

INTRODUÇÃO À PROGRAMAÇÃO DE DISPOSITIVOS MÓVEIS

Alunos: Gildo, Kadmyel e Wender



QUAIS TELAS VAI TER?

1. Login
2. Cadastro
3. Menu
4. Pedido
5. Finalizar pedido



Logo
Pizzaria

Login:

Senha:

Login

Cadastrar



Logo
Pizzaria

Nome:

Email:

Senha:

Cadastrar



Logo
Pizzaria

Nome:

Sobrenome:

Email:

Senha:

Cadastrar

Voltar

Logo
Pizzaria

Fazer Pedido

Sair

Foto
da
Pizza

Nome da Pizza-Tamanho

Ingredientes:.....
.....
.....

Preço: R\$ 00,00

Quantidade: 0

+

-

Foto
da
Pizza

Nome da Pizza-Tamanho

Ingredientes:.....
.....
.....

Preço: R\$ 00,00

Total

R\$ 0,00

Menu

Finalizar
Pedido

Endereço:

Telefone:

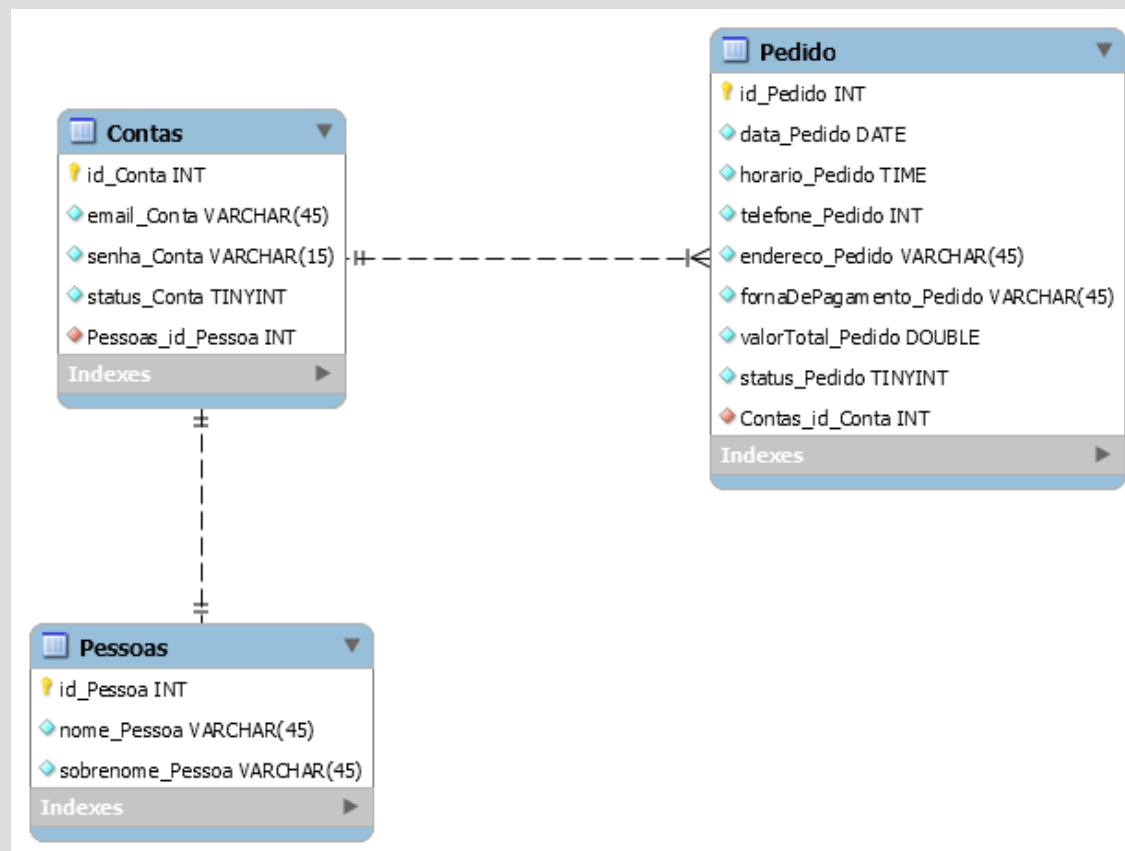
Forma de
pagmaneto:

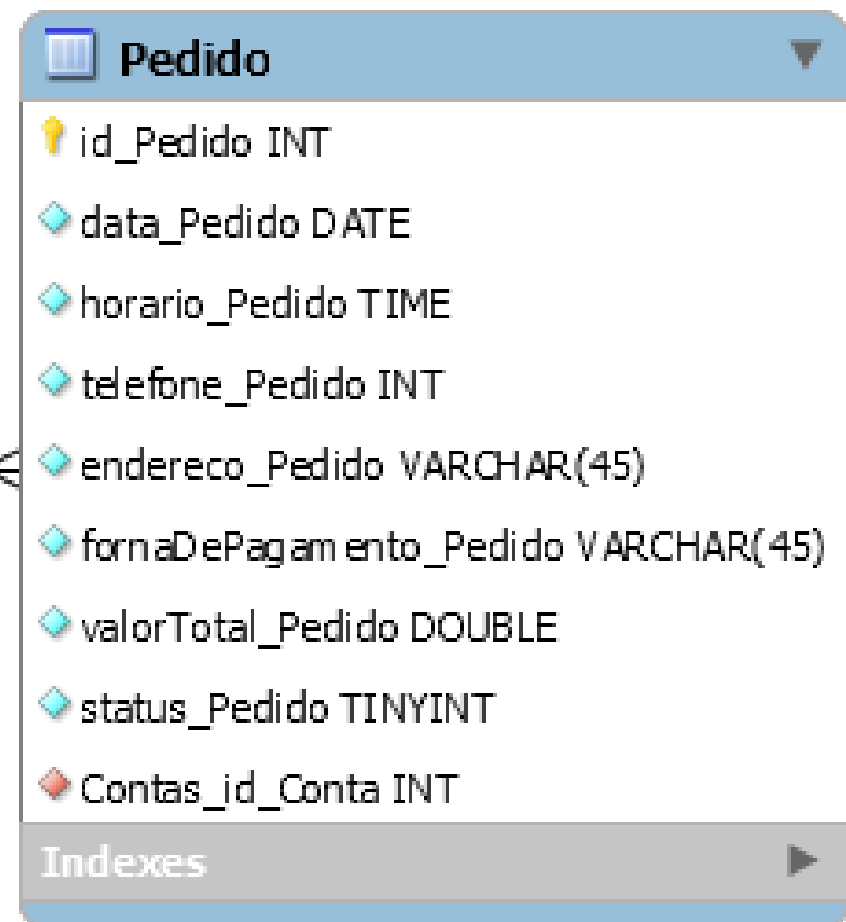
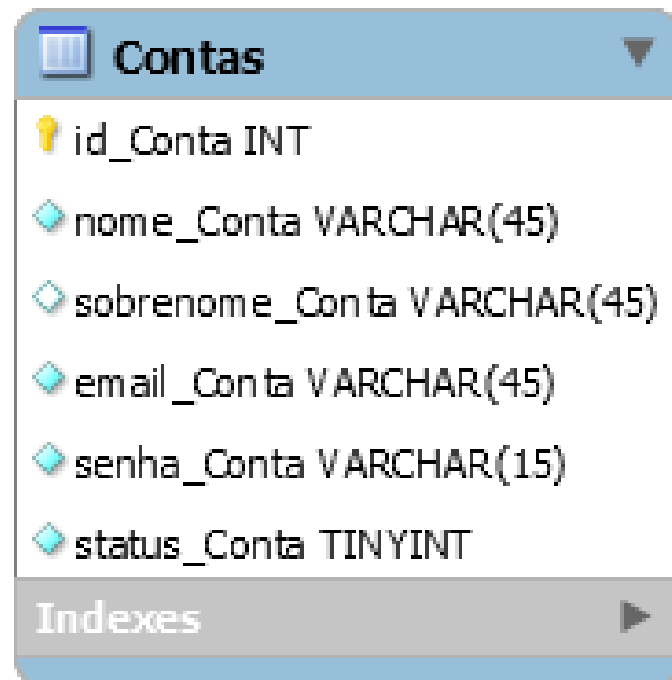
Menu

