Digital Business Enablement

Challenge Sprint 3: Grupo Calíope

Integrantes

Cristine Ramiro d'Arc Acocella Piccolotto Vasconcellos: RM88251

Jonathan Felix: RM88082

Marcos Vinicius Mendes Ferreira: RM86904

Marcos Maciel: RM88267

Priscila Nastacio: RM: 88849

Descrição do projeto

O e-commerce vem ganhando cada vez mais espaço no volume de vendas e já existem diversas empresas que sequer possuem lojas físicas. Embora as vendas digitais possuam uma série de vantagens em relação às vendas em lojas físicas, tais como disponibilidade 24/7, variedade de produtos e comodidade, ela carece de uma peça-chave que, em muitos casos, é a diferença entre a venda ser feita ou não: o vendedor.

Um bom vendedor (i) pode apresentar uma gama de produtos para o cliente avaliar conforme seu pedido inicial, das mais baratas às mais caras; (ii) sugere itens complementares; (iii) oferece mais detalhes e contexto/inspiração sobre os produtos; e (iv) se vale dos mais diversos argumentos para tentar persuadir o cliente a comprar um determinado produto.

Hoje, sites de vendas já buscam trazer boa parte desses elementos para seu processo de venda, com seções contendo descrições e detalhes dos produtos, itens que geralmente são comprados junto, outros produtos correlatos, etc. No entanto, toda essa comunicação é “muda”, ou seja, o usuário precisa buscar ativamente as informações, as descrições costumam ser técnicas demais e a compra deixa de ser uma experiência e se torna uma mera transação financeira.

Por meio da **Calíope**, reimaginamos a jornada de compra do cliente ao trazer uma assistente virtual que cumpre o papel de “vendedora”, aproximando ainda mais a compra online da tradicional compra física. . A Calíope atenderá seus clientes tanto por texto quanto por voz, a ser escolhido pelo(a) cliente, além de se valer de Inteligência Artificial para prever os potenciais produtos que possam interessar o(a)s clientes com base em outras vendas anteriormente realizadas.

Com isso, buscamos ajudar empresas de todos os tamanhos a aumentar o volume de vendas de suas lojas online ao mesmo tempo em que tornamos do ato de comprar digitalmente um ato mais humano e prazeroso.

