

# Análise, Modelagem e Projeto Orientados a Serviços para Provedor de Logística de E-commerce

Este documento apresenta a análise, modelagem e projeto orientado a serviços para o provedor de logística do projeto de e-commerce proposto na disciplina DM112, tendo como único insumo para análise o comando do Trabalho 1.

## 1. Análise Orientada a Serviços

### 1.1. Identificação dos Serviços

**Serviço de Consulta de Pedidos:** Permite que o entregador consulte a lista de pedidos em situação de logística. Consideramos pedidos em situação de logística, aqueles que já passaram pelas etapas anteriores, como pagamento, e se encontram na etapa de logística. Seriam os pedidos prontos para serem entregues.

**Serviço de Registro de Entrega:** Permite que o entregador registre a entrega de um pedido, incluindo informações como identificação do pedido, CPF da pessoa que recebeu e data/hora da entrega.

Para fins didáticos, consideramos que só ocorre uma única ação de entrega por pedido, ou seja, cada pedido pode gerar uma única entrega, não havendo mais de uma entrega por pedido.

**Serviço de Notificação por E-mail:** O sistema envia um e-mail de notificação para o cliente quando o pedido é entregue.

### 1.2. Definição de escopo

**Fronteira 1:** Interação entre o entregador e o sistema para consulta da lista de pedidos.

**Fronteira 2:** Interação entre o entregador e o sistema para registro da entrega do pedido.

**Fronteira 3:** Interação do sistema com o servidor de e-mails para envio de notificações.

## 1.3. Análise dos Requisitos

**Requisito Funcional 1:** O sistema deve fornecer uma interface para que o entregador consulte a lista de pedidos pendentes de entrega.

**Requisito Funcional 2:** O sistema deve permitir que o entregador registre a entrega de um pedido, coletando as informações necessárias.

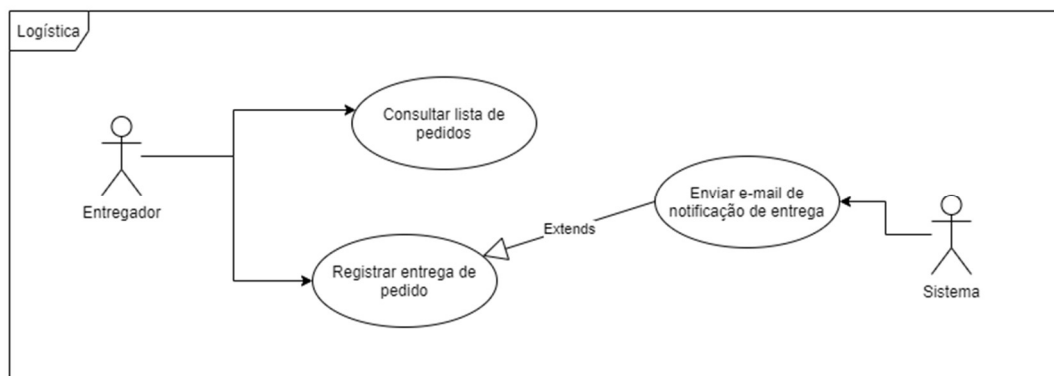
**Requisito Funcional 3:** O sistema deve integrar-se com um servidor de e-mails para enviar notificações aos clientes sobre a entrega de seus pedidos.

## 2. Modelagem Orientada a Serviços

### 2.1. Diagrama de Casos de Uso

Atores: Entregador, Sistema

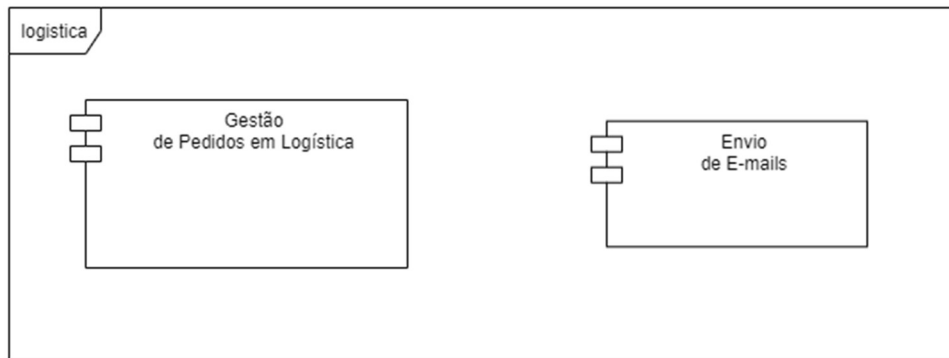
Casos de Uso: Consultar Lista de Pedidos, Registrar Entrega de Pedido, Enviar Email de Notificação.



