def adicionar\_item\_saida():

global saida\_produtos

nome\_produto = entrada\_nome\_qtde\_saida.get().strip()

qtde\_retirada = entrada\_qtde\_retirada.get().strip()

if not nome\_produto or not qtde\_retirada:

showinfo("Aviso", "Preencha todos os campos!")

return

try:

qtde = int(qtde\_retirada)

if qtde <= 0:

showinfo("Aviso", "A quantidade deve ser maior que zero!")

return

except ValueError:

showinfo("Aviso", "Quantidade inválida!")

return

# Adiciona o item à lista de saída

item\_texto = f"{nome\_produto} - -{qtde} unidades"

# Verifica se já existe na lista para atualizar a quantidade

encontrado = False

for i, item in enumerate(saida\_produtos):

if item['nome'] == nome\_produto:

saida\_produtos[i]['quantidade'] += qtde

encontrado = True

break

if not encontrado:

# Adiciona novo item ao vetor

saida\_produtos.append({

'nome': nome\_produto,

'quantidade': qtde

})

# Atualiza a exibição na interface

atualizar\_lista\_saida()

# Limpa o campo de quantidade

entrada\_qtde\_retirada.delete(0, tk.END)

def atualizar\_lista\_saida():

# Limpa todos os widgets do frame\_scroll4

for widget in frame\_scroll4.winfo\_children():

widget.destroy()

# Adiciona cada item do vetor à interface

for i, item in enumerate(saida\_produtos):

label = customtkinter.CTkLabel(frame\_scroll4, text=f"{item['nome']} - -{item['quantidade']} unidades")

label.grid(row=i, column=0, pady=5, padx=0, sticky="w")

botao\_remover = customtkinter.CTkButton(

frame\_scroll4,

text="🗑️",

width=5,

command=lambda idx=i: remover\_item\_saida(idx)

)

botao\_remover.grid(row=i, column=1, pady=5, padx=5)

def remover\_item\_saida(indice):

global saida\_produtos

if 0 <= indice < len(saida\_produtos):

saida\_produtos.pop(indice)

atualizar\_lista\_saida() linha 434