



UNIVERSIDADE CATÓLICA DE ANGOLA  
FACULDADE DE ENGENHARIA

# **Perfumaria Pier Parfum**

Luanda, 2021



UNIVERSIDADE CATÓLICA DE ANGOLA  
FACULDADE DE ENGENHARIA

CURSO: ENGENHARIA INFORMÁTICA

SALA: 104

TURMA: EI2A

PERÍODO: NOITE

ANO: 4º.

NÚMERO DE ELEMENTOS: 1

DOCENTE

---

Pedro Mbote

## **Integrantes do grupo**

21099 - Eufrânio Diogo

# Índice

<b>Integrantes do grupo</b>	<b>3</b>
<b>Introdução</b>	<b>5</b>
<b>Tecnologias</b>	<b>6</b>
Front-end	6
HTML	6
CSS	6
JS	6
Back-end	7
Python	7
Postgresql	7
<b>Estrutura projecto</b>	<b>8</b>
Estrutura organizacional	8
<b>Estrutura de navegação</b>	<b>9</b>
<b>Estrutura aplicação</b>	<b>10</b>
<b>Base de dados</b>	<b>11</b>
Diagrama Relacional	11
Diagrama de entidade e relacionamento	12
Diagrama de classes	13
<b>Instruções</b>	<b>14</b>
<b>Mockups</b>	<b>15</b>
<b>Conclusão</b>	<b>16</b>

## Introdução

Este website foi construído para representar uma marca de perfume **Pierer Parfum** de origem francesa com o principal objectivo de autopromoção, venda de seus produtos e um contacto mais próximo com relação o seu público alvo.

Que no final do website, teremos o website dividido em duas partes nomeadamente: Front-end, Back-end. Sendo que o Front-end responsável pela a apresentação e interação com usuário e o Back-end focado na implementação da lógica de negócio.

# Tecnologias

No processo de construção do projecto foram usadas várias tecs que foram separadas em tecs que umas actuaram na parte do Front-end do projecto e outras que foram usadas dentro do Back-end.

O projecto foi constituído por:

- HTML 46.2%
- CSS 21.5%
- Python 17.8%
- JavaScript 14.5%

Dados retirados do repositório do projecto hospedado em [github.com](https://github.com).

## Front-end

### HTML

O HTML foi usado para implementação estrutural das páginas, no caso a organização estrutural do conteúdo a ser apresentado.

### CSS

CSS sendo aplicada para a estilização completa do website.

### JS

Em particular JS foi das coisas que mais foram usadas dentro do webapp, sendo usado para comunicar diversas vezes com o Back-end para procurar determinada informação no Back-end, verificação de autenticidade do usuário, renderização condicional e entre outros aspectos implementados dentro do projecto com JS.

## **Back-end**

### Python

O Python foi usada para implementação da lógica de negócio da aplicação, desde os critérios de segurança aos critérios de entrega, inserção, alteração da dados através da API desenvolvida, a renderização de páginas no servidor e módulos de segurança da app, para verificação da conta, gestão de JWTs.

### Postgresql

Postgresql foi a base de dados relacional escolhida para o armazenamento de todas as informações relacionados com o projecto.

# Estrutura projecto

## Estrutura organizacional

A organização de um projecto encontrasse nos pontos mais importantes para a construção de uma boa aplicação, não basta a produção do código para que a aplicação rode e sim também uma boa organização, para que as coisas tenham lógica e não seja complicado de gerir depois que o software começa a crescer.

Organização do projecto:

/pierer-parfum

/blue\_prints -> Pasta para blue\_prints

/controllers -> Pasta onde guardamos os controllers da API para a comunicação ajax entre o front-end e o back-end da aplicação

/static -> Pasta onde se localiza os arquivos estáticos para acesso publico na aplicação

/assets -> Pasta que guarda tudo que tem a ver com imagens, fonts, vídeos, e outros arquivos de mídia

