



A aula interativa do Módulo 4 – Bootcamp Arquiteto Cloud Computing em breve!

Importante:

- 1) Não se esqueça de acessar a aula com seu e-mail cadastrado no ambiente de aprendizagem do IGTI e seu nome completo. Entrou com os dados errados? Saia da sala e entre novamente com os dados corretos.
- 2) Para sua frequência ser computada, responda a enquete no ambiente de aprendizagem, no horário indicado pelo professor. A enquete ficará disponível por 10 minutos.
- 3) Utilize o Chat para interagir com os colegas durante a aula, a ferramenta Raise Hands para pedir a palavra, e, em caso de dúvidas sobre o conteúdo, utilize o Q&A (perguntas e respostas) para que o professor tutor possa respondê-las.

Informações importantes

- Acesse a aula com o seu nome completo e com o mesmo e-mail utilizado no cadastro do ambiente de aprendizagem do IGTI. Entrou com os dados errados? É só sair da sala e entrar com os dados corretos!
- As presenças das Aulas Interativas são computadas através de uma enquete, que será realizada no ambiente de aprendizagem do IGTI. Para sua frequência ser computada, quando solicitado pelo professor, você deverá ir até a seção “Enquete de presença da Aula Interativa”, localizada na Área Acadêmica da disciplina, e responder a enquete. Essa seção estará logo abaixo da que contiver o link para a Aula Interativa em questão. A enquete ficará no ar por 10 minutos e sua nota estará disponível ao término desse tempo.
- Utilize o chat para interagir com os colegas durante a aula interativa! Em caso de dúvidas sobre o conteúdo, é só postá-las no Q&A que o tutor irá respondê-las! Lembre-se que você ainda pode esclarecer as suas dúvidas nos fóruns disponibilizados no Ambiente de Aprendizagem.

Informações importantes

- Ah! E se você não conseguir assistir a aula interativa, não se preocupe! Sua gravação ficará disponível no Área Acadêmica, juntamente com os slides utilizados pelo professor, em até 24 horas úteis após o término da aula. Você também poderá realizar a atividade de reposição para recuperar os pontos de presença!
- Se você precisar solicitar prorrogação e/ou 2^a oportunidade para entrega de atividades, saiba que isso é realizado somente mediante a apresentação de atestado médico ou de óbito de parentes de 1º grau.
- Para melhor experiência nas aulas interativas, sugerimos que você baixe o aplicativo do Zoom no seu computador.



