

## **Projeto Final - Avaliação A3**

### **PROJETO DE PROTEÇÃO CONTRA GOLPES NO PIX**

Relatório técnico apresentado na UC  
Sistemas Distribuídos e Mobile orientado  
pelo Prof. MSc Flávio Henrique da Silva.

**Curitiba**

**2025**

**INTEGRANTES DO GRUPO**

Carlo Augusto Viel - 172221510

Gabriel Gomes Fernandes - 172317728

Gabriel Luiz Klein - 172312555

Mateus Morais Lima – 172220483

1 INTRODUÇÃO .....	4
2 DESENVOLVIMENTO .....	4
2.1 Divisão das Tarefas .....	5
2.2 Estrutura do Projeto .....	5
2.3 Explicação da Aplicação/Software.....	5
2.4 Orientações de execução da Aplicação/Software .....	8
2.5 Repositório .....	8
3 CONCLUSÃO .....	8
REFERÊNCIAS.....	9

## 1 INTRODUÇÃO

O avanço da tecnologia tem impactado diretamente a forma como lidamos com transações financeiras, principalmente com o uso do Pix, que se tornou um dos meios de pagamento mais utilizados no Brasil. No entanto, com essa popularização, também cresceram os casos de golpes e fraudes envolvendo pagamentos via Pix. Pensando nisso, o projeto foi desenvolvido com o objetivo de criar uma aplicação simples e funcional que permita cadastrar e verificar chaves Pix, ajudando a identificar se uma chave é confiável, suspeita ou ainda não está cadastrada.

A ideia surgiu dentro do Programa Dual do Bradesco, como uma forma prática de aplicar os conhecimentos adquiridos em sala de aula, como lógica de programação, manipulação de banco de dados. Ao longo do desenvolvimento, foi possível trabalhar em equipe, enfrentar desafios reais de codificação e reforçar conceitos importantes da área de tecnologia.

Neste relatório, apresentamos todas as etapas do projeto, desde a ideia inicial até a construção da aplicação final, explicando o funcionamento do sistema, as ferramentas utilizadas e os principais aprendizados durante o processo. A proposta, além de servir como exercício prático, também busca uma solução simples para um problema real: aumentar a segurança nas transações com Pix.

## 2 DESENVOLVIMENTO

Esse projeto foi feito com a ideia de criar um programa simples em Python que ajuda a cadastrar e verificar chaves Pix, pensando na segurança das transações. A gente usou a biblioteca Tkinter para criar uma interface gráfica amigável, e o SQLite como banco de dados local para armazenar as informações.

Ao longo do relatório, explicamos o porquê de escolher esse tema, quais eram os objetivos do projeto e o que a gente queria resolver com ele. Depois disso, mostramos como o código foi estruturado, quais arquivos e funções principais existem, e como tudo se conecta pra fazer o sistema funcionar.

Também tem uma parte explicando como usar o programa — como cadastrar uma nova chave Pix e como verificar se ela é confiável, suspeita ou não cadastrada. A gente comenta ainda sobre os desafios que enfrentamos durante o desenvolvimento, como

lidar com algumas funções que não estavam funcionando como esperado e o trabalho de organizar tudo em uma interface só.

Por fim, mostramos o que conseguimos alcançar com o projeto e deixamos algumas ideias para melhorias no futuro, como adicionar mais opções ao sistema ou integrá-lo com uma base de dados online.

## 2.1 Divisão das Tarefas

A equipe fez a seguinte divisão.

Carlo Viel e Gabriel Fernandes ficaram com a parte do código e o Gabriel Klein e o Mateus Lima ficaram com a parte do relatório.

## 2.2 Estrutura do Projeto

O projeto foi feito em Python, usando a biblioteca Tkinter para criar a interface gráfica e o SQLite3 como banco de dados. A ideia principal era permitir o cadastro e a verificação de chaves Pix, separando tudo em arquivos para deixar o código mais organizado.

