

# Modelagem do sistema

## PyCommerce

**Disciplina:** Programação Orientada a Objetos

**Discente:** Renan do Nascimento Lisboa

**Matrícula:** 20231014040028

### 1. Descrição

PyCommerce é um sistema que consiste em um comércio eletrônico que visa facilitar a venda de produtos na internet utilizando menus intuitivos para garantir o pleno acesso dos usuários. Este conta com um sistema de cadastros de produtos e categorias feitos pelo usuário administrador, consulta de produtos e categorias pelo visitante e criação de pedido e realização da compra feita pelo cliente.

#### 1.1 Lista de requisitos funcionais

**Criar perfil:** Permite que um visitante possa criar seu perfil para poder realizar compras.

**Gerenciar perfil:** Permite que um usuário atualize ou remova seu perfil.

**Gerenciar produtos:** O administrador poderá criar, atualizar ou remover os produtos do sistema.

**Gerenciar categorias:** O administrador poderá criar, atualizar ou remover as categorias do sistema.

**Ver categorias:** Qualquer usuário poderá visualizar as categorias disponíveis no sistema.

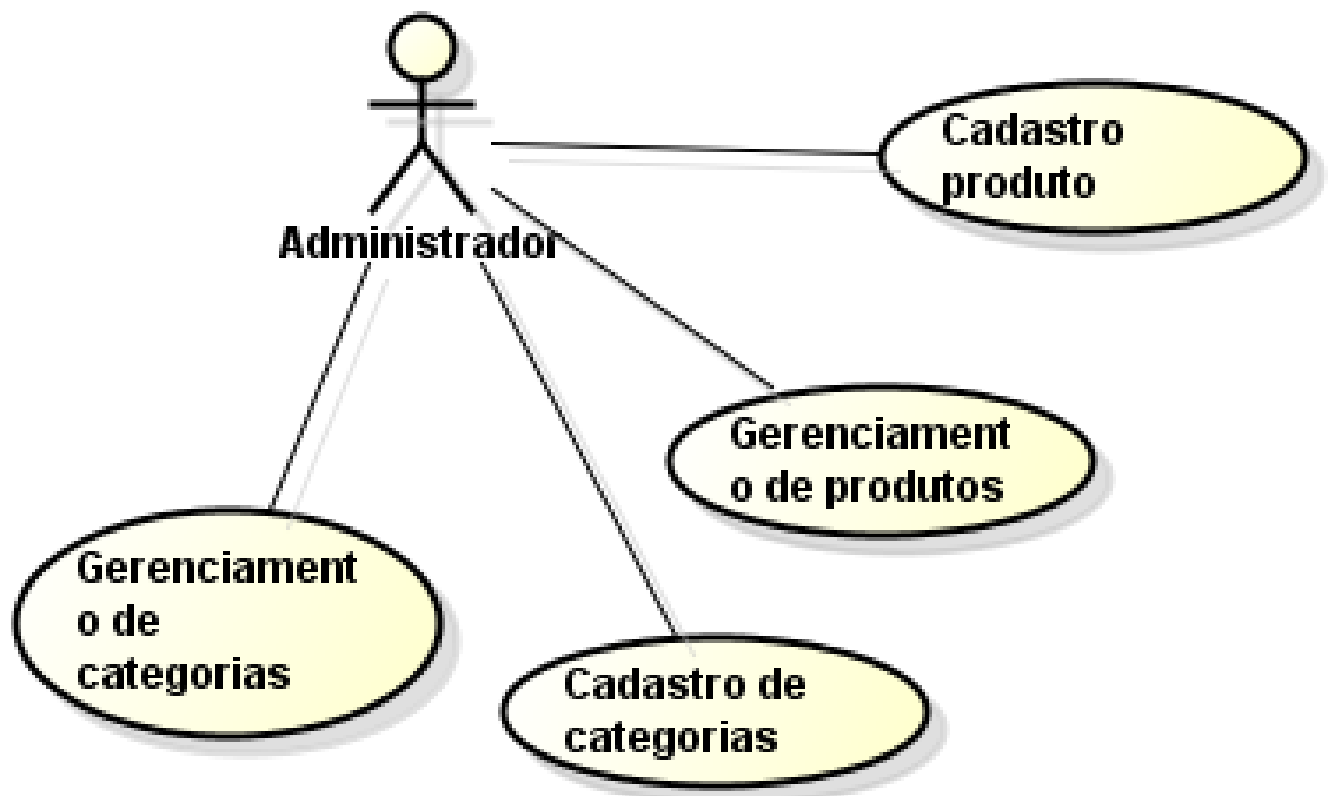
**Ver produtos:** Qualquer usuário poderá listar um ou vários produtos disponíveis, incluindo os detalhes de cada um.

