

Manual do Usuário - SISAQUI

Sistema de Aquisição Universal para Balanças

Laboratório de Medição Mecânicas (LMM) - IPT

Gustavo dos Santos Ribeiro
Victor Nascimento Pereira

Janeiro de 2026

Conteúdo

| | | |
|-----------|---|----------|
| 1 | Introdução | 3 |
| I | Manual para Usuário Final | 3 |
| 2 | Instalação para Usuários Finais | 3 |
| 2.1 | Requisitos do Sistema | 3 |
| 2.2 | Instalação | 3 |
| 3 | Uso do Aplicativo | 3 |
| 3.1 | Inicialização | 3 |
| 3.2 | Interface do Aplicativo | 4 |
| 3.3 | Configuração do Arquivo de Dados | 4 |
| 3.4 | Leitura e Salvamento de Dados | 5 |
| 3.5 | Comandos Adicionais | 5 |
| 3.6 | Encerramento | 6 |
| 4 | Troubleshooting para Usuários Finais | 6 |
| 4.1 | Erro de Conexão Serial | 6 |
| 4.2 | Erro 30 - Porta Serial Ocupada | 6 |
| 4.3 | Arquivo CSV Não Abre | 6 |
| 4.4 | Problemas Gerais | 7 |
| II | Manual para Desenvolvedores | 7 |
| 5 | Instalação para Desenvolvedores | 7 |
| 5.1 | Requisitos do Sistema | 7 |
| 5.2 | Instalação do Ambiente | 7 |
| 5.3 | Geração do Executável | 8 |

| | |
|---|----------|
| 6 Estrutura do Código | 8 |
| 6.1 Principais Arquivos | 8 |
| 6.2 Classe Principal: AppBalanca | 8 |
| 6.2.1 Métodos Principais | 8 |
| 6.3 Comunicação Serial | 8 |
| 7 Desenvolvimento e Modificações | 9 |
| 7.1 Adicionando Novas Funcionalidades | 9 |
| 7.2 Debugging | 9 |
| 7.3 Versionamento | 9 |
| 8 Troubleshooting para Desenvolvedores | 9 |
| 8.1 Erros de Importação | 9 |
| 8.2 Problemas de Serial | 9 |
| 8.3 Erros de Build | 9 |
| 8.4 Contribuição | 10 |

1 Introdução

O SISAQUI (Sistema de Aquisição Universal) é uma aplicação desenvolvida especificamente para o Laboratório de Medidas Mecânicas (LMM) do Instituto de Pesquisas Tecnológicas (IPT), com o objetivo de automatizar a coleta de dados de balanças Sartorius via conexão serial. Este software facilita medições metrológicas precisas, salvando dados em arquivos CSV para análise posterior.

Desenvolvido por Gustavo dos Santos Ribeiro e Victor Nascimento Pereira, o SISAQUI oferece uma interface intuitiva para usuários finais e um código modular para desenvolvedores, garantindo eficiência e confiabilidade nas operações laboratoriais.

Parte I

Manual para Usuário Final

2 Instalação para Usuários Finais

Para usuários finais do laboratório, o SISAQUI é distribuído como um executável stand-alone, não requerendo instalação de dependências adicionais.

2.1 Requisitos do Sistema

- Sistema Operacional: Windows 10 ou superior
- Espaço em disco: Mínimo 50 MB
- Porta serial disponível (COM) e driver do cabo sendo usado para conexão com a balança Sartorius

2.2 Instalação

1. Acesse a pasta de rede "automatização de balanças".
2. Abra o executável `app_balanca.exe`.
3. Execute o arquivo diretamente. Não é necessária instalação adicional.

Nota: Certifique-se de que o antivírus não bloqueie a execução do arquivo. Se necessário, adicione uma exceção para o executável.

3 Uso do Aplicativo

3.1 Inicialização

1. Ligue a balança Sartorius e aguarde de 30 segundos a 1 minuto para que ela estabilize completamente.
2. Conecte o cabo serial (USB) da balança ao computador.

3. Abra o aplicativo SISAQUI executando o arquivo `app_balanca.exe`.
4. Selecione a porta COM correspondente à balança no menu suspenso.
5. Clique em "Conectar" para estabelecer a comunicação serial.