Diagrama de arquitetura atualizado

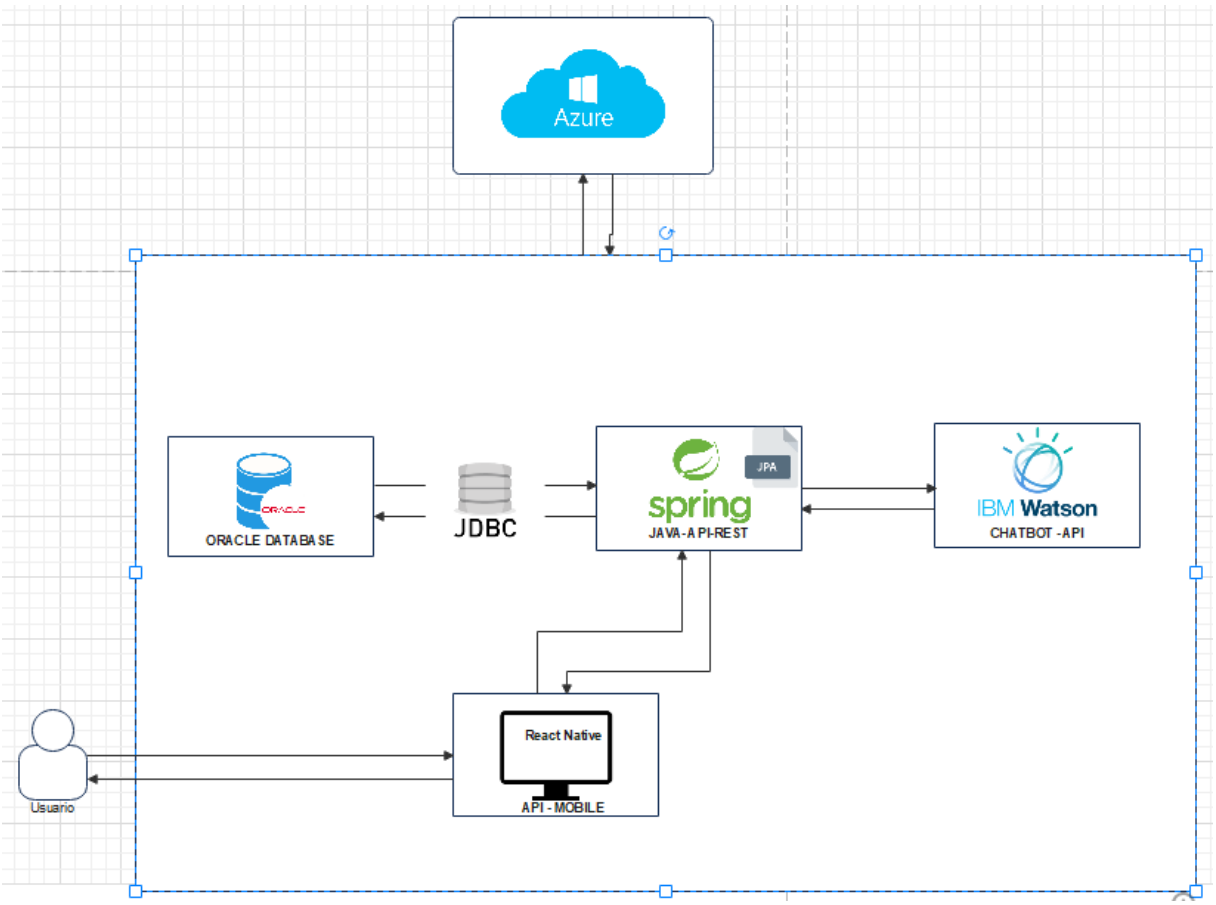

Tabela dos endpoints

Endpoint	Descrição
POST - localhost:8080/api/pedidovenda/{idUsuario}	**Realizar uma compra de produtos, passando como parâmetros o identificador do usuário na url e no RequestBody, a lista de itens**
GET- localhost:8080/api/pedidovenda/{idPedido}	**Consultar todos os itens de um pedido (Necessário rodar o primeiro endpoints antes, para realizar um pedido.)**
GET- localhost:8080/api/produto/{idProduto}	**Buscar produto por meio de seu ID**
GET- localhost:8080/api/produto/	**Buscar todos os produtos que estão registrados.**

A coleção do Insomnia contendo as requisições, juntamente com o json para o método post, está no repositório do projeto.

A principal funcionalidade do software é a realização da compra por um usuário identificado.

No PedidoVendaController, desenvolvemos o seguinte endpoint:

```
@PostMapping("/api/pedidovenda/{idUsuario}")
  public ResponseEntity<PedidoVenda> compra(@PathVariable Long idUsuario, @RequestBody
List<ItemPedidoVenda> itemPedidoVenda) {

    Usuario usuario = serviceUsuario.findById(idUsuario);
    PedidoVenda pedidoVenda = service.compra(usuario, itemPedidoVenda);

    return ResponseEntity.status(HttpStatus.OK).body(pedidoVenda);
  }
```

Este é responsável por realizar uma compra de diversos produtos por um usuário. Passamos como parâmetros o identificador do usuário e no body uma lista de itens, que denominamos itemPedidoVenda.

Buscamos o objeto usuário pelo seu identificador e, juntamente com a lista de itens, passamos para a classe PedidoVendaService, através do método compra.

```
public PedidoVenda compra(Usuario usuario, List<ItemPedidoVenda> itemPedidoVenda) {
    BigDecimal valorTotalItem = new BigDecimal(0);
    PedidoVenda pedidoVenda = new PedidoVenda();
    ItemPedidoVenda newItemPedidoVenda = new ItemPedidoVenda();
    repo.save(pedidoVenda);

    for (ItemPedidoVenda item : itemPedidoVenda) {
      valorTotalItem = valorTotalItem.add(valorTotal(item.getQuantidadePedida(),
item.getProduto().getPrecoUnitario()));
      pedidoVenda.setEmpresa(item.getProduto().getEmpresa());
      pedidoVenda.setUsuario(usuario);
      pedidoVenda.setDataPedidoVenda(Instant.now());

      newItemPedidoVenda = new ItemPedidoVenda(pedidoVenda,
item.getProduto(),item.getQuantidadePedida(),item.getProduto().getPrecoUnitario(),
      valorTotal(item.getQuantidadePedida(), item.getProduto().getPrecoUnitario()));
      // item.setPedidoVenda(pedidoVenda)
      repositoryItem.save(newItemPedidoVenda);
    }
    pedidoVenda.setValorTotalPedidoVenda(valorTotalItem);
```

```
  repo.save(pedidoVenda);
  pedidoVenda.setItemPedidoVendas(itemPedidoVenda);
  return pedidoVenda;

 }
```