### 2.2. Diagrama de Componentes

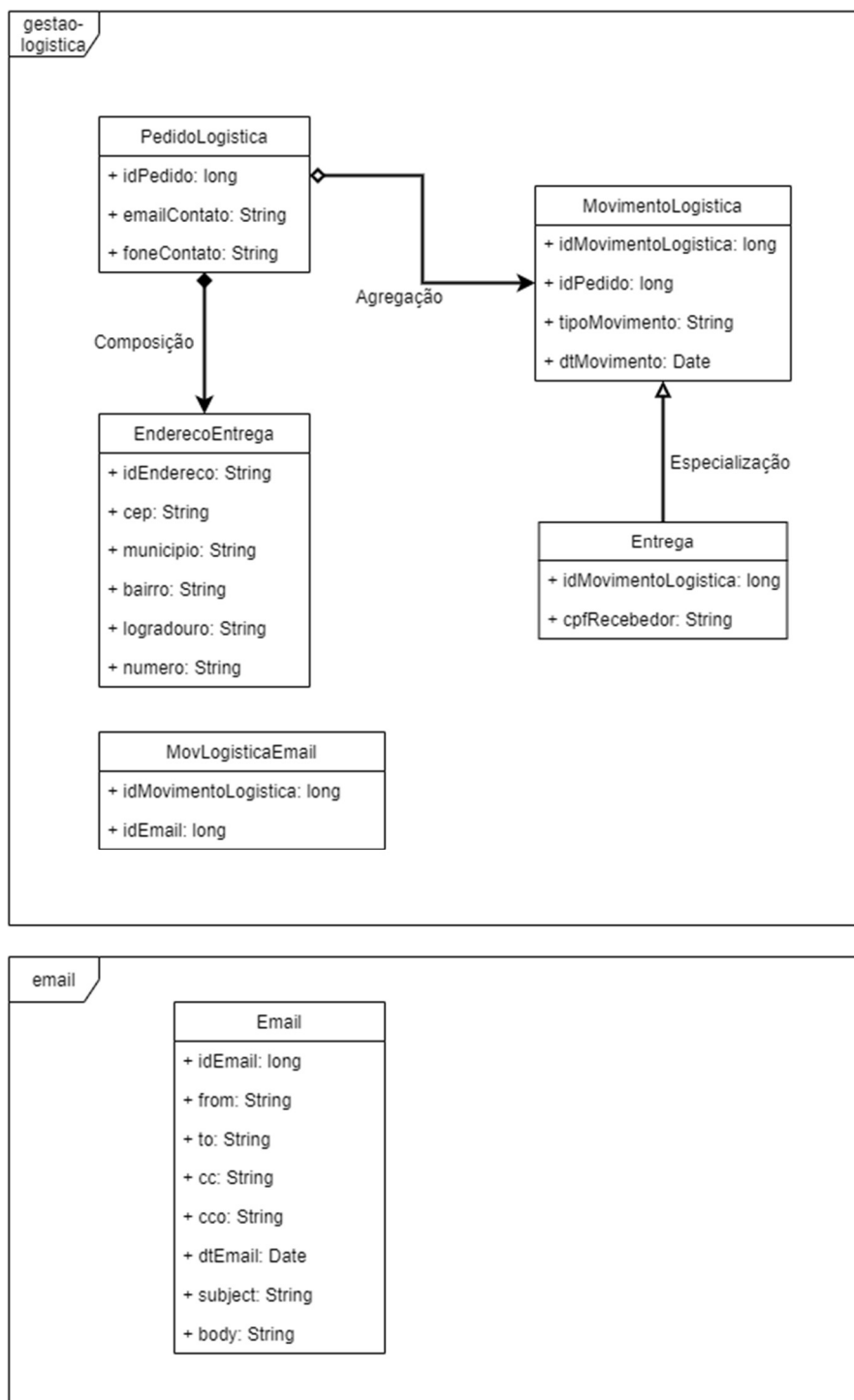
**Gestão de Pedidos em Logística:** Componente contempla os casos de uso de Consultar Lista de Pedidos e Registrar Entrega de Pedido. Esse componente incluir recuperação, validação e armazenamento de dados. Provê a interação entre o entregador e o sistema.

**Envio de E-mails:** Componente genérico que contém a integração com o provedor de e-mail e a implementação para dar suporte ao envio de e-mails pelo sistema.



## 2.3. Modelo de Dados

A entidade **PedidoLogistica** tem um **EnderecoEntrega** e representa um pedido em situação de logística, mas não a totalidade dos dados do pedido antes de entrar na situação de logística. Essa entidade pode ter um **MovimentoLogistica** que por sua vez representa os vários tipos de movimento que podem ocorrer em logística, como entrega, devolução e outros e contém a data em que o movimento ocorreu. MovimentoLogistica tem uma especialização chamado **Entrega** que representa a entrega propriamente dita, sendo a entrega um tipo de MovimentoLogistica. A entrega tem o registro do CPF do receptor.



### 3. Projeto Orientado a Serviços

#### 3.1. Especificação dos Serviços

Serviço de Consulta de Pedidos:

Operação: consultarPedidos

Entrada: Nenhuma

Saída: Lista de pedidos (idPedido, municipio, bairro, logradouro, numero, cep)

Serviço de Registro de Entrega:

Operação: registrarEntrega

Entrada: idPedido, dtMovimento, cpfReebedor

Saída: idMovimentoLogistica

Serviço de Notificação por E-mail:

Operação: enviarEmail

Entrada: from, to, subject, body

Saída: idEmail

## 3.2. Arquitetura da Solução

Implementação de uma API REST para os serviços de consulta e registro.

Integração com um serviço de envio de e-mails (ex: AWS SES, SendGrid).

Banco de dados para armazenar os dados de pedidos e entregas.

## 3.3. Stack de Tecnologias

Linguagem de programação: Java (com Spring e Spring Boot).

Banco de dados: MySQL.

Serviço de e-mail: AWS SES, SendGrid, Mailgun ou SMTP do Gmail.

## Considerações Finais

Este projeto orienta a criação de uma solução modular e escalável, facilitando a manutenção e a integração com outros sistemas. A abordagem orientada a serviços

garante a flexibilidade e a reutilização dos componentes, otimizando o processo de desenvolvimento, testes e entrega, assim como a sustentação da solução em produção.