

Engenharia de Softwares Escaláveis

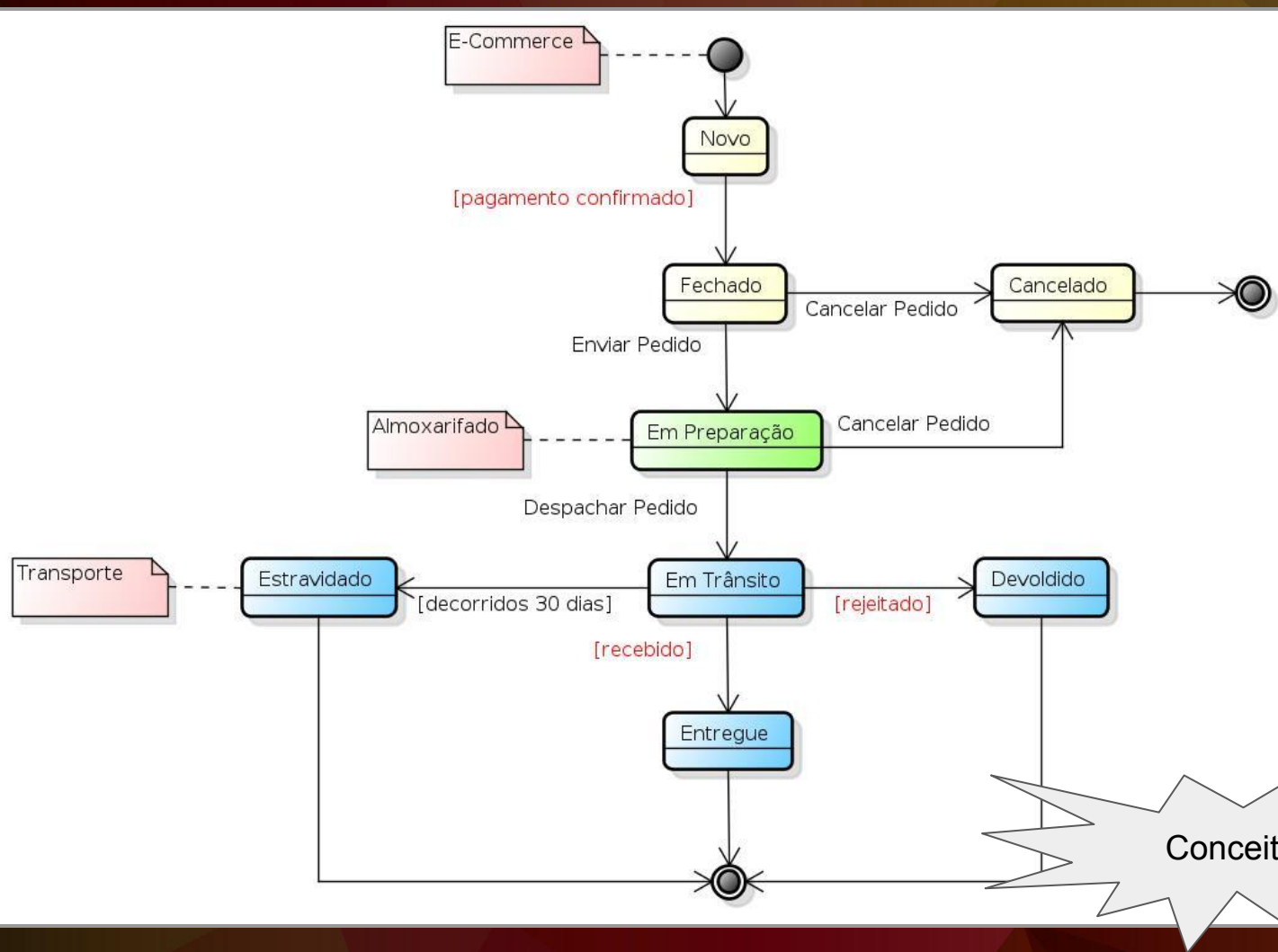
Domain-Driven Design (DDD) e Arquitetura de
Softwares Escaláveis com Java

