

DOCUMENTAÇÃO DO CÓDIGO DO TRABALHO DE DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES  
EM PYTHON

# ESTOQUE DE PRODUTOS

Dupla: Júlia Vassimon da Silva e Daniele de Carvalho Vieira  
Machado.

Neste trabalho estaremos usando a linguagem de programação Python com a biblioteca Tkinter na utilização de interface e para trabalhar com Banco de Dados foi usado MySQL com a biblioteca mysql.connector.

## Conexão com o Banco de Dados:

Primeiro foi preciso conectar o banco de dados com Python.

```
class AppBd:
    def __init__(self):
        self.conn = mysql.connector.connect(
            host='127.0.0.1',
            user='root',
            password='35020349',
            database='produtos'
        )
        self.cursor = self.conn.cursor()
```

Nesse bloco de código tem as informações para a conexão, o “cursor” é criado para interagir com o banco de dados.

## Operações no Banco de Dados:

```
def select_data(self):
    self.cursor.execute("SELECT * FROM produto")
    return self.cursor.fetchall()

def insert_product(self, codigo, nome, preco):
    def update_data(self, codigo, nome, preco):
    def delete_data(self, codigo):
```

- select\_data: Executa uma consulta SQL para selecionar todos os registros da tabela "produto" e retorna os resultados.
- insert\_product: Insere um novo produto na tabela "produto" com base nos parâmetros fornecidos (código, nome, preço).

- `update_data`: Atualiza os dados de um produto na tabela "produto" com base nos parâmetros fornecidos (código, nome, preço).
- `delete_data`: Exclui um produto da tabela "produto" com base no código fornecido.

## Interface Gráfica:

```
class Bd:
    def __init__(self, win):
```

- `Bd` é uma classe que lida com a interface gráfica da aplicação.
- Inicializa a interface com rótulos, entradas de texto, botões e uma Treeview para exibir a lista de produtos.
- `treeProdutos` é um widget Treeview usado para exibir dados em formato de tabela.
- Os botões (Cadastrar, Atualizar, Excluir, Limpar, Calcular Acréscimo) são configurados com funções associadas.

```
def apresentar_itens_selecionados(self, event):
def carregar_dados_iniciais(self):
def ler_campos(self):
def cadastrar_produto(self):
def atualizar_produto(self):
def excluir_produto(self):
def limpar_tela(self):
def calcular_acrescimo(self):
```

- Essas funções manipulam a interface gráfica com base nas interações do usuário.
- `apresentar_itens_selecionados` é chamada quando um item na Treeview é selecionado e exibe os detalhes desse item nas entradas de texto.
- `carregar_dados_iniciais` carrega os dados iniciais da base de dados e os exibe na Treeview.
- As funções `f_cadastrar_produto`, `f_atualizar_produto`, e `f_excluir_produto` são chamadas pelos botões correspondentes e interagem com o banco de dados.
- `f_limpar_tela` limpa os campos de entrada de texto.
- `calcular_acrescimo` calcula um acréscimo de 10% no preço do produto selecionado.

E por último é a execução da aplicação:

```
if __name__ == "__main__":  
    root = tk.Tk()  
    app = Bd(root)  
    root.title("Estoque de produtos")  
    root.geometry("700x550")  
    root.mainloop()
```

- O código verifica se o script está sendo executado como um programa principal.
- Cria uma instância da classe Bd (interface gráfica) e inicia a aplicação Tkinter com root.mainloop().