**Fazer pedido:** O usuário com conta cadastrada e logada poderá adicionar itens de um produto ao carrinho.

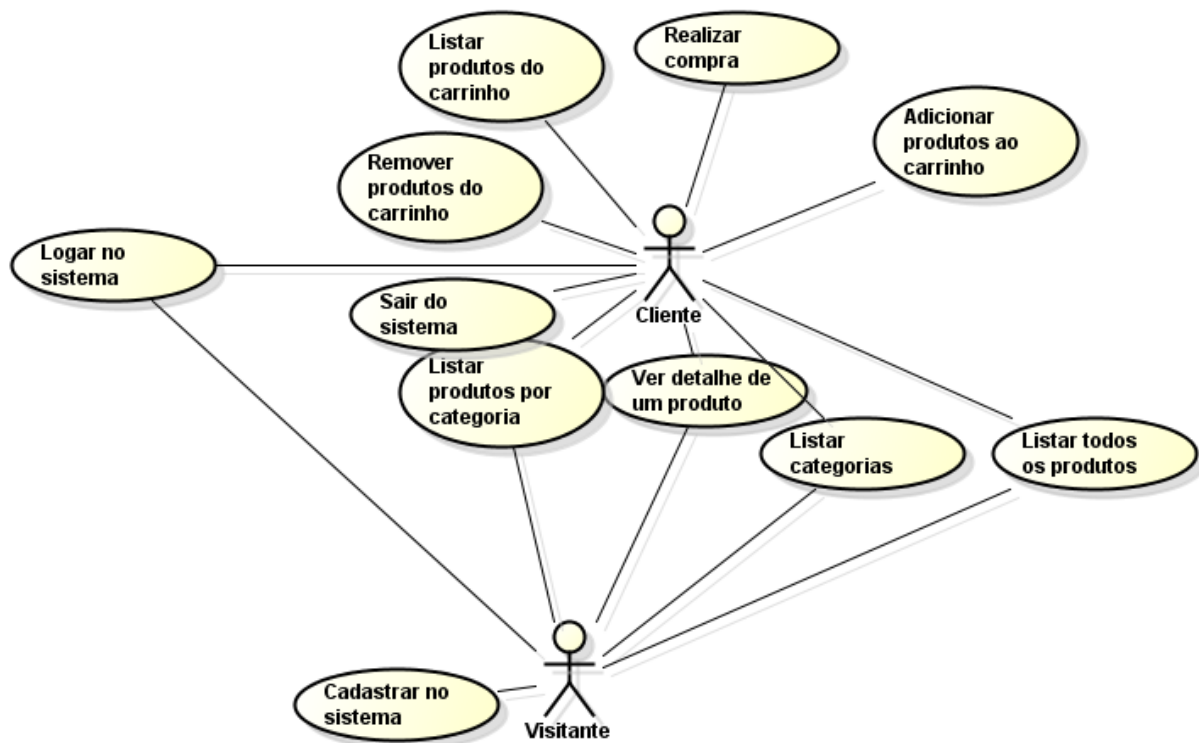
**Gerenciar pedido:** O usuário com conta cadastrada e logada poderá listar e atualizar, desfazer os itens disponíveis no carrinho e realizar a compra.

