Documentação Completa – FinIO API (Finance Input/Output)

Introdução

A **FinIO API (Finance Input/Output)** é um microserviço para controle de fluxo financeiro multiempresa. Permite o registro de **entradas**, **saídas**, **pagamentos extras** e **parcelas**, com classificação por categorias.

Este serviço é utilizado por sistemas que compartilham uma chave de autenticação JWT e identificam a empresa via `idEmpresa`. Toda a movimentação financeira gerenciada por este microserviço é isolada por empresa.

Objetivos Funcionais

- Cadastrar entradas financeiras (recorrentes ou únicas)
- Registrar saídas fixas, variáveis ou parceladas
- Classificar saídas (Impulso, Obrigatória, etc.)
- Inserir pagamentos extras avulsos
- Gerar relatórios de fluxo de caixa, por período e por classificação

---

Tecnologias Utilizadas

- Java 21 ou superior
- Spring Boot 3.x.x ou superior
- **Spring Web**, **Spring Data JPA**, **Jakarta Validation**
- **Spring Security + JWT** (autenticação com chave compartilhada)
- **Lombok** (boilerplate reduction)
- **Flyway** (versionamento do banco)
- **SpringDoc / OpenAPI** (Swagger UI)
- **PostgreSQL**
- **Maven ou Gradle**

Estrutura de Pastas (MVC + Interfaces)

```
bash
CopiarEditar
src/
└── main/
    ├── java/com/finio/
    │   ├── controller/         # REST Controllers
```

```
│   ├── service/            # Interfaces dos serviços
│   │   └── imp/            # Implementações dos serviços
│   ├── repository/         # JPA Repositories
│   ├── entity/             # Entidades JPA
│   │   └── enums/          # Enumerações
│   ├── dto/                    # Data Transfer Objects
│   ├── convert/                # DTO <-> Entity Converters
│   ├── config/             # Configurações do projeto
│   │   └── security/       # JWT e segurança
└── resources/
    ├── db/migration/       # Scripts Flyway
    ├── application.yml     # Configuração Spring Boot
    └── schema.sql          # Estrutura opcional (DDL inicial)
```

Dependências Necessárias

xml
CopiarEditar

```xml
<!-- Spring Web -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<!-- Spring Data JPA -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<!-- Validação -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>

<!-- JWT -->
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-api</artifactId>
  <version>0.11.5</version>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-impl</artifactId>
  <version>0.11.5</version>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-jackson</artifactId>
  <version>0.11.5</version>
  <scope>runtime</scope>
</dependency>

<!-- Lombok -->
```

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
</dependency>

<!-- Flyway -->
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>

<!-- Swagger/OpenAPI -->
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.0.4</version>
</dependency>

<!-- PostgreSQL Driver -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
</dependency>
```

Fluxos Funcionais

Entrada Recorrente

1. Cadastra-se uma entrada com frequência = MENSAL.
2. Sistema registra data prevista de recebimento.
3. Quando recebido, preenche `data_recebimento`.
4. Futuras entradas podem ser geradas automaticamente.

Saída Parcelada

1. Usuário define uma saída com `quantidade_parcelas`.
2. O valor total é dividido e armazenado em `parcela_saida`.
3. Cada parcela possui `data_vencimento`.
4. O pagamento preenche `data_pagamento`.

Pagamento Avulso

1. Registro isolado de uma despesa avulsa.
2. Classificado como “Impulso”, “Obrigatória”, etc.
3. Impacta diretamente nos relatórios de comportamento financeiro.

Relatórios

1. Entradas e saídas são agrupadas por período.

2. Exibição de saldos, totais por categoria e classificação.
3. Pode ser usado em dashboards externos (ex: Power BI).

Endpoints REST

Entradas

- `GET /api/entradas`
- `GET /api/entradas/{id}`
- `POST /api/entradas`
- `PUT /api/entradas/{id}`
- `DELETE /api/entradas/{id}`
- `GET /api/entradas/periodo`

Saídas

- `GET /api/saidas`
- `GET /api/saidas/{id}`
- `POST /api/saidas`
- `PUT /api/saidas/{id}`
- `DELETE /api/saidas/{id}`
- `GET /api/saidas/periodo`

Parcelas

- `GET /api/saidas/{id}/parcelas`
- `PUT /api/parcelas/{id}/registrar-pagamento`

Pagamentos Extras

- `GET /api/pagamentos-extras`
- `POST /api/pagamentos-extras`
- `PUT /api/pagamentos-extras/{id}`
- `DELETE /api/pagamentos-extras/{id}`

Grupos de Categoria

- `GET /api/grupos`
- `POST /api/grupos`
- `PUT /api/grupos/{id}`
- `DELETE /api/grupos/{id}`

Enums

- `GET /api/enums/frequencias`

- `GET /api/enums/classificacoes`
- `GET /api/enums/tipos-grupo`

Relatórios

- `GET /api/relatorios/fluxo-caixa`
- `GET /api/relatorios/resumo-mensal`
- `GET /api/relatorios/classificação`

`Script apoio`

