

# **METODOLOGIA MODULAR NO DESENVOLVIMENTO DE SISTEMAS DESKTOP EM PYTHON: ESTUDO DE CASO GF INFORMÁTICA**

Gustavo de Lima  
Fernando Bordin

## **1. INTRODUÇÃO**

O desenvolvimento de sistemas de gestão comercial vem se tornando cada vez mais complexo, especialmente no contexto de aplicações desktop. Esse crescimento traz um desafio recorrente: como manter a organização do código, facilitar futuras manutenções e ainda permitir que o software evolua de forma contínua? Entre as diversas abordagens existentes, a arquitetura modular tem se mostrado uma das mais eficazes para lidar com essa questão, pois permite dividir o sistema em partes menores, coesas e independentes, garantindo clareza estrutural e reduzindo o acoplamento entre os componentes.

O presente artigo apresenta o GF Informática como estudo de caso — um sistema desktop desenvolvido em Python para o gerenciamento de ordens de serviço e cadastros em assistências técnicas de computadores. Desde o início do projeto, a modularização foi adotada como princípio central, orientando a divisão do sistema em módulos específicos, como banco de dados, interface gráfica e controle de processos. Essa organização visou garantir escalabilidade, estabilidade e facilidade de evolução.

O objetivo deste trabalho é analisar de forma prática como a metodologia modular foi aplicada no desenvolvimento do GF Informática e de que maneira essa escolha contribuiu para a eficiência do código, clareza estrutural e capacidade de adaptação a novas demandas. Busca-se, assim, demonstrar os benefícios reais dessa abordagem no contexto de aplicações desktop em Python, destacando sua relevância para a criação de sistemas robustos e sustentáveis.

---

## **2. REVISÃO DE LITERATURA / FUNDAMENTAÇÃO TEÓRICA**

A modularidade é um conceito essencial na engenharia de software moderna. Em linhas simples, modularizar significa dividir um sistema complexo em partes menores, cada uma com uma função bem definida. Essa divisão facilita o desenvolvimento, os testes e a manutenção, além de tornar o código mais claro e escalável.

De acordo com Pressman (2016), sistemas modulares apresentam duas características fundamentais: alta coesão, que ocorre quando os elementos internos de um módulo estão intimamente relacionados entre si, e baixo acoplamento,

quando a dependência entre módulos é mínima. Esses princípios favorecem a reutilização de código e permitem que o software evolua sem comprometer o restante da aplicação.

A modularização também dialoga bem com metodologias ágeis e abordagens iterativas, nas quais o sistema é construído gradualmente, por meio da entrega contínua de módulos funcionais. Essa forma de desenvolvimento reduz riscos, facilita correções e garante maior flexibilidade diante de mudanças de requisitos.

No caso do Python, essa filosofia é naturalmente apoiada pela própria linguagem, que oferece suporte nativo à criação de módulos e pacotes. Isso permite que o desenvolvedor estruture o código em camadas claras, separando responsabilidades e facilitando a manutenção. O GF Informática segue exatamente essa lógica, distribuindo suas funções entre módulos de interface, regras de negócio e persistência de dados — um reflexo direto dos princípios teóricos que sustentam a engenharia de software moderna.

---

### 3. METODOLOGIA DO ESTUDO DE CASO

O estudo de caso faz parte da análise do GF Informática, um sistema desktop em Python desenvolvido para auxiliar assistências técnicas de computadores na gestão de ordens de serviço, cadastros e relatórios. A proposta foi criar uma solução leve, acessível e eficiente para pequenas e médias empresas.

O desenvolvimento foi realizado em Python puro, utilizando Tkinter para a interface gráfica e PostgreSQL como banco de dados relacional. Contudo, o grande diferencial do projeto está na adoção da arquitetura modular, que definiu toda a estrutura de pastas, nomenclaturas e padrões de comunicação entre as partes do sistema.

A organização do projeto foi planejada de forma a separar claramente as responsabilidades, seguindo a estrutura de pastas:

- **database/**: camada responsável pela persistência dos dados e comunicação com o PostgreSQL;
- **services/**: responsável pela lógica de negócio e operações CRUD;
- **ui/**: camada de interface gráfica com o usuário, construída com Tkinter;
- **utils/**: funções auxiliares e reutilizáveis, como geração de relatórios e formatações.