Pedido

Finalizar Pedido

MER





Contas
id_Conta INT
nome_Conta VARCHAR(45)
sobrenome_Conta VARCHAR(45)
email_Conta VARCHAR(45)
senha_Conta VARCHAR(15)
status_Conta TINYINT
Indexes

Pedido
id_Pedido INT
data_Pedido DATE
horario_Pedido TIME
telefone_Pedido INT
endereco_Pedido VARCHAR(45)
formaDePagamento_Pedido VARCHAR(45)
valorTotal_Pedido DOUBLE
status_Pedido TINYINT
Contas_id_Conta INT
pizza_Ads_Pedido INT
pizza_Baiana_Pedido INT
pizza_Bolonhesa_Pedido INT
pizza_Calabresa_Pedido INT
pizza_FrangoCom Catupiry_Pedido INT
pizza_Ituiutaba_Pedido INT
pizza_Mucarela_Pedido INT
pizza_QuatroQueijos_Pedido INT
pizza_TresQueijos_Pedido INT
pizza_Brocolis_Pedido INT
pizza_Champignon_Pedido INT
pizza_CincoQueijos_Pedido INT
pizza_Salaminho_Pedido INT
pizza_Vegetariana_Pedido INT
pizza_Acai_Pedido INT
pizza_Banana_Pedido INT
pizza_Brigadeiro_Pedido INT
pizza_Prestigio_Pedido INT
pizza_RomeuAndJulieta_Pedido INT
Indexes



TRECHOS DOS CÓDIGOS



```
<Login>:
  BoxLayout:
    orientation: 'vertical'
    padding: 0
    spacing: 0

    Image:
      source: "../imgs/pizzaLogo.png"
      allow_stretch: False
      size_hint_y: None
      height: 150

    GridLayout:
      cols: 2
      spacing: 0

      Label:
        # id:
        text: "E-mail"
      TextInput:
        id: log
        font_size: 30
        multiline: False # Apenas uma linha é permitida
        write_tab: False # Não pode apertar tap e dar varios espaços, não mais
      Label:
        # id:
        text: "Senha"
      TextInput:
        id: sen
        font_size: 30
        password: True
```

```
Button:
  text: 'Login'
  on_release: root.teste_integrarLogin()
  on_release: app.root.current = 'menu'
Button:
  text: 'Cadastrar'
  on_release: app.root.current = 'cadastrar'
```

```
class Login(Screen):  
    def teste_integrarLogin(self):  
        connection = bancoDeDados.create_server_connection('localhost', 'root', '')  
        email = self.ids.log.text  
        senha = self.ids.sen.text  
        bancoDeDados.procurar(connection, email, senha)
```

```
# Cria a conexão com o Banco de dados
def create_server_connection(host_name, user_name, user_password):
    connection = None
    try:
        connection = mysql.connect(
            host= host_name,
            user = user_name,
            password = user_password
        )
        print("MySQL Database connection successful")
    except Error as err:
        print('Erro ao conectar o MySQL', err)
    return connection
```

```
# Mostra a versão do Banco de dados
def version_database(connection):
    if connection.is_connected():
        versao = connection.get_server_info()
        print('Conectado ao MySQL Server versão ', versao)
        cursor = connection.cursor()
        cursor.execute('SELECT DATABASE();')
        resultado = cursor.fetchone()
        print('Você está conectado ao banco de dados: ', resultado)

# Destroi a conexão com o Banco de dados
def server_connection_destroy(connection):
    if (connection.is_connected()):
        cursor = connection.cursor()
        cursor.close()
        connection.close()
        connection = None
        print('Conexão MySQL está fechado')
    return connection
```

```
# Cria o banco de dados
def create_database(connection, query):
    cursor = connection.cursor()
    try:
        cursor.execute(query)
        print("Database created successfully")
    except Error as err:
        print(f"Error: '{err}'")
```

```
# Cria a tabela Contas
def create_table_Contas(connection):
    cursor = connection.cursor()
    cursor.execute('USE pizzariaKivy')
    cursor.execute('CREATE TABLE IF NOT EXISTS Contas'
        '(id_Conta INT NOT NULL AUTO_INCREMENT PRIMARY KEY, '
        'nome_Conta VARCHAR(255) NOT NULL, '
        'sobrenome_Conta VARCHAR(255) NOT NULL, '
        'email_Conta VARCHAR(255) NOT NULL, '
        'senha_Conta VARCHAR(255) NOT NULL, '
        'status_Conta TINYINT(1) NOT NULL DEFAULT 0, '
        'CONSTRAINT unique_email_Conta UNIQUE (email_Conta))'
    )

# Inserir dado na tabela Contas
def insert_table_Contas(connection, nome, sobrenome, email, senha):
    cursor = connection.cursor()
    cursor.execute('USE pizzariaKivy')
    cursor.execute('INSERT INTO Contas'
        '(nome_Conta, sobrenome_Conta, email_Conta, senha_Conta) VALUES'
        '(%s, %s, %s, %s);', (nome, sobrenome, email, senha)
    )

    print("Inserted", cursor.rowcount, "row(s) of data.")
    connection.commit()
    cursor.close()
    connection.close()
    print("Done.")
```

```

def procurar(connection, email, senha):
    cursor = connection.cursor()
    cursor.execute('USE pizzariaKivy')
    cursor.execute('SELECT senha_Conta FROM Contas WHERE email_Conta = "{}"'.format(email))
    senha_bd = cursor.fetchall()
    print(senha_bd[0][0])
    print(senha)
    cursor.execute('SELECT status_Conta FROM Contas WHERE email_Conta = "{}"'.format(email))
    status_Conta = cursor.fetchall()
    print(status_Conta[0][0])

    # try:
    #     cursor = connection.cursor()
    #     cursor.execute('USE pizzariaKivy')
    #     cursor.execute('SELECT senha_Conta FROM Contas WHERE email_Conta = "{}"'.format(email))
    #     senha_bd = cursor.fetchall()
    # except:
    #     print("Erro ao validar o login")

    if senha_bd[0][0] == senha and status_Conta[0][0] == 0:
        print("Login realizado com sucesso!!!")
    else:
        print("Falha ao fazer o Login")
        sys.exit()