Soluções para Desenvolvimento

Primeira Aula Interativa

Prof.ª Analía Irigoyen

Soluções para Desenvolvimento

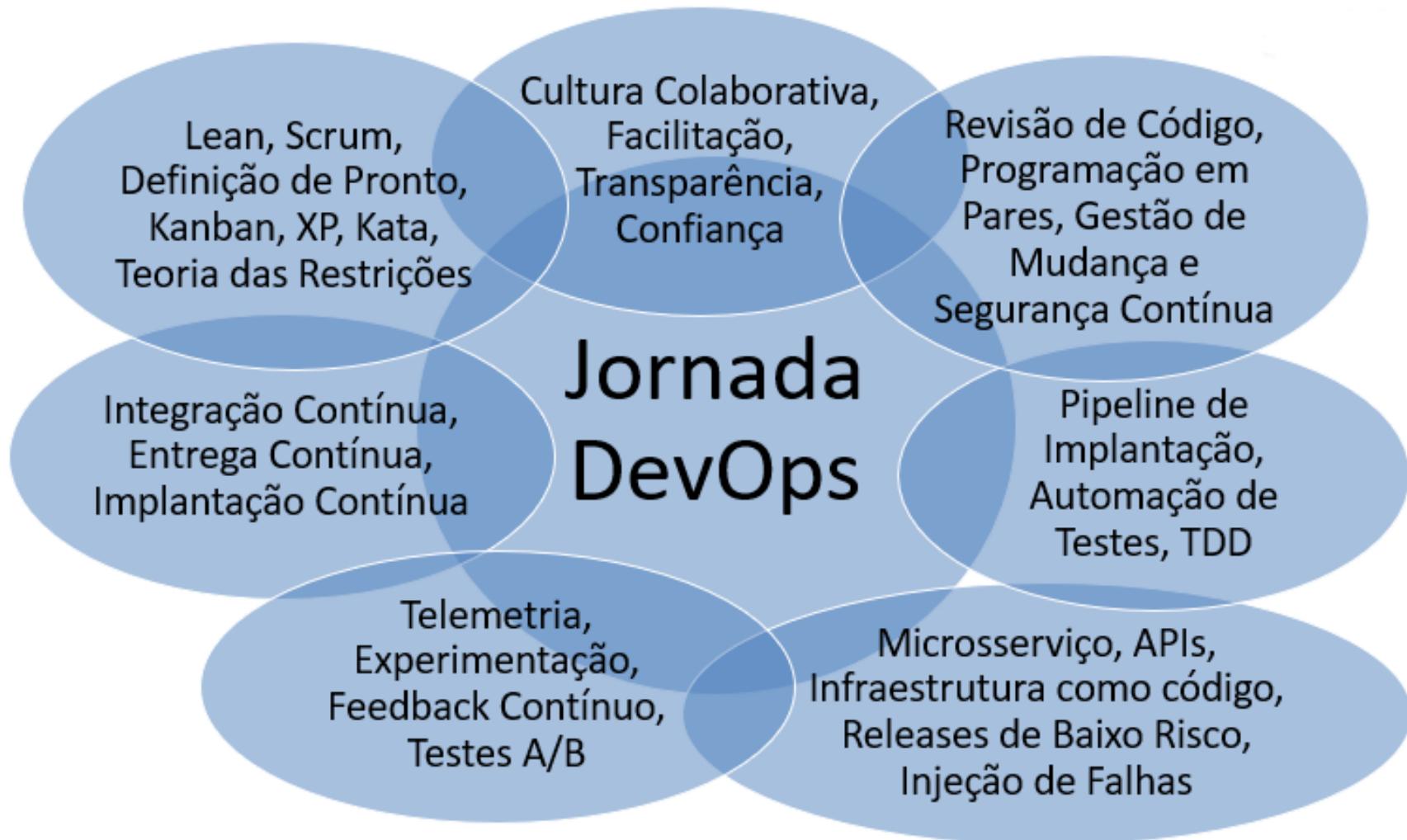
Primeira Aula Interativa – Módulo 4 e Case Prático

Prof.^a Analia Irigoyen

Por que DevOps?



Mindset aceitável no passado	Mindset digital exige DevOps
Filas no checkin para embarcar	Checkin via app
Blockbuster multa por atraso	Netflix entrega vídeo sob demanda
TV domina conteúdo e audiência	Netflix domina conteúdo e audiência
Guia 4 rodas para viagem	Google compra Waze por U\$ 1 Bi
Momento kodak em papel	Facebook compra Instagram por U\$ 1 Bi
Comunicação por e-mail	Facebook compra WhatsApp por U\$ 22 Bi
Filas para transações bancárias	Transações na palma da mão
Filas no orelhão	Qtd de celular supera qtd de pessoas
Táxi acessível para poucos	Uber acessível para muitos
Anúncio caro: TV, jornal e revista	Domínio do Google e mídias sociais
Reembolso médico em papel	Reembolso digital e ágil
Linear e analógico	Exponencial e digital



