



INSTITUTO SUPERIOR POLITÉCNICO PRIVADO DA CATEPA

(Aprovado pelo Decreto nº 132/17 de 19 de Junho)

Departamento de Ciência da Engenharia

PROGRAMAÇÃO 1

CONVERSOR DE UNIDADES

Autores:

Félix Maurício Cupatia Domingos (Chefe)

Filipe Salvador Domingos

Luciano Correia Cassua

Turma: Única

Sala: 5

Engenharia Electrotécnica

Professor

Ovaldo Queta

Malanje, Fevereiro de 2026

FÉLIX MAURÍCIO CUPATIA DOMINGOS (CHEFE)

FILIPE SALVADOR DOMINGOS

LUCIANO CORREIA CASSUA

CONVERSOR DE UNIDADES

Projecto da disciplina de Programação I apresentado ao Professor do Instituto Superior Politécnico Privado da Catepa como requisito parcial para a obtenção da avaliação semestral.

Professor: Ovaldo Queta

Malanje, Fevereiro de 2026

ÍNDICE

INTRODUÇÃO.....	3
1. DESCRIÇÃO GERAL DO PROJECTO	4
2. TKINTER.....	4
3. UNIDADES DE CONVERSÃO SUPORTADAS	5
4. TESTES E RESULTADOS	5
5. CÓDIGO-FONTE COMPLETO.....	5
CONCLUSÃO.....	9

INTRODUÇÃO

Este documento detalha o desenvolvimento de um Conversor de Unidades implementado em Python com a biblioteca Tkinter. Esta versão foi concebida com uma estrutura linear, focando na simplicidade e no fluxo sequencial do código, o que é ideal para fins didáticos na disciplina de Programação I do ISCAT. O objetivo é proporcionar uma ferramenta funcional para a conversão de unidades, ao mesmo tempo que serve como um exemplo claro de programação GUI sem o uso de abstrações complexas.

Objectivos

Geral

Desenvolver uma aplicação gráfica em Python que permita a conversão de diferentes unidades de medida, aplicando estruturas de controlo fundamentais e princípios básicos de programação orientada à interface gráfica.

Específicos

- Caracterizar os valores de entrada segundo a categoria e a unidade seleccionada pelo utilizador.
- Implementar a lógica de conversão para unidades de comprimento, massa e temperatura.
- Utilizar estruturas de decisão e repetição para garantir a execução correta das conversões.
- Apresentar os resultados de forma clara e organizada na interface gráfica.
- Proporcionar uma interação simples e intuitiva entre o utilizador e o sistema.

1. DESCRIÇÃO GERAL DO PROJECTO

O presente projecto consiste no desenvolvimento de um Conversor de Unidades em Python, utilizando a biblioteca Tkinter para a criação de uma interface gráfica de utilizador. A aplicação tem como finalidade permitir que o utilizador converta valores entre diferentes unidades de medida de forma simples, rápida e eficiente.

O sistema está organizado por categorias de conversão, nomeadamente Comprimento, Massa e Temperatura, possibilitando a escolha da unidade de origem e da unidade de destino. O utilizador introduz o valor a ser convertido, selecciona as opções desejadas e, através de um comando, o programa processa os dados e apresenta o resultado automaticamente.

A interface foi concebida de forma intuitiva, com campos de entrada, listas suspensas e botões que facilitam a interação com o programa. Internamente, o projecto aplica estruturas de decisão e operações matemáticas para garantir a correta transformação dos valores entre as unidades seleccionadas.

Além de cumprir uma função prática, o projecto tem também um carácter didático, servindo como apoio ao aprendizado de programação com interfaces gráficas, reforçando conceitos fundamentais da disciplina de Programação I.

2. TKINTER

O Tkinter é a biblioteca padrão do Python para o desenvolvimento de interfaces gráficas de utilizador (GUI), permitindo criar aplicações visuais e interativas. Através dos seus componentes, chamados widgets, é possível construir janelas, botões, campos de texto, listas suspensas e outros elementos que facilitam a comunicação entre o utilizador e o sistema.

A aplicação inicia-se com a criação da janela principal, onde são organizados os elementos gráficos por meio de gestores de layout como o pack e o grid. As acções são associadas aos widgets, possibilitando que o programa responda aos eventos do utilizador, como cliques e inserções de dados. O funcionamento da interface é mantido pelo mainloop, que garante que a janela permaneça activa. Neste projecto, o Tkinter é utilizado para disponibilizar uma interface simples e eficiente para a conversão de unidades.