```



```
connection = create_server_connection('localhost', 'root', '')
create_database_query = "CREATE DATABASE IF NOT EXISTS pizzariaKivy;"
create_database(connection, create_database_query)
version_database(connection)
create_table_Contas(connection)
server_connection_destroy(connection)
```



```
class Pedido(Screen):
    # PIZZAS TRADICIONAIS

    ## Funções da Pizza Ads
    def pizza_Ads(self): # Ela será chamada pelas funções incrementar e decrementar
        valorUnitarioDaPizza_Ads = (float(self.ids.valor_Ads.text)) # Pega o valor da pizza e passe ele de str para float
        quantidadeDePizza_Ads = (float(self.ids.quantidade_Ads.text)) # Pega o valor do contador da pizza
        valorTotalDaPizza_Ads = quantidadeDePizza_Ads * valorUnitarioDaPizza_Ads
        # Faz o calculo do total da pizza multiplicando o valor e a quantidade
        return valorTotalDaPizza_Ads # Retorna o valor total da pizza
```

```

def incrementar_Ads(self): # Será chamada sempre que for apertado o botão de + da pizza
    if (int(self.ids.quantidade_Ads.text)) < quantidadeMaximaDePizzaDeUmSabor: # Enquanto a quantidade for menor que 10 vai acontecer...
        self.ids.quantidade_Ads.text = str(int(self.ids.quantidade_Ads.text) + 1)
        # transforma em int soma 1 e transforma o novo valor em str
        self.pizza_Ads() # Chama a função para atualizar o valor total do pedido

    ### Pizzas Tradicionais
    pizza_Ads = self.pizza_Ads() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Baiana = self.pizza_Baiana() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Bolonhesa = self.pizza_Bolonhesa() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Calabresa = self.pizza_Calabresa() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_FrangoComCatupiry = self.pizza_FrangoComCatupiry() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Ituiutaba = self.pizza_Ituiutaba() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Mucarela = self.pizza_Mucarela() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_QuatroQueijos = self.pizza_QuatroQueijos() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_TresQueijos = self.pizza_TresQueijos() # Atribui a variavel a função que retorna o valor total da pizza

    ### Pizzas Especiais
    pizza_Brocolis = self.pizza_Brocolis() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Champignon = self.pizza_Champignon() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_CincoQueijos = self.pizza_CincoQueijos() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Salaminho = self.pizza_Salaminho() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Vegetariana = self.pizza_Vegetariana() # Atribui a variavel a função que retorna o valor total da pizza

    ### Pizzas Doces
    pizza_Acai = self.pizza_Acai() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Banana = self.pizza_Banana() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Brigadeiro = self.pizza_Brigadeiro() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Prestigio = self.pizza_Prestigio() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_RomeuAndJulieta = self.pizza_RomeuAndJulieta() # Atribui a variavel a função que retorna o valor total da pizza

    self.valorTotalDoPedido(pizza_Ads, pizza_Baiana, pizza_Bolonhesa, pizza_Calabresa, pizza_FrangoComCatupiry, pizza_Ituiutaba, pizza

