

# Relatório Trabalho FBD Final

Equipe:

Giovana Souza de Araujo

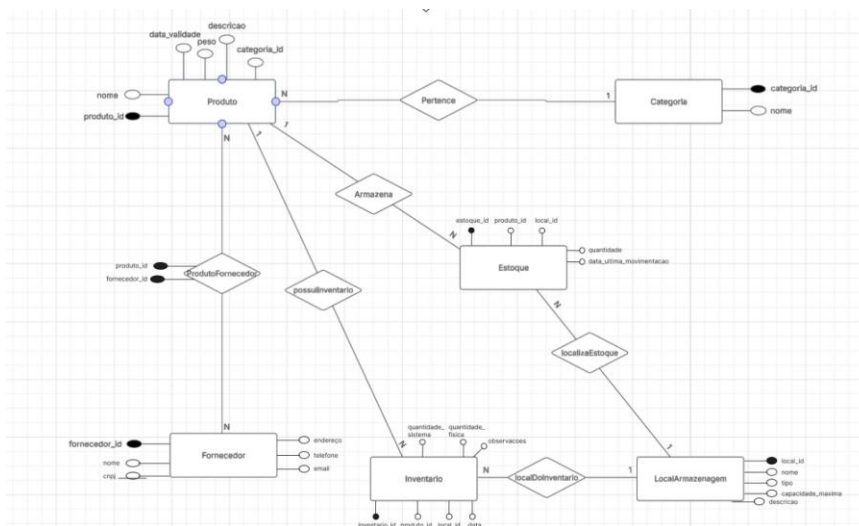
Juvenal da Costa Lavres da Conceição

Gabriel Gomes Marques

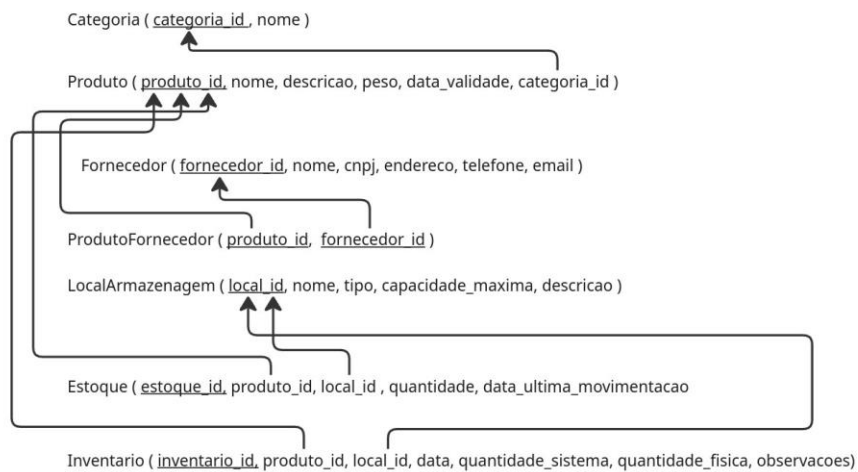
Rafael de Abreu Barbosa

Bryan Adrian Souza Roseno

## DER



## MER



# Inserções

## Inserção de Produto

```
# Inserir produto
cur.execute("""
    INSERT INTO Produto (produto_id, nome, descricao, peso, data_validade, categoria_id)
    VALUES (%s, %s, %s, %s, %s, %s)
""", tuple(entrys_produto[c].get() for c in campos_produto))

# Inserir associação com fornecedor
cur.execute("""
    INSERT INTO ProdutoFornecedor (produto_id, fornecedor_id)
    VALUES (%s, %s)
""", [
    entrys_produto["produto_id"].get(),
    entry_fornecedor_id.get()
])
```

Aqui é o código da função que insere novos produtos e ele também insere um novo ProdutoFornecedor para o novo produto.

produto_id	nome	descricao	peso	data_validade	categoria_id
102	Notebook Dell	Notebook 15 p			2
103	Detergente	Detergente neu			3
104	Refrigerante Cola	Garrafa 2L	2.00	2025-11-15	4
105	Caderno 100 folhas	Capa dura	0.40	None	5
106	...	...	...	...	...

Inserção do Produto Macarrão 200g

## Inserção de Fornecedor

```
con = conectar()
cur = con.cursor()
cur.execute("""
    INSERT INTO Fornecedor (fornecedor_id, nome, cnpj, endereco, telefone, email)
    VALUES (%s, %s, %s, %s, %s, %s)
""", tuple(entradas_for[c].get() for c in campos_for))
con.commit()
cur.close()
con.close()
listar_fornecedores()
messagebox.showinfo("Sucesso", "Fornecedor inserido com sucesso.")
except Exception as e:
    messagebox.showerror("Erro", str(e))
```

Aqui é o código da função que insere novos Fornecedores.

