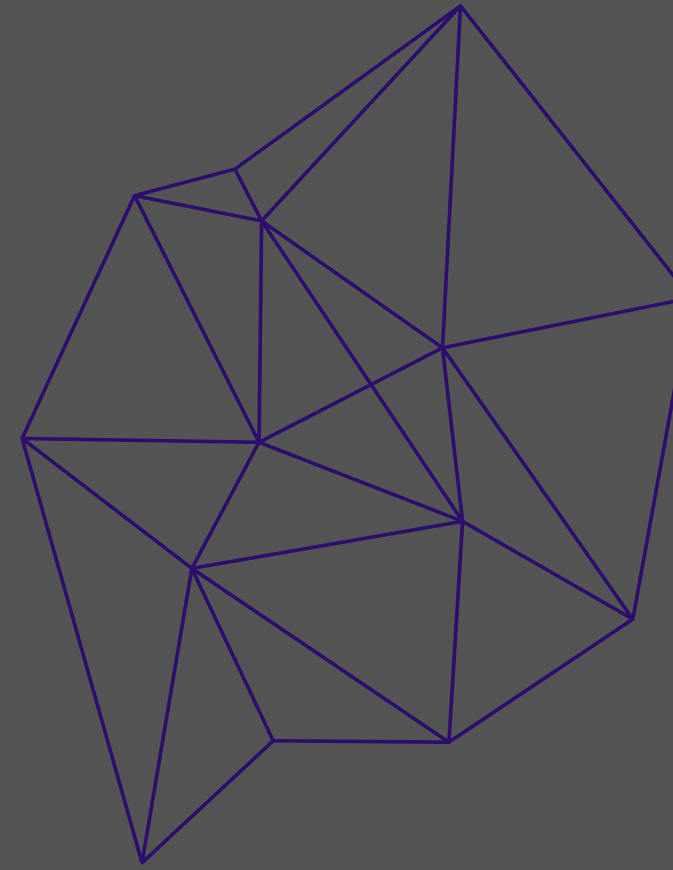


2ª Trilha

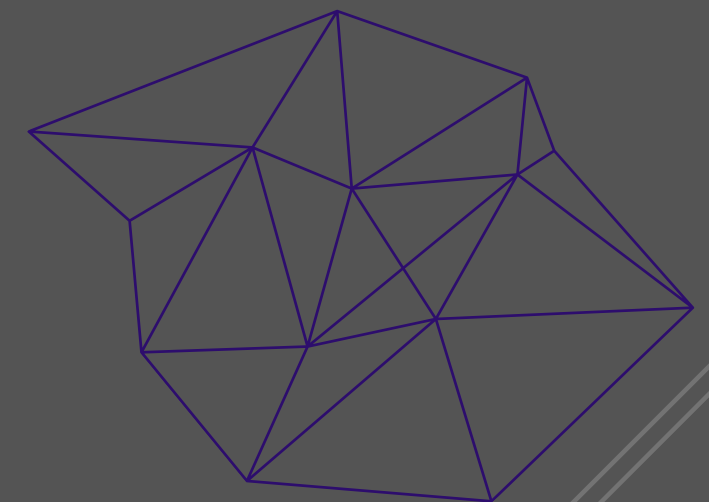
DevOps



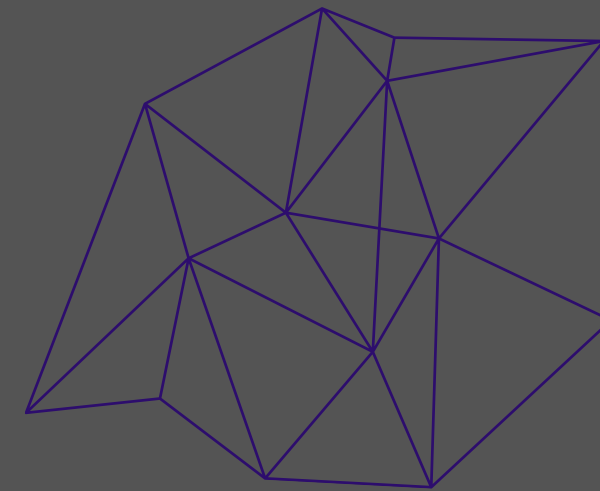
Pedro Amaral

Gerência de Configuração

- > Sistema de Controle de Versão
- > Sistema de Controle de Modificação
- > Sistema de Gerenciamento de Construção



