

**Olist**

Rafaela Aparecida Teixeira

**TechStartPro**

Desafio proposto pela Olist, para a vaga de Analista  
Desenvolvedor Jr.

**Curitiba**

**2020**

## Sumário

1.	Introdução.....	3
2.	Configuração de Ambiente .....	4
2.1.	Instruções de instalação (configuração) e teste .....	4
2.2.	Descrição do ambiente de trabalho .....	4
3.	Descrição.....	5
3.1.	Categoria .....	5
3.2.	Produto.....	5
3.3.	Test.....	5
4.	Conclusão.....	7

## **1. Introdução**

Desafio proposto pela equipe de tecnologia da Olist, com o objetivo de implementar uma solução, no qual um aplicativo armazena produtos e categorias. No qual é desenvolvido um sistema que importa dados de um arquivo CSV, para um banco de dados.

## 2. Configuração de Ambiente

### 2.1. Instruções de instalação (configuração) e teste

- Python 3.7 ou superior.
- MySQL 8.0 ou superior. Connector/C++ 8.0.22 ou superior.
- Prompt/Terminal dentro da pasta do projeto e execute o comando `pip install -r requirements.txt` para instalar todas as dependências.
- Configurar o arquivo `.env`:
  - `MYSQL_HOST = [PREENCHER COM O HOST]`  
`MYSQL_PORT = [PREENCHER COM A PORT]`  
`MYSQL_DATABASE = [PREENCHER COM O NOME DO BANCO UTILIZADO NA APLICAÇÃO]`  
`MYSQL_USER = [PREENCHER COM O NOME DO USUARIO]`  
`MYSQL_PASSWORD = [PREENCHER COM A SENHA]`
- Executar o comando no prompt `python manage.py makemigrations` para criar a migração.
- Executar o comando no prompt `python manage.py migrate` para realizar a migração.
- Executar o comando no prompt `python manage.py test olist.tests.test_models` para realizar o test do arquivo `test.models`.
- Rodar o comando no prompt `python manage.py test olist.tests.test_views` para realizar o test do arquivo `test.Views`.
- Para iniciar o servidor, execute o comando `python manage.py runserver`.

### 2.2. Descrição do ambiente de trabalho

- Sistema Operacional Windows 10 Home Single Language.
- Configuração do computador:
  - Processador Intel Core I7-7500U (Placa de Vídeo Integrada), Memória 8 GB RAM.
  - Visual Studio Code 1.50.1

### 3. Descrição

Projeto desenvolvido na linguagem Python com o Framework Django para armazenamento de dados de produtos e categorias.

O projeto foi desenvolvido seguindo as seguintes diretrizes:

#### 3.1. Categoria

- **Models:**
  - Id: Integer(PK)
  - name: String(max\_length= 50)
- **Views:**
  - Foram utilizados dois métodos para importação dos dados via arquivo csv:
    - save\_data()
    - import\_csv()
- **Template:**
  - Pagina html criada com Boodstrap 4, jquery e cloudflare:
    - category\_import.html
    - base.html

#### 3.2. Produto

- **Models:**
  - Id: Integer(PK)
  - name: String(max\_length= 50)
  - description: String(max\_length= 1000)
  - value: Double
  - category = models.ManyToManyField(Category)
- **Views:**
  - Foram utilizados 3 métodos para realização do CRUD:
    - forms()
    - list()
    - delete()
- **Template:**
  - Pagina html criada com Boodstrap 4, jquery e cloudflare:
    - forms.html
    - list.html
    - base.html
- **Forms:**
  - Foi criado um objeto de product para criação do formulario
- **Filter:**
  - Foi criado um objeto de product para criação dos filtros.

#### 3.3. Test

- **Models:**
  - Tests unitarios da camada modelo:
    - setUpTestData()
    - test\_category\_name\_label()
    - test\_category\_name\_max\_length()
    - test\_product\_get()
    - test\_product\_name\_max\_length()

- test\_product\_description\_max\_length()
- **Views:**
  - Tests unitários da camada views:
    - setUpTestData()
    - test\_list\_response\_200()
    - test\_import\_csv\_response\_200()
    - test\_forms\_response\_200()

## **4. Conclusão**

O desafio foi concluído com todas os requisitos entregues, seguindo boas práticas de programação, criação de testes automatizados, projeto alocado em repositório no GitHub, etc.