The screenshot displays a software application window titled "Sistema de Gestão de Armazém". The interface includes a menu bar with options: Produto, Fornecedor, Estoque, Categoria, LocalArmazenagem, Inventário, and Consultas Avançadas. Below the menu, there are three buttons: "Inserir Fornecedor", "Atualizar Fornecedor", and "Remover Fornecedor". A search bar is located below these buttons, with labels "Buscar por ID:" and "Buscar por Nome:". The main area contains a table with columns: fornecedor\_id, nome, cnpj, endereco, telefone, and email. The table lists 8 suppliers. A dialog box titled "Sucesso" is open in the center, displaying the message "Fornecedor inserido com sucesso." and an "OK" button. At the bottom of the window, there is a button labeled "Listar Fornecedores" and a Windows activation notice: "Ativar o Windows. Acesse Configurações para ativar o Windows."

fornecedor_id	nome	cnpj	endereco	telefone	email
1	Fornecedor Alimentos LTDA	11.111.111/0001-11		8599001122	alimentos@forn.com
2	Tech Eletrônicos	22.222.222/0001-22		8599002233	eletronicos@tech.com
3	Limpa Tudo S.A.	33.333.333/0001-33		8599003344	contato@limpatudo.com
4	Bebidas Brasil	44.444.444/0001-44		8599004455	bebidas@brasil.com
5	Escola Fácil	55.555.555/0001-55		8599005566	vendas@escolafacil.com
6	Delicias do Sul	66.666.666/0001-66	Rua F, 600	8599006677	contato@deliciasdosul.com
7	Electronic World	77.777.777/0001-77	Av. G, 700	8599007788	suporte@eletronicworld.com
8	Fornecedora Eletronicos	88.888.888/0001-88	Rua F, 888	8899008899	suporte@eletronicFor.com

Inserção do Fornecedor "Fornecedora Eletrônicos".

## Inserção de Inventario

```
on = conectar()
ur = con.cursor()
ur.execute("""
    INSERT INTO Inventario (inventario_id, produto_id, local_id, data, quantidade_sistema, quantidade_fisica, observacoes
    VALUES (%s, %s, %s, CURRENT_DATE, %s, %s, %s)
""", [
    entrys_inventario["inventario_id"].get(),
    entrys_inventario["produto_id"].get(),
    entrys_inventario["local_id"].get(),
    entrys_inventario["quantidade_sistema"].get(),
    entrys_inventario["quantidade_fisica"].get(),
    entrys_inventario["observacoes"].get()
])
on.commit()
ur.close()
con.close()
```

Aqui é o código da função que insere novos Inventarios.

The screenshot shows a web application titled "Sistema de Gestão de Armazém". The "Inventário" tab is active, displaying a form for adding new inventory items. The form fields are: inventario\_id (15), produto\_id (115), local\_id (5), quantidade\_sistema (12), quantidade\_fisica (12), and observacoes (OK). Below the form are buttons for "Inserir Inventário", "Atualizar Inventário", and "Remover Inventário". A search bar at the bottom allows searching by ID or local ID. A table below the search bar lists 15 inventory records with columns: inventario\_id, produto\_id, local\_id, data, quantidade\_sistema, quantidade\_fisica, and observacoes. A success message dialog box is overlaid on the table, stating "Sucesso" and "Inventário inserido com sucesso." with an "OK" button. A "Listar Inventários" button is at the bottom of the table.

	inventario_id	produto_id	local_id	data	quantidade_sistema	quantidade_fisica	observacoes
1	101	1	1		00	48	Conferir quantidade real e
2	102	2	2		0	10	Estoque OK
3	103	3	3		0	80	Sem divergência
4	104	5	5		0	19	1 unidade danificada
5	105	2	2		0	62	Erro de contagem anterior
6	106	1	1	2025-07-25	40	40	OK
7	107	1	1	2025-07-25	60	59	1 unidade faltando
8	108	5	5	2025-07-25	25	25	OK
9	109	2	2	2025-07-25	15	15	OK
10	110	4	4	2025-07-25	8	8	OK
11	111	3	3	2025-07-25	70	70	OK
12	112	3	3	2025-07-25	90	89	1 unidade faltando
13	113	6	6	2025-07-25	30	30	OK
14	114	6	6	2025-07-25	45	46	1 unidade a mais
15	115	5	5	2025-07-29	12	12	OK

Inserção do Inventario do Produto 115 no Local 5.

