**Iury Alves Pedrosa**

**1. Garantia de Qualidade (QA)**

**Pergunta:** Explique o que você entende por Garantia de Qualidade (QA) e qual a sua importância no desenvolvimento de software.

R: Quando se fala em software, estamos nos referindo a um conjunto de algoritmos, que por muitas vezes, executar processos matemáticos complexos e extensos. O processo de criação de software evoluiu bastante até os dias do hoje, onde os programadores gozem de uma gama de ferramentas para construir software a nível comercial e industrial, porém, nem sempre foi assim, visto que a poucos anos, a construção de algoritmos era mais rudimentar e bem menos complexa, sendo possível destrinchar um software inteiro e visualizar todos os seus passos. No entanto, nos dias de hoje, com reuso de códigos e surgimento de frame works, não é mais possível, em pouco tempo, entender o funcionamento de um algoritmo, acarretando em problemas profundos e de difícil acesso. Em razão disso, ferramentas de testes e de qualidade surgiram para aprimorar e dar confiabilidade ao desenvolvimento de softwares.  
  
Segundo Pressman, Garantia de Qualidade envolve um conjunto de atividades projetadas para avaliar o processo pelo qual os produtos são desenvolvidos ou fabricados. No contexto do desenvolvimento de software, isso significa garantir que os processos e procedimentos estão em vigor e são seguidos durante o ciclo de vida do software, resultando em produtos de alta qualidade.

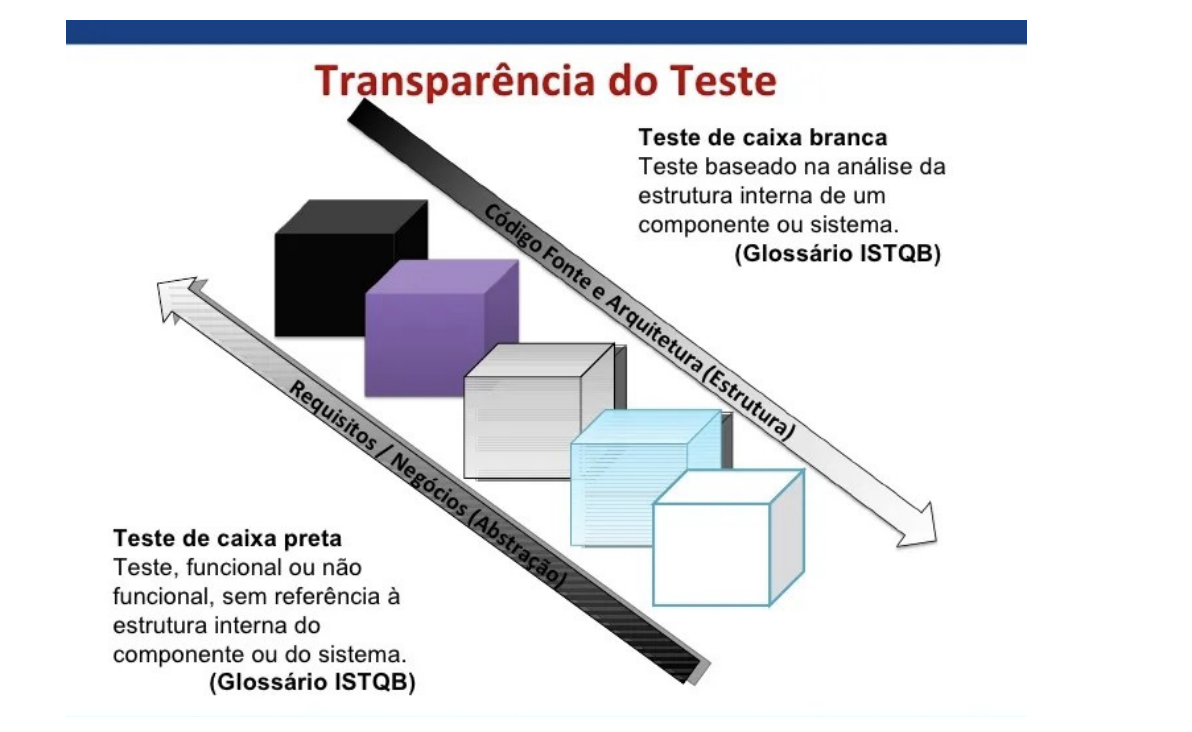
Pressman destaca a importância de métricas para a garantia da qualidade em todas as fases do desenvolvimento, incluindo análise, design, codificação, teste e manutenção. Para ele, um software de qualidade não é apenas aquele que está livre de defeitos, mas também aquele que atende ou excede as expectativas dos clientes e usuários finais em termos de desempenho, confiabilidade e usabilidade.