## 2. Diagrama de casos de uso

### 2.1 Administrador

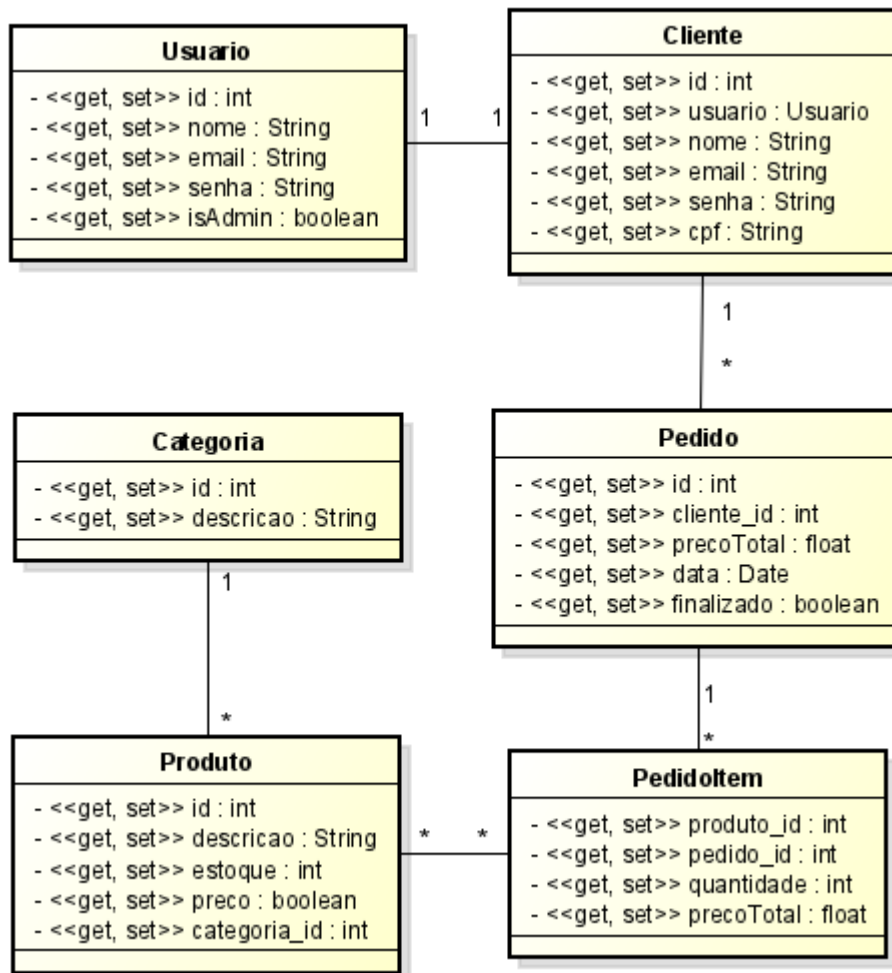


## 2.2 Visitante e cliente

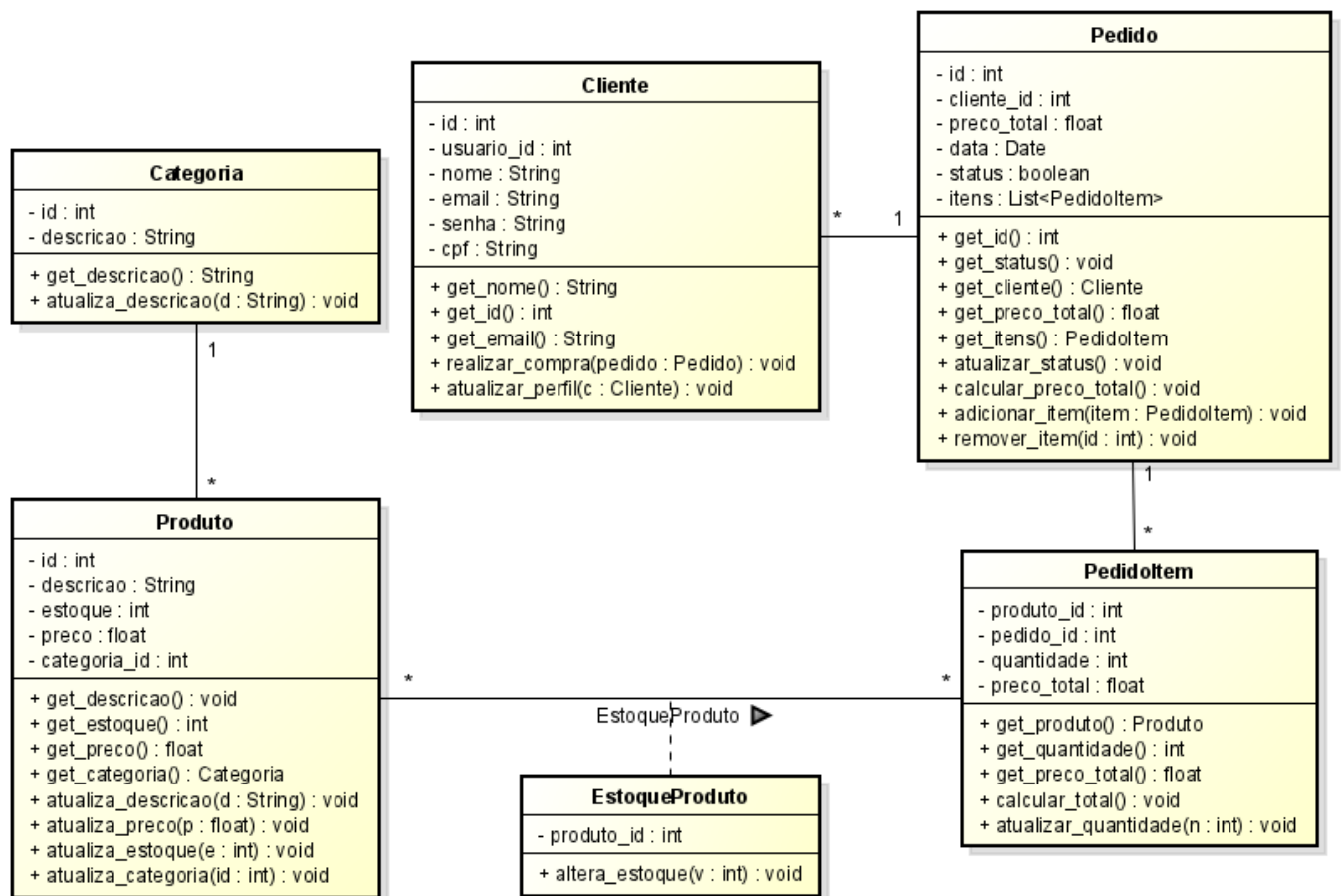