O desenvolvimento seguiu um processo incremental, permitindo testar cada módulo isoladamente antes de integrá-lo ao sistema principal. Essa abordagem trouxe clareza, estabilidade e maior previsibilidade na evolução do código.

---

#### 4. APlicaÇÃO DA ARQUITETURA MODULAR

A modularização do GF Informática foi aplicada desde as primeiras linhas de código, guiada pela ideia de separar responsabilidades e reduzir o acoplamento entre as partes do sistema. Essa prática resultou em um código mais limpo, de fácil manutenção e preparado para crescer.

O projeto foi dividido nos quatro módulos principais — database, services, ui e utils — todos coordenados por um arquivo principal (main.py), responsável por conectar as camadas. Essa estrutura facilitou a comunicação entre os componentes e permitiu que cada parte evoluísse sem impactar o restante.

Entre os resultados mais marcantes dessa abordagem estão:

- **Manutenção simplificada:** Alterações pontuais em um módulo não interferem nos demais, devido ao baixo acoplamento.
- **Facilidade de expansão:** Novos recursos, como controle financeiro, podem ser adicionados mantendo o padrão estrutural e a coesão.
- **Testes localizados:** Cada módulo pode ser testado individualmente, o que reduz o tempo de depuração e garante a qualidade do código.
- **Maior segurança e estabilidade:** A lógica de banco de dados é isolada da interface, evitando falhas na apresentação afetarem a integridade dos dados.

Essa estrutura não apenas organizou o projeto, mas também trouxe clareza ao processo de desenvolvimento, tornando o código mais previsível e confiável.

---

#### 5 ANÁLISE E RESULTADOS

A implementação da arquitetura modular no GF Informática mostrou resultados expressivos. Desde o início, a divisão em módulos autônomos trouxe uma visão mais clara do sistema, facilitando o entendimento das responsabilidades e reduzindo a complexidade do código.

Na prática, isso se refletiu em um código mais legível e fácil de manter. Adicionar novas funcionalidades, como o módulo de controle de técnicos, tornou-se uma tarefa simples, sem necessidade de alterar partes já consolidadas.

Outro ponto positivo foi a estabilidade alcançada: mudanças na estrutura do banco de dados não afetaram a interface gráfica ou a lógica principal. Essa independência reforçou a confiabilidade do sistema e permitiu o uso eficiente de testes unitários.

Por outro lado, o processo revelou desafios, como a necessidade de documentação clara e definição precisa das interfaces entre módulos. Esses pontos mostraram-se essenciais para manter a coesão do projeto e facilitar a colaboração futura.

De modo geral, o caso do GF Informática confirmou que a modularização é uma estratégia viável e poderosa, mesmo em projetos desktop feitos em Python puro. Ela demonstrou que não é preciso recorrer a frameworks complexos para alcançar organização e escalabilidade — basta aplicar bons princípios de engenharia de software.

---

## 6. CONCLUSÃO E PERSPECTIVAS FUTURAS

O desenvolvimento do GF Informática demonstrou, na prática, o valor da arquitetura modular como base para sistemas bem estruturados e duradouros. Essa abordagem proporcionou uma organização clara, com responsabilidades bem definidas e um fluxo de desenvolvimento mais fluido e previsível.

A modularização se mostrou essencial para reduzir o tempo de manutenção, facilitar o crescimento do sistema e aumentar a estabilidade do código. Com a estrutura atual, o projeto está preparado para receber novas funcionalidades e até mesmo evoluir para outros contextos, como a criação de uma versão web baseada em APIs com FastAPI ou Django.

Em um cenário futuro, o sistema também pode incorporar testes automatizados, injeção de dependências e uma arquitetura de plugins, permitindo personalizações e expansões sob demanda.

Mais do que uma técnica de organização, a modularização provou ser uma filosofia de desenvolvimento sustentável. O caso do GF Informática evidencia como uma boa estrutura pode transformar o modo de construir software — tornando-o mais limpo, escalável e pronto para o futuro.

---

## **REFERÊNCIAS**

LUTZ, Mark. *Programming Python*. 4th ed. Sebastopol: O'Reilly Media, 2010.

PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 8. ed. Porto Alegre: AMGH Editora, 2016.