# Teste Desenvolvedor UDS

Elaborar uma aplicação com os seguintes casos de uso:

1. Manter Produto: CRUD simples da entidade Produto (id: UUID, código: String, nome: String, preçoUnitário: Double).
   1. Todos os atributos são obrigatórios.
   2. ID, Código e nome não podem se repetir.
   3. Preço é monetário e deve ser maior que zero.
   4. Deve ser possível pesquisar os produtos por qualquer atributo, exceto pelo id.
2. Manter Pessoa: CRUD simples da entidade Pessoa (id: UUID, nome: String, dataNascimento: Date).
   1. Todos os atributos são obrigatórios.
   2. ID e Nome não podem se repetir.
   3. Deve ser possível pesquisar as pessoas por qualquer atributo, exceto pelo id.
3. Registrar Pedido de Venda: permite incluir ou excluir um pedido de venda, sabendo que o Pedido (id: UUID, cliente: Pessoa, número: Integer/Autoincremento, emissão: Date, total: Double) contém um ou mais ItemPedido (id: UUID, produto: Produto, quantidade : Double, preçoUnitário : Double/recuperado do produto, percentualDesconto : Double, total : Double).
   1. Todos os atributos são obrigatórios.
   2. IDs não podem se repetir.
   3. Todos os valores numéricos devem ser tratados com duas casas decimais e devem ser > 0 (exceto desconto, que pode ser >= 0).
   4. Preço e totais são monetários.
   5. Deve ser possível pesquisar os pedidos por qualquer atributo, exceto pelo id.

Detalhes importantes:

Implemente nas tecnologias e *frameworks* para as quais você está se candidatando.

Se o seu teste seletivo for para *front-end*, implemente a aplicação com dados transientes (em memória) ou em armazenamento local ou usando uma mock API.

Se o seu teste seletivo for para *back-end*, implemente a aplicação como uma API REST com mapeamento objeto relacional em um banco de dados MySQL ou Postgresql.

Se o seu teste seletivo for *full stack* implemente conforme descrito em back-end + front-end.

Todo seu código deverá estar disponível no Github.

O uso de testes automatizados será considerado na avaliação.

Caso não consiga implementar todos os casos de uso, entregue o máximo que conseguir com a melhor qualidade possível.

Esquemas do banco de dados

CREATE TABLE public.pessoa

(

id uuid NOT NULL,

nome character varying(50) NOT NULL,

datanascimento date NOT NULL,

PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

);

ALTER TABLE public.pessoa

OWNER to postgres;

CREATE TABLE public.produto

(

id uuid NOT NULL,

codigo character varying(50) NOT NULL,

nome character varying(50) NOT NULL,

valorunitario money NOT NULL,

CONSTRAINT produto\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

ALTER TABLE public.produto

OWNER to postgres;

CREATE TABLE public.pedido

(

id uuid NOT NULL,

numero bigserial NOT NULL,

emissao time without time zone NOT NULL,

total money NOT NULL,

cliente\_id uuid NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (cliente\_id)

REFERENCES public.pessoa (id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

)

WITH (

OIDS = FALSE

);

ALTER TABLE public.pedido

OWNER to postgres;

CREATE TABLE public.itempedido

(

id uuid NOT NULL,

quantidade numeric(9, 2) NOT NULL,

valorunitario money NOT NULL,

percentualdesconto numeric(3, 2) NOT NULL,

total money NOT NULL,

pedido\_id uuid NOT NULL,

produto\_id uuid NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (pedido\_id)

REFERENCES public.pedido (id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (produto\_id)

REFERENCES public.produto (id) MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE

)

WITH (

OIDS = FALSE

);

ALTER TABLE public.itempedido

OWNER to postgres;