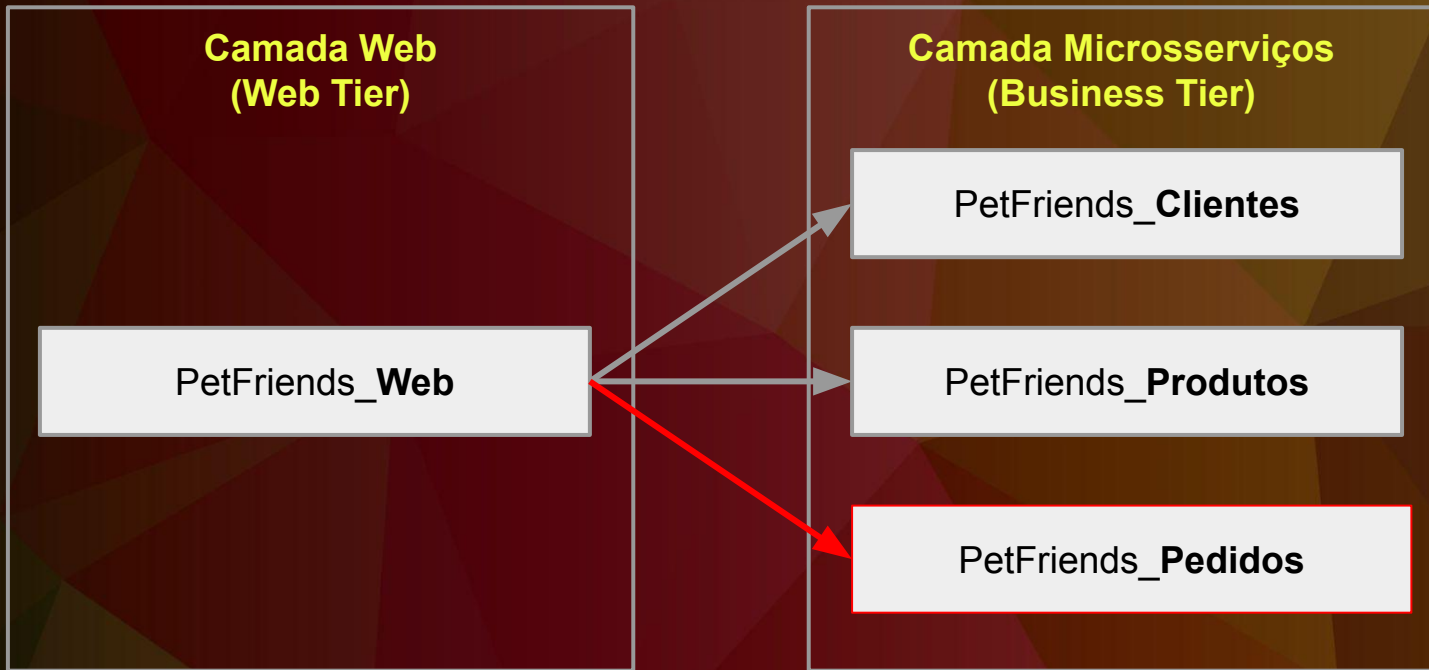
Agenda

Etapas 7: Prévia do AT.