## Remoções

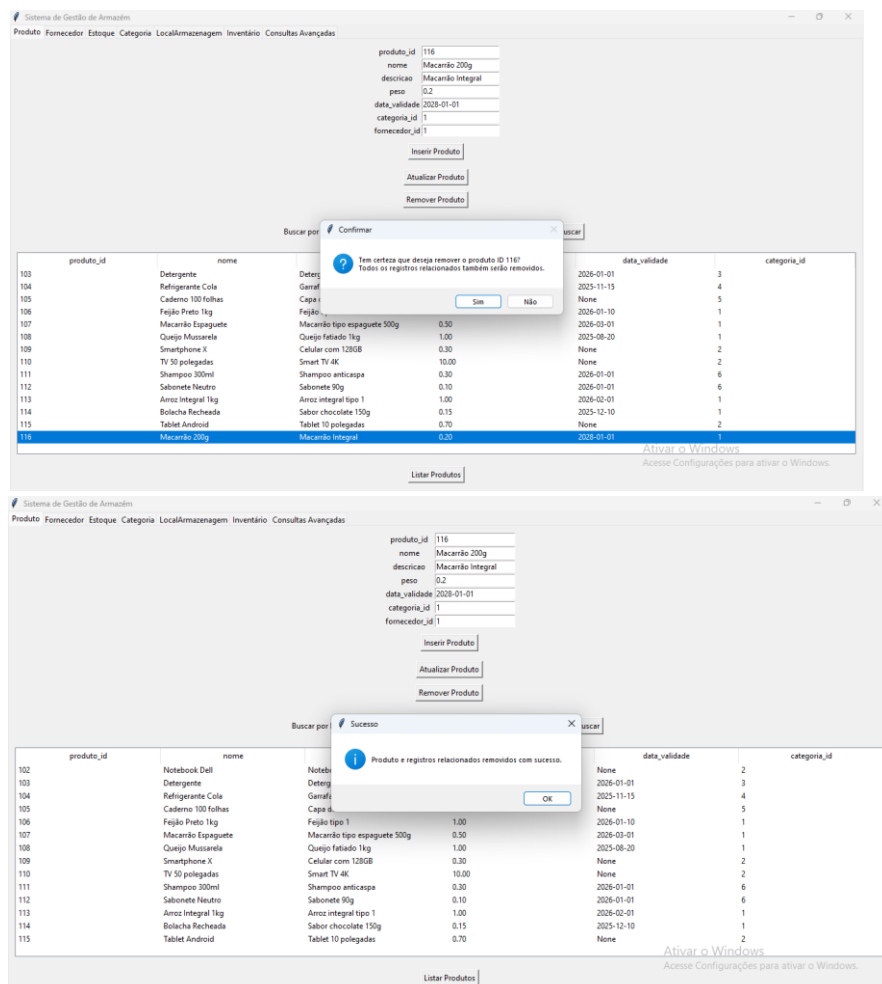
### Remoção de Produto

```
try:
    con = conectar()
    cur = con.cursor()

    # Agora pode remover o produto
    cur.execute("DELETE FROM Produto WHERE produto_id = %s", (produto_id,))

    con.commit()
    cur.close()
    con.close()
    listar_produtos()
    messagebox.showinfo("Sucesso", "Produto e registros relacionados removidos com sucesso.")
except Exception as e:
    messagebox.showerror("Erro", str(e))
```

Código de Remoção do Produto, além disso ao remover um produto também é removido todos os itens de outras tabelas que envolvam esse Produto.

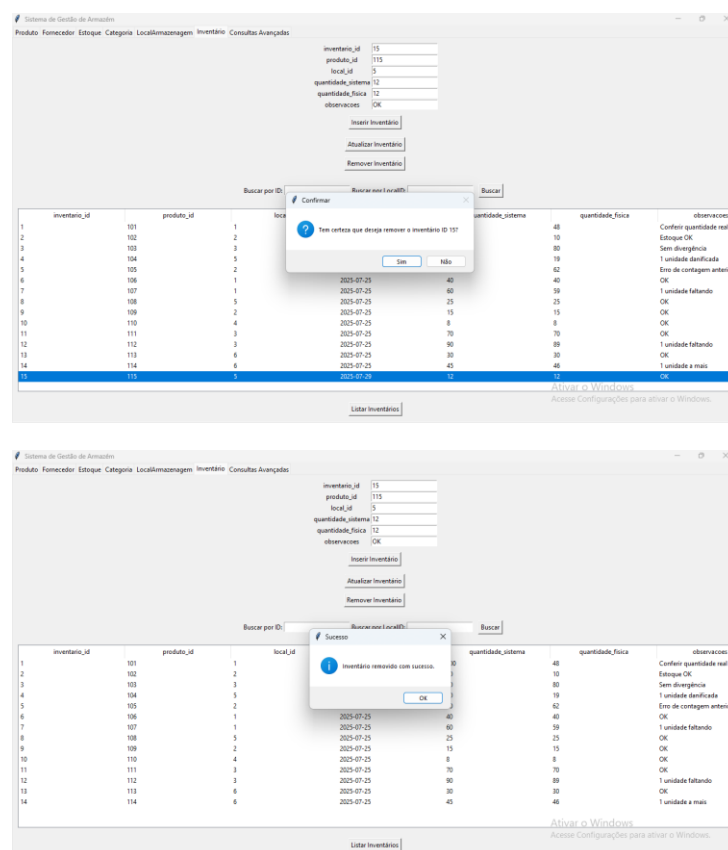


Pedido de Confirmação da remoção e a remoção bem-sucedida.