```

```
def decrementar_Ads(self):
    if (int(self.ids.quantidade_Ads.text)) > 0:
        self.ids.quantidade_Ads.text = str(int(self.ids.quantidade_Ads.text) - 1)

    ### Pizzas Tradicionais
    pizza_Ads = self.pizza_Ads() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Baiana = self.pizza_Baiana() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Bolonhesa = self.pizza_Bolonhesa() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Calabresa = self.pizza_Calabresa() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_FrangoComCatupiry = self.pizza_FrangoComCatupiry() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Ituiutaba = self.pizza_Ituiutaba() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Mucarela = self.pizza_Mucarela() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_QuatroQueijos = self.pizza_QuatroQueijos() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_TresQueijos = self.pizza_TresQueijos() # Atribui a variavel a função que retorna o valor total da pizza

    ### Pizzas Especiais
    pizza_Brocolis = self.pizza_Brocolis() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Champignon = self.pizza_Champignon() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_CincoQueijos = self.pizza_CincoQueijos() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Salaminho = self.pizza_Salaminho() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Vegetariana = self.pizza_Vegetariana() # Atribui a variavel a função que retorna o valor total da pizza

    ### Pizzas Doces
    pizza_Acai = self.pizza_Acai() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Banana = self.pizza_Banana() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Brigadeiro = self.pizza_Brigadeiro() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_Prestigio = self.pizza_Prestigio() # Atribui a variavel a função que retorna o valor total da pizza
