

2ª Avaliação – 2023/1

Crie um aplicativo que, utilizando os web-services disponibilizados na URL **`http://argo.td.utfpr.edu.br/clients/ws/<recurso>`** permitam que um usuário realize as seguintes operações:

- Cadastrar novos clientes ou consulte e atualize dados de clientes já cadastrados (busca por cpf). Ao cadastrar um cliente, deve ser informado o CEP do mesmo, e usar o web-service da **`viacep.com.br`** para obter os dados correspondentes ao endereço.
- Consulte os setores e os produtos disponíveis para venda. Esta consulta deve ser usada na hora de criar um novo pedido.
- Crie um pedido, adicionando produtos cadastrados, e o envie para a API. Deve ser possível selecionar o setor, e em seguida selecionar produtos deste setor, informar a quantidade e adicionar ao pedido. Também é possível alterar um pedido, adicionando, alterando ou removendo itens. A lista de itens alterada deve ser enviada completa, substituindo a anterior.

O código do vendedor a ser usado deve ser o RA do aluno (já estão todos cadastrados, não há web-service para cadastro).

Ao enviar um novo pedido, não é necessário informar os ids do Pedido e dos itens, podendo ser atribuídos valores aleatórios, pois o servidor irá ignorá-los (no POST). No PUT, deverá ser enviado o pedido com o id original, porém os **ids dos itens** serão ignorados. Exemplo do JSON para envio de um pedido:

```
{  "cpfCliente": "12332133300",
    "data": "02/05/2023 15:31:37",
    "idVendedor": 1742841,
    "itens": [ {      "idProduto": 8,
                    "precoUnitario": 45.49,
                    "quantidade": 20.0 },
                { "idProduto": 9,
                    "precoUnitario": 23.79,
                    "quantidade": 15.0 },
                { "idProduto": 11,
                    "precoUnitario": 3455.99,
                    "quantidade": 3.0
                } ],
    "percentualDesconto": 8.0 }
```

As datas serão formatadas sempre usando a máscara **`dd/MM/yyyy HH:mm:ss`**, tanto no envio como no retorno. Configurar o GSON corretamente.

Classes do lado do servidor:

```
public class Setor {
    private Long id;
    private String descricao;
    private Double margem;
}

public class Cliente {
    private String cpf;
    private String nome;
    private String cep;
    private String logradouro;
    private String complemento;
    private String bairro;
    private String numero;
    private String localidade;
    private String uf;
    private String ddd;
    private String telefone;
}

public class Produto {
    private Long id;
    private String descricao;
    private Double preco;
    private Double estoque;
    private Setor setor;
}

public class ItemPedido {
    private Long id;
    private Long idPedido;
    private Long idProduto;
    private Double quantidade;
    private Double precoUnitario;
}

public class Pedido {
    private Long id;
    private Long idVendedor;
    private String cpfCliente;
    private Date data;
    private Double percentualDesconto;
    List<ItemPedido> itens;
}
```

Para buscar os produtos de um setor, utilize o parâmetro **setor** na busca:

<http://argo.td.utfpr.edu.br/clients/ws/produto?setor=2>

Para buscar os pedidos, é necessário informar o código do vendedor, o cpf do cliente, ou ambos:

<http://argo.td.utfpr.edu.br/clients/ws/pedido?vendedor=1231234&cpf=12312312300>

Todos os recursos obedecem ao padrão:

- A URL raiz suporta GET para listar (apenas pedido exige parâmetro de busca), e POST insere um novo objeto no BD. Exemplo para listar os setores

<http://argo.td.utfpr.edu.br/clients/ws/setor>

- A URL raiz, acrescentada de um **id** (cpf no caso do cliente) suporta GET para consultar, PUT para alterar (não é possível alterar o **id**), e DELETE remove um objeto no BD. Ex: para alterar um pedido com id = 43, enviar PUT para a URL, contendo o JSON com os novos dados do pedido. O id do pedido enviado **deve ser** 43.

<http://argo.td.utfpr.edu.br/clients/ws/pedido/43>

Requisitos:

- Usar layouts diferenciados para dispositivo na vertical ou na horizontal.
- Ao entrar no aplicativo, se não houver conexão com a internet, avisar de que não é possível usar o App de forma offline.