

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SÃO PAULO**

JOÃO PEDRO VIANA RODRIGUES

TO DO LIST

**CAMPOS DO JORDÃO
2025**

RESUMO

Este relatório descreve o desenvolvimento de uma aplicação desktop de produtividade, criada utilizando o framework Qt em C++. O projeto visa proporcionar uma ferramenta eficiente para organização pessoal, combinando uma lista de tarefas interativa com a técnica de gestão de tempo Pomodoro. O software incorpora persistência de dados via JSON e elementos de interface gráfica moderna. A implementação foi realizada sob uma abordagem orientada a objetos, garantindo a modularidade, encapsulamento e a manutenibilidade do código.

INTRODUÇÃO

Este trabalho apresenta o processo de concepção e desenvolvimento do "Gerenciador de Tarefas e Foco", um software que auxilia o usuário na organização de suas atividades diárias. O objetivo principal é criar uma aplicação robusta e funcional, utilizando os princípios da programação orientada a objetos e os recursos visuais do Qt Widgets.

Objetivos:

- Desenvolver um sistema de gestão de tarefas (CRUD) em C++ utilizando o Qt Framework.
- Implementar um cronômetro Pomodoro integrado para auxiliar no foco do usuário.
- Aplicar conceitos de Orientação a Objetos, como herança e encapsulamento, para criar widgets personalizados (TaskWidget).
- Utilizar persistência de dados (JSON) para salvar e carregar o estado das tarefas automaticamente.

Metodologia

O desenvolvimento iniciou-se com a definição da arquitetura do projeto e o design da interface gráfica. A escolha da linguagem C++ e do Qt Creator foi motivada pela performance e pela capacidade de criação de interfaces ricas baseadas em eventos.

Ferramentas Utilizadas:

- **Linguagem de Programação:** C++
- **Compilador:** MinGW 64-bit / GCC.
- **Framework Gráfico:** Qt 6 (Módulo Widgets).
- **IDE:** Qt Creator / Visual Studio Code.
- **Persistência:** Biblioteca QJsonDocument (JSON).

Descrição do Projeto: O sistema é estruturado em uma janela principal dividida em dois módulos essenciais para o fluxo de trabalho do usuário:

- **Módulo de Tarefas (To-Do List):** Localizado à esquerda da interface. Permite ao usuário digitar uma nova tarefa e adicioná-la à lista. As tarefas são separadas visualmente em "Pendentes" e "Concluídas".
- **Módulo de Foco (Pomodoro):** Localizado à direita. Apresenta um cronômetro digital e controles para alternar entre os modos "Trabalho" (25 min), "Pausa Curta" (5 min) e "Pausa Longa" (15 min).
- **Widget Personalizado:** Cada tarefa na lista não é apenas texto, mas um objeto interativo contendo um *checkbox* e um botão de exclusão.