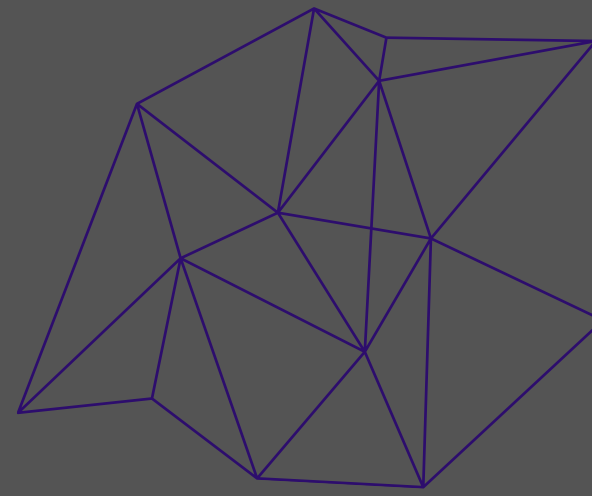
Sistema de Controle de Versão



Evita conflitos no software, como sobrescrita, duplicação e perda de linhas de código.

Exemplos de Ferramentas: Github, Gitlab, BitBucket etc...

Sistema de Controle de Modificação



Armazena informações coletadas das solicitações de modificações e repassa para aqueles que estão autorizados, como por exemplo, os participantes que iram realizar a modificação.

Exemplos de Ferramentas: Bugzilla, Jira etc...

Sistema de Gerenciamento de Construção

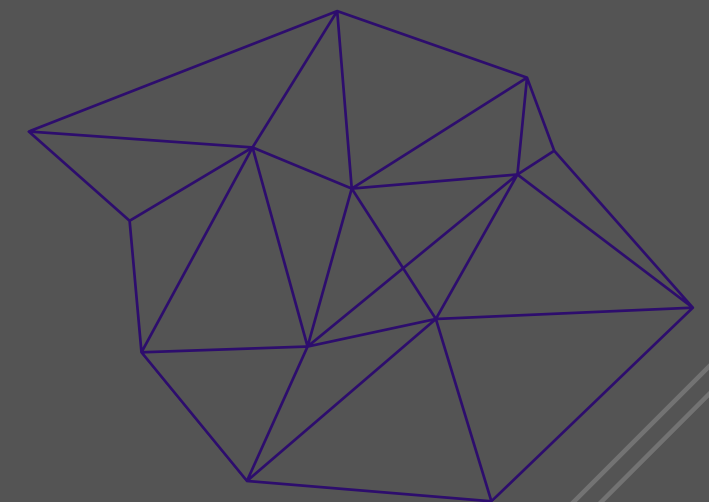
Automatiza, testa e empacota a aplicação.

Como por exemplo um código em Java para um arquivo .jar

Exemplos de Ferramentas: Jenkins, Apache Ant, Maven, etc...

