo()

- **Descrição:** Cadastra um novo veículo no sistema. Verifica se existem locais cadastrados e se o limite de veículos não foi atingido. Solicita placa, modelo, define status inicial como disponível (0) e associa a um local válido.

- **Parâmetros:** nenhum

- **Retorno:** void (não retorna valor)

Função listarVeiculo()

- **Assinatura:** void listarVeiculo(void)
- **Descrição:** Lista todos os veículos cadastrados no sistema, exibindo índice, placa, modelo, status ("Disponível" ou "Ocupado") e índice do local associado. Se nenhum veículo estiver cadastrado, exibe mensagem informativa.
- **Parâmetros:** nenhum
- **Retorno:** void (não retorna valor)

Função editarVeiculo()

- **Assinatura:** void editarVeiculo(void);
- **Descrição:** Permite editar os dados de um veículo existente. Lista todos os veículos, solicita o índice do veículo a ser editado, valida o índice e permite atualizar placa, modelo, status e local associado. Valida o novo local antes de atualizar.
- **Parâmetros:** nenhum
- **Retorno:** void (não retorna valor)

Função apagarVeiculo()

- **Assinatura:** void apagarVeiculo(void)
- **Descrição:** Remove um veículo existente do sistema. Lista todos os veículos cadastrados, solicita o índice do veículo a ser removido, valida o índice e, se válido, remove o veículo deslocando os elementos subsequentes no array para preencher a posição removida. Atualiza o contador de veículos.
- **Parâmetros:** nenhum
- **Retorno:** void (não retorna valor)

Função cadastrarPedido()

- **Assinatura:** void cadastrarPedido(void)
- **Descrição:** Cadastra um novo pedido no sistema. Verifica se o limite máximo de pedidos foi atingido e se existem locais cadastrados. Atribui um ID único, solicita local de origem, local de destino e peso do pedido, e incrementa o contador global de pedidos.
- **Parâmetros:** nenhum (opera em variáveis globais)
- **Retorno:** void (não retorna valor)

Função listarPedido()

- **Assinatura:** void listarPedido(void)
- **Descrição:** Lista todos os pedidos cadastrados no sistema, exibindo índice, ID, origem (nome do local), destino (nome do local) e peso. Se nenhum pedido estiver cadastrado, exibe mensagem informativa. Valida automaticamente os índices de origem e destino para evitar acesso inválido a locais.
- **Parâmetros:** nenhum (opera em variáveis globais)
- **Retorno:** void (não retorna valor)

Função editarPedido()

- **Assinatura:** void editarPedido()
- **Descrição:** Permite editar um pedido existente. Lista todos os pedidos cadastrados, solicita o índice do pedido a ser editado, valida o índice e permite atualizar o local de origem, local de destino e peso do pedido selecionado.
- **Parâmetros:** nenhum (opera em variáveis globais)
- **Retorno:** void (não retorna valor)

Função listarVeiculo()

- **Assinatura:** void apagarPedido(void)
- **Descrição:** Remove um pedido existente do sistema. Lista todos os pedidos cadastrados, solicita o índice do pedido a ser removido, valida o índice e, se válido, remove o pedido deslocando os elementos subsequentes no array para preencher a posição removida. Atualiza o contador global de pedidos.
- **Parâmetros:** nenhum (opera em variáveis globais)
- **Retorno:** void (não retorna valor)

Função distancia()

Assinatura: float distancia(int i, int j);

Descrição: Calcula a distância euclidiana entre dois locais usando suas coordenadas (X, Y) armazenadas no sistema. A função acessa diretamente o array global locais usando os índices fornecidos.

Parâmetros:

i, tipo int, índice do primeiro local no array locais

j, tipo int, índice do segundo local no array locais

Retorno: float (distância calculada entre os locais)

Função calcularRota()

- **Assinatura:** void calcularRota(void)
- **Descrição:** Calcula a rota ideal para um pedido, selecionando o veículo disponível mais próximo do local de origem do pedido. Solicita o ID do pedido, valida suas informações e encontra o veículo ótimo com base na distância euclidiana.
- **Parâmetros:** void (não retorna valor)
- **Retorno:** void (não retorna valor)

Função menuLocal()

Assinatura: void menuLocais(void)

Descrição: Exibe e gerencia um menu interativo para operações relacionadas a locais. Oferece opções para cadastrar, listar, editar e apagar locais, além de permitir retornar ao menu anterior. O menu permanece ativo até que o usuário selecione a opção "Voltar" (0).

Parâmetros: nenhum

Retorno: void (não retorna valor)

Função menuVeiculo()

Assinatura: void menuVeiculos(void)

Descrição: Exibe e gerencia um menu interativo para operações relacionadas a veículos. Oferece opções para cadastrar, listar, editar e apagar veículos, além de permitir retornar ao menu anterior. O menu permanece ativo até que o usuário selecione a opção "Voltar" (0).

Parâmetros: nenhum (opera em variáveis globais)

Retorno: void (não retorna valor)

Função menuPedido()

- **Assinatura:** void menuPedidos()
- **Descrição:** Exibe e gerencia um menu interativo para operações relacionadas a pedidos. Oferece opções para cadastrar, listar, editar, apagar pedidos e calcular rotas, além de permitir retornar ao menu anterior. O menu permanece ativo até que o usuário selecione "Voltar" (0).
- **Parâmetros:** nenhum (opera em variáveis globais)
- **Retorno:** void (não retorna valor)

Função salvarDados()

- **Assinatura:** void salvarDados()
- **Descrição:** Salva todos os dados do sistema em arquivos binários separados para locais, veículos e pedidos. Cada arquivo armazena primeiro a quantidade de registros seguida pelos próprios registros. Exibe uma mensagem de confirmação após a conclusão do processo.
- **Parâmetros:** nenhum
- **Retorno:** void (não retorna valor)

Testes do software

Casos de testes do software: função restaurarDados()

Número	Varáveis de Entrada	Valores válidos	Resultado Esperado	Valores inválidos	Resultado Esperado
1	Arquivos de dados	.dats existentes	Dados carregados	Arquivo não existe	Vetores vazios
2	Estrutura de dados	Structs compatíveis	Vetores preenchidos	Estrutura inválida	Ignora ou trata erro
3	Qnt. de registros	Até MAX registros	Registros restaurados	Acima limite MAX	Exibe alerta
4	Dados parciais	Registros válidos	Dados restaurados	Formatação inválida	Erro de leitura
...					

Casos de testes do software: função menu()

Número	Varáveis de Entrada	Valores válidos	Resultado Esperado	Valores inválidos	Resultado Esperado
1	Opção escolhida	1 a N	Executa função	Letra/num inválido	Solicita entrada
2	Fluxo completo	Navegar nas funções	Menu de cada função	Fora do intervalo	Mensagem de erro
3	Repetição comandos	Números repetidos	Executa a função	Símbolos especiais	Solicita entrada
4	Entrada c/ espaços	"1" ou "2"	Ignora espaços	Entrada vazia	Solicita entrada
...					