/fonts

/images

/css -> Arquivos CSS

/js -> Localização de arquivos JS para uso nas páginas HTML

/auth -> localização de todos arquivos JS para autenticação e segurança

/templates -> Pasta para a localização dos templates para o flask

app.py -> Arquivo inicial do programa

DB\_STRUCTURE.txt -> Pequena descrição textual da base de dados

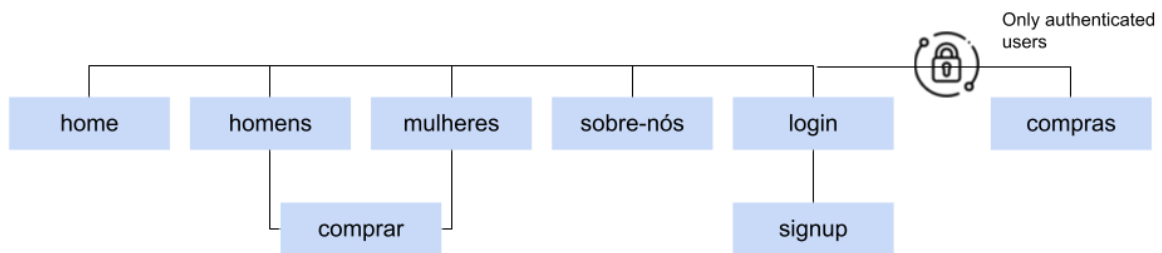
requirements.txt -> Arquivo de dependências para colocar a app running.

