### API Java para Integração com o Transfere Gov

Vou criar uma API em Java usando Spring Boot para integrar com o sistema Transfere Gov, abrangendo as funcionalidades solicitadas: pagamento de bolsistas, projetos, operações financeiras e compras.

## Estrutura do Projeto

```java file="pom.xml"

...

```

## Aplicação Principal

```java file="src/main/java/com/api/transferegov/TransfereGovApplication.java"

...

```

## Configuração de Segurança

```java file="src/main/java/com/api/transferegov/config/SecurityConfig.java"

...

```

## Configuração do Cliente TransfereGov

```java file="src/main/java/com/api/transferegov/config/TransfereGovConfig.java"

...

```

## Modelos (DTOs)

### Bolsista

```java file="src/main/java/com/api/transferegov/model/Bolsista.java"

...

```

### Projeto

```java file="src/main/java/com/api/transferegov/model/Projeto.java"

...

```

### Transação Financeira

```java file="src/main/java/com/api/transferegov/model/TransacaoFinanceira.java"

...

```

### Compra

```java file="src/main/java/com/api/transferegov/model/Compra.java"

...

```

## Repositórios

```java file="src/main/java/com/api/transferegov/repository/BolsistaRepository.java"

...

```

```java file="src/main/java/com/api/transferegov/repository/ProjetoRepository.java"

...

```

```java file="src/main/java/com/api/transferegov/repository/TransacaoFinanceiraRepository.java"

...

```

```java file="src/main/java/com/api/transferegov/repository/CompraRepository.java"

...

