



Organização Projeto-A3-GQS

Apresentação do Projeto A3-GQS: Sistema Bancário

Participantes:

- João Luiz da Silva - 82420546
- Gabriel Fernandes Bispo - 823143267
- Pedro Zorzete Brisolla de Campo - 823147399
- Pedro Henrique Braz Lira - 823117291



Contribuição da Organização

Projeto-A3-GQS



Repositório GitHub

Criação do A3-Sistema-Bancario
e configuração inicial.



Código Legado

Inclusão do código original para
refatoração de Clean Code.



Merge Inicial

Integração main e develop para histórico consolidado.

Contribuição de Pedro Braz no Código

Refatoração de Classes

Refatorou Cliente e PessoaFisica em app.py, adicionando type hints e List["Conta"].



Modularização

Extração de Cliente, Erro e PessoaFisica para src/dominio.

Exceção de Domínio

Implementou RegraDeNegocioError para sinalizar violações de regras de negócio.



Projeto Reorganizado

Estrutura mais clara, app.py menos carregado, lógica de domínio em módulos.

Encapsulamento

Atributos privados e acesso via @property para entidades de cliente.



Classe de Erro Dedicada

Consolidação do tratamento de erros de regra de negócio em erro.py.

Ajustes Finais

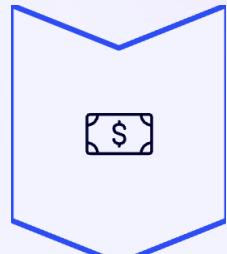
Corrigiu mensagens de erro e formatou datas no histórico de transações.



Merge de PR

Integração da refatoração de usuário ao branch develop.

Contribuição de Gabriel Bispo no Código



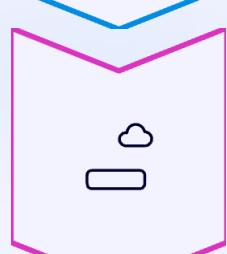
Refatoração de Contas

Melhoria da coesão em Conta e Historico, ajuste de acesso ao histórico.



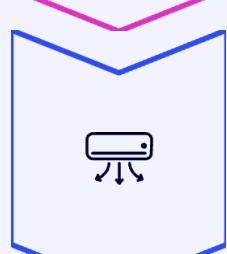
Modularização de Operações

Criação de operacoes/contaCorrente.py, extraindo lógica de limites.



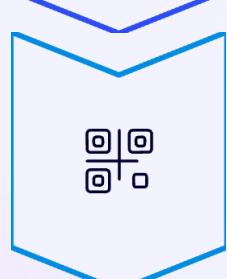
Liderança Arquitetural

Modularização de operações para pacotes independentes de transações e contas.



Testes Unitários

Criação da pasta tests com arquivos de teste e configuração do pytest.ini.



Remoção de Arquivos Antigos

Consolidação da nova estrutura modular e testável.

Contribuição de Pedro Zorzete no Código



Criação de Pacotes e Lógica de Transações

Criação do pacote 'operacoes' (Transacao, Deposito, Saque) e implementação da lógica para registrar transações bem-sucedidas.



Modularização e Ajustes de Importações

Finalização da modularização geral do sistema, garantindo a organização e o correto funcionamento das dependências.



Merge e Integração de Código

Realização do merge da PR "refatorar-operacoes", integrando importantes melhorias ao projeto.