Prévia do AT





**Camada Microsserviços
(Business Tier)**

PetFriends_**C**lientes

PetFriends_**P**rodutos

PetFriends_**P**edidos

**Camada Microsserviços
(Business Tier)**

PetFriends_**A**lmojarifado

PetFriends_**T**ransporte

Conceitos



1. Transformar monólitos em microsserviços eficazes, aplicando princípios de DDD e técnicas de decomposição:
 - a. Implemente a classe entity e seu respectivo repository que represente o agregado mais representativo do microsserviço PetFriends_Almojarifado.
 - b. Implemente um exemplo de Value Object a ser usado na classe entity da questão anterior.
 - c. Implemente a classe entity e seu respectivo repository que represente o agregado mais representativo do microsserviço PetFriends_Transporte.
 - d. Implemente um exemplo de Value Object a ser usado na classe entity da questão anterior.
2. Projetar softwares usando "domain events":
 - a. O módulo PetFriends_Web foi desenvolvido em ReactJS, acessando os microsserviços de Clientes, Produtos e Pedidos de forma síncrona, via REST API. Que funcionalidade síncrona executada pelo cliente é diretamente afetada pelos eventos de domínio?
 - b. Explique de forma sucinta qual é a diferença de enviar eventos somente com o ID do agregado e enviar um payload completo?
 - c. A partir da resposta dada na questão anterior, como você projetaria o evento a ser enviado pelo PetFriends_Pedido para o PetFriends_Almojarifado?
 - d. A partir da resposta dada na questão anterior, como você projetaria o evento a ser enviado pelo PetFriends_Pedido para o PetFriends_Transporte?
3. Desenvolver microsserviços event-driven e com outros padrões de comunicação assíncrona:
 - a. No microsserviço PetFriends_Almojarifado, implemente a classe de configuração para o tratamento de mensagens para receber os eventos do PetFriends_Pedidos.
 - b. No microsserviço PetFriends_Almojarifado, implemente o serviço que receberá os eventos do PetFriends_Pedidos.
 - c. No microsserviço PetFriends_Transporte, implemente a classe de configuração para o tratamento de mensagens para receber os eventos do PetFriends_Pedidos.
 - d. No microsserviço PetFriends_Transporte, implemente o serviço que receberá os eventos do PetFriends_Pedidos.
4. Implementar testes e observabilidade em microsserviços com Zipkin, Spring Cloud Sleuth e ELK Stack:
 - a. Explique de forma sucinta o que é um Gateway de Serviço, suas vantagens e desvantagens.
 - b. O que é ID de Correlação e quais são os seus pré-requisitos?
 - c. Qual é a função do Spring Cloud Sleuth e sua relação com o serviço Zipkin?
 - d. Explique de forma sucinta o que é Agregador de Logs, suas vantagens e desvantagens.

Assim que terminar, salve seu trabalho em PDF nomeando o arquivo conforme a regra "nome_sobrenome_DR4_AT.PDF" e poste como resposta a este AT.

Assinaturas - Pub/Sub - Sof x +

console.cloud.google.com/cloudpubsub/subscription/list?authuser=0&hl=pt&project=softwares-esca... ☆

Administrativo Conteúdo Ferramentas Livros Localhost | Todos os favoritos

Google Cloud Software Escaláveis top Pesquisa

Pub/Sub

Assinaturas + CRIAR ASSINATURA EXCLUIR OCULTAR PAINEL DE INFORMAÇÕES

LISTA MÉTRICAS

Filtro Filtrar inscrições

<input type="checkbox"/>	Estado	Código da assinatura ↑	Tipo de envio		
<input type="checkbox"/>	✓	dr4_sub	Pull	⋮	▼
<input type="checkbox"/>	✓	PetFriends_Almojarifado	Pull	⋮	▼
<input type="checkbox"/>	✓	PetFriends_Pedidos	Pull	⋮	▼
<input type="checkbox"/>	✓	PetFriends_Transporte	Pull	⋮	▼

Selecione uma assinatura

PERMISSÕES MARCADORES

Selecione pelo menos um recurso.

Setup

PetFriends_Pedidos - Apache NetBeans IDE 22

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

1588/1596MB

Projects - Infnet - Eng de Soft ... x

Files Services

Navigator

- DesignTatico
- DR4_E2_A1
- DR4_E2_A2
- DR4_E3_A2
- DR4_E4_A1
- DR4_E5_A1
- DR4_E6_A2_Stream
- PetFriends_Pedidos**
 - Source Packages
 - br.edu.infnet
 - br.edu.infnet.controllers
 - br.edu.infnet.pedidos.domain
 - br.edu.infnet.pedidos.eventos
 - br.edu.infnet.pedidos.infra.message
 - br.edu.infnet.pedidos.infra.repository
 - br.edu.infnet.pedidos.infra.service
 - Test Packages
 - Other Sources
 - src/main/resources
 - <default package>
 - application.properties
 - softwares-escalaveis-a12fc70b8812.json
 - static
 - templates
 - Dependencies
 - Runtime Dependencies
 - Test Dependencies
 - Project Files