Testes de Software

- > Teste de Unidade, Integração, Funcional, P2P, Aceitação, Desempenho, Fumaça



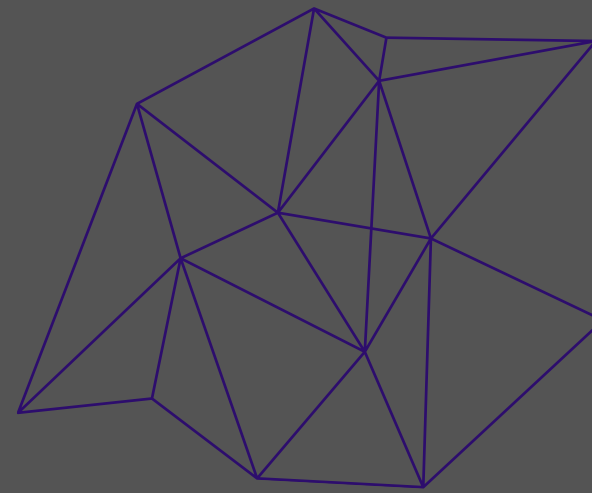
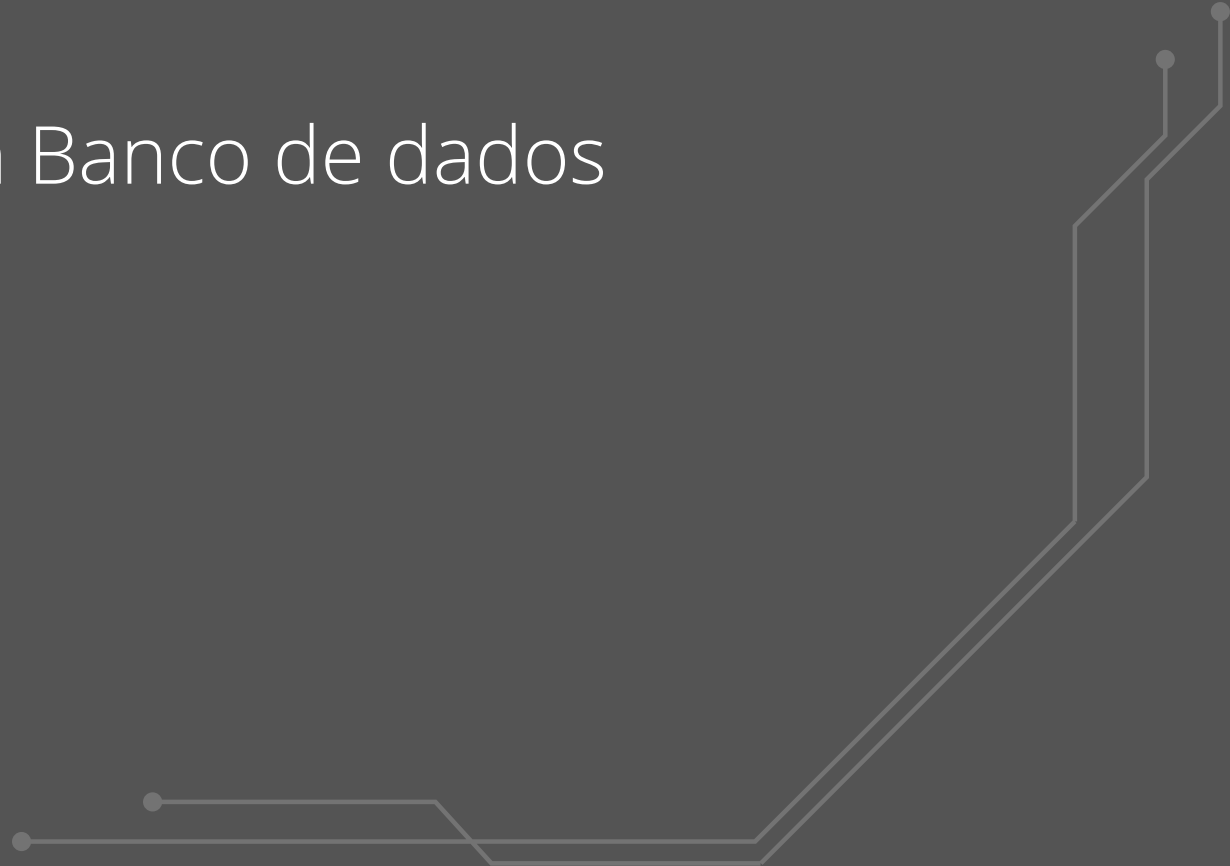


Teste de Unidade

Teste de métodos e funções de forma isoladas;
Custo Baixo, fácil automatização e Execução rápida;

Teste de Integração

Teste de integração como por exemplo se conexão com Banco de dados e microserviços funcionam de maneira esperada;
Custo maior, várias partes do software em execução;





Teste Funcional

Focado nos requisitos de negócio, verificando apenas a saída de uma ação.
Esperar resultado específicos de um requisito que o produto possui.