3.2 Interface do Aplicativo

A interface principal do SISAQUI é dividida em painéis intuitivos, facilitando o uso por usuários do laboratório.

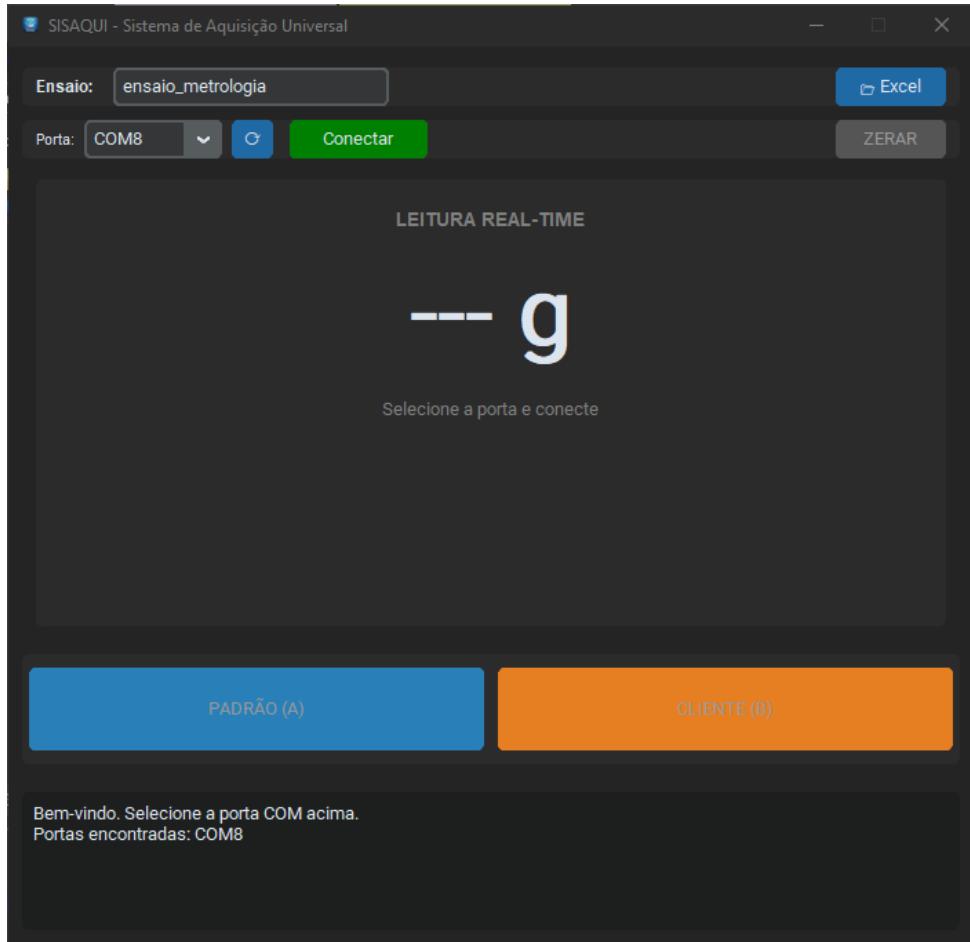


Figura 1: Interface principal do SISAQUI mostrando os painéis de arquivo, conexão, display e ações.

3.3 Configuração do Arquivo de Dados

- No campo "Ensaio:", insira o nome do arquivo onde você quer que os dados sejam salvos (ex.: `ensaio_metrologia`).
- Os dados serão salvos automaticamente na subpasta `dados_coletados/` no arquivo com o nome escolhido (ex.: `ensaio_metrologia`).

3.4 Leitura e Salvamento de Dados

- Após conectar, a leitura em tempo real será exibida no painel central.
- Use os botões "PADRÃO (A)" ou "CLIENTE (B)" para salvar medições.
- Atalhos de teclado:
 - A ou a: Salvar como Padrão (A)
 - B ou b: Salvar como Cliente (B)
 - Espaço: Salvar como Genérico