New Project...
Ctrl+Shift+N

Open Project...
Ctrl+Shift+O


New File...
Ctrl+N

Open File...
-

Go to File...
Alt+Shift+O

Show Dashboard
Alt+Shift+W

Test Results Search Results Output Usages



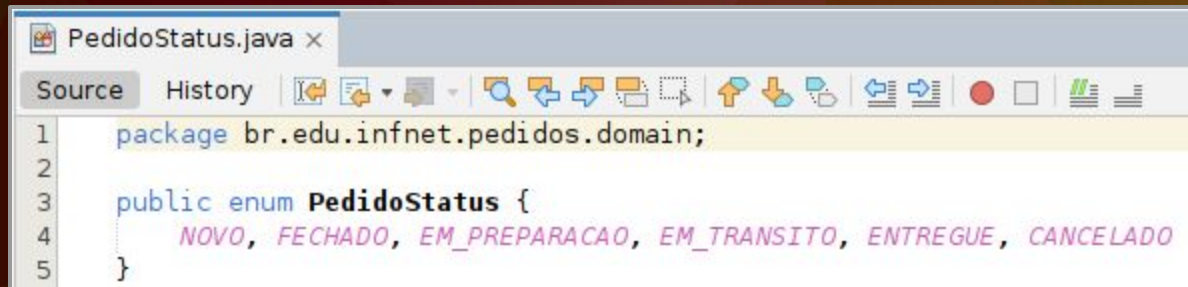
Setup

pom.xml [PetFriends_Pedidos] x

Source Graph Effective History

```
20 <dependencies>
21   <dependency>
22     <groupId>org.springframework.boot</groupId>
23     <artifactId>spring-boot-starter-data-jpa</artifactId>
24   </dependency>
25   <dependency>
26     <groupId>org.springframework.boot</groupId>
27     <artifactId>spring-boot-starter-integration</artifactId>
28   </dependency>
29   <dependency>
30     <groupId>org.springframework.boot</groupId>
31     <artifactId>spring-boot-starter-web</artifactId>
32   </dependency>
33   <dependency>
34     <groupId>com.google.cloud</groupId>
35     <artifactId>spring-cloud-gcp-starter-pubsub</artifactId>
36   </dependency>
37   <dependency>
38     <groupId>org.springframework.integration</groupId>
39     <artifactId>spring-integration-http</artifactId>
40   </dependency>
41   <dependency>
42     <groupId>org.springframework.integration</groupId>
43     <artifactId>spring-integration-jpa</artifactId>
44   </dependency>
45   <dependency>
46     <groupId>com.h2database</groupId>
47     <artifactId>h2</artifactId>
48     <scope>runtime</scope>
49   </dependency>
50   <dependency>
51     <groupId>org.springdoc</groupId>
52     <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
53     <version>2.6.0</version>
54   </dependency>
```

Setup



```
PedidoStatus.java x
Source History
1 package br.edu.infnet.pedidos.domain;
2
3 public enum PedidoStatus {
4     NOVO, FECHADO, EM_PREPARACAO, EM_TRANSITO, ENTREGUE, CANCELADO
5 }
```

Domínio

```
ValorMonetario.java x
Source History
1 package br.edu.infnet.pedidos.domain;
2
3 import ...4 lines
7
8 public class ValorMonetario implements Serializable {
9
10     private final BigDecimal quantia;
11
12     public ValorMonetario(BigDecimal quantia) {
13         if (quantia == null || quantia.signum() < 0) {
14             throw new IllegalArgumentException("Valor Monetário não pode ser negativo");
15         }
16         this.quantia = quantia.setScale(2, RoundingMode.HALF_UP);
17     }
18
19     public BigDecimal getQuantia() {
20         return this.quantia;
21     }
22
23     public ValorMonetario somar(ValorMonetario outro) {
24         if (outro == null) {
25             throw new IllegalArgumentException("Outro valor não pode ser nulo");
26         }
27         return new ValorMonetario(this.quantia.add(outro.getQuantia()));
28     }
29
30     public ValorMonetario subtrair(ValorMonetario outro) {
31         if (outro == null) {
32             throw new IllegalArgumentException("Outro valor não pode ser nulo");
33         }
34         return new ValorMonetario(this.quantia.subtract(outro.getQuantia()));
35     }
36
37     @Override
38     public boolean equals(Object objeto) {
39         final ValorMonetario outro = (ValorMonetario) objeto;
40         return Objects.equals(this.quantia, outro.getQuantia());
41     }
42 }
```

Domínio

```
Pedido.java x
Source History
1 package br.edu.infnet.pedidos.domain;
2
3 import ...19 lines
22
23 @Entity
24 @Table(name = "pedidos", catalog = "DR4_1", schema = "PUBLIC")
25 public class Pedido implements Serializable {
26
27     private static final long serialVersionUID = 1L;
28     @Id
29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     @Column(nullable = false)
31     private Long id;
32     @Column(name = "ORDER_DATE")
33     @Temporal(TemporalType.DATE)
34     private Date orderDate;
35     @JsonIgnoreProperties(value = "orderId")
36     @OneToMany(mappedBy = "orderId", cascade = CascadeType.ALL)
37     private List<ItemPedido> itemList;
38     @Column(name = "CUSTOMER_ID")
39     private Long customerId;
40     @Convert(converter = PedidoStatusConverter.class)
41     @Column(name = "STATUS")
42     private PedidoStatus status;
43     @Convert(converter = ValorMonetarioConverter.class)
44     @Column(name = "VALOR_TOTAL")
45     private ValorMonetario valorTotal;
46
```