Este método instancia um objeto do tipo PedidoVenda, necessário para guardar cada objeto Item.

Depois, fazemos um loop para percorrer a lista de itens e guardá-los no banco, relacionando cada item pedido com o mesmo id do PedidoVenda.

Esse é o post:

Post: localhost:8080/api/pedidovenda/1

Json:

```
[{
    "produto":{
        "empresa": {
            "id": 1,
            "cnpj": 1234,
            "razaoSocial": "fiap",
            "nomeFantasia": "caliope"
        },
        "nrSku": 10,
        "nome": "Tunica Bege",
        "precoUnitario": 150.99,
        "quantidade": 20,
        "descricao": "Tunica Bege linda",
        "nomeMarca": "marca"
    },
    "quantidadePedida": 2
},
    { "produto":{
        "empresa": {
            "id": 1,
            "cnpj": 1234,
            "razaoSocial": "fiap",
            "nomeFantasia": "caliope"
        },
        "nrSku": 10,
        "nome": "Tunica Bege",
        "precoUnitario": 150.99,
        "quantidade": 20,
        "descricao": "Tunica Bege linda",
        "nomeMarca": "marca"
    },
    "quantidadePedida": 2
}
]
```

Esse é o retorno esperado (o objeto PedidoVenda com uma lista de itens comprados, que foi passado no body!):


```
{
    "id": 1,
    "empresa": {
        "id": 1,
        "cnpj": 1234,
        "razaoSocial": "fiap",
        "nomeFantasia": "caliope"
    },
    "usuario": {
        "id": 1,
        "nomeUsuario": "Jonathan",
        "sobrenomeUsuario": "Felix",
        "email": "example@gmail.com",
        "senha": "dev12",
        "cpf": 5000,
        "genero": "MASCULINO"
    },
    "dataPedidoVenda": "2022-08-20T17:52:52.697998Z",
    "valorTotalPedidoVenda": 603.96,
    "itemPedidoVendas": [
        {
            "id": {
                "idPedidoVenda": 0,
                "idProduto": 0
            },
            "produto": {
                "id": 0,
                "empresa": {
                    "id": 1,
                    "cnpj": 1234,
                    "razaoSocial": "fiap",
                    "nomeFantasia": "caliope"
                },
                "nrSku": 10,
                "nome": "Tunica Bege",
                "precoUnitario": 150.99,
                "quantidade": 20,
                "descricao": "Tunica Bege linda",
                "nomeMarca": "marca"
            },
            "quantidadePedida": 2,
            "valorDescontoItemVenda": null,
            "valorUnitarioVenda": null,
            "valorTotalItemVenda": null
        },
        {
            "id": {
                "idPedidoVenda": 0,
                "idProduto": 0
            },
            "produto": {
                "id": 0,
                "empresa": {
                    "id": 1,
```

```
                    "cnpj": 1234,
                    "razaoSocial": "fiap",
                    "nomeFantasia": "caliope"
                },
                "nrSku": 10,
                "nome": "Tunica Bege",
                "precoUnitario": 150.99,
                "quantidade": 20,
                "descricao": "Tunica Bege linda",
                "nomeMarca": "marca"
            },
            "quantidadePedida": 2,
            "valorDescontoItemVenda": null,
            "valorUnitarioVenda": null,
            "valorTotalItemVenda": null
        }
    ]
}
```

Link GitHub

https://github.com/cristineacocella/sprint3_digital