P2P (Ponta a Ponta)

Simula o uso cotidiano de um usuário com o sistema, como fazer login como verificação do status de uma compra.

Alto custo e difícil de fazer uma atualização caso seja automatizado.





Teste de Aceitação

Testa todos os requisitos de negócio. Replicar o comportamento do usuário

Pode testar desempenho e aceitar/rejeitar alterações baseado no cumprimento de metas.

Teste de Desempenho

Testa o desempenho de uma carga de trabalho específica.

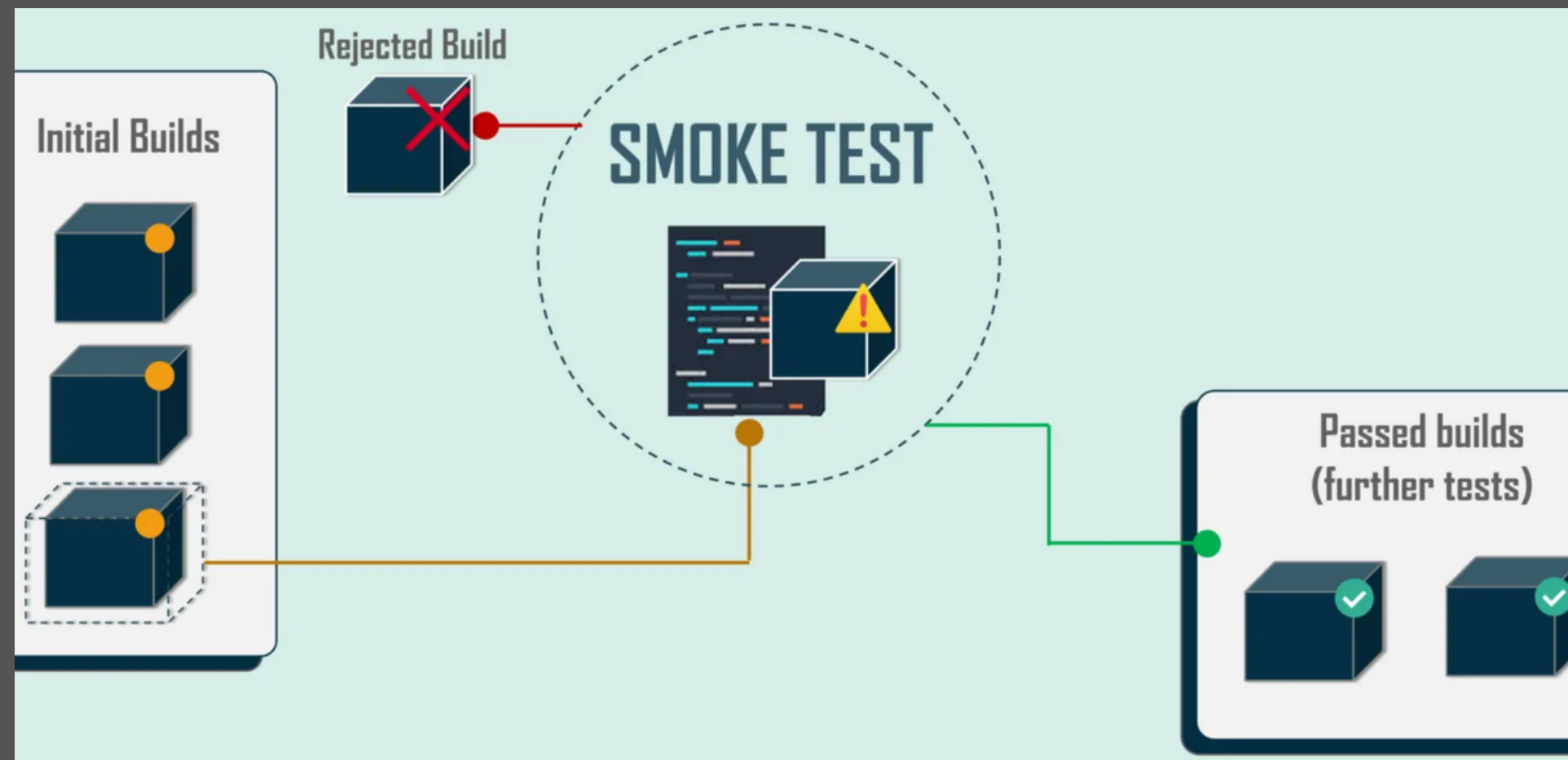
Velocidade, Confiabilidade, Escalabilidade.



