

Avaliação ItHappens

Allyson Cardozo da Silva

Tecnologia:

Código desenvolvido em Java versão 11(LTS) com Spring Boot JPA

IDE Spring Tool Suíte com Banco de dados em PostgreSQL.

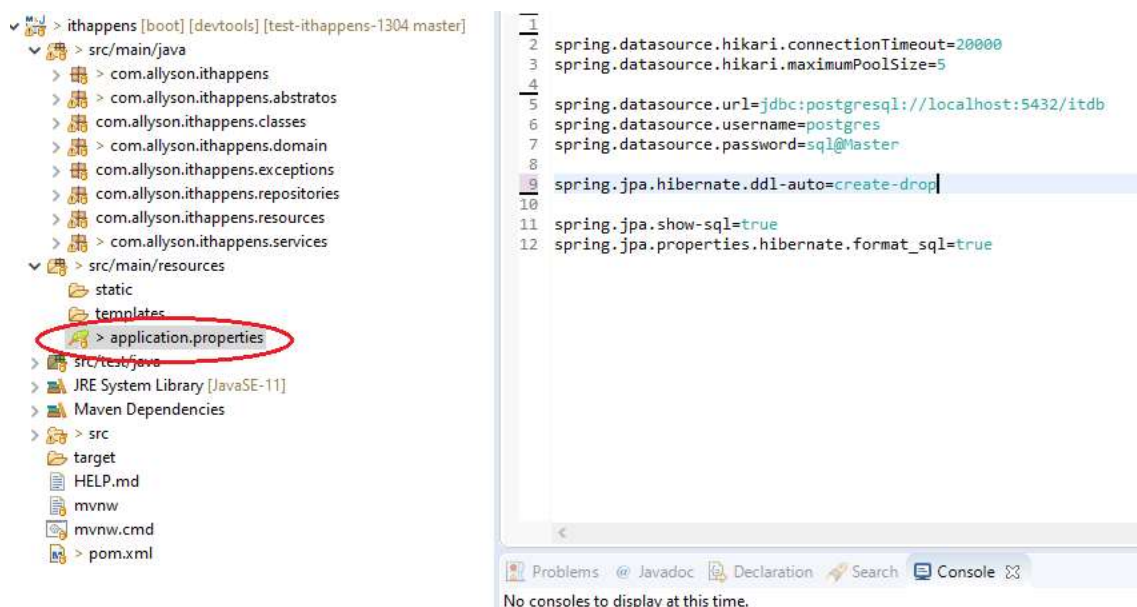
Objetivo:

Criar um software que seja capaz de criar uma Venda com regras de entrada e saída no estoque através de uma metodologia elaborada.

Informações sobre o Projeto:

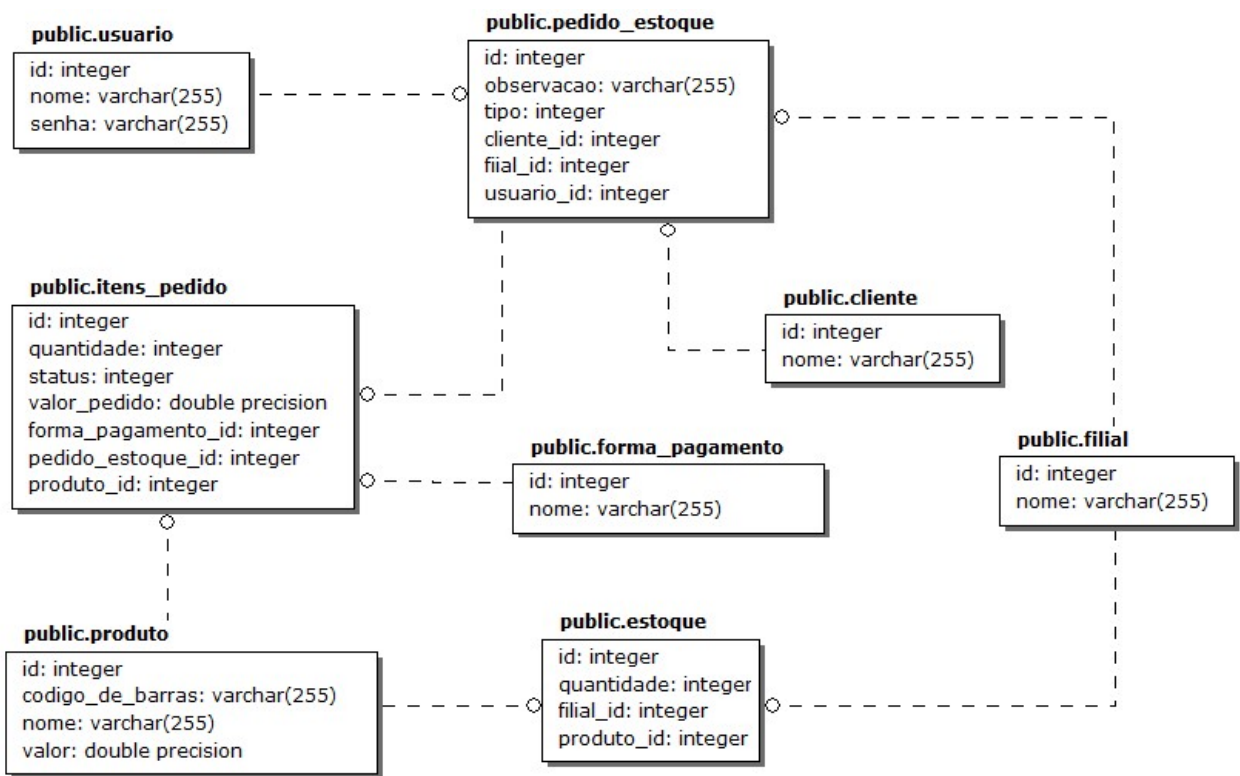
Configuração:

O código desenvolvido trata as regras de desenvolvimento voltado orientação a objetos. A conexão com o banco de dados se dá através do arquivo **application.properties** como demonstrado na figura abaixo:



Para que se consiga executar na maquina local, no postgresql, é necessário antes de mais nada ter o banco de dados vazio já criado no postgree para que a aplicação faça a criação das tabelas automaticamente bem como popular com dados fictícios algumas tabelas para serem usadas como teste.