3. UNIDADES DE CONVERSÃO SUPORTADAS

A tabela abaixo detalha as unidades suportadas pelo conversor, organizadas por categoria:

Categoria	Unidades Suportadas
Comprimento	Quilômetro (km), Metro (m), Centímetro (cm)
Massa	Quilograma (kg), Grama (g)
Temperatura	Celsius (°C), Fahrenheit (°F)

4. TESTES E RESULTADOS

Para validar a precisão das conversões, foram realizados testes com valores conhecidos. A tabela a seguir apresenta alguns dos casos de teste e os resultados esperados e obtidos:

Categoria	Valor de Entrada	Unidade de Origem	Unidade de Destino	Resultado Esperado	Resultado Obtido
Comprimento	1	Quilômetro (km)	Metro (m)	1000	1000.0000
Comprimento	100	Centímetro (cm)	Metro (m)	1	1.0000
Massa	1	Quilograma (kg)	Grama (g)	1000	1000.0000
Temperatura	0	Celsius (°C)	Fahrenheit (°F)	32	32.0000
Temperatura	100	Celsius (°C)	Fahrenheit (°F)	212	212.0000

Todos os testes foram bem-sucedidos, confirmando a correta implementação da lógica de conversão para as unidades e categorias especificadas.

5. CÓDIGO-FONTE COMPLETO

O código-fonte completo da aplicação conversor.py, com comentários detalhados em cada linha, é apresentado abaixo:

```

import tkinter as tk # Importa a biblioteca principal do Tkinter para a interface gráfica
from tkinter import ttk, messagebox # Importa widgets modernos e caixas de diálogo

# --- CONFIGURAÇÃO DA JANELA PRINCIPAL ---
janela = tk.Tk() # Cria a janela principal do programa
janela.title("Conversor de Unidades - ISCAT ") # Define o título da janela
janela.geometry("400x600") # Define a largura e altura da janela
janela.resizable(False, False) # Bloqueia o redimensionamento da janela

# --- TÍTULO DA INTERFACE ---
titulo = tk.Label(janela, text="Conversor de Unidades", font=("Arial", 16, "bold"), pady=20) # Cria o texto do título
titulo.pack() # Adiciona o título à janela

# --- SEÇÃO DE CATEGORIA ---
moldura_cat = ttk.LabelFrame(janela, text=" Selecione a Categoria ", padding=10) # Cria uma moldura para a categoria
moldura_cat.pack(padx=20, pady=10, fill="x") # Posiciona a moldura na janela

categorias = ["Comprimento", "Massa", "Temperatura"] # Lista de categorias disponíveis
var_categoria = tk.StringVar(value=categorias[0]) # Variável para armazenar a categoria escolhida
combo_categoria = ttk.Combobox(moldura_cat, textvariable=var_categoria, values=categorias, state="readonly") # Cria a lista de categorias
combo_categoria.pack(fill="x") # Adiciona a lista à moldura

# --- SEÇÃO DE CONVERSÃO ---
moldura_conv = ttk.LabelFrame(janela, text=" Dados da Conversão ", padding=10) # Cria a moldura para os dados
moldura_conv.pack(padx=20, pady=10, fill="x") # Posiciona a moldura

tk.Label(moldura_conv, text="Valor:").grid(row=0, column=0, sticky="w", pady=5) # Rótulo para o campo de valor
campo_valor = ttk.Entry(moldura_conv) # Cria a caixa de entrada para o número
campo_valor.grid(row=0, column=1, sticky="ew", pady=5) # Posiciona a caixa de entrada

tk.Label(moldura_conv, text="De:").grid(row=1, column=0, sticky="w", pady=5) # Rótulo para unidade de origem
var_origem = tk.StringVar() # Variável para a unidade de origem
combo_origem = ttk.Combobox(moldura_conv, textvariable=var_origem, state="readonly") # Lista de origem
combo_origem.grid(row=1, column=1, sticky="ew", pady=5) # Posiciona a lista

```

de origem

```
tk.Label(moldura_conv, text="Para:").grid(row=2, column=0, sticky="w", pady=5)
# Rótulo para unidade de destino
var_destino = tk.StringVar() # Variável para a unidade de destino
combo_destino = ttk.Combobox(moldura_conv, textvariable=var_destino, state="readonly") # Lista de destino
combo_destino.grid(row=2, column=1, sticky="ew", pady=5) # Posiciona a lista de destino
```