## Remoção de Inventário

```
try:
    con = conectar()
    cur = con.cursor()
    cur.execute("DELETE FROM Inventario WHERE inventario_id = %s", (inventario_id,))
    con.commit()
    cur.close()
    con.close()
    listar_inventarios()
    messagebox.showinfo("Sucesso", "Inventário removido com sucesso.")
except Exception as e:
    messagebox.showerror("Erro", str(e))
```

## Código de Remoção do Inventário.

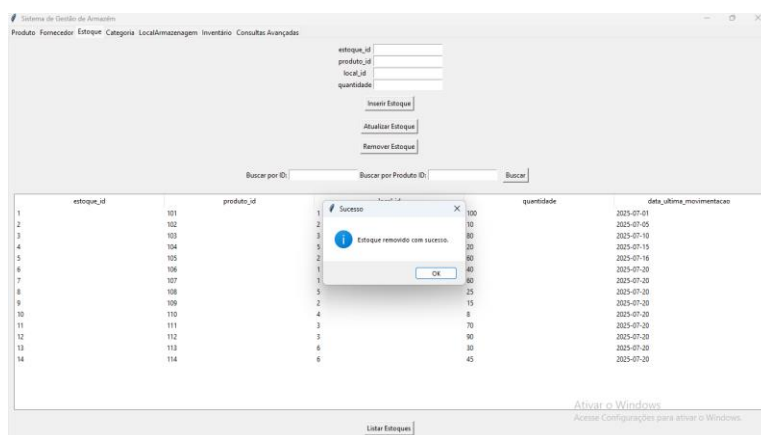
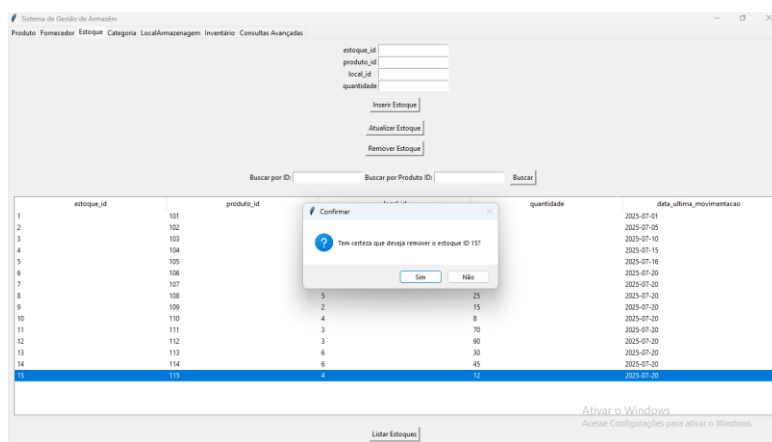


Pedido de Confirmação da remoção e a remoção bem-sucedida.

## Remoção Estoque

```
try:
    con = conectar()
    cur = con.cursor()
    cur.execute("DELETE FROM Estoque WHERE estoque_id = %s", (estoque_id,))
    con.commit()
    cur.close()
    con.close()
    listar_estoques()
    messagebox.showinfo("Sucesso", "Estoque removido com sucesso.")
except Exception as e:
    messagebox.showerror("Erro", str(e))
```