Domínio



The image shows a screenshot of a Java IDE window titled "Pedido.java". The window has a "Source" tab and a "History" tab. The code is as follows:

```
47 public Pedido() {  
48     this.orderDate = new Date();  
49     this.status = PedidoStatus.NOVO;  
50     this.valorTotal = new ValorMonetario(BigDecimal.ZERO);  
51 }  
52  
53 public Pedido(Long id) {  
54     this.id = id;  
55     this.orderDate = new Date();  
56     this.status = PedidoStatus.NOVO;  
57     this.valorTotal = new ValorMonetario(BigDecimal.ZERO);  
58 }  
59  
60 public Pedido(Long id, Long customerId) {  
61     this.id = id;  
62     this.customerId = customerId;  
63     this.orderDate = new Date();  
64     this.status = PedidoStatus.NOVO;  
65     this.valorTotal = new ValorMonetario(BigDecimal.ZERO);  
66 }
```

Domínio

Pedido.java x

Source History

```
141 public void adicionarItem(Long productId, int quantidade, double unitValue) {
142     if (productId == null) {
143         throw new IllegalArgumentException("Produto inválido");
144     }
145     if (quantidade <= 0) {
146         throw new IllegalArgumentException("Quantidade precisa ser positiva");
147     }
148     if (this.status != PedidoStatus.NOVO) {
149         throw new IllegalStateException("Não é possível inserir itens em um pedido em andamento");
150     }
151     ItemPedido itemPedido = new ItemPedido();
152     itemPedido.setOrderId(this);
153     itemPedido.setProductId(productId);
154     itemPedido.setQuantity(quantidade);
155     itemPedido.setTotal(new ValorMonetario(new BigDecimal(unitValue * quantidade)));
156     if (this.itemList == null) {
157         this.itemList = new ArrayList<>();
158     }
159     this.itemList.add(itemPedido);
160     this.setValorTotal(this.valorTotal.somar(itemPedido.getTotal()));
161 }
```

Domínio

Pedido.java x

Source

History



```
163 public void fecharPedido() {
164     if (this.status != PedidoStatus.NOVO) {
165         throw new IllegalStateException("Não é possível fechar um pedido que não é novo");
166     }
167     if (this.itemList.isEmpty()) {
168         throw new IllegalStateException("Não é possível fechar um pedido vazio");
169     }
170     this.status = PedidoStatus.FECHADO;
171 }
172
173 public void cancelarPedido() {
174     if (this.status != PedidoStatus.FECHADO) {
175         throw new IllegalStateException("Não é possível cancelar um pedido que não esteja fechado");
176     }
177     this.status = PedidoStatus.CANCELADO;
178 }
179
180 public void enviarPedido() {
181     if (this.status != PedidoStatus.FECHADO) {
182         throw new IllegalStateException("Não é possível enviar um pedido que não esteja fechado");
183     }
184     this.status = PedidoStatus.EM_PREPARACAO;
185 }
186 }
```

Domínio

```
ItemPedido.java x
Source History
1 package br.edu.infnet.pedidos.domain;
2
3 import ...9 lines
12
13 @Entity
14 @Table(name = "itens_pedidos", catalog = "DR4_1", schema = "PUBLIC")
15 public class ItemPedido implements Serializable {
16
17     private static final long serialVersionUID = 1L;
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     @Column(nullable = false)
21     private Long id;
22     private Integer quantity;
23     @Column(precision = 20, scale = 2)
24     private ValorMonetario total;
25     @Column(name = "PRODUCT_ID")
26     private Long productId;
27     @JoinColumn(name = "ORDER_ID", referencedColumnName = "ID")
28     @ManyToOne
29     private Pedido orderId;
30
31     public ItemPedido() {
32     }
33
34     public ItemPedido(Long id) {
35         this.id = id;
36     }
37 }
```

