

Universidade Federal de Santa Catarina  
2024.1

Aluno: André Pinheiro Paes//(23205038)

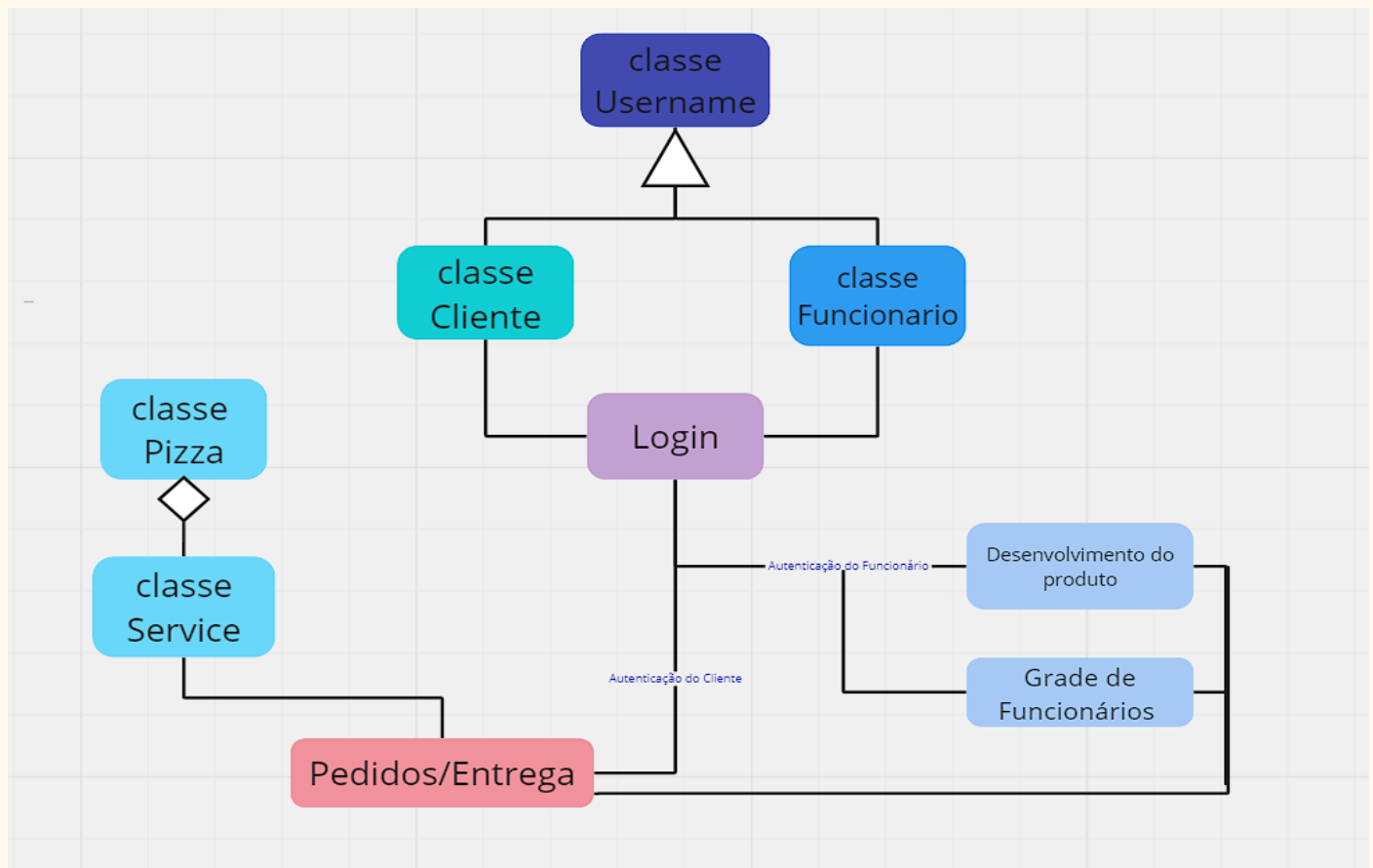
Professor: Alex Sandro Roschildt Pinto

Turma: INE5404-02208A

# Programação Orientada a Objetos II

## Prova I: Modelagem e implementação de um sistema de pizzeria

### Introdução: Modelagem UML



## **1. Classe Username:**

As classes Cliente e Funcionario herdam desta afim de compartilhar os atributos “nome” e “senha”.

Atributos:

nome: String que representa o nome de usuário.

senha: String que representa a senha associada ao nome de usuário, criando assim um login

`__init__(nome, senha)`: Método especial de inicialização (construtor) que define os atributos na criação de objetos.

## **2. Classe Cliente:**

Atributos:

clientes\_registrados: Dicionário usado para armazenar login de clientes registrados.