## 3. Diagrama de classes com modelo das entidades

### 3.1 Diagrama de modelos

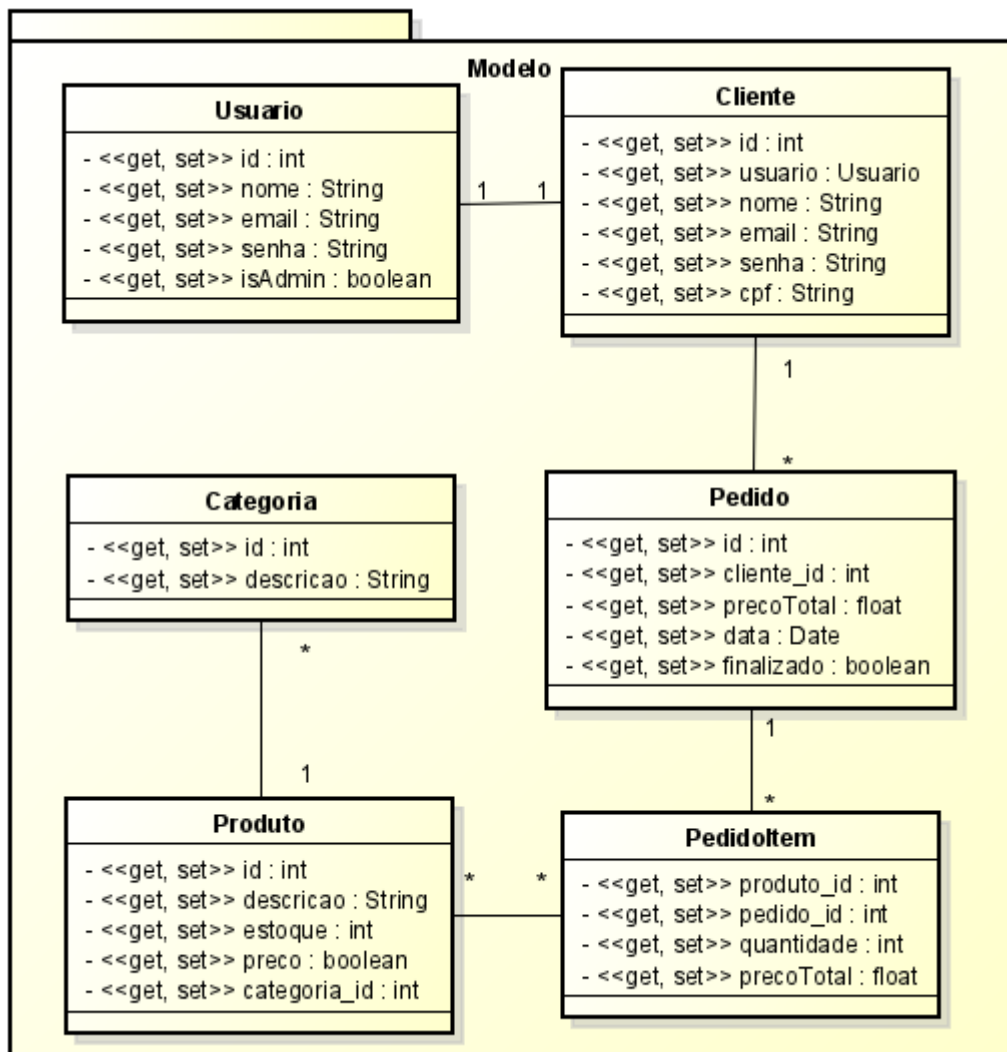
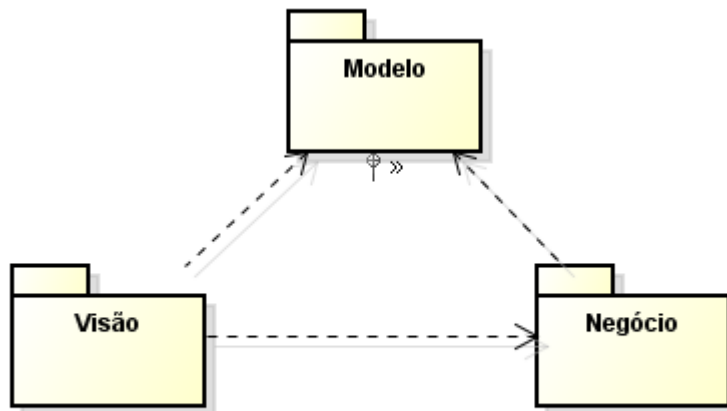