sm\_bd.sql -> Arquivo de backup da base de dados



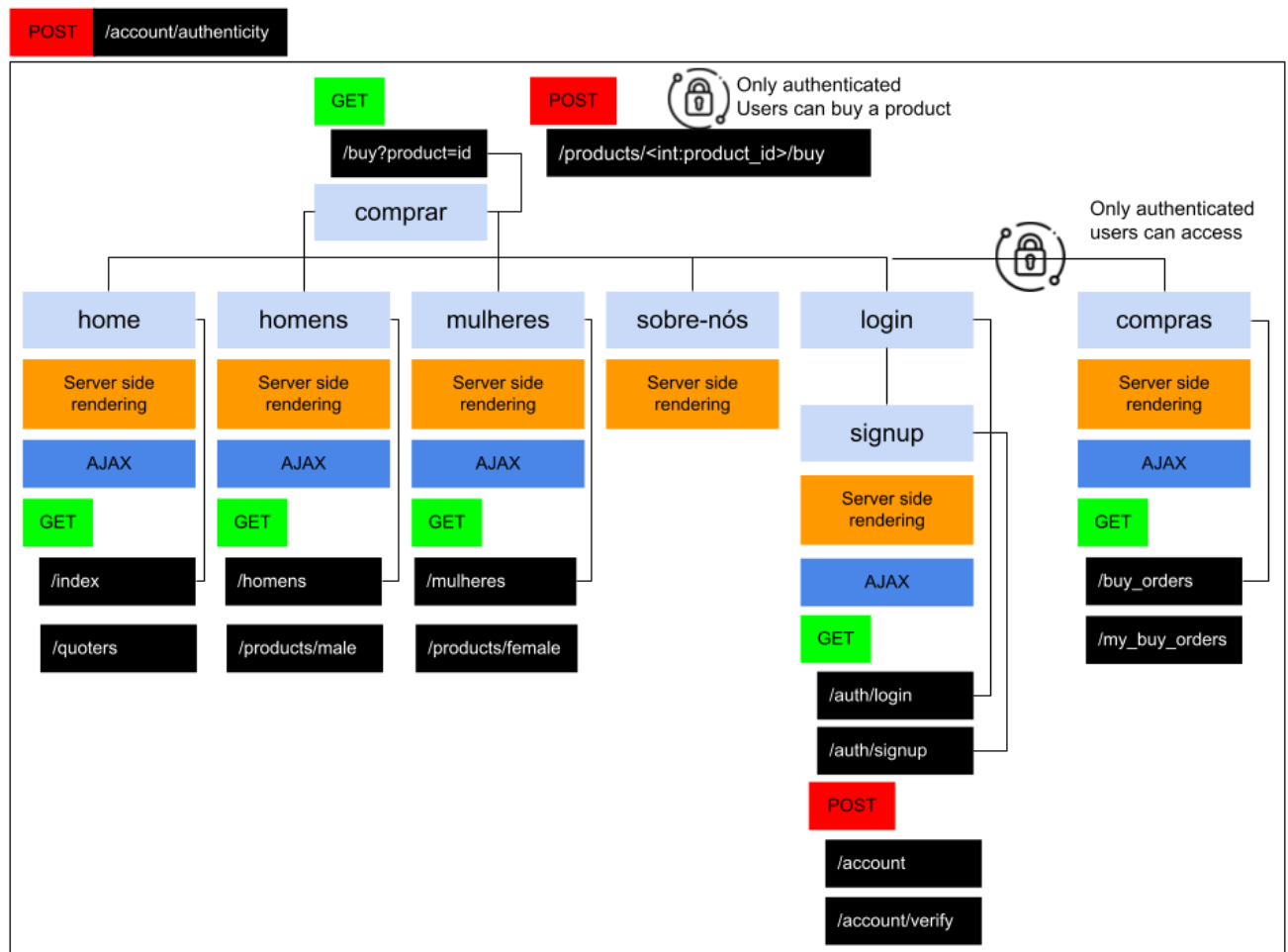
## Estrutura de navegação

O projecto possui no total 8 páginas webs descritas por: **home**, **homens**, **mulheres**, **compras**, **about**, **login**, **signup**, **comprar**. Sendo que as páginas login e cadastro são páginas que redirecionam o usuário para outra stack de navegação de de páginas onde o usuário navega entre a página de login e cadastro, e as outras páginas como home, homens e mulheres mantêm o usuário dentro da mesma stack de navegação.



## Estrutura aplicação

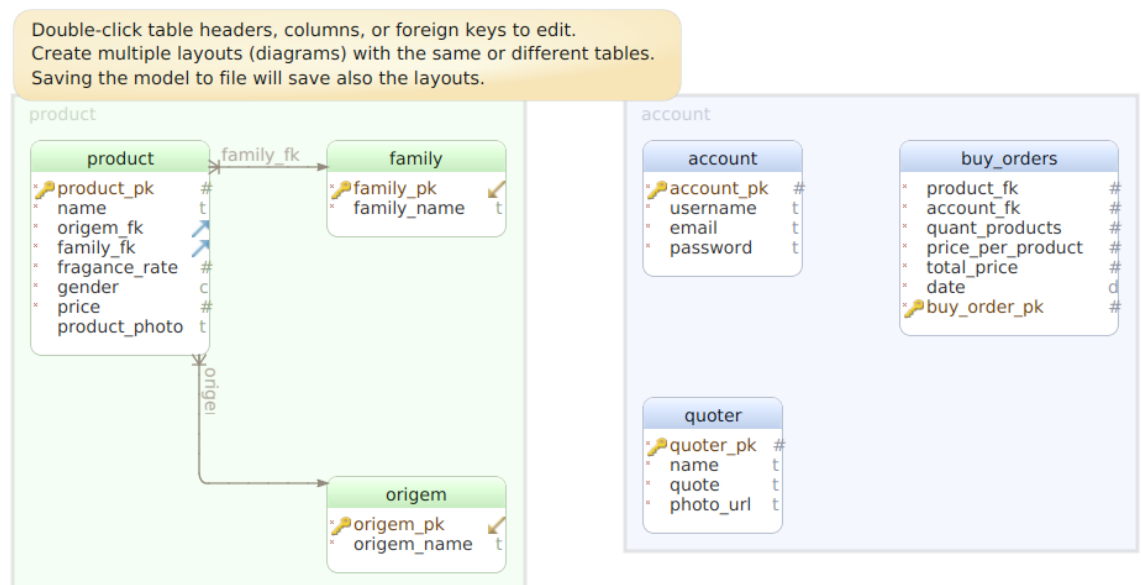
Nesta figura é referenciada a estrutura arquitetónica de toda aplicação, interligando o frontend e o backend, e todos os métodos referenciados para o seu devido funcionamento.