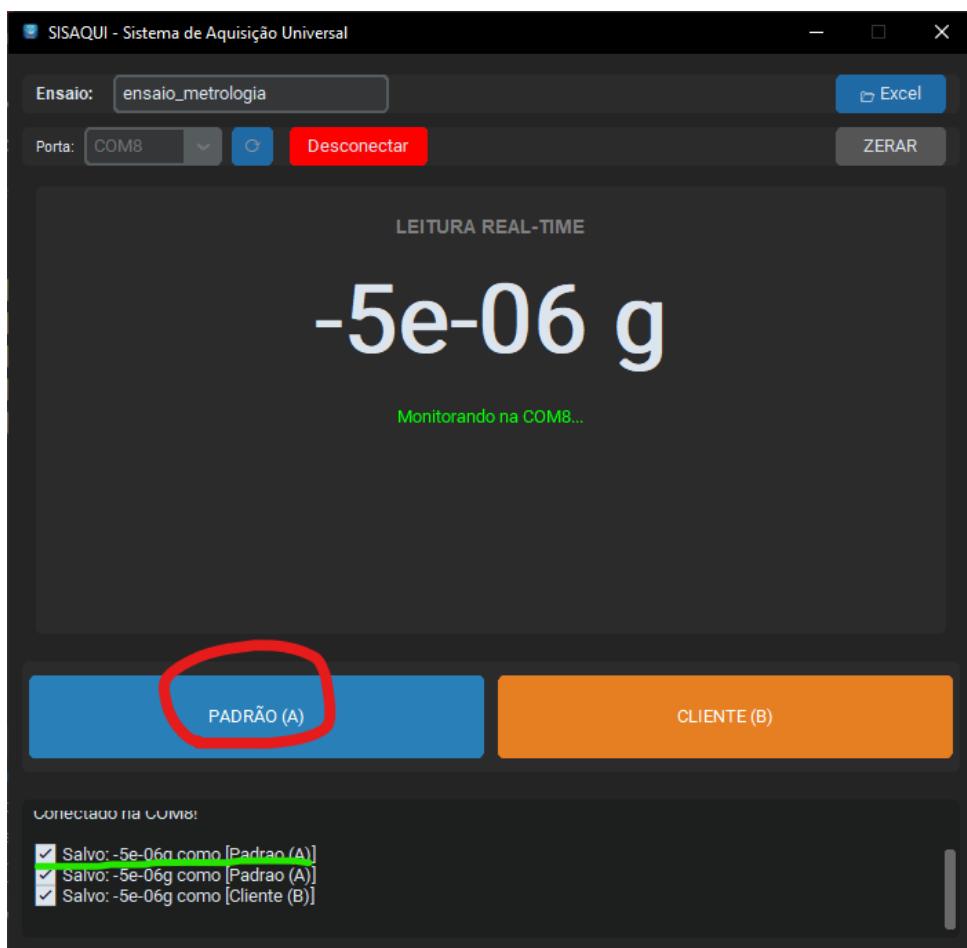


Figura 2: Demonstração do salvamento de dados com botões destacados.

Note que ao salvar uma medição o painel de log do aplicativo indica que esta foi salva e qual seu tipo (Padrão, Cliente ou Genérico).

3.5 Comandos Adicionais

- Botão "ZERAR": Tara a balança (envia comando de tara).
- Botão "Excel": Abre o arquivo CSV no Microsoft Excel (se instalado).

3.6 Encerramento

- Clique em "Desconectar" antes de fechar o aplicativo para liberar a porta serial.
- O aplicativo fecha automaticamente a conexão serial ao ser fechado.

Jamais puxe o cabo serial enquanto o aplicativo estiver conectado, para evitar erros de comunicação, sempre clique em "Desconectar" antes de mexer no cabo.

4 Troubleshooting para Usuários Finais

4.1 Erro de Conexão Serial

- Verifique se a balança está ligada e conectada corretamente à porta COM selecionada.
- Certifique-se de que nenhum outro programa está usando a porta serial.
- Reinicie o aplicativo e tente conectar novamente.

4.2 Erro 30 - Porta Serial Ocupada

O erro 30 indica que a porta serial está bloqueada. Siga estes passos:

1. **Desligue a balança:** Desconecte a energia da balança.
2. **Remova o cabo serial:** Desconecte o cabo USB/serial da balança e do computador.
3. **Espere 30 segundos:** Aguarde pelo menos 30 segundos.
4. **Conekte apenas a alimentação:** Reconecte o cabo de alimentação da balança (sem o cabo serial).
5. **Ligue a balança:** Ligue a balança e aguarde estabilização.
6. **Espere 1 minuto:** Aguarde 1 minuto completo.
7. **Conekte o cabo serial:** Agora conecte o cabo USB/serial.
8. **Reinic peace o aplicativo:** Abra o SISAQUI novamente e tente conectar.