Domínio

```
EstadoPedidoMudou.java x
Source History
1 package br.edu.infnet.pedidos.eventos;
2
3 import ...3 lines
4
5
6
7 public class EstadoPedidoMudou implements Serializable {
8
9     private Long idPedido;
10    private PedidoStatus estado;
11    private Date momento;
12
13    public EstadoPedidoMudou() {
14    }
15
16    public EstadoPedidoMudou(Long idPedido, PedidoStatus estado) {
17        this.idPedido = idPedido;
18        this.estado = estado;
19        this.momento = new Date();
20    }
21
22    public EstadoPedidoMudou(Long idPedido, PedidoStatus estado, Date momento) {
23        this.idPedido = idPedido;
24        this.estado = estado;
25        this.momento = momento;
26    }
27
28    public Long getIdPedido() {
29        return idPedido;
30    }
31
32    public void setIdPedido(Long idPedido) {
33        this.idPedido = idPedido;
34    }
35}
```


Eventos e
Mensageria

```
EstadoPedidoMudouSerializer.java x
Source History
1 package br.edu.infnet.pedidos.infra.message;
2
3 import ...6 lines
9
10 public class EstadoPedidoMudouSerializer extends StdSerializer<EstadoPedidoMudou> {
11
12     public EstadoPedidoMudouSerializer() {
13         super(EstadoPedidoMudou.class);
14     }
15
16     @Override
17     public void serialize(EstadoPedidoMudou evento, JsonGenerator jgen, SerializerProvider provider) throws IOException {
18         jgen.writeStartObject();
19         jgen.writeNumberField("idPedido", evento.getIdPedido());
20         jgen.writeStringField("estado", evento.getEstado().toString());
21         SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
22         String data = sdf.format(evento.getMomento());
23         jgen.writeStringField("momento", data);
24     }
25 }
```

Eventos e
Mensageria



```
1  package br.edu.infnet.pedidos.infra.message;
2
3  import ...12 lines
15
16  public class EstadoPedidoMudouDeserializer extends StdDeserializer<EstadoPedidoMudou> {
17
18      public EstadoPedidoMudouDeserializer() {
19          super(EstadoPedidoMudou.class);
20      }
21
22      @Override
23      public EstadoPedidoMudou deserialize(JsonParser jp, DeserializationContext dc) throws IOException, JacksonException {
24          EstadoPedidoMudou evento = null;
25          JsonNode node = jp.getCodec().readTree(jp);
26          SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
27          try {
28              evento = new EstadoPedidoMudou(
29                  node.get("idPedido").asLong(),
30                  PedidoStatus.valueOf(node.get("estado").asText()),
31                  sdf.parse(node.get("momento").asText())
32              );
33          } catch (ParseException e) {
34              throw new IOException("Erro na data");
35          }
36          return evento;
37      }
38  }
```



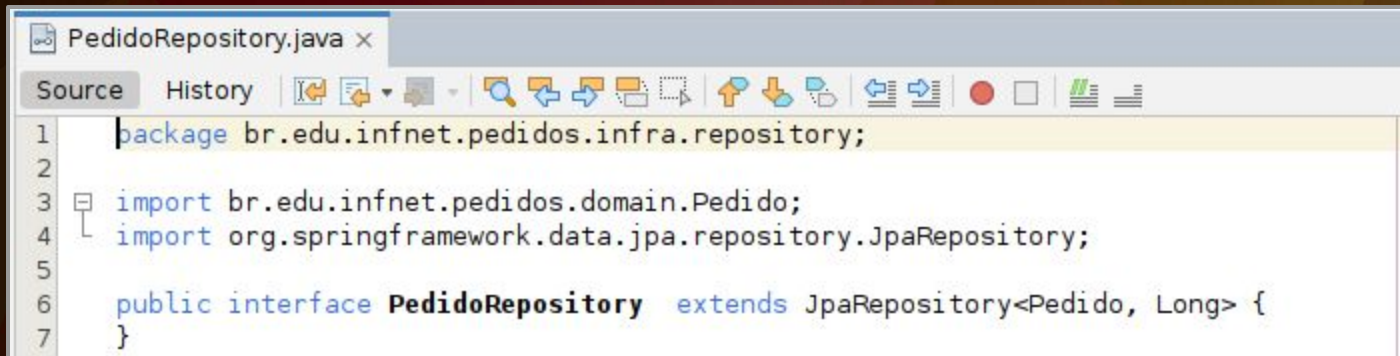
Eventos e
Mensageria

```
PetFriendsPedidosMessageConfig.java x
Source History
1 package br.edu.infnet.pedidos.infra.message;
2
3 import ...12 lines
15
16 @Configuration
17 public class PetFriendsPedidosMessageConfig {
18
19     @Bean
20     public JacksonPubSubMessageConverter estadoMudouConverter() {
21         ObjectMapper objectMapper = new ObjectMapper();
22         SimpleModule simpleModule = new SimpleModule();
23         simpleModule.addSerializer(EstadoPedidoMudou.class, new EstadoPedidoMudouSerializer());
24         simpleModule.addDeserializer(EstadoPedidoMudou.class, new EstadoPedidoMudouDeserializer());
25         objectMapper.registerModule(simpleModule);
26         return new JacksonPubSubMessageConverter(objectMapper);
27     }
}
```