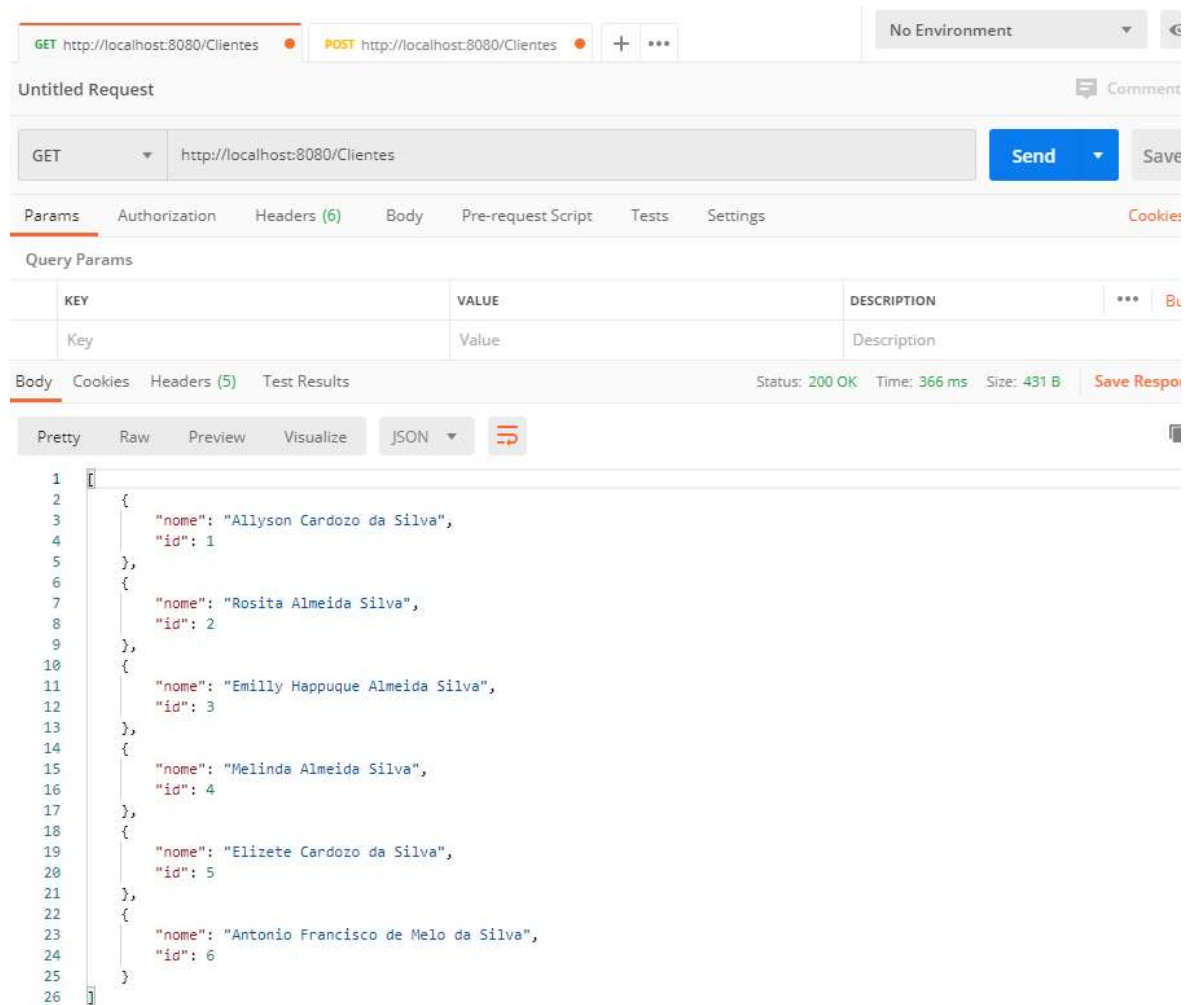
Modelo do Banco de dados:



Rotas:

A forma disponibilizada para consultar, salvar, atualizar ou excluir dados do banco de dados foi através dos controlers ou resources criados dentro do projetos. Essas rotas são regras documentadas para que outras aplicações possam utilizar funcionalidades desta sem precisar conhecer detalhes da nossa implementação. Para testar essas rotas, utilizamos o **POSTMAN** como descrito na imagem abaixo:

Busca de Clientes



The screenshot shows the Postman interface for a GET request to `http://localhost:8080/Clientes`. The request is successful, returning a 200 OK status with a response time of 366 ms and a size of 431 B. The response body is displayed in JSON format, showing an array of 6 client objects.

```
1 [
2   {
3     "nome": "Allyson Cardozo da Silva",
4     "id": 1
5   },
6   {
7     "nome": "Rosita Almeida Silva",
8     "id": 2
9   },
10  {
11    "nome": "Emilly Happuque Almeida Silva",
12    "id": 3
13  },
14  {
15    "nome": "Melinda Almeida Silva",
16    "id": 4
17  },
18  {
19    "nome": "Elizete Cardozo da Silva",
20    "id": 5
21  },
22  {
23    "nome": "Antonio Francisco de Melo da Silva",
24    "id": 6
25  }
26 ]
```

Busca de ItensPedido:

GET http://localhost:8080/ItensPedi...

POST http://localhost:8080/Clientes

+

...

No Environment

Untitled Request

GET

http://localhost:8080/ItensPedido

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION	**
Key	Value	Description	

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 78 msSize: 40.32 KBSave

PrettyRawPreviewVisualizeJSON

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

```
},
  "pedidoEstoque": {
    "tipo": "saida",
    "filial": {
      "nome": "Filial de Teste",
      "id": 1
    },
    "usuario": {
      "senha": "$@%\"{!}%",
      "nome": "Allyson",
      "id": 1
    },
    "observacao": "CPF na nota, Senhor?",
    "cliente": {
      "nome": "Elizete Cardozo da Silva",
      "id": 5
    },
    "id": 5
  },
  "status": "ativo",
  "valorPedido": 19.9,
  "quantidade": 10,
  "formaPagamento": {
    "nome": "À VISTA",
    "id": 1
  },
  "produto": {
    "valor": 1.99,
    "nome": "Sabonete",
    "codigoDeBarras": "11111111111",
    "id": 1
  },
  "id": 5
}
```

Busca de PedidoEstoque pelo id;

GET http://localhost:8080/PedidoEst...