    pizza_RomeuAndJulieta = self.pizza_RomeuAndJulieta() # Atribui a variavel a função que retorna o valor total da pizza

    self.valorTotalDoPedido(pizza_Ads, pizza_Baiana, pizza_Bolonhesa, pizza_Calabresa, pizza_FrangoComCatupiry, pizza_Ituiutaba, pizz
```

```
## Função que faz a contabilidade do pedido
def valorTotalDoPedido(self, pizza_Ads, pizza_Baiana, pizza_Bolonhesa, pizza_Calabresa, pizza_FrangoComCatupiry, pizza_Ituiutaba, pizza_M
    valorTotalDoPedido = pizza_Ads + pizza_Baiana + pizza_Bolonhesa + pizza_Calabresa + pizza_FrangoComCatupiry + pizza_Ituiutaba + pizza
    valorTotalDoPedidoFormatado = "{:.2f}".format(valorTotalDoPedido) # Deixa o valor total do pedido com apenas duas casas decimais
    self.ids.valorTotalDoPedido.text = str(valorTotalDoPedidoFormatado) # Mostra o valor total do pedido na tela

    arquivo = open('C:/Users/noteb/Desktop/Pizzaria_Kivy/valorTotalDoPedido.txt', 'w')
    arquivo.write(valorTotalDoPedidoFormatado)
    arquivo.close()
```



```
class Finalizar_Pedido(Screen):
    def spinner_Clicado(self, value):
        self.ids.forma_de_pagamento.text = value
        # pass

    def atualizarValorDoPedido(self):
        data = datetime.today().strftime('%Y-%m-%d')
        hora = datetime.today().strftime('%H:%M:%S')

        print(data)
        print(type(data))
        print(hora)
        print(type(hora))

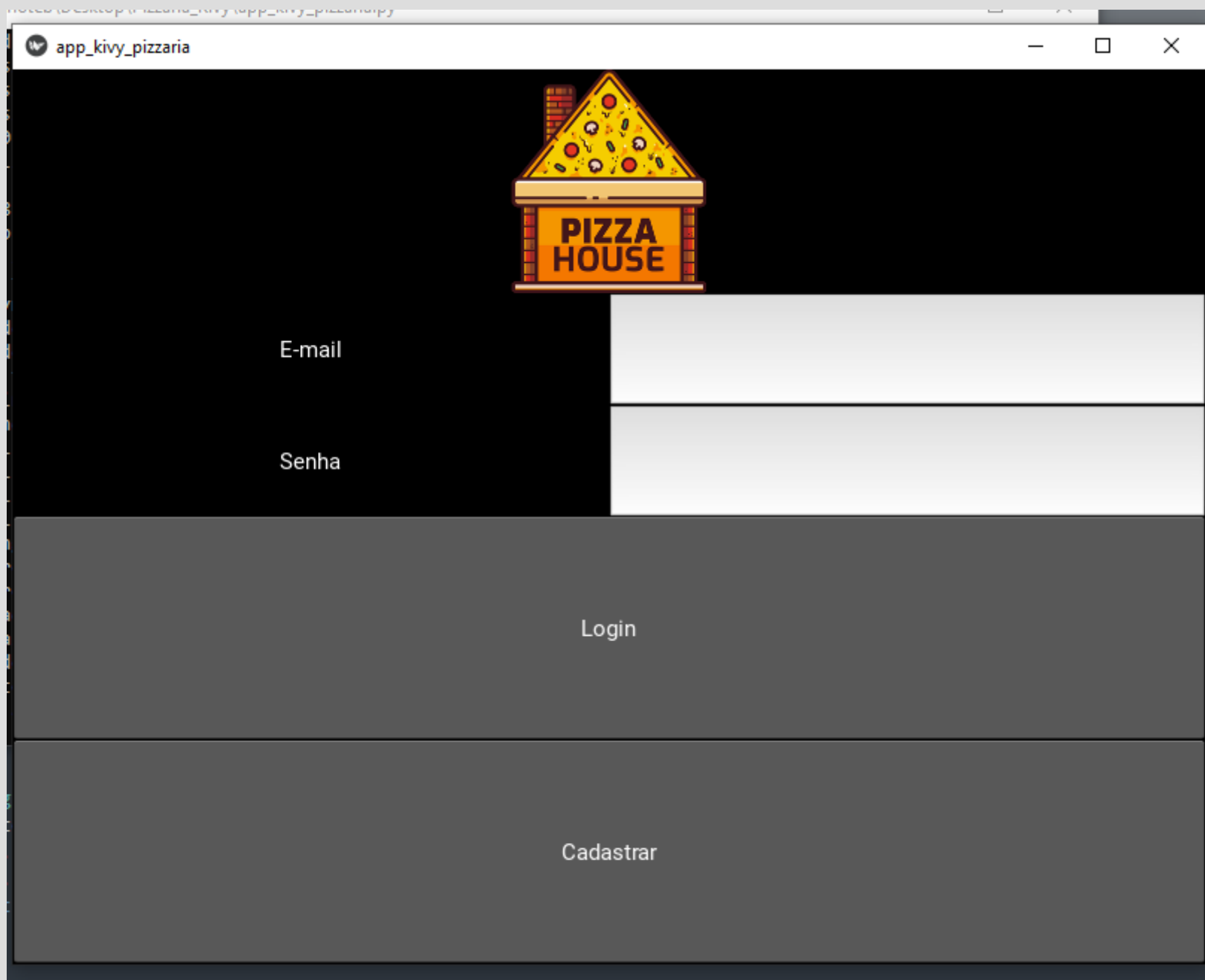
        arquivo = open('C:/Users/noteb/Desktop/Pizzaria_Kivy/valorTotalDoPedido.txt', 'r')
        print(arquivo.readline())
        arquivo.close()

class app_kivy_pizzaria(App):
    def build(self):
        return Gerente_das_telas()

    def notificando(self):
        return plyer.notification.notify(title='app_kivy_pizzaria', message="O app_kivy_pizzaria está em execução")

if __name__ == '__main__':
    app_kivy_pizzaria().notificando()
    app_kivy_pizzaria().run()
```

