

# Implementação de software desktop em Python

Hélder S. Frota<sup>1</sup>, Fábio S. Ruver,<sup>1</sup> Josiel M. Figueiredo<sup>1</sup>,

<sup>1</sup>Universidade Federal do Mato Grosso (UFMT) - Cuiabá, MT – Brazil

{helder.frota,Fabio.ruver}@sou.ufmt.br, josiel@ic.ufmt.br

**Abstract.** *This work presents the development of software for desktop use aimed at generating labels, using Python as the main language. The application seeks to optimize and automate the label creation and printing process, facilitating product management and control. The software uses a graphical interface and executable generator, developed with libraries such as CustomTkinter and Pyinstaller, which allows the creation of graphical interfaces and the generation of the executable file. Furthermore, the solution features Web API usage capabilities focused on automation, integrating data from ERP systems such as BLING to speed up the filling and formatting of labels, minimizing errors and saving time.*

**Resumo.** *Este trabalho apresenta o desenvolvimento de um software para uso em desktop voltado para a geração de etiquetas, utilizando Python como linguagem principal. A aplicação busca otimizar e automatizar o processo de criação e impressão de etiquetas, facilitando a gestão e controle de produtos. O software utiliza uma interface gráfica e gerador de executável, desenvolvida com bibliotecas como CustomTkinter e Pyinstaller, que permite a criação de interfaces gráficas e a geração do arquivo executável. Além disso, a solução conta com recursos de uso de API Web com foco em automação, integrando dados de sistemas ERP como a BLING para agilizar o preenchimento e a formatação das etiquetas, minimizando erros e economizando tempo.*

## 1. Introdução

Embora a maior parte das soluções e demandas de softwares atuais estejam voltadas para o desenvolvimento de software web, ainda há um nicho significativo para softwares desktop. Essas soluções atendem a demandas específicas, onde a robustez e a integração local com sistemas legados são cruciais. Mesmo sendo um software desktop, a necessidade de interação com serviços web via APIs continua a ser relevante, garantindo a integração com plataformas externas e a comunicação em tempo real. Este trabalho descreve a implementação de um software desktop utilizando Python, abordando desde a modelagem eficaz das informações até a integração com outros sistemas, envolvendo aspectos relacionados com a interface, a modularização e o sistema operacional de execução.

O contexto envolvido neste trabalho é de um ambiente real relacionado com a cadeia têxtil, que tem tido uma crescente demanda por rastreabilidade e a automação de processos surge como uma solução primordial nesse cenário, pois permite não apenas a integração fluida de informações, mas também a geração precisa de informações que permitem acompanhar o trajeto do algodão até o consumidor final, permitindo que o mesmo possa pela etiqueta de sua roupa, saber todas as informações de rastreabilidade.

A integração de sistemas, incluindo a utilização de tecnologias modernas como sistemas ERP e plataformas de integração, como Bling, torna-se um ponto crucial. Através do uso de APIs, visamos a manipulação eficiente e segura de dados, garantindo sua qualidade, segurança e conformidade com as regulamentações aplicáveis.

## **2. Metodologia**

Diferentemente de um software web no qual é possível separar o código fonte entre front-end e back-end, até mesmo com linguagens diferentes, um software desktop normalmente é implementado em uma única linguagem. Dessa forma, a modularização, versionamento e a interface são elementos essenciais para a construção do software proposto, visando não apenas a eficiência operacional, mas também a escalabilidade e flexibilidade necessárias para acompanhar as demandas em constante evolução.

Assim a linguagem Python foi escolhida, principalmente pela facilidade de empacotamento do software para gerar um executável, o que facilita sua distribuição e uso em ambientes desktop sem a necessidade de instalações adicionais de dependências, tornando a solução mais acessível e prática para o usuário final.

### **2.1. Aquisição dos Dados**

Uma gestão eficiente é fundamental para garantir sua competitividade e sustentabilidade no mercado atual. Nesse contexto, sistemas de Planejamento de recursos empresariais (ERP) desempenham um papel crucial ao integrar e automatizar processos em diferentes áreas organizacionais, proporcionando uma visão abrangente e atualizada das operações empresariais (SHEHAB, 2004).

A comunicação entre o sistema e as plataformas foi por meio de Interface de Programação de Aplicação Web (API-Web) que já se tornou padrão na integração de sistemas e na disponibilização de serviços e dados entre diferentes plataformas e aplicações. (KLAMT, 2011).