Eventos e
Mensageria

```
PetFriendsPedidosMessageConfig.java x
Source History
29 @Bean
30 public MessageChannel inputMessageChannel() {
31     return new PublishSubscribeChannel();
32 }
33
34 @Bean
35 public PubSubInboundChannelAdapter inboundChannelAdapter(
36     @Qualifier("inputMessageChannel") MessageChannel messageChannel, PubSubTemplate pubSubTemplate) {
37
38     pubSubTemplate.setMessageConverter(estadoMudouConverter());
39     PubSubInboundChannelAdapter adapter = new PubSubInboundChannelAdapter(pubSubTemplate, "PetFriends_Pedidos");
40     adapter.setOutputChannel(messageChannel);
41     adapter.setAckMode(AckMode.MANUAL);
42     adapter.setPayloadType(EstadoPedidoMudou.class);
43     return adapter;
44 }
45 }
```

Eventos e
Mensageria



```
1 package br.edu.infnet.pedidos.infra.repository;
2
3 import br.edu.infnet.pedidos.domain.Pedido;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface PedidoRepository extends JpaRepository<Pedido, Long> {
7 }
```

Persistência

```
PedidoStatusConverter.java x
Source History
1 package br.edu.infnet.pedidos.infra.repository;
2
3 import br.edu.infnet.pedidos.domain.PedidoStatus;
4 import jakarta.persistence.AttributeConverter;
5 import jakarta.persistence.Converter;
6
7 @Converter(autoApply = true)
8 public class PedidoStatusConverter implements AttributeConverter<PedidoStatus, String> {
9
10     @Override
11     public String convertToDatabaseColumn(PedidoStatus pedidoStatus) {
12         return pedidoStatus.toString();
13     }
14
15     @Override
16     public PedidoStatus convertToEntityAttribute(String pedidoStatus) {
17         return PedidoStatus.valueOf(pedidoStatus);
18     }
19 }
```

Persistência

```
ValorMonetarioConverter.java x
Source History
1 package br.edu.infnet.pedidos.infra.repository;
2
3 import br.edu.infnet.pedidos.domain.ValorMonetario;
4 import jakarta.persistence.AttributeConverter;
5 import jakarta.persistence.Converter;
6 import java.math.BigDecimal;
7
8 @Converter(autoApply = true)
9 public class ValorMonetarioConverter implements AttributeConverter<ValorMonetario, BigDecimal> {
10
11     @Override
12     public BigDecimal convertToDatabaseColumn(ValorMonetario valorMonetario) {
13         return valorMonetario.getQuantia();
14     }
15
16     @Override
17     public ValorMonetario convertToEntityAttribute(BigDecimal quantia) {
18         return new ValorMonetario(quantia);
19     }
20 }
```

Persistência

Assinaturas - Pub/Sub - Sof x H2 Console x Swagger UI x +

localhost:8080/h2-console/login.do?jsessionId=0ad1f7dac2f40cfe14f0f389cc4c61c5

Administrativo Conteúdo Ferramentas Livros Localhost Todos os favoritos

Auto commit Auto complete Off Auto select On

jdbc:h2:file:/home/armeniocardos

- ITENS_PEDIDOS
- PEDIDOS
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM PEDIDOS

SELECT * FROM PEDIDOS;

ID	CUSTOMER_ID	ORDER_DATE	STATUS	VALOR_TOTAL
1	4321	2024-09-06	FECHADO	280.00

(1 row, 1 ms)

Edit

Persistência

Assinaturas - Pub/Sub - Sof x H2 Console x Swagger UI x +

localhost:8080/h2-console/login.do?jsessionid=0ad1f7dac2f40cfe14f0f389cc4c61c5

Administrativo Conteúdo Ferramentas Livros Localhost Todos os favoritos

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:file:/home/armeniocardos

- ITENS_PEDIDOS
- PEDIDOS
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM ITENS_PEDIDOS
```