## Código de Remoção do Estoque.



## Pedido de Confirmação da remoção e a remoção bem-sucedida.

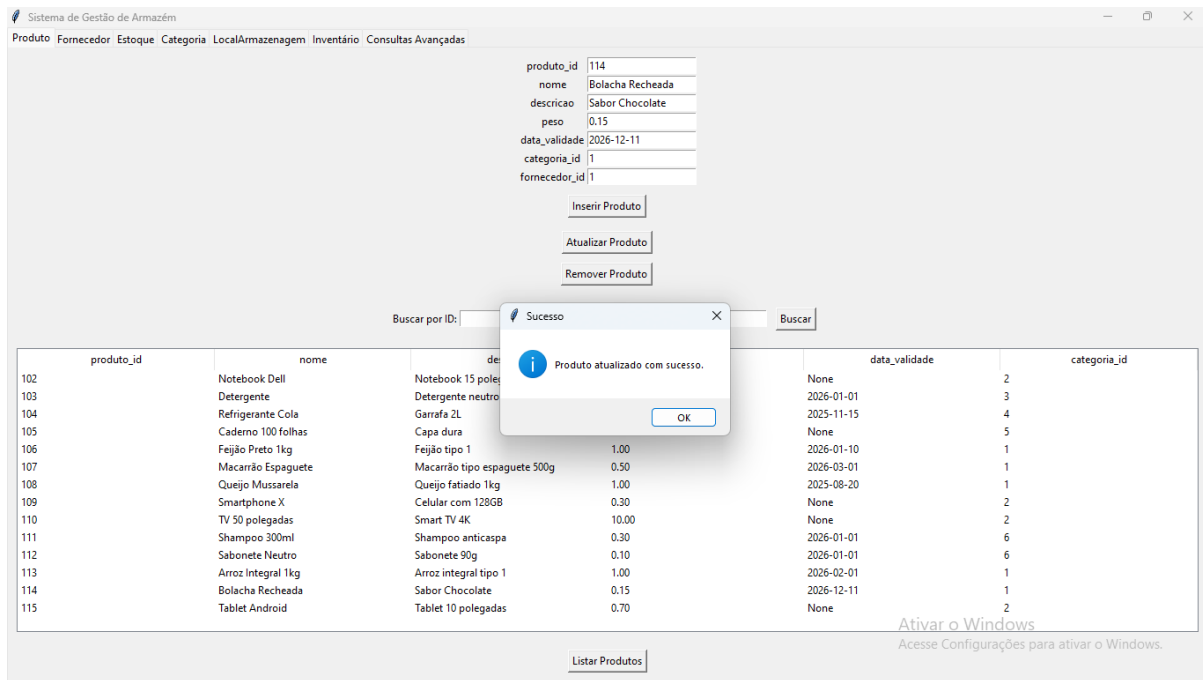
## Atualização

### Atualização de Produto

```
def atualizar_produto():
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("""
            UPDATE Produto
            SET nome = %s,
                descricao = %s,
                peso = %s,
                data_validade = %s,
                categoria_id = %s
            WHERE produto_id = %s
        """, (
            entrys_produto["nome"].get(),
            entrys_produto["descricao"].get(),
            entrys_produto["peso"].get(),
            entrys_produto["data_validade"].get(),
            entrys_produto["categoria_id"].get(),
            entrys_produto["produto_id"].get()
        ))
```

Função que atualiza um Produto.



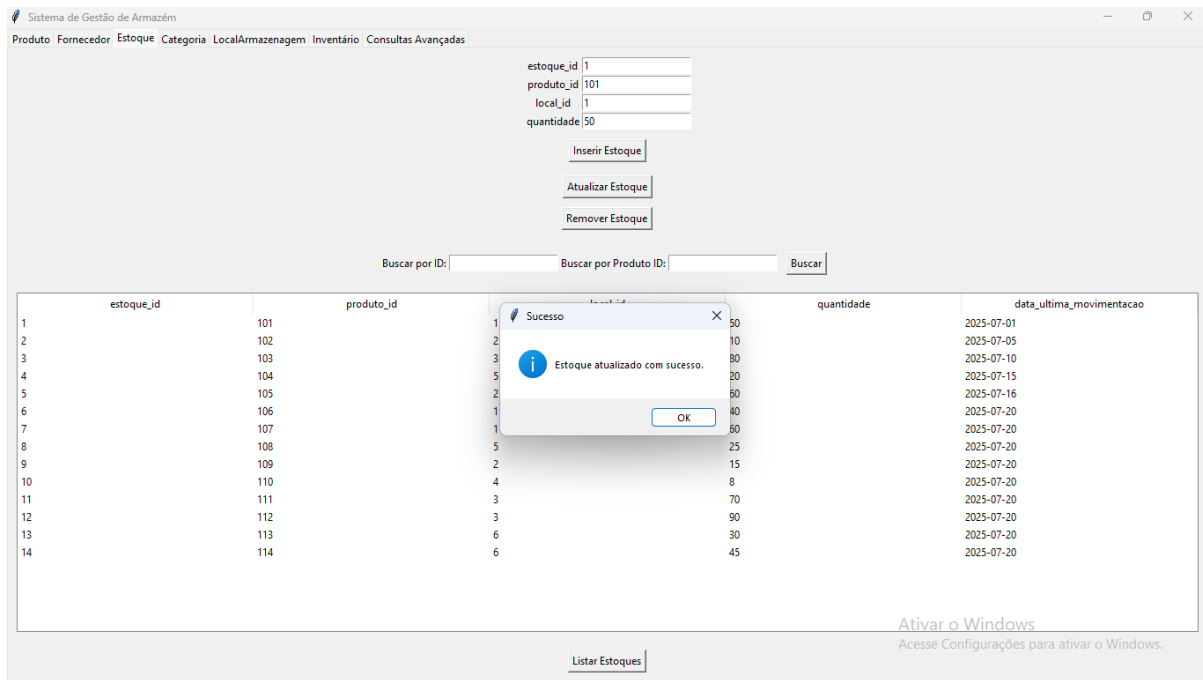


Produto 114 teve a data de validade atualizada

## Atualização de Estoque

```
def atualizar_estoque():
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("""
            UPDATE Estoque
            SET produto_id = %s,
              local_id = %s,
              quantidade = %s
            WHERE estoque_id = %s
        """, (
            entrys_estoque["produto_id"].get(),
            entrys_estoque["local_id"].get(),
            entrys_estoque["quantidade"].get(),
            entrys_estoque["estoque_id"].get()
        ))
        con.commit()
        cur.close()
        con.close()
```

Função de atualização para estoque, além disso através de uma trigger também atualiza o campo "quantidade\_sistema" e adiciona uma observação no "observacoes" do Inventário