### **2.2. Bling**

O Bling [www.bling.com.br](http://www.bling.com.br) é um sistema ERP (Enterprise Resource Planning) brasileiro voltado para a gestão integrada de pequenas e médias empresas. Ele oferece uma ampla variedade de funcionalidades, que incluem controle de estoque, emissão de notas fiscais, gerenciamento de pedidos, fluxo de caixa, compras, vendas, e integração com marketplaces e e-commerce. Essa integração facilita a administração centralizada de operações comerciais, reduzindo a necessidade de múltiplos sistemas e otimizando processos internos. O software Bling facilita a emissão de notas fiscais, boletos, gerência de estoque, ordens de compra etc, além de realizar integrações com as principais plataformas de e-commerce e marketplaces.

O Bling também possui recursos de automação, como a sincronização automática de dados com diversas plataformas, como lojas virtuais e transportadoras. A interface do sistema é intuitiva, e a plataforma é configurável, permitindo que as empresas adaptem o ERP às suas necessidades específicas. Além disso, oferece integração via API, possibilitando a conexão com outras ferramentas e sistemas, o que contribui para a melhoria do fluxo de trabalho e a produtividade empresarial.

## 2.3. Biblioteca CustomTkinter

A criação da interface desktop em Python utilizando a biblioteca CustomTkinter permite o desenvolvimento de GUIs (Interfaces Gráficas do Usuário) intuitivas e modernas. O CustomTkinter foi escolhido por ser uma extensão popular do Tkinter, amplamente utilizada entre desenvolvedores de software, além de contar com uma comunidade ativa e vasta documentação disponível.

Essa biblioteca oferece uma estética mais atual e personalizável, mantendo a simplicidade do Tkinter, o que a torna acessível tanto para iniciantes quanto para desenvolvedores experientes que buscam criar interfaces gráficas sofisticadas sem complexidade excessiva.

Com ela, foi possível criar janelas, botões, campos de entrada e outros elementos de forma simples e eficiente, graças à sua sintaxe amigável e integração perfeita com o Python. A documentação detalhada, disponível no site oficial <https://customtkinter.tomschimansky.com/>, facilita o aprendizado e acelera o desenvolvimento de interfaces modernas em projetos Python. A Figura 1 ilustra um exemplo complexo de GUI desenvolvida utilizando a biblioteca CustomTkinter.

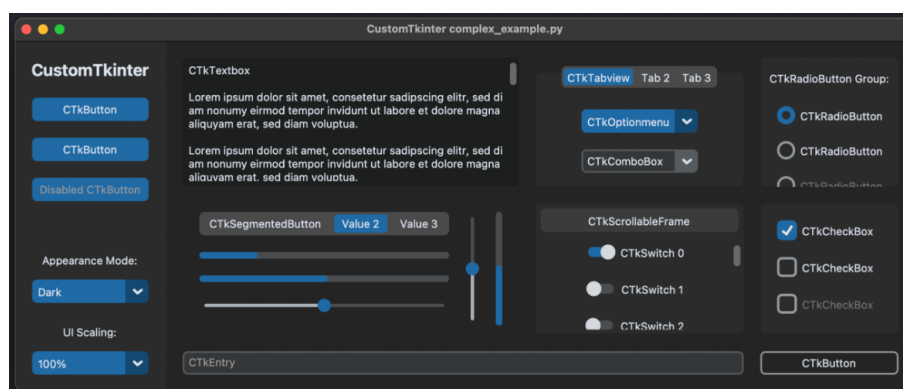


Figure 1. Exemplo GUI da Biblioteca CustomTkinter

## 2.4. Empacotamento

Todo software executável em um ambiente desktop é um arquivo em código objeto, já entendido e processado diretamente pelo sistema operacional. No entanto, como o Python é uma linguagem interpretada, o código fonte precisa ser interpretado em tempo de execução, o que normalmente requer a instalação do interpretador Python. Para evitar essa dependência, é necessário gerar um código objeto a partir do script Python, e isso pode ser feito através de ferramentas como o PyInstaller.