**2. Testes de Caixa Branca e Caixa Preta  
Pergunta:** Diferencie testes de caixa branca e testes de caixa preta.

R: Caixa Branca refere-se á um teste “transparente”. Possui acesso ao código fonte, conhecendo a estrutura interna do produto. Os níveis (estratégias) de teste caixa preta são de Unidade. Sendo analisados e possibilitando que sejam escolhidas partes específicas de um componente para ser avaliados, permitindo uma

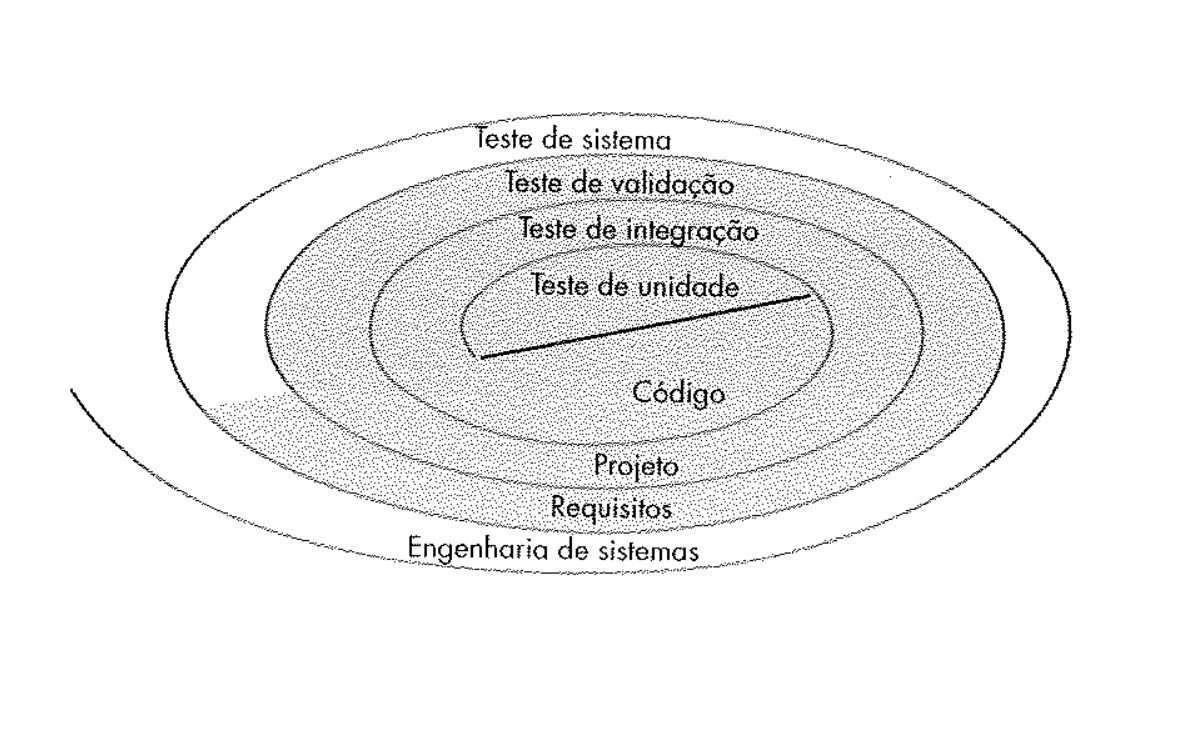
busca precisa do comportamento da estrutura.

Já os testes Caixa Preta, baseiam-se nos requisitos básicos do software, sendo o foco nos requisitos da aplicação, ou seja, nas ações que deve desempenhar. Os níveis (estratégias) de teste caixa preta são: Integração, Sistema, Aceitação. Se resumem em testes de entrada e saída.