Métodos:

register\_clit(): Método para registrar um novo cliente no dicionário “clientes\_registrados”.

aut\_cliente(): Método para autenticar um cliente com base no nome de usuário e senha armazenados na lista.

## **3. Classe Funcionario:**

Atributos:

funcionarios\_registrados: Dicionário usado para armazenar funcionários registrados.

Métodos:

register\_funcionario(): Método para registrar um novo funcionário no dicionário “funcionarios\_registrados”.

autenticar\_funcionario(): Método para verificar um funcionário com base no nome de usuário e senha armazenados na lista.

#### **4. Classe Pizza:**

Atributos:

flavor: String que representa o sabor da pizza.

Price: Valor em ponto flutuante que representa o preço da pizza.

Métodos:

`__init__(flavor, Price)`: Método de inicialização (construtor) que define os atributos flavor e Price durante a criação de objetos.

#### **5. Classe Service:**

A classe Service é responsável por gerenciar e calcular os pedidos de pizza.

Atributos:

`Pizzas_disponiveis`: Lista de objetos “Pizza” que representa os sabores e preços das pizzas disponíveis.

Métodos:

`mostrar_Pizza_realizados()`: Método para exibir e enumerar os sabores e preços das pizzas.

`orderr_pizza(indice, quant)`: Método para registrar um pedido de pizza com base no índice (sabor escolhido) e quantidade definidos pelo cliente.

Herança:

A relação de herança (Cliente e Funcionario herdam de Username) indica que as subclasses “Cliente” e “Funcionario” compartilham os atributos e comportamentos da classe base “Username”.

Associação:

Os dicionários estáticos (`clientes_registrados` em Cliente e `funcionarios_registrados` em Funcionario) representam uma associação entre as classes e a estrutura de armazenamento dos dados dos clientes e funcionários registrados.

#### **Manual de Instrução:**

Utilização da linguagem python (versão: 3.11), desenvolvido na IDE “Pycharm”.

### **1.1 Login como Cliente:**

Selecione a opção 1 no menu principal, insira seu nome de usuário e senha. Se autenticado com sucesso, você poderá acessar o menu de pedidos de pizza. Caso não tenha cadastrado nenhum usuário, selecione a opção 3 para adicionar seu cadastro na lista.

### **1.2 Realização de Pedidos:**

Após realizar o login como cliente com sucesso, você poderá escolher opções de pizzas disponíveis, selecionando a opção 1 será listado todos os sabores disponíveis e seus respectivos preços.

Selecione a opção 2 para realizar seu pedido, escolha o sabor da pizza desejado (indicando o número correspondente) e a quantidade desejada. O sistema confirma o pedido e calcula o valor total.

### **2. Login como Funcionário:**

Selecione a opção 2 no menu principal, insira seu nome de usuário e senha. Se autenticado com sucesso, você poderá acessar seu cronograma de trabalho e atualizações do dia. Caso não tenha cadastrado nenhum usuário, selecione a opção 4 para adicionar seu cadastro na lista.

### **3. Encerramento do Sistema:**

Em qualquer momento, escolha a opção 3 no menu principal para sair do sistema se estiver na tela de pedidos, e opção 2 caso esteja na tela de funcionários

### **4. Observações Importantes:**

- Certifique-se de inserir corretamente seu nome de usuário e senha durante o login e registro.
- Os pedidos de pizza estão sujeitos à disponibilidade dos sabores listados.
- Para dúvidas ou problemas, entre em contato com nosso suporte técnico.

