



# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## **Entrega continuada 2/2**

Programação Web Back End

Nome: Bhreno Venditti de Oliveira Barboza- RA:01242072

Nome: Diego Furtado Lima - RA:01242042

Nome: Eduardo Ohata do Reis - RA:01242110

Nome: Guilherme Vieira Moreira de Paiva - RA:01242068

Nome: Lucca Barbosa Oliveira - RA:01242125

Nome: Vinicius Aoki Eguchi - RA:01242095

## Padrão de Projeto: Observer

Devido ao projeto ser focado no controle de estoque de uma adega, foi escolhida o padrão Observer. Com o intuito de gerar alertas para quantidade baixa em estoque.

Com isso o cliente saberá quais são os produtos que precisa comprar para repor o estoque.

Classes:

- Variante.java

A entidade da tabela variante, onde descreve os tipos diferentes de produtos (recipiente ou sabor por exemplo), é a classe que guarda a quantidade em estoque

- Produto.java

A entidade que possui a quantidade minima em estoque, já como um produto pode conter várias variantes (vários sabores de Gin, por exemplo)

- VarianteService.java

Classe java que guarda a regra de negócio

Possui o método: patchQuantidadeRemove, responsável por quando é feito uma venda, diminuindo a quantidade em estoque, e dispara o evento caso a quantidade em estoque esteja mais baixa que a quantidade minima.

- EstoqueBaixoEvento.java

Evento que é criado pela service, representa quando o estoque fica abaixo do minimo definido no sistema

- QuantidadeBaixaObserver.java

Classe responsável por enviar o alerta para o Front.

Prints:

EstoqueBaixoEvento

```
1  package sptech.art8.evento;
2
3  import lombok.Getter;
4  import org.springframework.context.ApplicationEvent;
5  import sptech.art8.model.Variante;
6
7  @Getter
8  public class EstoqueBaixoEvento extends ApplicationEvent {
9      private final Variante variante;
10
11      public EstoqueBaixoEvento(Object source, Variante variante) {
12          super(source);
13          this.variante = variante;
14      }
15  }
```

QuantidadeBaixaObserver

```
1  package sptech.art8.observer;
2
3  import org.springframework.context.event.EventListener;
4  import org.springframework.stereotype.Component;
5  import sptech.art8.evento.EstoqueBaixoEvento;
6  import org.slf4j.Logger;
7  import org.slf4j.LoggerFactory;
8
9  @Component
10 public class QuantidadeBaixaObserver {
11     private static final Logger logger = LoggerFactory.getLogger(QuantidadeBaixaObserver.class);
12
13     @EventListener
14     public void registraQtdBaixa(EstoqueBaixoEvento evento) {
15         logger.info("Variante: {} com estoque baixo", evento.getVariante().toString());
16     }
17 }
```

## VarianteService

```
public int patchQuantidadeRemove(int id, int quantidade) {
    Variante variante = repository.findById(id)
        .orElseThrow(() -> new DependenciaNaoEncontradaException("Variante"));

    if (variante.getQuantidade() < quantidade) {
        throw new IllegalArgumentException("Quantidade insuficiente em estoque");
    }

    int novaQuantidade = variante.getQuantidade() - quantidade;
    int resultado = repository.patchQuantidadeRemove(id, quantidade);

    if (novaQuantidade <= variante.getProduto().getQuantidade()) {
        publisher.publishEvent(new EstoqueBaixoEvento(this, variante));
    }

    return resultado;
}
```