POST http://localhost:8080/Clientes

+

...

No Environment

Untitled Request

GET

http://localhost:8080/PedidoEstoque/1

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (5)

Test Results

Status: 200 OK Time: 75 ms Size: 379 B Save

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

"tipo": "saida",

"filial": {

"nome": "Filial de Teste",

"id": 1

},

"usuario": {

"senha": "\$@%\"{!l}#",

"nome": "Allyson",

"id": 1

},

"observacao": "CPF na nota, Senhor?",

"cliente": {

"nome": "Allyson Cardozo da Silva",

"id": 1

},

"id": 1

Alguns Sql's

```
--consulta que retorne todos os produtos com  
--quantidade maior ou igual a 100
```

```
SELECT p.*  
FROM  
    estoque as e  
    left join produto p on e.produto_id = p.id  
where e.quantidade >= 100
```

```
-- Consulta que retorna todos os produtos  
--que têm estoque para a filial de código 60
```

```
SELECT p.*  
FROM  
    estoque as e  
    left join produto p on e.produto_id = p.id  
where e.filial_id = 60
```

```
--Consulta que lista todos os campos para o domínio PedidoEstoque  
--ItensPedido filtrando apenas o produto de código 7993
```

```
SELECT  
    pe.id,  
    pe.observacao,  
    pe.tipo,  
    pe.cliente_id,  
    pe.filial_id,  
    pe.usuario_id,  
    ip.id as item_pedido_id,  
    ip.quantidade,  
    ip.status,  
    ip.valor_pedido,  
    ip.forma_pagamento_id,  
    ip.pedido_estoque_id,  
    ip.produto_id  
FROM  
    itens_pedido ip  
    left join pedido_estoque pe on ip.pedido_estoque_id = pe.id  
where ip.produto_id = 7993
```



```
--consulta retorna todos os pedidos com suas  
--respectivas formas de pagamento
```

```
SELECT  
    ip.id,  
    cl.nome as nome_cliente,  
    ip.produto_id,  
    p.nome nome_produto,  
    p.valor as valor_unitario,  
    ip.forma_pagamento_id,  
    fp.nome formapagamento_nome,  
    ip.quantidade as qtde_pedido,  
    ip.valor_pedido as valor_total  
FROM  
    itens_pedido ip  
    left join pedido_estoque pe on ip.pedido_estoque_id = pe.id  
    left join cliente cl on pe.cliente_id = cl.id  
    left join produto p on ip.produto_id = p.id  
    left join forma_pagamento fp on ip.forma_pagamento_id = fp.id  
|
```

```
-- consulta que soma e bate os valores da capa do pedido  
-- com os valores dos itens de pedido
```

```
SELECT  
    pe.id,  
    pe.tipo,  
    fi.nome,  
    cl.nome as nome_cliente,  
    pe.usuario_id,  
    (select sum(coalesce(ip.valor_pedido,0))  
     from itens_pedido ip  
     where ip.pedido_estoque_id = pe.id) total_itempedido,  
    pe.valor total_valorpedido  
FROM  
    pedido_estoque pe  
    left join cliente cl on pe.cliente_id = cl.id  
    left join filial fi on pe.fiial_id = fi.id  
|
```

```
-- consulta para soma o total dos itens por pedido e que filtra  
-- apenas os pedidos no qual a soma total da quantidade de itens  
--de pedido seja maior que 10
```

```
select ip.produto_id,  
    p.nome,  
    sum(ip.quantidade) qtde_itenspedido  
from itens_pedido ip  
    left join pedido_estoque pe on ip.pedido_estoque_id = pe.id  
    left join produto p on ip.produto_id = p.id  
group by  
    ip.produto_id,  
    p.nome  
having count(ip.id) > 10|
```