O processo de empacotamento com o PyInstaller é flexível e funciona tanto em Windows quanto em Linux. Essa ferramenta cria um executável que encapsula todas as bibliotecas e recursos necessários, garantindo que o software possa ser executado de maneira eficiente em diversos ambientes, sem a necessidade de ter o Python instalado. No Windows, o PyInstaller gera um arquivo .exe, facilitando a execução do software com um simples clique. Para empacotar o software, siga os passos abaixo:

```
pip install pyinstaller
```

```
pyinstaller --onefile nome_do_script.py
```

O parâmetro `--onefile` gera um único executável, encapsulando todos os módulos e dependências necessários. Caso deseje que o console não seja exibido durante a execução do software, adicione o parâmetro `--noconsole` antes do nome do script.

### 3. Resultados e discussões

#### 3.1. Interface gráfica

A Figura 2, Figura 3 e Figura 4 mostram a interface criada para o software e suas seguintes abas. Pode-se observar como os índices são representados e como estão dispostos, ficando claro o tipo de dados utilizado.

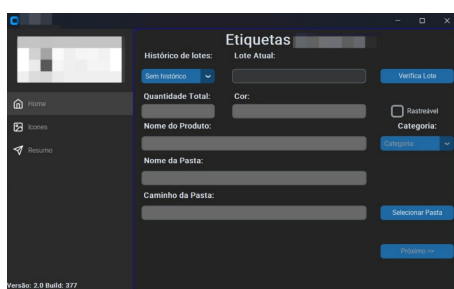


Figure 2. Interface criada para o software.

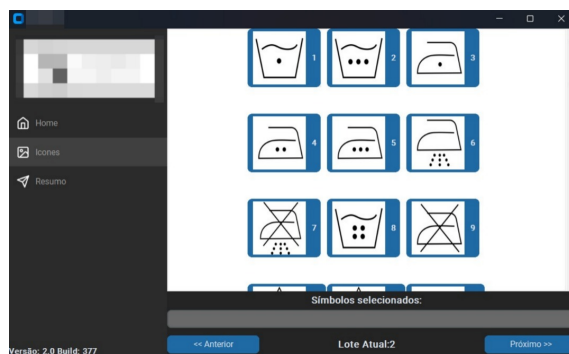


Figure 3. Seleção de ícones da ABNT ISO 3758 2013.

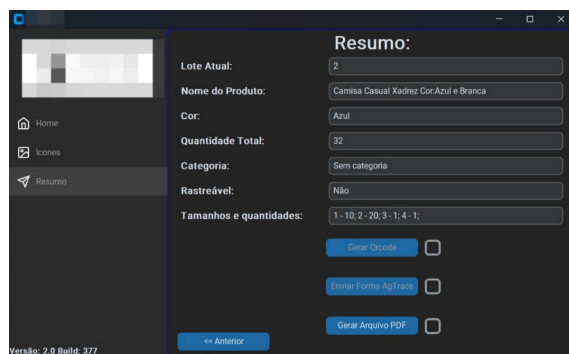


Figure 4. Resumo para conferir os dados.

Ao final do processo, após uma série de escolhas realizadas pelo usuário, o sistema gera uma etiqueta no formato PDF, contendo informações obtidas através das buscas realizadas na API, assim como as preferências selecionadas pelo usuário na interface.

#### **4. Conclusão**

O desenvolvimento do software para geração de etiquetas em Python, utilizando CustomTkinter e PyInstaller, provou ser uma solução eficiente para otimizar processos. Com uma interface gráfica e integração com a Bling, o software facilita a criação de etiquetas, minimizando erros. A geração de executáveis torna-o acessível a usuários sem conhecimentos técnicos avançados.

Como trabalhos futuros, pretende-se implementar outros processos internos da empresa envolvendo aspectos de controle de estoque.

#### **5. Agradecimentos**

Agradecemos à FAPEMAT e à UFMT pelas bolsas de iniciação científica do Programa PIBIC recebidas pelos autores Helder Schneider Frota e Fábio Schmitz Ruver.

As referências a seguir foram utilizadas: [Foundation 2023], [Bling 2023], [Shehab et al. 2004], [Klamt and Von K. 2011].

#### **References**

Bling (2023). Api reference documentation.

Foundation, P. S. (2023). Python documentation.

Klamt, S. and Von K., A. (2011). An application programming interface for cellnetanalyzer. *Biosystems*, 105(2):162–168.

Shehab, E. M., Sharp, M. W., Supramaniam, L., and Spedding, T. A. (2004). Enterprise resource planning: An integrative review. *Business Process Management Journal*, 10(4):359–386.