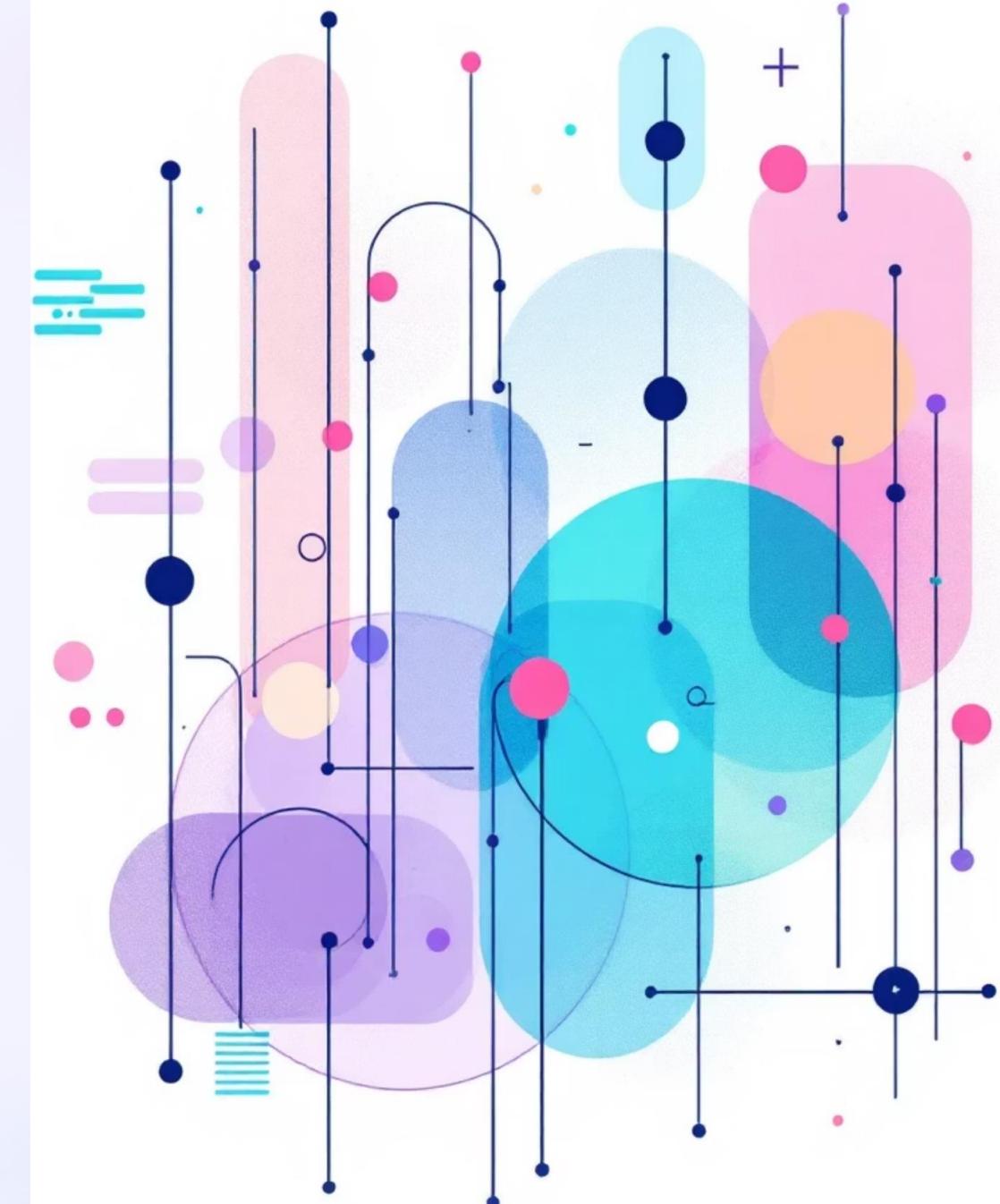
Revisão e Documentação

Atuação na revisão de código, correção de problemas e elaboração do README oficial do projeto.



Manutenção Estrutural

Garantia da consistência estrutural do código através de manutenções contínuas.



Contribuição de João Luiz da Silva no Código

- Limpeza final do arquivo de interface src/app/cli.py
- Deixou o arquivo de CLI mais enxuto e legível
- Contribuição na etapa de polimento da aplicação

Conclusão

As contribuições para a refatoração e a aplicação dos princípios de Clean Code foram cruciais para aprimorar o projeto. Este processo não só elevou a **qualidade do código**, tornando-o mais robusto e eficiente, mas também melhorou significativamente a **manutenibilidade** e a **legibilidade**. A **estrutura modular** do sistema bancário foi reforçada, facilitando futuras atualizações e a colaboração da equipe.

O trabalho colaborativo e a dedicação em seguir as melhores práticas de desenvolvimento, incluindo a refatoração contínua, são essenciais para construir e manter sistemas complexos e de alta performance.