```
SELECT * FROM ITENS_PEDIDOS;
```

ID	PRODUCT_ID	QUANTITY	TOTAL	ORDER_ID
1	1234	2	200.00	1
2	1235	1	50.00	1
3	1236	1	30.00	1

(3 rows, 2 ms)

Edit

Persistência

```
PedidoService.java x
Source History
1 package br.edu.infnet.pedidos.infra.service;
2
3 import ...14 lines
17
18 @Service
19 public class PedidoService {
20
21     private static final Logger LOG = LoggerFactory.getLogger(PedidoService.class);
22
23     @Autowired
24     private PubSubTemplate pubSubTemplate;
25     @Autowired
26     JacksonPubSubMessageConverter converter;
27     @Autowired
28     private PedidoRepository repository;
29
30     public Pedido obterPorId(long id) {
31         return repository.getReferenceById(id);
32     }
33
34     public Pedido fecharPedido(long id) {
35         Pedido pedido = repository.getReferenceById(id);
36         pedido.fecharPedido();
37         pedido = repository.save(pedido);
38         enviar(new EstadoPedidoMudou(pedido.getId(), PedidoStatus.FECHADO));
39         return pedido;
40     }
41
42     public Pedido criarPedido(Pedido pedido) {
43         Pedido retorno = null;
44         if (pedido != null) {
45             retorno = repository.save(pedido);
46             enviar(new EstadoPedidoMudou(pedido.getId(), PedidoStatus.NOVO));
47         }
48         return retorno;
49     }
}
```

Serviço

```
PedidoService.java x
34 public Pedido fecharPedido(long id) {
42 public void enviar(EstadoPedidoMudou estado) {
43     pubSubTemplate.setMessageConverter(converter);
44     pubSubTemplate.publish("dr4_topic", estado);
45     LOG.info("***** Mensagem Publicada ---> " + estado);
46 }
47
48 @ServiceActivator(inputChannel = "inputMessageChannel")
49 public void receber(EstadoPedidoMudou payload,
50     @Header(GcpPubSubHeaders.ORIGINAL_MESSAGE) ConvertedBasicAcknowledgeablePubsSubMessage<EstadoPedidoMudou> message) {
51
52     LOG.info("***** Mensagem Recebida ---> " + payload);
53     message.ack();
54 }
55 }
```

Serviço

```
PedidoController.java x
Source History
1 package br.edu.infnet.controllers;
2
3 import ...8 lines
11
12 @RestController
13 @RequestMapping(value = "/pedidos")
14 public class PedidoController {
15
16     @Autowired
17     private PedidoService service;
18
19     @GetMapping("/{id}")
20     public Pedido obterPorId(@PathVariable(value = "id") long id){
21         return service.obterPorId(id);
22     }
23
24     @PatchMapping("/fechar-pedido/{id}")
25     public Pedido fecharPedido(@PathVariable(value = "id") long id){
26         return service.fecharPedido(id);
27     }
28 }
```

Controller

Assinaturas - Pub/Sub - Sof x Swagger UI x +

localhost:8080/swagger-ui/index.html

Administrativo | Conteúdo | Ferramentas | Livros | Localhost | Todos os favoritos

Swagger. Supported by SMARTBEAR

/v3/api-docs Explore

OpenAPI definition v0 OAS 3.0

/v3/api-docs

Servers

http://localhost:8080 - Generated server url

pedido-controller

PATCH /pedidos/fechar-pedido/{id}

GET /pedidos/{id}

Controller

Assinaturas - Pub/Sub - Sof x Swagger UI x +

localhost:8080/swagger-ui/index.html#/pedido-controller/obterPorId

Administrativo | Conteúdo | Ferramentas | Livros | Localhost | Todos os favoritos

200

Response body

```
{
  "quantia": 200,
  "productId": 1234,
  {
    "id": 2,
    "quantity": 1,
    "total": {
      "quantia": 50
    },
    "productId": 1235
  },
  {
    "id": 3,
    "quantity": 1,
    "total": {
      "quantia": 30
    },
    "productId": 1236
  }
},
{
  "customerId": 4321,
  "status": "FECHADO",
  "valorTotal": {
    "quantia": 280
  },
  "hibernateLazyInitializer": {}
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Fri, 06 Sep 2024 22:32:50 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Controller