O Sistema de Pizzaria Vesúvio's foi projetado para conciliar uma experiência proveitosa tanto na realização de pedidos, quanto no gerenciamento de pessoas. Nossa preocupação é o seu bem-estar, aproveite a experiência e bom apetite! 🍕

```
#Classe Username serve de base para armazenar nome e senha

class Username:

    def __init__(self, nome, senha):

        self.nome = nome

        self.senha = senha

#Classe Cliente que herda os atributos de Username

class Cliente(Username):

    clientes_registrados = {}

    def register_clit(self):

        #Cria um login para o cliente e registra na lista
        "clientes_registrados"

        Cliente.clientes_registrados[self.nome] = self.senha

        print()

        print('MAMMA MIA, A NEW CUSTOMER !!!')

    def aut_cliente(self):

        #Verifica o cadastro do cliente
```

```
        if self.nome in Cliente.clientes_registrados and
Cliente.clientes_registrados[self.nome] == self.senha:

            return True

        else:

            return False


#Classe Funcionario que herda os atributos de Username
class Funcionario(Username):

    funcionarios_registrados = {}

    def register_funcionario(self):

        #Cria um login para o funcionario e registra na lista
"funcionarios_registrados"

        Funcionario.funcionarios_registrados[self.nome] = self.senha

        print()

        print('Thank you for being part of this family')

    def autenticar_funcionario(self):

        #Verifica o cadastro do funcionario

        if self.nome in Funcionario.funcionarios_registrados and
Funcionario.funcionarios_registrados[self.nome] == self.senha:

            return True

        else:
```

```
        return False
```

```
# Classe Pizza para definir a variavel sabores e preços das pizzas
```

```
class Pizza:
```

```
    def __init__(self, flavor, Price):
```

```
        self.flavor = flavor
```

```
        self.Price = Price
```

```
class Service:
```

```
    def __init__(self):
```

```
        # Lista de pizzas sabor/preço
```

```
        self.Pizzas_disponiveis = [
```

```
            Pizza('Margherita', 10.99),
```

```
            Pizza('Vegetarian', 11.49),
```

```
            Pizza('Hawaiian', 11.99),
```

```
            Pizza('Pepperoni', 12.99),
```

```
            Pizza('Buffalo Chicken', 12.99),
```

```
            Pizza('BBQ Chicken', 13.49),
```

```
            Pizza('Supreme', 14.99),
```

```
            Pizza('Meat Lovers', 15.99)
```

```
        ]
```

```
    def mostrar_Pizza_realizados(self):
```

```

        # Mostra as opções de pizzas

        print('=== Pizza >> flavors and price << ===')

        for i, Pizza in enumerate(self.Pizzas_disponiveis):

            print(f'{i+1}. {Pizza.flavor} -- price:
${Pizza.Price:.2f}')

    def orderr_pizza(self, indice, quant):

        # Registra o pedido da pizza selecionada e quantidade

        Pizza_escolhido = self.Pizzas_disponiveis[indice-1]

        valor_total = Pizza_escolhido.Price*quant

        print(f'You chose the Pizza >> {Pizza_escolhido.flavor} in
{quant} units, with a total value of ${valor_total:.2f}.')

        print('Your order has been completed')

        print()

        print('You made an offer i cant refuse.... thank you!')

# Instância da classe Service para gerenciar os pedidos
ordeer = Service()

def main():

    while True:

        print()

        print("=== Vesuvio's Pizza: (Login) ===")

        print("1. Login Client")

```



```
print("2. Login Employee")

print("3. Register New Client")

print("4. Register New Employee")

print()

choice = input("Choose the option: ")


if choice == '1':

    # Login do cliente

    nome = input("Username Client: ")

    senha = input("Password Client: ")

    cliente = Cliente(nome, senha)

    if cliente.aut_cliente():

        # Cliente autenticado, exibe menu de pedidos

        print()

        print("Successful!!! What's today's special?.")

        while True:

            print()

            print("==== Choose your Pizza =====")

            print('1. Menu')

            print('2. Make your order')

            print('3. Quit')
```

```
selec = input('Choose the option: ')

if selec == '1':

    #printa o menu

    ordeer.mostrar_Pizza_realizados()

    continue

elif selec == '2':

    #inputs de escolha do sabor e quantidade de
pizzas

    flav_Pizza = int(input('Which flavor:'))

    amount = int(input('How many pizzas?:'))

    ordeer.orderr_pizza(flav_Pizza,amount)

elif selec == '3':

    print('Leaving the system...')

    break

else:

    print("Login Error!!!.")

    break

elif choice == '2':
```

```
# Login do funcionário

nome = input("Username Employee: ")
senha = input("Password Employee: ")
funcionario = Funcionario(nome, senha)

if funcionario.autenticar_funcionario():
    # Funcionário autenticado, exibe as opções
    print()
    print("Have a great job!")
    while True:
        print()
        print("==== Choose your Service  =====")
        print('1. Menu for waiters')
        print('2. Rotation of the day')
        print('3. Quit')
        selecc = input('Choose the option: ')

        if selecc == '1':
            # printa o menu
            ordeer.mostrar_Pizza_realizados()
            continue

        if selecc == '2':
```

```

        #Lista de empregados

        print('===Employee Rotation===')

        print((nome)+'(new kid) == 08:00 --> 05:30
PM')

        print('Tony Soprano == 08:00 AM --> 10:35 PM')

        print('Christopher Moltisanti == 12:00 AM -->
11:40 PM')

        print('Adriana La Cerva == 07:00 AM --> 04:00
PM')

        print('Jennifer Melfi == 05:00 AM --> 08:20
PM')

        continue

    elif selecc == '3':

        print('Leaving the system...')

        break

    else:

        print("Login Error!!!.")

        break

elif choice == '3':

    #Cria login cliente

    nome = input("Username New Client: ")

    senha = input("Password New Client: ")

    cliente = Cliente(nome, senha)

```

```
        cliente.register_clit()

elif choice == '4':

    #Cria login funcionario

    nome = input("Username New Employee: ")

    senha = input("Password New Employee: ")

    funcionario = Funcionario(nome, senha)

    funcionario.register_funcionario()

if __name__ == "__main__":

    main()
```