Claro, Aloisio! Aqui está a **documentação atualizada** do seu projeto de Conversor Numérico Multibase com interface gráfica em Python. Ela inclui:

* ✅ Visão geral do projeto
* 👤 Manual do usuário final
* 🧑‍💻 Especificação técnica para desenvolvedores
* 🔧 Guia de manutenção e expansão
* 📄 Créditos e autoria

**📘 Documentação do Projeto: Conversor Numérico Multibase com Interface Gráfica**

**🧩 Visão Geral**

Este projeto é um aplicativo em Python com interface gráfica que permite converter números entre diferentes sistemas de numeração:

* Decimal ↔ Binário
* Binário ↔ Decimal
* Octal ↔ Binário
* Binário ↔ Octal
* Hexadecimal ↔ Binário
* Binário ↔ Hexadecimal
* Decimal ↔ Octal
* Decimal ↔ Hexadecimal
* Octal ↔ Hexadecimal
* Hexadecimal ↔ Octal

A interface foi desenvolvida com Tkinter, oferecendo uma experiência intuitiva e acessível para qualquer usuário, sem necessidade de comandos no terminal.

**👤 Manual do Usuário Final**

**🖥️ Requisitos**

* Python 3 instalado
* Sistema operacional compatível com Tkinter (Windows, macOS, Linux)

**🚀 Como Executar**

1. Salve o código em um arquivo chamado conversor\_gui.py
2. Execute com o comando:

python conversor\_gui.py

**🧭 Como Usar**

1. Escolha a **base de origem** e a **base de destino** em campos separados.
2. Digite o número no campo "Número".
3. (Opcional) Informe o número de casas decimais para conversões com frações.
4. Clique em **Converter**.
5. Veja o resultado destacado em fonte grande.
6. Se as bases forem iguais, o programa exibirá um alerta.

**🧑‍💻 Especificação Técnica para Desenvolvedores**

**🔧 Tecnologias Utilizadas**

* **Python 3.x**
* **Tkinter** (GUI padrão do Python)

**📂 Estrutura do Código**

**1. Funções de Conversão**

Cada função é isolada e pode ser testada individualmente:

* decimal\_para\_binario(numero\_decimal, casas\_fracionarias)
* binario\_para\_decimal(binario\_str)
* decimal\_para\_octal(numero\_decimal)
* octal\_para\_decimal(octal\_str)
* decimal\_para\_hex(numero\_decimal)
* hex\_para\_decimal(hex\_str)
* binario\_para\_octal(binario\_str)
* octal\_para\_binario(octal\_str)
* binario\_para\_hex(binario\_str)
* hex\_para\_binario(hex\_str)

**2. Função Principal: converter()**

* Lê os dados da interface
* Verifica se as bases são diferentes
* Executa a função de conversão correspondente
* Exibe o resultado ou alerta de erro

**3. Interface Gráfica**

* Criada com Tkinter e ttk para widgets modernos
* Campos:
  + Combobox para base de origem
  + Combobox para base de destino
  + Entry para número
  + Entry para casas decimais
  + Button para converter
  + Label com fonte 20 para exibir resultado

**🔧 Guia de Manutenção e Expansão**

**📌 Para adicionar novas conversões**

1. Crie uma nova função de conversão.
2. Adicione a base ao Combobox se necessário.
3. Atualize a função converter() com a nova lógica.

**📌 Para melhorar a interface**

* Adicionar histórico de conversões
* Exportar resultados para arquivo .txt ou .csv
* Adicionar botão de limpar campos
* Adicionar validação visual (ex: bordas vermelhas para erro)

**📌 Para transformar em aplicativo web**

* Use Flask, Streamlit ou Gradio
* Reaproveite as funções de conversão como backend
* Crie rotas ou componentes para entrada e saída

**📄 Créditos e Autoria**

**Criado por:** Aloisio Bomfim  
**Data:** Outubro de 2025  
**Licença sugerida:** MIT License (livre uso e modificação com atribuição)

Se quiser, posso preparar um README pronto para GitHub, gerar um PDF da documentação ou empacotar o projeto como instalador .exe. Me avise como prefere seguir!