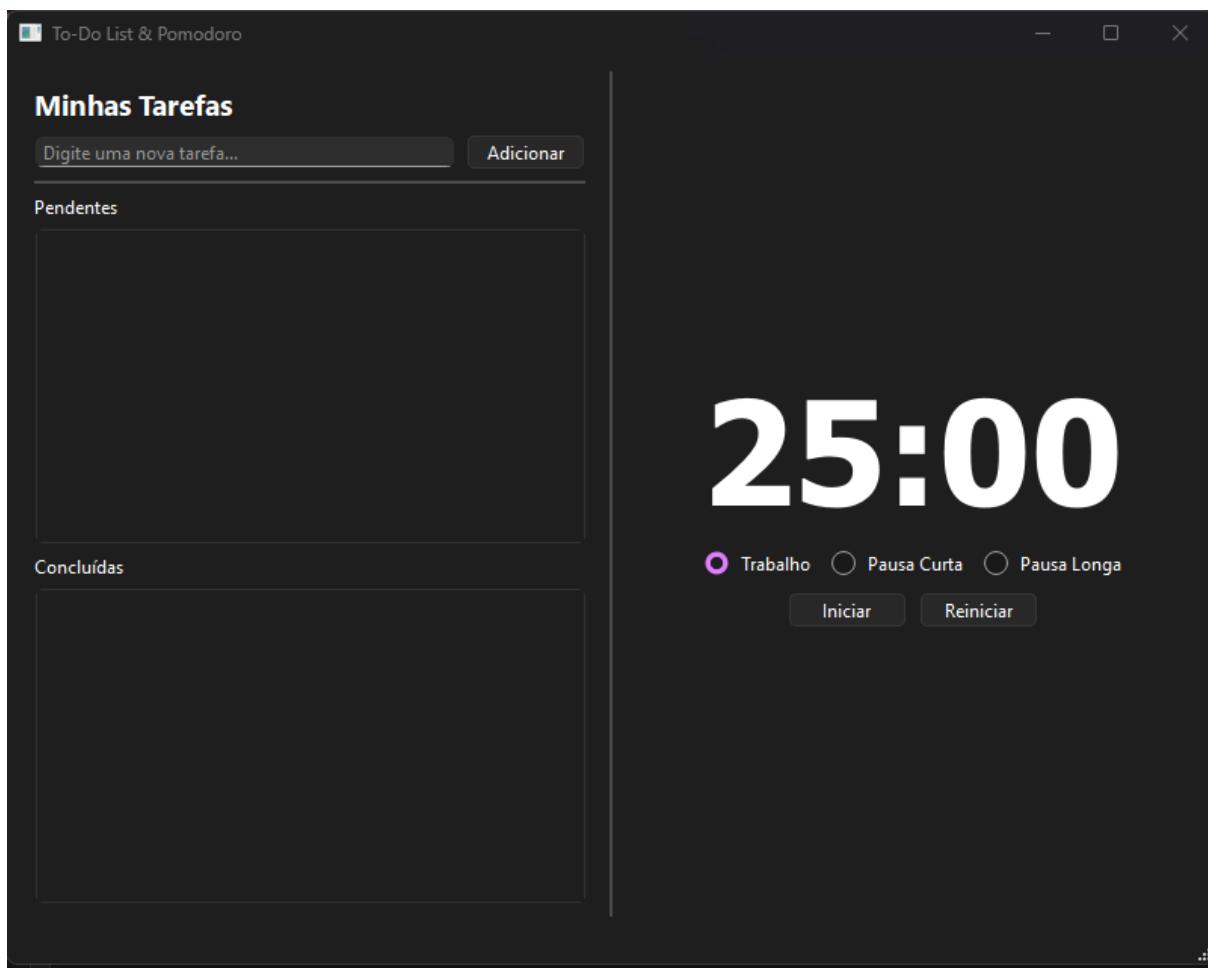
Arquitetura Orientada a Objetos: A lógica do sistema é gerenciada pelas seguintes classes:

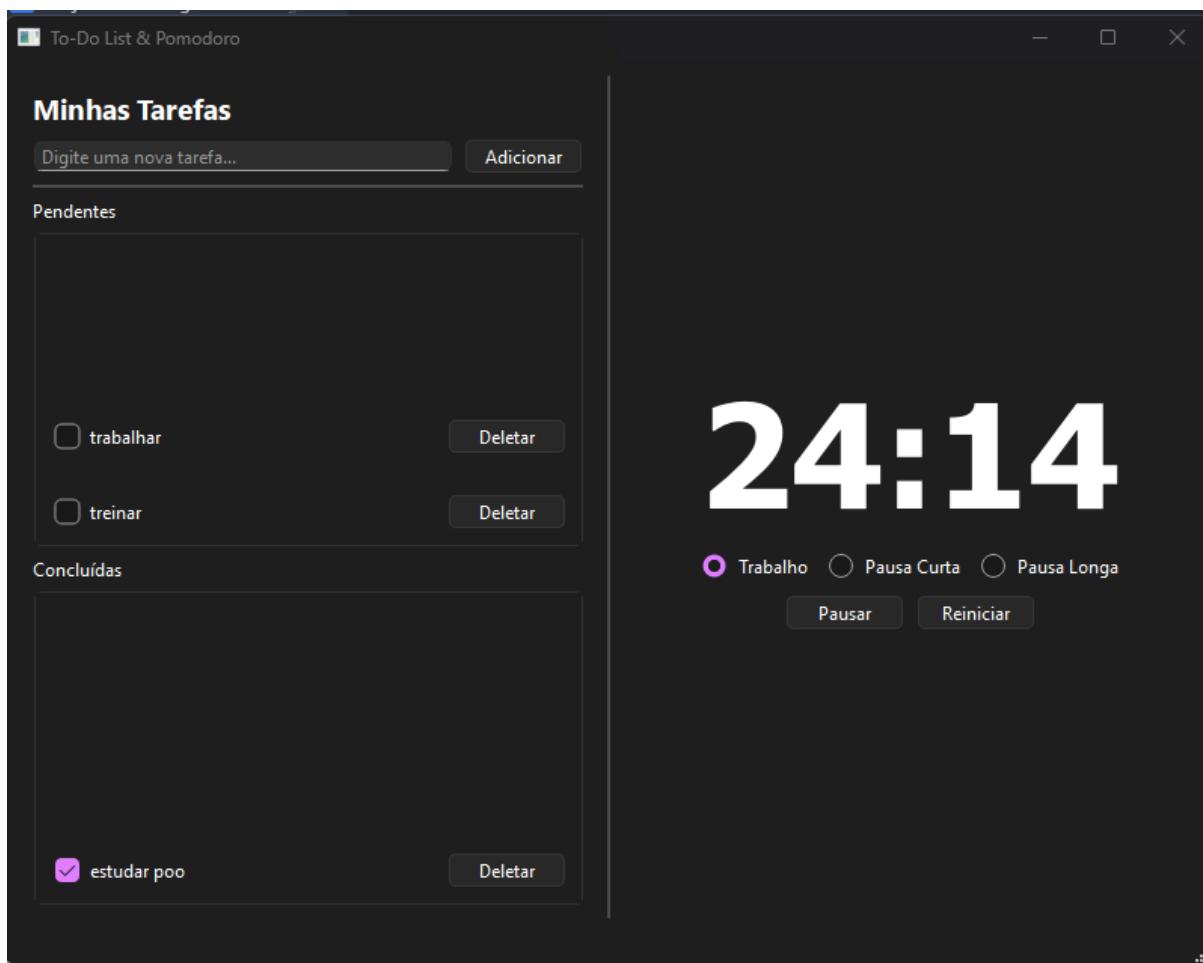
- **MainWindow (Herda de QMainWindow):** Classe principal que gerencia o ciclo de vida da aplicação, a persistência de dados (Salvar/Carregar JSON) e a lógica do cronômetro (QTimer).
- **TaskWidget (Herda de QWidget):** Classe responsável por representar uma única tarefa. Ela encapsula os componentes visuais (Checkbox e Botão) e emite sinais personalizados (statusChanged, taskRemoved) para comunicar alterações à janela principal.
- **Sistema de Sinais e Slots:** Utilizado extensivamente para conectar as ações do usuário (cliques) às funções do código, garantindo um baixo acoplamento entre os componentes.

Resultados Obtidos

O desenvolvimento resultou em um software funcional que demonstra a aplicação prática de C++ moderno e bibliotecas gráficas. Foram superados desafios como a configuração do ambiente de compilação (qmake), a manipulação de arquivos JSON e a criação de widgets customizados.

Capturas de Tela do sistema:





Conclusão

O projeto "Gerenciador de Tarefas e Foco" foi concluído com êxito, entregando todas as funcionalidades propostas. A utilização da abordagem orientada a objetos permitiu que a interface fosse modular, facilitando a manutenção e a expansão futura. A integração com o sistema de arquivos para salvar as tarefas em JSON garantiu que o software fosse útil em cenários reais de uso.

Sugestões para Possíveis Melhorias:

- **Notificações de Sistema:** Implementar alertas sonoros ou pop-ups quando o tempo do Pomodoro acabar.
- **Categorias de Tarefas:** Permitir criar etiquetas (tags) para filtrar tarefas por "Trabalho", "Estudo", etc.

- **Tema Escuro:** Adicionar uma opção para alternar as cores da interface (Dark Mode).

8. Referências Bibliográficas

- QT DOCUMENTATION. Qt Widgets C++ Classes. Disponível em:
<https://doc.qt.io/qt-6/qtwidgets-index.html>.
- C++ REFERENCE. cppreference.com. Disponível em:
<https://en.cppreference.com/w/cpp>.
- JSON.org. Introducing JSON. Disponível em:
<https://www.json.org/json-en.html>.