**3. Testes de Regressão  
Pergunta:** O que são testes de regressão? Por que são importantes?

R:O teste de regrassão está dentro da estratégia de testes de Integração.



Visa construir, sistematicamente a estrutura do programa enquanto testes são feitos para descobrir erros. Sendo assim, o teste de Regressão é a reexecução de algum subconjunto de testes que já foi guiado, pois a chegada de novos componentes ao programa pode resultar em erros que antes funcionavam.

**4. Automação de Testes em Aplicações Streamlit  
Pergunta:** Como você abordaria a automação de testes para uma aplicação

Streamlit? Quais ferramentas você usaria?

R: Em que pese minha experiência profissional com python tenha envolvido apenas a construção de automações de nível intermediário com dataframes e do lado do back end, após pesquisa entendi que o python tem uma ferramenta nativa chamada AppTest, que é um framework de testes automatizados para aplicações Streamlit.

**5. Importância da Documentação em QA**

**Pergunta:** Por que a documentação é importante no processo de Garantia de Qualidade (QA)? Como a documentação de testes contribui para o sucesso de um projeto de software?

R: É documentação é crucial para garantir a qualidade e continuidade perene do software, pois fornece a base de compreensão de um software. Além disso, serve para substituição fácil de desenvolvedores responsáveis pelo software, bem como para clareza ao explicar o funcionamento para usuários. A falta de documentação é um dos principais causadores de falhas em programas.

**Parte 2: Habilidades Práticas**

**1. Revisão de Código**

**Instrução:** Abaixo está um trecho de código Python usado em uma aplicação Streamlit. Identifique qualquer erro ou problema potencial que possa afetar a qualidade ou desempenho da aplicação.

*import streamlit as st*

*def calculate\_interest(principal, rate, time):*

*# Suponha que esta função tenha uma lógica complexa*

*return (principal \* rate \* time) / 100*

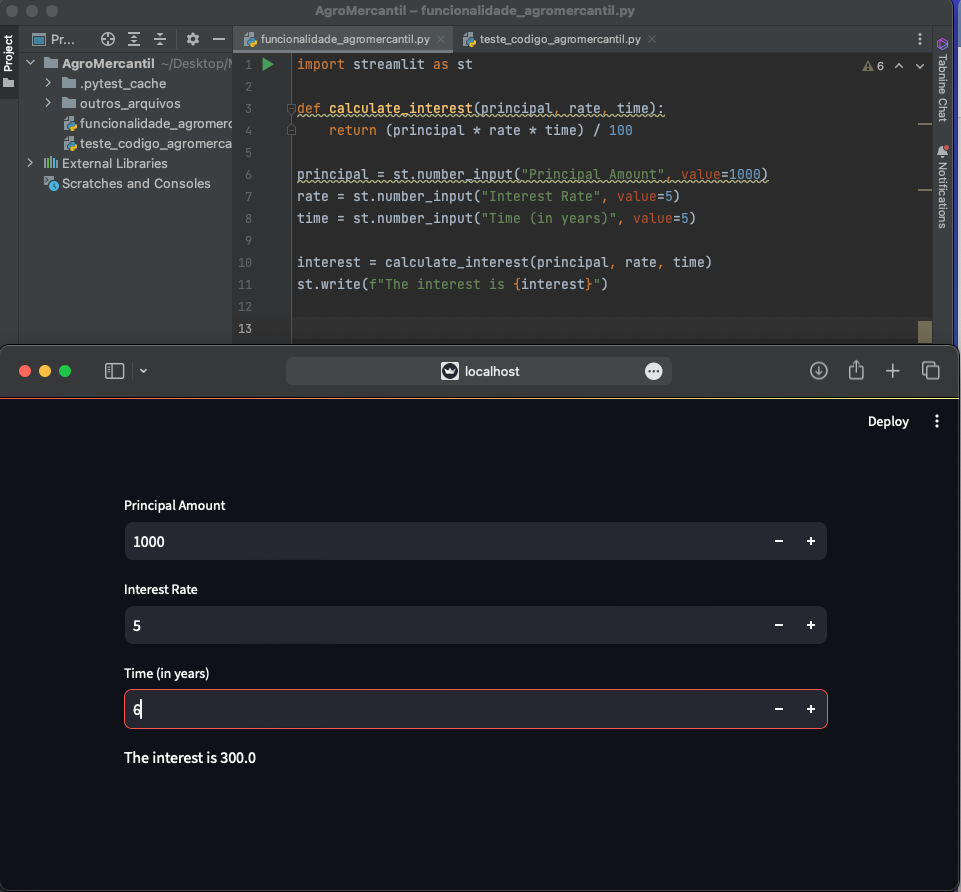
*principal = st.number\_input("Principal Amount", value=1000)*