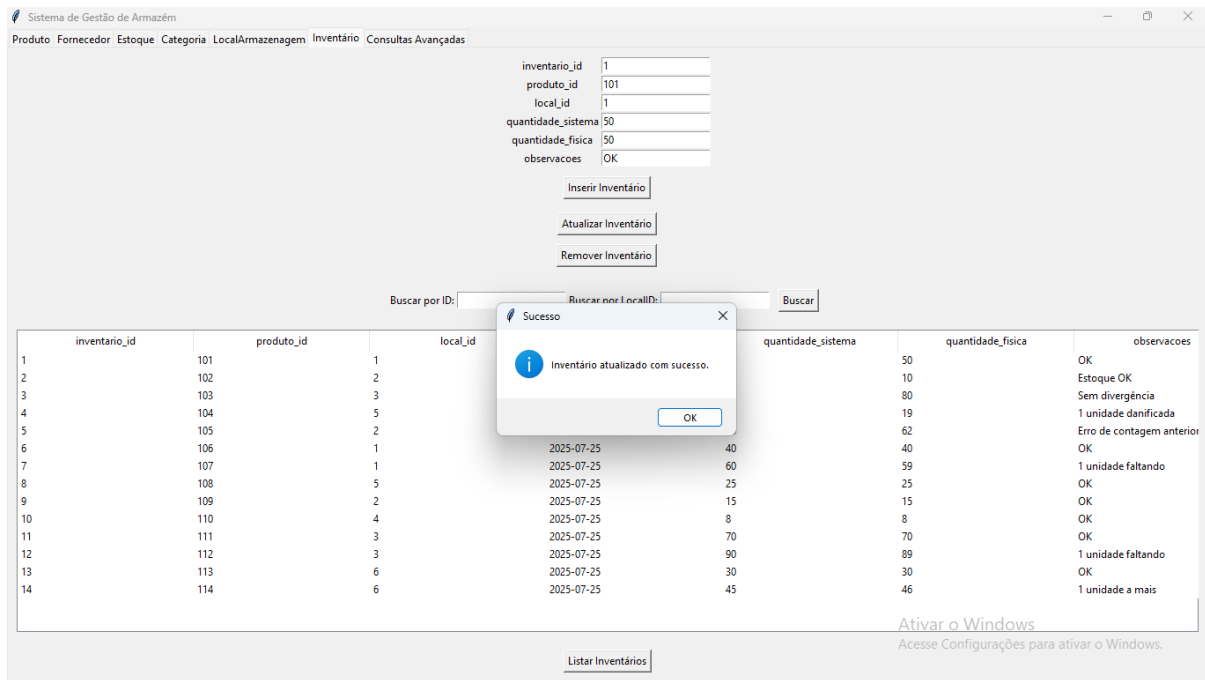


Quantidade do estoque 1 atualizada de 100 para 50.

## Atualização de Inventário

```
def atualizar_inventario():
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("""
            UPDATE Inventario
            SET produto_id = %s,
              local_id = %s,
              quantidade_sistema = %s,
              quantidade_fisica = %s,
              observacoes = %s
            WHERE inventario_id = %s
        """, (
            entrys_inventario["produto_id"].get(),
            entrys_inventario["local_id"].get(),
            entrys_inventario["quantidade_sistema"].get(),
            entrys_inventario["quantidade_fisica"].get(),
            entrys_inventario["observacoes"].get(),
            entrys_inventario["inventario_id"].get()
        ))
```

Função que atualiza o inventário



Inventario 1 atualizado trocando a quantidade real de 48 para 50, para condizer com o estoque.

## Listar

## Listar Produtos

```
def listar_produtos():
    for i in tree_produto.get_children():
        tree_produto.delete(i)
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("SELECT * FROM Produto p ORDER BY p.produto_id ASC")
        for row in cur.fetchall():
            tree_produto.insert("", tk.END, values=row)
        cur.close()
        con.close()
    except Exception as e:
        messagebox.showerror("Erro", str(e))
```

## Função de Listar todos os Produtos

produto_id	nome	descricao	peso	data_validade
------------	------	-----------	------	---------------

	produto_id	nome	descricao	peso	data_validade
101		Arroz 5kg	Arroz branco tipo 1	5.00	2025-12-31
102		Notebook Dell	Notebook 15 polegadas	2.20	None
103		Detergente	Detergente neutro 500ml	0.50	2026-01-01
104		Refrigerante Cola	Garrafa 2L	2.00	2025-11-15
105		Caderno 100 folhas	Capa dura	0.40	None
106		Feijão Preto 1kg	Feijão tipo 1	1.00	2026-01-10
107		Macarrão Espaguete	Macarrão tipo espaguete 500g	0.50	2026-03-01
108		Queijo Mussarela	Queijo fatiado 1kg	1.00	2025-08-20
109		Smartphone X	Celular com 128GB	0.30	None
110		TV 50 polegadas	Smart TV 4K	10.00	None
111		Shampoo 300ml	Shampoo anticaSPA	0.30	2026-01-01
112		Sabonete Neutro	Sabonete 90g	0.10	2026-01-01

Ativar o Windows  
Acesse Configurações para mais informações

Listar Produtos

Lista antes e depois de acionar a função de Listar

Listar Fornecedores

```
def listar_fornecedores():
    for i in tree_forn.get_children():
        tree_forn.delete(i)
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("SELECT * FROM Fornecedor f ORDER BY f.fornecedor_id ASC")
        for row in cur.fetchall():
            tree_forn.insert("", tk.END, values=row)
        cur.close()
        con.close()
    except Exception as e:
        messagebox.showerror("Erro", str(e))
```