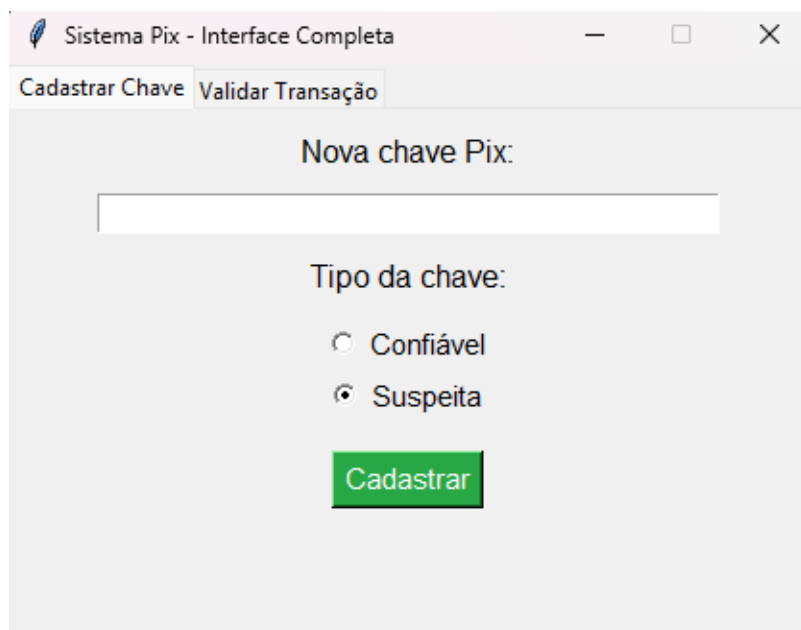
No arquivo main.py fica a tela principal, onde o usuário digita a chave Pix e o sistema verifica se ela é confiável, suspeita ou não cadastrada.

O verificar.py faz essa verificação no banco de dados. O db\_config.py cuida da conexão com o banco e cria a tabela, se ainda não existir. Já o cadastro.py tem uma tela separada para cadastrar novas chaves Pix.

As bibliotecas que usamos foram: tkinter para a interface, sqlite3 para o banco de dados local e logging para registrar os eventos do sistema. Tudo foi pensado para ser simples, funcional e fácil de manter, com cada parte do código cuidando de uma tarefa específica.

## 2.3 Explicação da Aplicação/Software

- **Funcionalidade da aba principal:** Ao abrir o programa, a janela principal será exibida com o título "Validador de Pix". Nela, o usuário encontrará um campo para digitar a chave Pix e um botão "Verificar".



Sistema Pix - Interface Completa

Cadastrar Chave Validar Transação

Nova chave Pix:

Tipo da chave:

☐ Confiável

☒ Suspeita

Cadastrar

### Funcionalidade da aba Validar Transação:

O usuário deve digitar uma chave PIX (CPF, telefone, e-mail, etc.) no campo disponível.

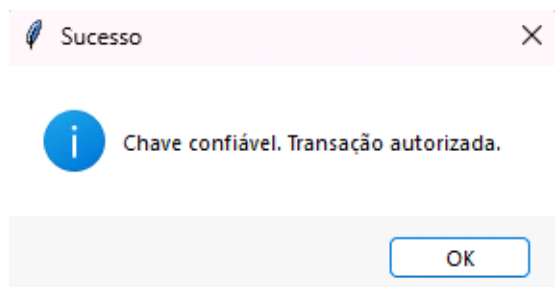
Ao clicar no botão “Verificar”, o sistema consulta o banco de dados para identificar o status da chave.

Os possíveis resultados são:

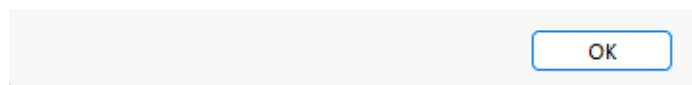
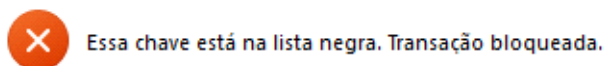
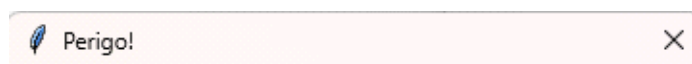
Confiável: Mensagem positiva autorizando a transação. (Imagem 01)

Suspeita: Alerta bloqueando a transação. (Imagem 02)

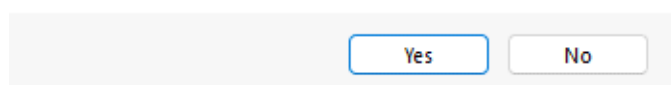
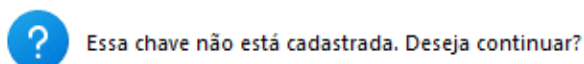
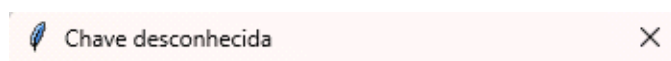
Não Cadastrada: Pergunta se o usuário deseja continuar mesmo assim. (Imagem 03)



(Imagem 01)



(Imagem 02)



(Imagem 03)

