Modelagem do Banco de Dados para o Tracking Service

# 1. Estrutura das Tabelas

## 1.1 Packs

Guarda as informações dos pacotes rastreados.

|  |  |  |
| --- | --- | --- |
| Campo | Tipo de Dado | Descrição |
| id | BIGINT (PK) | ID único do pacote. |
| description | VARCHAR(255) | Descrição do conteúdo do pacote. |
| sender | VARCHAR(255) | Nome do remetente. |
| recipient | VARCHAR(255) | Nome do destinatário. |
| is\_holliday | BOOLEAN | Indica se a entrega será afetada por feriados. |
| fun\_fact | VARCHAR(255) NULL | Informações extras sobre o pacote. |
| estimated\_delivery\_date | VARCHAR(50) NULL | Data estimada da entrega. |
| status | ENUM | Status atual do pacote (CREATED, IN\_TRANSIT, DELIVERED). |
| delivered\_at | DATETIME NULL | Data e hora da entrega (se já entregue). |
| created\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP | Data de criação do registro. |
| updated\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP | Data da última atualização. |
| is\_active | BOOLEAN DEFAULT TRUE | Define se o pacote está ativo no sistema. |

📌 Índices:

* Índice no campo status para otimizar buscas por pacotes em determinado estado.

# 1.1.2 PackEvents

Armazena os eventos associados aos pacotes, permitindo rastreamento detalhado.

|  |  |  |
| --- | --- | --- |
| Campo | Tipo de Dado | Descrição |
| id | BIGINT (PK) | Identificador único do evento. |
| pack\_id | BIGINT (FK) | Identificador do pacote relacionado. |
| location | VARCHAR(255) | Local onde ocorreu o evento. |
| description | TEXT | Descrição do evento ocorrido. |
| event\_date\_time | DATETIME DEFAULT CURRENT\_TIMESTAMP | Data e hora do evento. |
| created\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP | Data de criação do registro. |
| updated\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP | Data da última atualização. |
| is\_active | BOOLEAN DEFAULT TRUE | Define se o evento está ativo. |

## 📌 Índices:

Índice no campo pack\_id (idx\_pack\_events\_pack\_id) para otimizar buscas de eventos por pacote.

## 📌 Relacionamento:

pack\_id é uma chave estrangeira referenciando packs(id), com on delete cascade.

# 1.1.3 RequestLogs

Registra as requisições HTTP feitas ao sistema, permitindo auditoria e debug.

|  |  |  |
| --- | --- | --- |
| Campo | Tipo de Dado | Descrição |
| id | BIGINT (PK) | Identificador único do log. |
| method | VARCHAR(50) | Método HTTP (GET, POST, etc.). |
| endpoint | VARCHAR(255) | Endpoint requisitado. |
| request\_body | TEXT NULL | Corpo da requisição (se aplicável). |
| response\_body | TEXT NULL | Corpo da resposta (se aplicável). |
| status\_code | INT | Código de status HTTP da resposta. |
| created\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP | Data da requisição. |
| updated\_at | TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP | Última atualização do log. |
| is\_active | BOOLEAN DEFAULT TRUE | Indica se o log está ativo. |

Response to the user B\_5:

# 1.2 Estratégias de Persistência e Migração

O sistema utiliza Flyway para gerenciamento de migrações de banco de dados.

## 📌 Configuração Flyway:

* Benefícios:
* Garante controle de versões do esquema do banco.
* Facilita rollback e auditoria de mudanças.

# 1.3 Estratégia de Escalabilidade para Eventos de Rastreamento

## 📌 Processamento em Batch:

* O sistema implementa processamento em lote (batch processing) para otimizar a persistência dos eventos de rastreamento.
* Configuração do batch:
* A cada 1 minuto, 5.000 mensagens de eventos são consultadas e processadas em batch, reduzindo a sobrecarga no banco de dados e melhorando a escalabilidade.

## 📌 Otimizações:

* Fila de Mensagens: Permite processar eventos de forma assíncrona.
* Armazenamento de Logs: Garante rastreabilidade de cada requisição.

# 1.4 Gerenciamento de Conexões

O sistema utiliza HikariCP para gerenciar conexões de forma eficiente.

## 📌 Configuração do HikariCP:

* Benefícios:
* Garante maior eficiência no uso das conexões.
* Evita vazamento de conexões (leak-detection-threshold).
* Otimiza o tempo de resposta (connection-timeout e idle-timeout).

# 1.5 Abstração e Boas Práticas no Código

## 📌 Uso da EntityBase:

Todas as entidades estendem EntityBase, garantindo:

* Herança de Campos Comuns (id, createdAt, updatedAt, isActive).
* Execução automática de timestamps (@PrePersist e @PreUpdate).

## 📌 Classes Principais:

* PackEvent: Modela eventos de rastreamento.
* Pack: Modela pacotes e suas entregas.
* RequestLog: Modela registros de requisições HTTP.

## 🔹 Vantagens:

* Redução de código repetitivo.
* Maior coesão e reutilização de código.

# 1.6 Possíveis Melhorias Futuras

## Sugestões para Evolução

Mensageria escalável: Utilizar outras filas para garantir a escalabilidade da mensageria. Implementar mecanismos de validação para assegurar que mensagens foram lidas e salvas corretamente, ou lidas e recusadas, evitando perda de informações críticas.

Banco de dados: Introduzir um novo banco de dados dedicado ao armazenamento de dados arquivados, melhorando a eficiência e a organização do armazenamento de informações a longo prazo.

Implementação do Sonar: Integrar o Sonar para padronizar e melhorar a qualidade dos testes realizados pela equipe, garantindo consistência e cobertura adequada em todos os testes de código.

Escalabilidade: Utilizar Kubernetes para aumentar a escalabilidade do sistema, permitindo a adição dinâmica de instâncias conforme a demanda, o que melhora a disponibilidade e a resiliência dos serviços.

Observabilidade: Implementar ferramentas de observabilidade como Prometheus e Grafana para monitoramento contínuo, criação de alertas e definição de limites aceitáveis, garantindo a saúde do sistema e respostas rápidas a possíveis problemas.

Rate Limiting para APIs: Introduzir mecanismos de rate limiting para APIs, prevenindo ataques de DDoS e evitando sobrecarga no sistema, o que garante uma performance estável e confiável.

Implementação de Arquitetura baseada em Eventos: Adotar uma arquitetura baseada em eventos para melhorar a modularidade e a eficiência do sistema, permitindo um maior desacoplamento entre os componentes e uma mais fácil escalabilidade.

Melhoria dos testes: Investir na melhoria dos testes, aumentando a cobertura e a qualidade deles, o que resultará em um sistema mais robusto e confiável.

Implementação de segurança: Fortalecer a segurança do sistema, permitindo o acesso aos endpoints com base em autenticação por token e sessão, protegendo as informações sensíveis e garantindo a integridade dos dados.