```
-- Tabela de tipos de grupo (Entrada ou Saída)
CREATE TABLE enum_tipo_grupo (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   nome VARCHAR(50) NOT NULL UNIQUE
);
INSERT INTO enum_tipo_grupo (nome) VALUES
('ENTRADA'),
('SAIDA');


-- Tabela de frequências de entradas
CREATE TABLE enum_frequencia (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   nome VARCHAR(50) NOT NULL UNIQUE
);
INSERT INTO enum_frequencia (nome) VALUES
('UNICA'),
('QUINZENAL'),
('MENSAL');


-- Tabela de classificações de saídas
CREATE TABLE enum_classificacao (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   nome VARCHAR(50) NOT NULL UNIQUE
);
INSERT INTO enum_classificacao (nome) VALUES
('NECESSARIA'),
('OBRIGATORIA'),
('IMPULSO'),
('SUPÉRFLUA');



CREATE TABLE grupo_categoria (
```

```
  id BIGINT PRIMARY KEY AUTO_INCREMENT,
  descricao VARCHAR(100) NOT NULL,
  id_enum_tipo_grupo BIGINT NOT NULL,
  ativo BOOLEAN DEFAULT TRUE,
  FOREIGN KEY (id_enum_tipo_grupo) REFERENCES enum_tipo_grupo(id)
);
CREATE INDEX idx_grupo_categoria_tipo ON grupo_categoria(id_enum_tipo_grupo);

CREATE TABLE entrada (
  id BIGINT PRIMARY KEY AUTO_INCREMENT,
  idEmpresa BIGINT NOT NULL,
  id_grupo_categoria BIGINT NOT NULL,
  descricao VARCHAR(150),
  valor DECIMAL(10,2) NOT NULL,
  data_prevista DATE NOT NULL,
  data_recebimento DATE,
  id_enum_frequencia BIGINT NOT NULL,
  observacao TEXT,
  FOREIGN KEY (id_grupo_categoria) REFERENCES grupo_categoria(id),
  FOREIGN KEY (id_enum_frequencia) REFERENCES enum_frequencia(id)
);

CREATE INDEX idx_entrada_empresa ON entrada(idEmpresa);
CREATE INDEX idx_entrada_data_prevista ON entrada(data_prevista);
CREATE INDEX idx_entrada_grupo ON entrada(id_grupo_categoria);


CREATE TABLE saida (
  id BIGINT PRIMARY KEY AUTO_INCREMENT,
  idEmpresa BIGINT NOT NULL,
  id_grupo_categoria BIGINT NOT NULL,
  descricao VARCHAR(150) NOT NULL,
  valor_total DECIMAL(10,2),
  valor_parcela DECIMAL(10,2),
  quantidade_parcelas INT,
  data_compra DATE NOT NULL,
  data_cadastro DATE NOT NULL,
  data_pagamento DATE,
  id_enum_classificacao BIGINT,
  observacao TEXT,
  FOREIGN KEY (id_grupo_categoria) REFERENCES grupo_categoria(id),
  FOREIGN KEY (id_enum_classificacao) REFERENCES enum_classificacao(id)
);

CREATE INDEX idx_saida_empresa ON saida(idEmpresa);
CREATE INDEX idx_saida_data_compra ON saida(data_compra);
CREATE INDEX idx_saida_grupo ON saida(id_grupo_categoria);
CREATE INDEX idx_saida_classificacao ON saida(id_enum_classificacao);
```

```
-- Parcelas associadas a uma saída
CREATE TABLE parcela_saida (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   id_saida BIGINT NOT NULL,
   numero_parcela INT NOT NULL,
   valor DECIMAL(10,2) NOT NULL,
   data_vencimento DATE NOT NULL,
   data_pagamento DATE,
   observacao TEXT,
   FOREIGN KEY (id_saida) REFERENCES saida(id)
);

CREATE INDEX idx_parcela_saida_id_saida ON parcela_saida(id_saida);
CREATE INDEX idx_parcela_saida_vencimento ON parcela_saida(data_vencimento);



-
CREATE TABLE pagamento_extra (
   id BIGINT PRIMARY KEY AUTO_INCREMENT,
   idEmpresa BIGINT NOT NULL,
   descricao VARCHAR(150) NOT NULL,
   valor DECIMAL(10,2) NOT NULL,
   data_pagamento DATE NOT NULL,
   id_grupo_categoria BIGINT,
   id_enum_classificacao BIGINT,
   observacao TEXT,
   FOREIGN KEY (id_grupo_categoria) REFERENCES grupo_categoria(id),
   FOREIGN KEY (id_enum_classificacao) REFERENCES enum_classificacao(id)
);

CREATE INDEX idx_pagamento_extra_empresa ON pagamento_extra(idEmpresa);
CREATE INDEX idx_pagamento_extra_classificacao ON
pagamento_extra(id_enum_classificacao);
```