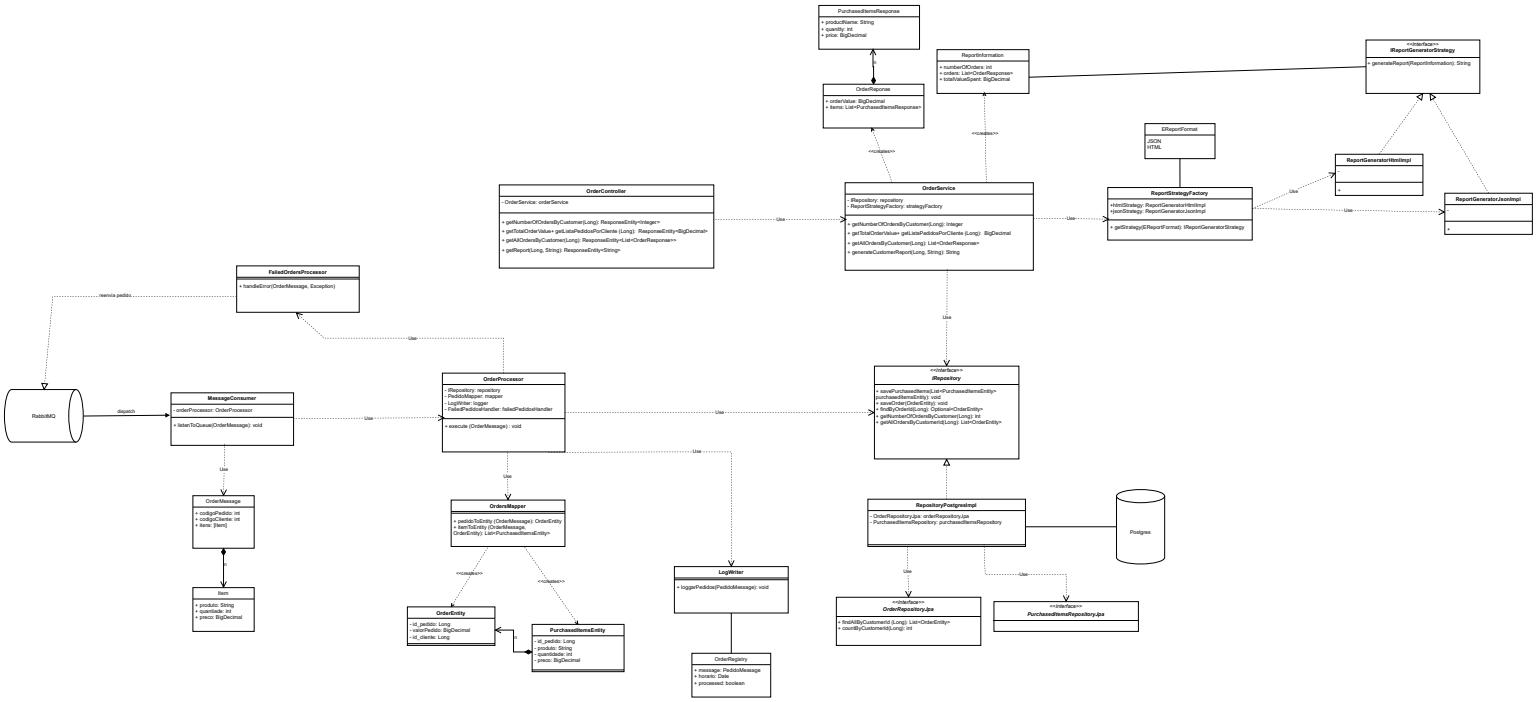
PurchasedItemsResponse
+ productName: String
+ quantity: int
+ price: BigDecimal
OrderResponse
+ orderValue: BigDecimal
+ items: List<PurchasedItemsResponse>
<<create>>
ReportInformation
+ numberOfOrders: int
+ orders: List<OrderResponse>
+ totalValueSpent: BigDecimal
<<create>>
<<Interface>>
IReportGeneratorStrategy
+ generateReport(ReportInformation): String
EReportFormat
JSON
HTML
ReportGeneratorHtmlImpl
+
ReportGeneratorJsonImpl
+
OrderController
- OrderService: orderService
+ getNumberOfOrdersByCustomer(Long): ResponseEntity<Integer>
+ getTotalOrderValue= getListaPedidosPorCliente (Long): ResponseEntity<BigDecimal>
+ getAllOrdersByCustomer(Long): ResponseEntity<List<OrderResponse>>
+ getReport(Long, String): ResponseEntity<String>
Use
OrderService
- IRepository: repository
- ReportStrategyFactory: strategyFactory
+ getNumberOfOrdersByCustomer(Long): Integer
+ getTotalOrderValue= getListaPedidosPorCliente (Long): BigDecimal
+ getAllOrdersByCustomer(Long): List<OrderResponse>
+ generateCustomerReport(Long, String): String
Use
ReportStrategyFactory
+HtmlStrategy: ReportGeneratorHtmlImpl
+JsonStrategy: ReportGeneratorJsonImpl
+ getStrategy(EReportFormat): IReportGeneratorStrategy
Use
Use
Use
FailedOrdersProcessor
+ handleError(OrderMessage, Exception)
reenvia pedido
Use
RabbitMQ
dispatch
MessageConsumer
- orderProcessor: OrderProcessor
+ listenToQueue(OrderMessage): void
Use
OrderProcessor
- IRepository: repository
- PedidoMapper: mapper
- LogWriter: logger
- FailedPedidoHandler: failedPedidoHandler
+ execute (OrderMessage): void
Use
Use
Use
<<Interface>>
IRepository
+ savePurchasedItems(List<PurchasedItemsEntity> purchasedItemsEntity): void
+ saveOrder(OrderEntity): void
+ findByOrderId(Long): Optional<OrderEntity>
+ getNumberOfOrdersByCustomer(Long): int
+ getAllOrdersByCustomerId(Long): List<OrderEntity>
Use
OrderMessage
+ codigoPedido: int
+ codigoCliente: int
+ itens: [Item]
Item
+ produto: String
+ quantidade: int
+ preco: BigDecimal
OrdersMapper
+ pedidoToEntity (OrderMessage): OrderEntity
+ itemToEntity (OrderMessage, OrderEntity): List<PurchasedItemsEntity>
<<create>>
<<create>>
OrderEntity
- id_pedido: Long
- valorPedido: BigDecimal
- id_cliente: Long
PurchasedItemsEntity
- id_pedido: Long
- produto: String
- quantidade: int
- preco: BigDecimal
LogWriter
+ loggerPedido(PedidoMessage): void
OrderRegistry
+ message: PedidoMessage
+ horario: Date
+ processed: boolean
RepositoryPostgreaImpl
- OrderRepository.jpa: orderRepository.jpa
- PurchasedItemsRepository: purchasedItemsRepository
Postgres
Use
Use
<<Interface>>
OrderRepository.jpa
+ findAllByCustomerId (Long): List<OrderEntity>
+ countByCustomerId(Long): int
<<Interface>>
PurchasedItemsRepository.jpa