4.3 Arquivo CSV Não Abre

- Certifique-se de que o Microsoft Excel ou um leitor de CSV compatível está instalado.
- Feche o arquivo se estiver aberto em outro programa.
- Verifique se o arquivo foi criado na subpasta `dados_coletados/`.

4.4 Problemas Gerais

- Consulte o painel de log inferior para mensagens de erro, bem como o painel da balança.
- Reinicie o computador se os problemas persistirem.
- Entre em contato com a equipe do LMM para suporte.

Parte II

Manual para Desenvolvedores

5 Instalação para Desenvolvedores

Para desenvolvimento e modificações no código, é necessário configurar o ambiente Python.

5.1 Requisitos do Sistema

- Python 3.8 ou superior
- Git para controle de versão
- Editor de código (recomendado: Visual Studio Code)

5.2 Instalação do Ambiente

1. Clone o repositório:

```
git clone https://github.com/gustaribeiro8/leitor-balanca-sartorius.git
cd automatizacao_balanca
```

2. Crie um ambiente virtual (recomendado):

```
python -m venv .venv
```

3. Ative o ambiente virtual:

- No Windows: .venv\Scripts\activate
- No Linux/Mac: source .venv/bin/activate

4. Instale as dependências:

```
pip install -r requirements.txt
```

5. Execute o aplicativo em modo desenvolvimento:

```
python app_balanca.py
```

5.3 Geração do Executável

Para criar um executável standalone:

```
python -m PyInstaller --noconsole --onefile --clean --icon=icone_sartorius.ico --add-
```

O executável será gerado na pasta dist/.

6 Estrutura do Código

6.1 Principais Arquivos

- `app_balanca.py`: Arquivo principal da aplicação.
- `leitura_balanca.py`: Módulo para comunicação serial (se separado).
- `icone_sartorius.ico`: Ícone da aplicação.
- `.gitignore`: Arquivo para ignorar arquivos no Git.

6.2 Classe Principal: AppBalanca

A classe `AppBalanca` herda de `ctk.CTk` e gerencia a interface gráfica.

6.2.1 Métodos Principais

- `__init__()`: Inicializa a interface e configurações.
- `criar_interface()`: Cria os elementos da GUI.
- `alternar_conexao()`: Conecta/desconecta da balança.
- `salvar_medida()`: Salva medições no CSV.
- `thread_monitoramento()`: Thread para leitura contínua.

6.3 Comunicação Serial

- Usa a biblioteca `pyserial` para comunicação.
- Configuração: 1200 baud, 7 bits, paridade ímpar, 1 stop bit.
- Comando de leitura: `\x1bP\r\n`
- Comando de tara: `\x1bf4_\r\n`

7 Desenvolvimento e Modificações

7.1 Adicionando Novas Funcionalidades

1. Analise os requisitos da nova funcionalidade e planeje as modificações necessárias.
2. Modifique a classe `AppBalanca` ou crie novos métodos conforme necessário.
3. Teste as alterações executando o script em modo desenvolvimento.
4. Verifique se a nova funcionalidade não quebra funcionalidades existentes.
5. Atualize a documentação, incluindo este manual, se aplicável.
6. Commite as mudanças no repositório Git com uma mensagem descritiva.

7.2 Debugging

- Use o painel de log para depurar.
- Adicione prints temporários para variáveis críticas.
- Verifique a porta serial com ferramentas como PuTTY.

7.3 Versionamento

- Use Git para controle de versão.
- Commite mudanças regularmente.
- Ignore arquivos temporários com `.gitignore`.

8 Troubleshooting para Desenvolvedores

8.1 Erros de Importação

- Certifique-se de que todas as dependências estão instaladas.
- Verifique a versão do Python (3.8+).

8.2 Problemas de Serial

- Teste a conexão serial com um terminal serial simples.
- Verifique se a balança responde aos comandos.

8.3 Erros de Build

- Limpe as pastas `build/` e `dist/` antes de rebuildar.
- Execute como administrador se houver permissões.

8.4 Contribuição

- Siga as boas práticas de código Python.
- Documente funções e classes.
- Teste em diferentes ambientes.