```

## Serviços

### Cliente TransfereGov

```java file="src/main/java/com/api/transferegov/service/TransfereGovClient.java"

...

```

### Serviço de Bolsistas

```java file="src/main/java/com/api/transferegov/service/BolsistaService.java"

...

```

### Serviço de Projetos

```java file="src/main/java/com/api/transferegov/service/ProjetoService.java"

...

```

### Serviço Financeiro

```java file="src/main/java/com/api/transferegov/service/FinanceiroService.java"

...

```

### Serviço de Compras

```java file="src/main/java/com/api/transferegov/service/CompraService.java"

...

```

## Controllers

### Controller de Bolsistas

```java file="src/main/java/com/api/transferegov/controller/BolsistaController.java"

...

```

### Controller de Projetos

```java file="src/main/java/com/api/transferegov/controller/ProjetoController.java"

...

```

### Controller Financeiro

```java file="src/main/java/com/api/transferegov/controller/FinanceiroController.java"

...

```

### Controller de Compras

```java file="src/main/java/com/api/transferegov/controller/CompraController.java"

...

```

## Configuração de Propriedades

```plaintext file="src/main/resources/application.properties"

...

```

## Diagrama da Arquitetura

```mermaid

Arquitetura da API.download-icon {

cursor: pointer;

transform-origin: center;

}

.download-icon .arrow-part {

transition: transform 0.35s cubic-bezier(0.35, 0.2, 0.14, 0.95);

transform-origin: center;

}

button:has(.download-icon):hover .download-icon .arrow-part, button:has(.download-icon):focus-visible .download-icon .arrow-part {

transform: translateY(-1.5px);

}

#mermaid-diagram-r88{font-family:var(--font-geist-sans);font-size:12px;fill:#000000;}#mermaid-diagram-r88 .error-icon{fill:#552222;}#mermaid-diagram-r88 .error-text{fill:#552222;stroke:#552222;}#mermaid-diagram-r88 .edge-thickness-normal{stroke-width:1px;}#mermaid-diagram-r88 .edge-thickness-thick{stroke-width:3.5px;}#mermaid-diagram-r88 .edge-pattern-solid{stroke-dasharray:0;}#mermaid-diagram-r88 .edge-thickness-invisible{stroke-width:0;fill:none;}#mermaid-diagram-r88 .edge-pattern-dashed{stroke-dasharray:3;}#mermaid-diagram-r88 .edge-pattern-dotted{stroke-dasharray:2;}#mermaid-diagram-r88 .marker{fill:#666;stroke:#666;}#mermaid-diagram-r88 .marker.cross{stroke:#666;}#mermaid-diagram-r88 svg{font-family:var(--font-geist-sans);font-size:12px;}#mermaid-diagram-r88 p{margin:0;}#mermaid-diagram-r88 .label{font-family:var(--font-geist-sans);color:#000000;}#mermaid-diagram-r88 .cluster-label text{fill:#333;}#mermaid-diagram-r88 .cluster-label span{color:#333;}#mermaid-diagram-r88 .cluster-label span p{background-color:transparent;}#mermaid-diagram-r88 .label text,#mermaid-diagram-r88 span{fill:#000000;color:#000000;}#mermaid-diagram-r88 .node rect,#mermaid-diagram-r88 .node circle,#mermaid-diagram-r88 .node ellipse,#mermaid-diagram-r88 .node polygon,#mermaid-diagram-r88 .node path{fill:#eee;stroke:#999;stroke-width:1px;}#mermaid-diagram-r88 .rough-node .label text,#mermaid-diagram-r88 .node .label text{text-anchor:middle;}#mermaid-diagram-r88 .node .katex path{fill:#000;stroke:#000;stroke-width:1px;}#mermaid-diagram-r88 .node .label{text-align:center;}#mermaid-diagram-r88 .node.clickable{cursor:pointer;}#mermaid-diagram-r88 .arrowheadPath{fill:#333333;}#mermaid-diagram-r88 .edgePath .path{stroke:#666;stroke-width:2.0px;}#mermaid-diagram-r88 .flowchart-link{stroke:#666;fill:none;}#mermaid-diagram-r88 .edgeLabel{background-color:white;text-align:center;}#mermaid-diagram-r88 .edgeLabel p{background-color:white;}#mermaid-diagram-r88 .edgeLabel rect{opacity:0.5;background-color:white;fill:white;}#mermaid-diagram-r88 .labelBkg{background-color:rgba(255, 255, 255, 0.5);}#mermaid-diagram-r88 .cluster rect{fill:hsl(0, 0%, 98.9215686275%);stroke:#707070;stroke-width:1px;}#mermaid-diagram-r88 .cluster text{fill:#333;}#mermaid-diagram-r88 .cluster span{color:#333;}#mermaid-diagram-r88 div.mermaidTooltip{position:absolute;text-align:center;max-width:200px;padding:2px;font-family:var(--font-geist-sans);font-size:12px;background:hsl(-160, 0%, 93.3333333333%);border:1px solid #707070;border-radius:2px;pointer-events:none;z-index:100;}#mermaid-diagram-r88 .flowchartTitleText{text-anchor:middle;font-size:18px;fill:#000000;}#mermaid-diagram-r88 .flowchart-link{stroke:hsl(var(--gray-400));stroke-width:1px;}#mermaid-diagram-r88 .marker,#mermaid-diagram-r88 marker,#mermaid-diagram-r88 marker \*{fill:hsl(var(--gray-400))!important;stroke:hsl(var(--gray-400))!important;}#mermaid-diagram-r88 .label,#mermaid-diagram-r88 text,#mermaid-diagram-r88 text>tspan{fill:hsl(var(--black))!important;color:hsl(var(--black))!important;}#mermaid-diagram-r88 .background,#mermaid-diagram-r88 rect.relationshipLabelBox{fill:hsl(var(--white))!important;}#mermaid-diagram-r88 .entityBox,#mermaid-diagram-r88 .attributeBoxEven{fill:hsl(var(--gray-150))!important;}#mermaid-diagram-r88 .attributeBoxOdd{fill:hsl(var(--white))!important;}#mermaid-diagram-r88 .label-container,#mermaid-diagram-r88 rect.actor{fill:hsl(var(--white))!important;stroke:hsl(var(--gray-400))!important;}#mermaid-diagram-r88 line{stroke:hsl(var(--gray-400))!important;}#mermaid-diagram-r88 :root{--mermaid-font-family:var(--font-geist-sans);}API JavaClienteAPI JavaControllersServicesRepositoriesBanco de DadosTransfereGovClientAPI Transfere Gov

```

## Conclusão

Esta API Java para integração com o Transfere Gov foi estruturada com as seguintes funcionalidades:

1. \*\*API para pagamento de bolsistas\*\*:

1. Cadastro e gerenciamento de bolsistas

2. Realização de pagamentos via Transfere Gov

3. Consulta de pagamentos realizados

2. \*\*API para projetos\*\*:

1. Cadastro e gerenciamento de projetos

2. Registro de projetos no Transfere Gov

3. Atualização de status de projetos

3. \*\*API para operações financeiras\*\*:

1. Registro e gerenciamento de transações financeiras

2. Realização de transferências via Transfere Gov

3. Consulta de saldo e extrato de projetos

4. Estorno de transações

4. \*\*API para compras\*\*:

1. Cadastro e gerenciamento de compras

2. Registro de compras no Transfere Gov

3. Atualização de status de compras

4. Consulta de compras realizadas

A API foi desenvolvida utilizando Spring Boot, com uma arquitetura em camadas (controllers, services, repositories) e integração com o sistema Transfere Gov através de um cliente REST. A segurança é implementada com Spring Security e JWT para autenticação.

Para utilizar esta API, é necessário configurar a chave de API do Transfere Gov no arquivo de propriedades ou como variável de ambiente.