--- RÓTULO DE RESULTADO ---

```
rotulo_resultado = tk.Label(janela, text="Resultado: -", font=("Arial", 12, "bold"), fg="blue", pady=10) # Cria o texto do resultado
rotulo_resultado.pack() # Adiciona o resultado à janela
```

--- LÓGICA DE ATUALIZAÇÃO DE UNIDADES ---

Esta parte define o que acontece quando a categoria muda

```
def atualizar_listas(event=None): # Nota: O Tkinter exige uma função para eventos, mas a lógica interna é linear
```

```
    escolha = var_categoria.get() # Verifica qual categoria foi selecionada
```

```
    if escolha == "Comprimento": # Se for comprimento
```

```
        unidades = ["Quilômetro (km)", "Metro (m)", "Centímetro (cm)"] # Define unidades de medida
```

```
    elif escolha == "Massa": # Se for massa
```

```
        unidades = ["Quilograma (kg)", "Gramma (g)"] # Define unidades de peso
```

```
    else: # Se for temperatura
```

```
        unidades = ["Celsius (°C)", "Fahrenheit (°F)"] # Define unidades de calor
```

```
    combo_origem["values"] = unidades # Atualiza a lista de origem
```

```
    combo_destino["values"] = unidades # Atualiza a lista de destino
```

```
    combo_origem.current(0) # Seleciona o primeiro item por padrão
```

```
    combo_destino.current(1) # Seleciona o segundo item por padrão
```

```
combo_categoria.bind("<<ComboboxSelected>>", atualizar_listas) # Liga a mudança de categoria à atualização
```

```
atualizar_listas() # Executa uma vez no início para preencher as listas
```

--- LÓGICA DE CONVERSÃO (EXECUTADA PELO BOTÃO) ---

```
def executar_conversao(): # Nota: O comando do botão exige uma função no Tkinter
```

```
    try: # Inicia tratamento de erro para valores não numéricos
```

```
        v = float(campo_valor.get()) # Pega o valor digitado e converte para decimal
```

```
        cat = var_categoria.get() # Pega a categoria atual
```

```
        orig = var_origem.get() # Pega a unidade de origem
```

```
        dest = var_destino.get() # Pega a unidade de destino
```

```
        res = 0.0 # Inicializa a variável de resultado
```



```

if orig == dest: # Se as unidades forem iguais
    res = v # O resultado é o próprio valor
elif cat == "Comprimento": # Lógica para Comprimento
    if "km" in orig: m = v * 1000 # De km para metros
    elif "cm" in orig: m = v / 100 # De cm para metros
    else: m = v # Já está em metros
    if "km" in dest: res = m / 1000 # De metros para km
    elif "cm" in dest: res = m * 100 # De metros para cm
    else: res = m # Fica em metros
elif cat == "Massa": # Lógica para Massa
    if "kg" in orig: res = v * 1000 # De kg para g
    else: res = v / 1000 # De g para kg
elif cat == "Temperatura": # Lógica para Temperatura
    if "°C" in orig: res = (v * 9/5) + 32 # De Celsius para Fahrenhe
it
        else: res = (v - 32) * 5/9 # De Fahrenheit para Celsius

rotulo_resultado.config(text=f"Resultado: {res:.4f}") # Mostra o res
ultado na tela
except ValueError: # Se o utilizador não digitar um número
    messagebox.showerror("Erro", "Insira um número válido!") # Mostra me
nsagem de erro

# --- BOTÃO CONVERTER ---
botao = ttk.Button(janela, text="CALCULAR CONVERSÃO", command=executar_conver
sao) # Cria o botão de ação
botao.pack(pady=10) # Adiciona o botão à janela

# --- RODAPÉ ---
tk.Label(janela, text="Félix D. Filipe D. Luciano C. - ISCAT", font=("Arial",
8, "italic")).pack(side="bottom", pady=10) # Texto final

janela.mainloop() # Mantém a janela aberta e funcionando

```

CONCLUSÃO

Esta versão Conversor de Unidades, desenvolvida em Python com Tkinter, cumpre os requisitos de funcionalidade e didáticos, oferecendo uma ferramenta prática e um exemplo claro de programação GUI. A documentação detalhada, incluindo o código-fonte completo e tabelas de validação, visa facilitar a compreensão e a avaliação do projecto. A simplicidade da estrutura do código, aliada aos comentários linha a linha, torna-o um recurso valioso para o aprendizado de programação de interfaces gráficas.