Função de Listar todos os Fornecedores

fornecedor_id	nome	cnpj	endereco	telefone	email

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Listar Fornecedores

	fornecedor_id	nome	cnpj	endereco	telefone	email
1		Fornecedor Alimentos LTDA	11.111.111/0001-11	Rua A, 100	8599001122	alimentos@fom.com
2		Tech Eletrônicos	22.222.222/0001-22	Av. B, 200	8599002233	eletronicos@tech.com
3		Limpa Tudo S.A.	33.333.333/0001-33	Rua C, 300	8599003344	contato@limpatudo.com
4		Bebidas Brasil	44.444.444/0001-44	Av. D, 400	8599004455	bebidas@brasil.com
5		Escola Fácil	55.555.555/0001-55	Rua E, 500	8599005566	vendas@escolafacil.com
6		Delícias do Sul	66.666.666/0001-66	Rua F, 600	8599006677	contato@deliciasdosul.com
7		Eletronic World	77.777.777/0001-77	Av. G, 700	8599007788	suporte@eletronicworld.com
8		Fornecedora Eletronicos	88.888.888/0001-88	Rua F, 888	8899008899	suporte@eletronicFor.com

Ativar o Windows  
Acesse Configurações para ativar o Windows.

Listar Fornecedores

Lista antes e depois de acionar a função de Listar

Listar Estoques

```
def listar_estoques():
    for i in tree_estoque.get_children():
        tree_estoque.delete(i)
    try:
        con = conectar()
        cur = con.cursor()
        cur.execute("SELECT * FROM Estoque e ORDER BY e.estoque_id ASC")
        for row in cur.fetchall():
            tree_estoque.insert("", tk.END, values=row)
        cur.close()
        con.close()
    except Exception as e:
        messagebox.showerror("Erro", str(e))
```

Buscar por ID:

Buscar por Produto ID:

Buscar

estoque_id	produto_id	local_id	quantidade	data_ultima_movimentacao
------------	------------	----------	------------	--------------------------

Ativar o Windows

Acesse Configurações para ativar o Windows.

Listar Estoques

Buscar por ID:

Buscar por Produto ID:

Buscar

1	estoque_id	101	produto_id	1	local_id	100	quantidade	data_ultima_movimentacao
2		102		2		10		2025-07-01
3		103		3		80		2025-07-05
4		104		5		20		2025-07-10
5		105		2		60		2025-07-15
6		106		1		40		2025-07-16
7		107		1		60		2025-07-20
8		108		5		25		2025-07-20
9		109		2		15		2025-07-20
10		110		4		8		2025-07-20
11		111		3		70		2025-07-20
12		112		3		90		2025-07-20
13		113		6		30		2025-07-20
14		114		6		45		2025-07-20

Ativar o Windows

Acesse Configurações para ativar o Windows.

Listar Estoques

Lista antes e depois de acionar a função de Listar

Busca

Busca por Produto\_Id e Nome em Produto

```
query = "SELECT * FROM Produto WHERE 1=1"
params = []

if id_busca:
    query += " AND produto_id = %s"
    params.append(id_busca)

if nome_busca:
    query += " AND nome ILIKE %s"
    params.append(f"%{nome_busca}%")

for i in tree_produto.get_children():
    tree_produto.delete(i)
```

## Função de busca

Buscar por ID: <input type="text" value="101"/> Buscar por Nome: <input type="text"/> <input type="button" value="Buscar"/>						
produto_id	nome	descricao	peso	data_validade		categoria_id
101	Arroz 5kg	Arroz branco tipo 1	5.00	2025-12-31		1

Busca pelo id 101 e nome "Arroz 5kg"

Busca por Fornecedor\_Id e Nome em Fornecedor

```
def buscar_fornecedor():
    id_busca = entry_busca_forn_id.get().strip()
    nome_busca = entry_busca_forn_nome.get().strip()

    query = "SELECT * FROM Fornecedor WHERE 1=1"
    params = []

    if id_busca:
        query += " AND fornecedor_id = %s"
        params.append(id_busca)

    if nome_busca:
        query += " AND nome ILIKE %s"
        params.append(f"%{nome_busca}%")

    for i in tree_forn.get_children():
        tree_forn.delete(i)
```

## Função de Busca de Fornecedores

Buscar por ID:

Buscar por Nome:

Buscar

	fornecedor_id	nome	cnnpj	endereço	telefone	email
1		Fornecedor Alimentos LTDA	11.111.111/0001-11	Rua A, 100	8599001122	alimentos@forn.com

Buscar por ID:

Buscar por Nome:

cedor Alimentos LTDA

Buscar

1	fornecedor_id	nome	cnpj	endereco	telefone	email
		Fornecedor Alimentos LTDA	11.111.111/0001-11	Rua A, 100	8599001122	alimentos@forn.com

Busca pelo ID 1 e nome “Fornecedor Alimentos LTDA”



## Busca por Loca\_id e Nome em LocalArmazenagem

```
def buscar_local():
    id_busca = entry_busca_local_id.get().strip()
    nome_busca = entry_busca_local_nome.get().strip()

    query = "SELECT local_id, nome, tipo, capacidade_maxima, descricao FROM LocalArmazenagem WHERE 1=1"
    params = []

    if id_busca:
        query += " AND local_id = %s"
        params.append(id_busca)

    if nome_busca:
        query += " AND nome ILIKE %s"
        params.append(f"%{nome_busca}%")

    for i in tree_local.get_children():
        tree_local.delete(i)
```

Função de busca por id e nome de um local de armazenagem.