## 3.2 Diagrama de classes



## 4. Diagrama de classes com as classes de negócio

# Arquitetura do sistema



## Negócio

<<static>>

### NCliente

- clientes : List<Cliente>

+ listar\_clientes() : List<Usuario>  
+ ver\_cliente(id : int) : Usuario  
+ cadastrar\_cliente(u : Usuario) : void  
+ excluir\_cliente(id : int) : void  
+ atualizar\_cliente(u : Usuario) : void

<<static>>

### NUsuario

- usuarios : List<Usuario>

+ listar\_usuarios() : List<Usuario>  
+ ver\_usuario(id : int) : Usuario  
+ cadastrar\_usuario(u : Usuario) : void  
+ excluir\_usuario(id : int) : void  
+ atualizar\_usuario(u : Usuario) : void

<<static>>

### NPedido

- pedidos : List<NPedido>

+ listar\_pedidos() : List<Pedido>  
+ ver\_pedido(id : int) : Pedido  
+ criar\_pedido(p : Pedido) : void  
+ excluir\_pedido(id : int) : void  
+ atualizar\_pedido(p : Pedido) : void

<<static>>

### NPedidoItem

- pedido\_itens : List<PedidoItem>

+ listar\_pedido\_itens() : List<PedidoItem>  
+ ver\_pedido\_item(id : int) : PedidoItem  
+ criar\_pedido\_item(p\_item : PedidoItem) : void  
+ excluir\_pedido\_item(id : int) : void  
+ atualizar\_pedido\_item(p : PedidoItem) : void

<<static>>

### NCategoria

- categorias : List<Categoria>

+ listar\_categorias() : List<Categoria>  
+ ver\_categoria(id : int) : Categoria  
+ criar\_categoria(c : Categoria) : void  
+ excluir\_categoria(id : int) : void  
+ atualizar\_categoria(c : Categoria) : void

<<static>>

### NProduto

- produtos : List<Produto>

+ listar\_produtos() : List<Produto>  
+ ver\_produto(id : int) : Produto  
+ cadastrar\_produto(p : Produto) : void  
+ excluir\_produto(id : int) : void  
+ atualizar\_produto(p : int) : void

## 5. Diagrama de classes da interface