Teste de Fumaça

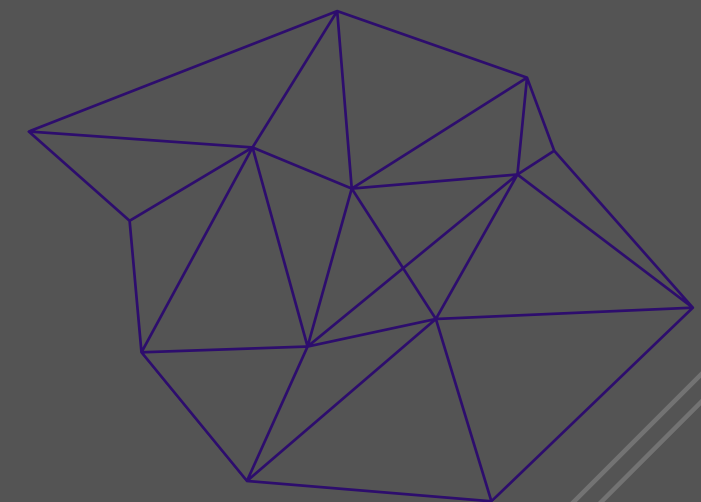
Testes básicos para funcionalidades básicas.

É o que determina se uma nova build precisa fazer testes mais caros ou se ela é rejeitada.



Arquitetura de Software

MVC
Cliente-Servidor
Microserviços
...



MVC (Model-View-Controller)

Projeto separado em 3 camadas:

Model -> Manipulação dos dados;

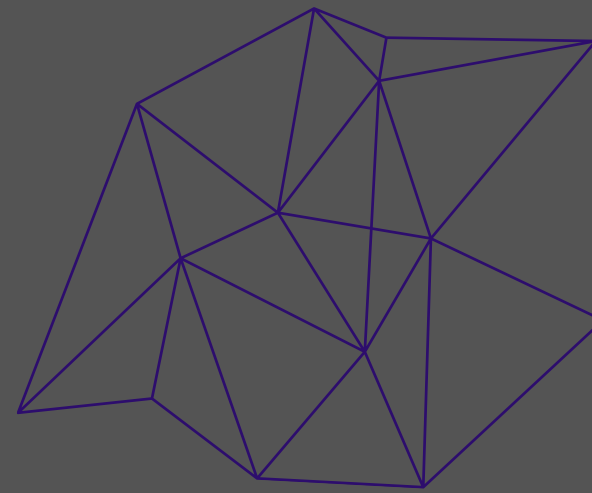
View -> O *front-end* onde captura e exibe os dados pro usuário;

Controller -> Fluxo de dados da Aplicação;

Cliente-Servidor

Cliente (Usuário) manda requisições para o Servidor.
Servidor processa e manda resposta para o cliente.


Centralizando o processamento no servidor, correndo risco de sobrecarga.





Microserviços

Estrutura modular;
Múltiplos serviços e componentes;
Escalabilidade e Independência dos módulos;



Bibliografia

<https://www.devmedia.com.br/gerencia-de-configuracao-de-software/9145>

<https://www.atlassian.com/br/continuous-delivery/software-testing/types-of-software-testing>

<https://pedrohjmartins.medium.com/smoke-testing-dfcd62903648>

<https://posdigital.pucpr.br/blog/tipos-de-arquitetura-de-software>