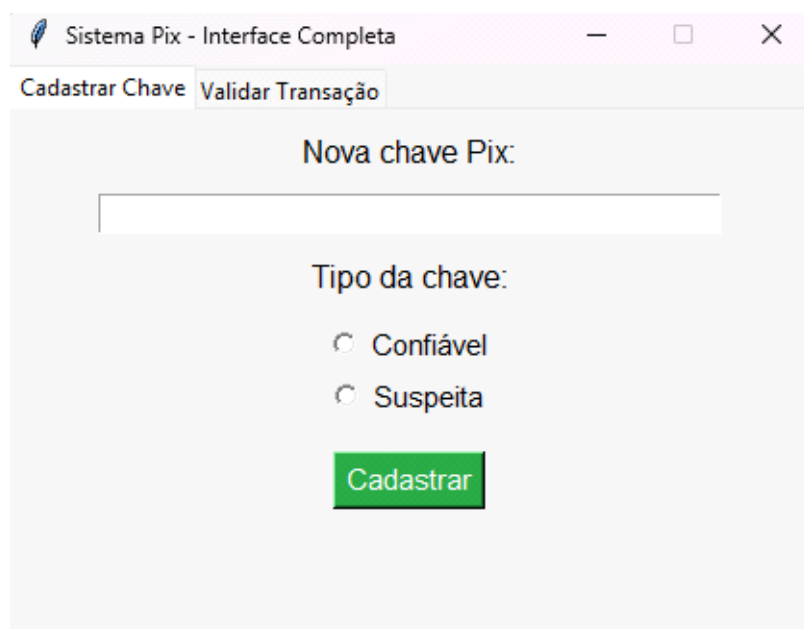
### Funcionalidade aba Cadastrar Chaves:

Caso o usuário deseje cadastrar uma nova chave Pix:

Deve abrir o arquivo cadastro.py (ou clicar em um atalho, se existir).

A janela de cadastro será exibida com os campos:

- Entrada para digitar a chave Pix.
- Seleção do tipo (Confiável ou Suspeita).
- Botão "Cadastrar".



## Mensagens do Sistema

Todas as ações geram mensagens na tela, informando se a chave foi cadastrada com sucesso, se já existia ou se houve algum erro durante o processo.

Além disso, o sistema registra todas as verificações e cadastros em um arquivo de log (logs.txt), que pode ser consultado posteriormente para análise de uso ou auditoria.

## 2.4 Orientações de execução da Aplicação/Software

Para executar o sistema de verificação e cadastro de chaves Pix, é necessário ter o Python instalado na máquina, preferencialmente a partir da versão 3.10. O primeiro passo é clonar ou baixar o projeto.

Com os arquivos em mãos, recomenda-se abrir o projeto em um editor de código como o Visual Studio Code, embora outros editores também possam ser utilizados. Em seguida, é necessário criar o banco de dados que armazenará as chaves Pix. Isso pode ser feito executando o arquivo db\_config.py no terminal, usando o comando `python db_config.py`. Esse passo criará automaticamente o arquivo pix.db e a tabela necessária para armazenar as informações.

O projeto foi desenvolvido apenas com bibliotecas nativas do Python, como tkinter para a interface gráfica, sqlite3 para o banco de dados e os para manipulação de diretórios. Sendo assim, não é necessário instalar dependências adicionais com o pip.

Após esses passos, basta executar o arquivo principal do sistema com o comando `python main.py`. Isso abrirá a interface gráfica da aplicação, onde é possível verificar e cadastrar chaves Pix de forma simples e prática.

## 2.5 Repositório

<https://github.com/carloviel/TrabalhoA3#>

## 3 CONCLUSÃO

O desenvolvimento deste projeto proporcionou uma experiência prática valiosa na aplicação dos conhecimentos adquiridos na Unidade Curricular de Sistemas Distribuídos e Mobile em parceria com o projeto Dual Bradesco. Ao longo do processo, foi possível planejar e implementar uma aplicação que permite o cadastro e a verificação de chaves Pix, contribuindo para a segurança de transações financeiras simuladas. A construção do sistema envolveu a criação de uma interface gráfica amigável, a manipulação de banco de dados e o uso de tratamento de erros para garantir sua estabilidade.



Com base na proposta inicial, foi possível atingir os objetivos esperados e entender melhor a importância de uma estrutura de código bem definida. A análise e execução do projeto também mostraram como a programação pode ser aplicada em situações reais, reforçando o papel da tecnologia como ferramenta de apoio à segurança digital. Por fim, este trabalho abre caminhos para possíveis melhorias futuras, como a integração com bancos de dados remotos, o uso de autenticação e o refinamento da interface para dispositivos móveis.

## REFERÊNCIAS

DEVMEDIA. Tkinter: Interfaces Gráficas em Python.

<https://www.devmedia.com.br/tkinter-interfaces-graficas-em-python/33956>

ChatGPT – Assistente do relatório

<https://chatgpt.com/>

REAL PYTHON. Logging in Python.

<https://realpython.com/python-logging/>