Buscar por ID: <input type="text"/> Buscar por Nome: <input type="text"/> <input type="button" value="Buscar"/>						
	local_id	nome	tipo	capacidade_maxima	descricao	
1		Prateleira A1	Prateleira	1000	Local Superior	

Buscar por ID: <input type="text"/> Buscar por Nome: <input type="text"/> <input type="button" value="Buscar"/>						
	local_id	nome	tipo	capacidade_maxima	descricao	
1		Prateleira A1	Prateleira	1000	Local Superior	

Busca pelo ID 1 e nome “Prateleira A1”.

## Buscas Avançadas

### 1-Quantidade de Produtos fornecidos por Fornecedor

```
def executar_consulta_fornecedores():
    try:
        conn = conectar()
        cur = conn.cursor()
        cur.execute("""
            SELECT
                f.nome AS fornecedor,
                COUNT(pf.produto_id) AS quantidade_produtos
            FROM
                Fornecedor f
            JOIN
                ProdutoFornecedor pf ON f.fornecedor_id = pf.fornecedor_id
            GROUP BY
                f.nome
            ORDER BY
                quantidade_produtos DESC;
        """)

        for item in tabela_fornecedores.get_children():
            tabela_fornecedores.delete(item)
```

Produtos por Fornecedor

Mostrar Produtos por Fornecedor

Fornecedor	Quantidade de Produtos
Delícias do Sul	7
Fornecedor Alimentos LTDA	6
Tech Eletrônicos	4
Eletronic World	3
Limpa Tudo S.A.	1
Escola Fácil	1

## 2-Total de produtos por tipo de local de armazenagem

```
def executar_consulta_locais():
    try:
        conn = conectar()
        cur = conn.cursor()
        cur.execute("""
            SELECT
                l.tipo AS categoria_local,
                SUM(e.quantidade) AS total_produtos
            FROM
                Estoque e
            JOIN
                LocalArmazenagem l ON e.local_id = l.local_id
            GROUP BY
                l.tipo
            ORDER BY
                total_produtos DESC;
        """)

        for item in tabela_locais.get_children():
            tabela_locais.delete(item)
```

Total de Produtos por Tipo de Local

Mostrar Total por Tipo de Local

Categoria de Local	Total de Produtos
Paleta	240
Prateleira	150
Estante	85
Depósito	75
Refrigerado	45
Corredor	8

### 3-Total de Estoque por Categoria de Produto

```
def executar_consulta_categorias():  
    try:  
        conn = conectar()  
        cur = conn.cursor()  
        cur.execute("""  
            SELECT  
                c.nome AS categoria,  
                SUM(e.quantidade) AS total_estoque  
            FROM  
                Estoque e  
            JOIN  
                Produto p ON p.produto_id = e.produto_id  
            JOIN  
                Categoria c ON c.categoria_id = p.categoria_id  
            GROUP BY  
                c.nome  
            ORDER BY  
                total_estoque DESC;  
        """)
```

#### Total de Produtos por Categoria

Mostrar Estoque por Categoria:

Categoria	Total em Estoque
Alimentos	250
Higiene Pessoal	160
Limpeza	80
Papelaria	60
Eletrônicos	33
Bebidas	20

#### 4-Produtos com mais de um fornecedor com o número de fornecedores

```
def executar_consulta_multifornecedores():  
    try:  
        conn = conectar()  
        cur = conn.cursor()  
        cur.execute("""  
            SELECT  
                p.nome,  
                COUNT(*) AS qtd_fornecedores  
            FROM  
                ProdutoFornecedor pf  
            JOIN  
                Produto p ON p.produto_id = pf.produto_id  
            GROUP BY  
                p.nome  
            HAVING  
                COUNT(*) > 1  
            ORDER BY  
                qtd_fornecedores DESC;  
        """)
```

##### Produtos com Mais de um Fornecedor

Mostrar Produtos com +1 Fornecedor

Produto	Qtd. de Fornecedores
Arroz Integral 1kg	2
Bolacha Recheada	2
Queijo Mussarela	2
Tablet Android	2
Macarrão Espaguete	2
Feijão Preto 1kg	2

## 5-Fornecedores com produtos em estoque

```
def executar_consulta_fornecedores_estoque():
    try:
        conn = conectar()
        cur = conn.cursor()
        cur.execute("""
            SELECT DISTINCT
                f.nome AS fornecedor
            FROM
                Fornecedor f
            JOIN
                ProdutoFornecedor pf ON pf.fornecedor_id = f.fornecedor_id
            JOIN
                Estoque e ON e.produto_id = pf.produto_id;
        """)

        for item in tabela_fornecedores_estoque.get_children():
            tabela_fornecedores_estoque.delete(item)

        for linha in cur.fetchall():
            tabela_fornecedores_estoque.insert('', 'end', values=linha)
```

### Fornecedores com Produtos em Estoque

Mostrar Fornecedores com Produtos no Estoque

Fornecedor

Eletronic World

Tech Eletrônicos

Escola Fácil

Limpa Tudo S.A.

Bebidas Brasil

Delícias do Sul