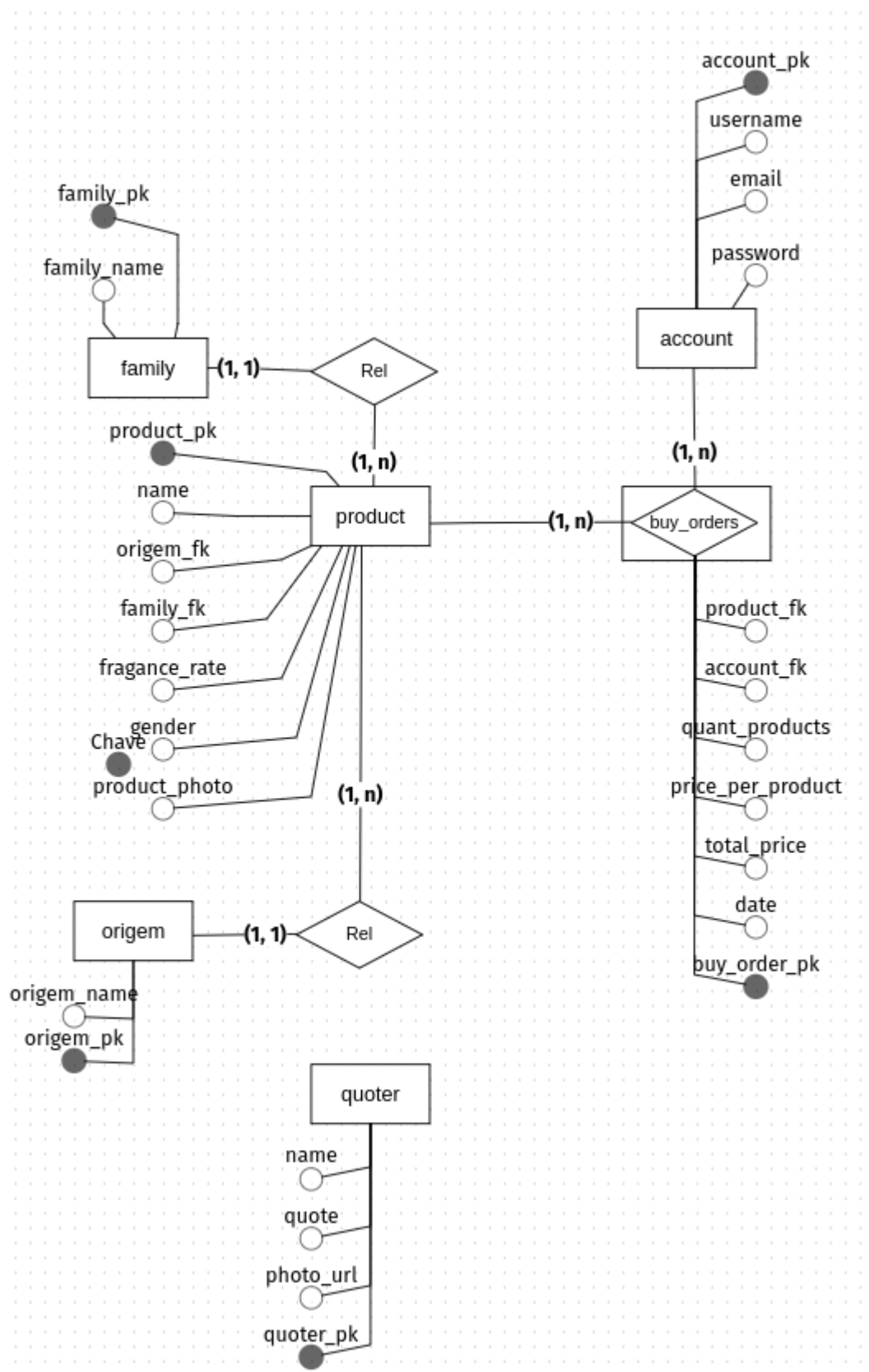
## Base de dados

A base de dados escolhida para o projecto foi o postgresql, uma base de dados relacional de fácil uso robusta, e que usa o SQL como linguagem de interação.

