

# 7pay - teste Henricke

**Ideia do projeto:** A ideia do projeto se consiste em consultar informações da API de despesas orçamentais do estado do Pernambuco utilizando um simples esquema de cache para diminuir o tempo de resposta

## **Informações técnicas:**

Versão do Java: 11

Spring Boot: 2.7.0

Banco de dados: H2

Swagger: 3.0.0

JUnit: 5

**O que foi feito:** O projeto tem como principal objetivo consultar dados da API de despesas do estado de Pernambuco de forma eficiente, utilizando um mecanismo de cache para reduzir o tempo de resposta. Para implementar esse cache, foi empregado o banco de dados H2. Além disso, a API foi devidamente documentada com o uso do Swagger, facilitando a compreensão e o uso por parte de desenvolvedores e usuários. Para garantir a qualidade do código e o funcionamento correto da API, foram criados testes unitários usando o framework JUnit. Esse conjunto de tecnologias e práticas visa fornecer uma solução eficaz e confiável para consultar e disponibilizar informações de despesas do estado de Pernambuco.

**Dificuldades:** O principal desafio que enfrentei durante o projeto concentrou-se em questões relacionadas ao banco de dados, mais precisamente em conflitos envolvendo a identificação única (ID) da entidade criada e problemas com os métodos do Repository. Esses desafios exigiram uma atenção cuidadosa para garantir a consistência dos dados e o correto funcionamento das operações de persistência.

**Solução do problema:** Para superar os desafios encontrados, desenvolvi soluções eficazes. Uma delas foi a implementação de um identificador interno, representado pelo atributo **internId**, que desempenhou um papel fundamental na identificação interna dos dados armazenados em cache no banco de dados. Essa abordagem permitiu uma melhor gestão e recuperação das informações, contribuindo para a resolução dos conflitos relacionados ao ID da entidade.

Além disso, para lidar com os problemas envolvendo os métodos do JpaRepository, adotei uma estratégia que envolveu a escrita manual de alguns métodos personalizados. Essa abordagem se mostrou altamente prática e eficiente, e pude aproveitar ao máximo a API de Streams do Java para simplificar e otimizar essas operações. Essas soluções não apenas resolveram os desafios

encontrados, mas também demonstraram a flexibilidade e a versatilidade do desenvolvimento em Java para atender às necessidades específicas do projeto.

Henricke Mendes - <https://github.com/riccskn>