COMO ESTÃO AS TELAS





Nome

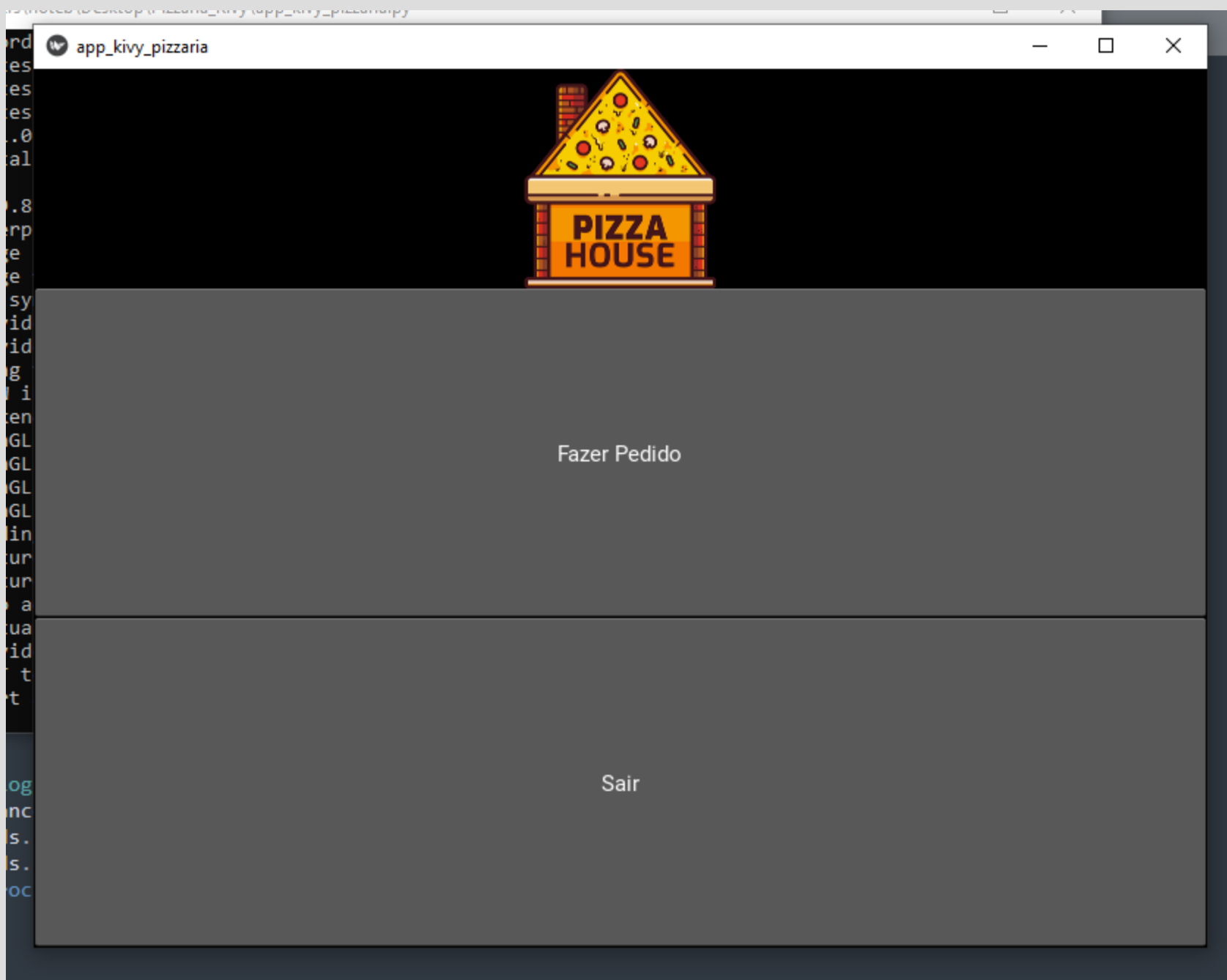
Sobrenome

Email

Senha

Cadastrar

Voltar



PIZZAS TRADICIONAIS

Ads

Muçarela, Presunto, Calabresa, Rodelas de Tomate, Cebola, Pimentão, Milho e Azeitonas.