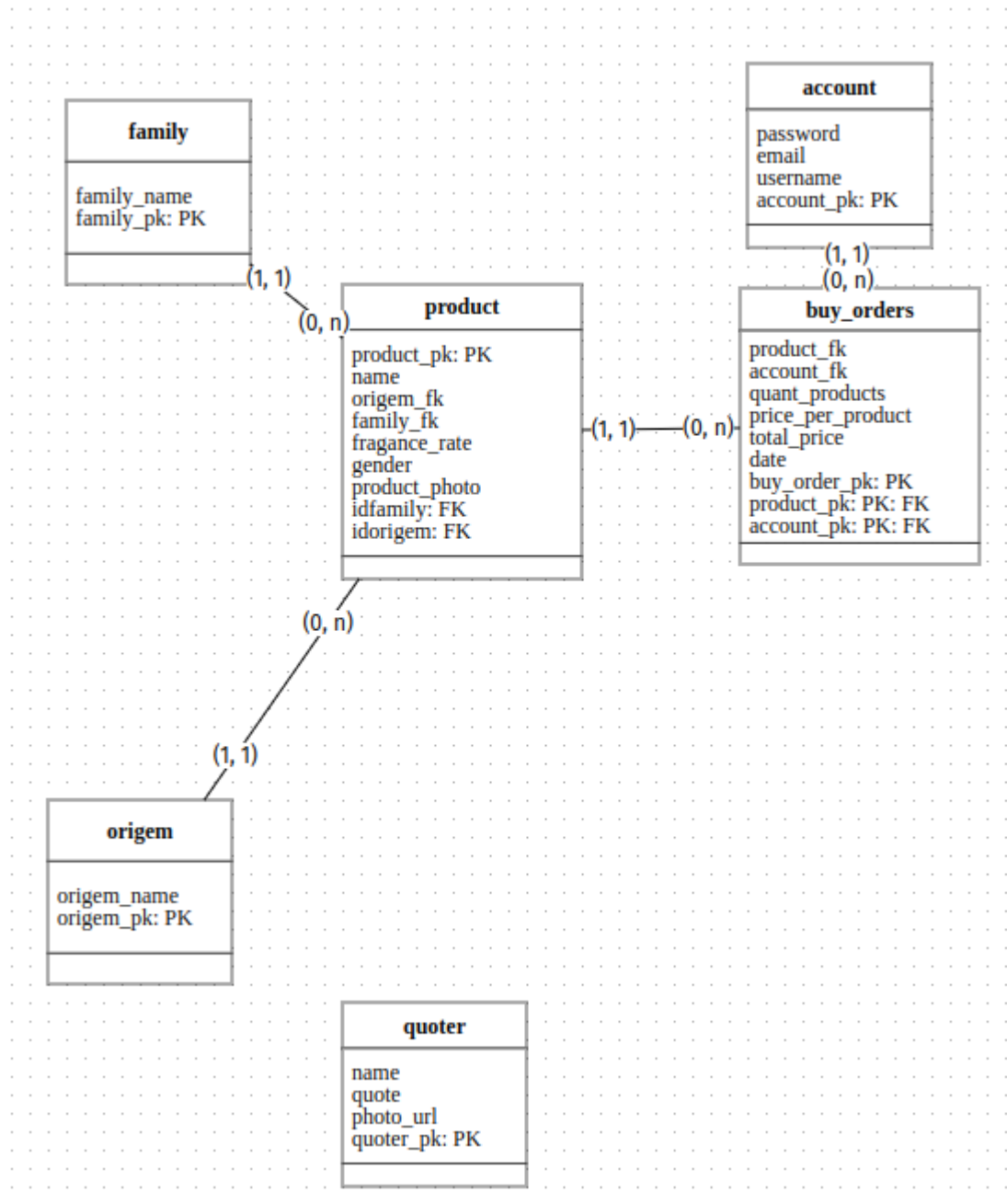
## Diagrama Relacional



## Diagrama de entidade e relacionamento



## Diagrama de classes



## Instruções

Aqui listo os métodos para colocar a aplicação a rodar em uma máquina local:

1. `pip3 install -r requirements.txt`
2. Criar uma base de dados no postgre chamada **sm\_bd**
3. Fazer o restore dos dados no banco de dados, sendo que o backup está dentro da pasta do projecto nomeado como **sm\_bd.sql**.
4. `python3.8 -m flask run`

## Mockups

Os mockups podem ser vistos com qualidade e visibilidade acessando: [Pierer Parfum Mockups on Figma](#).

## **Conclusão**

Concluimos que com Python usando flask no backend e HTML, CSS, JS no frontend é possível criarmos uma aplicação completa e com recursos básicos para os web apps hj criados.