*rate = st.number\_input("Interest Rate", value=5)*

*time = st.number\_input("Time (in years)", value=1)*

*interest = calculate\_interest(principal, rate, time)*

*st.write(f"The interest is {interest}")*

**

**2. Cenário de Teste**

**Instrução:** Escreva um cenário de teste detalhado para a funcionalidade de cálculo de juros demonstrada no trecho de código acima. Inclua os passos para a execução do teste, os dados de entrada necessários e os resultados esperados.

R: Cenário de teste de Taxa de Juros com casas decimais.

Cenário pensado para garantir que o algoritimo saiba calcularo juros corretamente com casas decimais, visto que é uma pratica recorrente no calculo de júros.   
  
**3. Automação de Teste**

**Instrução:** Com base no cenário de teste que você escreveu na questão anterior, crie um script de automação de teste usando uma ferramenta de sua escolha, adequado para uma aplicação Streamlit.

R:

*def test\_casas\_decimais():  
 principal = 1000  
 rate = 5.5 #entrada com casas decimais  
 time = 1.6  
  
 # O valor esperado  
 resultado\_esperado = 88  
  
 # Chama a função com os valores de ponto flutuante  
 juros\_calculados = funcionalidade\_agromercantil.calculate\_interest(principal, rate, time)  
  
 assert juros\_calculados == resultado\_esperado  
"""  
Teste de casas decimais executando corretamente  
"""*

**4. Criação de Documentação de Teste**

**Instrução:** Considerando o cenário de teste que você elaborou para a funcionalidade de cálculo de juros em uma aplicação Streamlit, crie um exemplo de documentação de teste. Essa documentação deve incluir:

Uma descrição do objetivo do teste.

Pré-requisitos ou configurações necessárias antes de executar o teste. Passos detalhados para a execução do teste.  
Dados de entrada para o teste.  
Resultados esperados.

Critérios para passar/falhar no teste.

R:

Objetivo do Teste:

Testar o calculo de juros com casas decimais.

Pré requisitos:

Ambiente python.

Pytest importado no código.

Import da função no arquivo de teste.

Passos do teste:

No terminal, na pasta do arquivo, inserir o comando pytest teste\_codigo\_agromercantil.py

Dados para a entrada:

Algum, ou todos os dados com casas decimais.

Ex: 1000; 5.5; 1.6.

Resultado esperado:

88

No teste em si o critério de falha e sucesso depende do resultado do calculo matemático.  
  
**Parte 3: Resolução de Problemas 1. Investigação de Bug**

**Instrução:** Imagine que um usuário relata um bug em uma aplicação Streamlit, onde o cálculo de juros não está considerando corretamente o valor do tempo (anos) inserido. Descreva os passos que você seguiria para reproduzir, investigar e sugerir uma correção para o problema.

R:

Primeiro eu ia reproduzir o erro que o usuário relatou, após isso iria ler a documentação da funcionalidade, verificando que o erro não é previsto e nem documentado, iria executar debug no código para encontrar o local de erro.

**2. Análise de Documentação de Teste Existente**

**Pergunta:** Imagine que você recebeu uma documentação de teste de uma aplicação Streamlit para revisão. Quais aspectos você avaliaria para garantir que a documentação é clara, completa e útil para os testadores? Descreva como você abordaria a melhoria dessa documentação.

R:

Eu avaliaria uma boa documentação a partir de boa formatação do texto, clareza nas explicações, ilustrações, bem como, que seja uma documentação sucinta. Além disso, deixaria claro o erro que está ocorrendo, como o encontrei, qual ambiente e dados de acessos eu utilizei.