Preço R\$ 1.50

Quantidade: 0

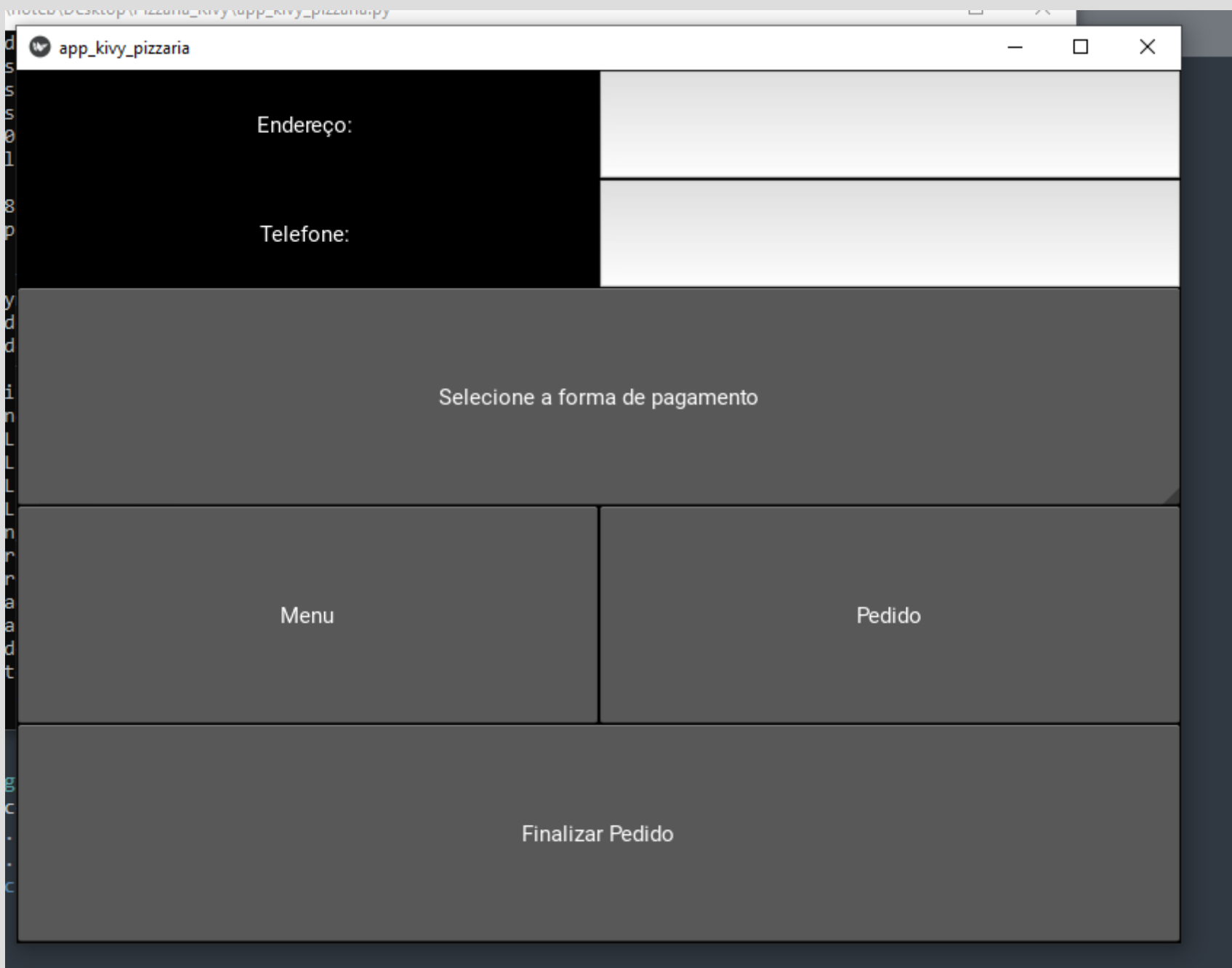
+

-

Total R\$ 0.00

Menu

Finalizar pedido



IDEIAS PARA O FUTURO

- Pop-up de aviso
- Salva a senha em Hash no banco de dados
- Tratamento de caracteres nos campos
- Definir horário para começar e parar de receber pedidos



ALGUMA DUVIDA?



“isso é tudo
pessoal!”

-loading-