Introdução e adoção do Devops – Conceitos Básicos

1	Os	2	En
Gl GitLab		Sp Splunk	
3 Fm	4 En	5 En	6 Fm
Gh GitHub	Dt Datical	XLr XebiaLabs XL Release	Aws AWS
11 Os	12 En	13 Os	14 En
Sv Subversion	Db DBMaestro	Dk Docker	Ur UrbanCode Release
19 En	20 En	15 Pd	16 Pd
Cw ISPW	Dp Delphix	17 Fm	18 Os
21 Os	22 Fm	19 Os	20 En
Jn Jenkins	Cs Codeship	Fn FitNesse	Ju JUnit
23 Os	24 Fr	25 Fr	26 Fm
Ka Karma	Su SoapUI	Ch Chef	Tf Terraform
27 En	28 Fr	29 En	30 En
XLd XebiaLabs XL Deploy	Ud UrbanCode Deploy	Ku Kubernetes	Cc CA CD Director
31 Os	32 Fm	33 En	34 Pd
Docke	Ur UrbanCode Release	Pr Plutora Release	Al Alibaba Cloud
35 Os	36 Os	37 En	38 Fm
Docker	Azure Functions	Kubernetes	Os OpenStack
Ld Lambda	Ic IBM Cloud	Cc CA CD Director	Ps Prometheus
39 Pd	40 Fm	41 Fr	42 Fr
Rg Redgate	Ba Bamboo	Vs VSTS	Se Selenium
43 Os	44 Pd	45 En	46 Os
Ja Jasmine	Sl Sauce Labs	An Ansible	Ru Rudder
47 En	48 Os	49 Os	50 Pd
Oc Octopus Deploy	Go GoCD	Ms Mesos	Gke GKE
51 Fm	52 Pd	53 Pd	54 En
At Artifactory	Rg Redgate	Ba Bamboo	Vs VSTS
55 Pd	56 Os	57 Os	58 Fm
Nx Nexus	Fw Flyway	Tr Travis CI	Tc TeamCity
59 Os	60 Fr	61 Fm	62 Pd
Ga Gatling	Tn TestNG	Tt Tricentis Tosca	Pe Perfecto
63 En	64 Os	65 Fm	66 En
Pu Puppet	Pa Packer	Cd AWS CodeDeploy	Ec ElectricCloud
67 Os	68 Pd	69 Os	70 Os
Ra Rancher	Aks AKS	Rk Rkt	Sp Spinnaker
71 Pd	72 Pd	73 Fm	74 En
Ir Iron.io	Mg Moogsoft	Bb BitBucket	Pf Perforce
75 Fm	76 Pd	77 Fr	78 Os
Cr Circle CI	Cb AWS CodeBuild	Cu Cucumber	Mc Mocha
79 Os	80 En	81 Os	82 Os
Lo Locust.io	Mf Micro Focus UFT	Sa Salt	Ce CFEngine
83 En	84 En	85 En	86 Pd
Eb ElasticBox	Ca CA Automic	De Docker Enterprise	Ae AWS ECS
87 Fm	88 Os	89 Os	90 Os
Hm Helm	Aw Apache OpenWhisk	Ls Logstash	



Follow @xebialabs

91 En	92 Os	93 Fm	94 En	95 En	96 Fm	97 Os	98 Os	99 Os	100 En	101 En	102 En	103 En	104 Os	105 Os
XLi XebiaLabs XL Impact	Ki Kibana	Nr New Relic	Dt Dynatrace	Dd Datadog	Ad AppDynamics	EI ElasticSearch	Ni Nagios	Zb Zabbix	Zn Zenoss	Cx Checkmarx SAST	Sg Signal Sciences	Bd BlackDuck	Sr SonarQube	Hv HashiCorp Vault
106 En	107 Pd	108 Fm	109 Fm	110 Fm	111 En	112 En	113 En	114 Pd	115 Pd	116 Os	117 Fm	118 En	119 En	120 En
Sw ServiceNow	Jr Jira	Tl Trello	Sk Slack	St Stride	Cn CollabNet VersionOne	Ry Remedy	Ac Agile Central	Og OpsGenie	Pd Pagerduty	Sn Snort	Tw Tripwire	Ck CyberArk	Vc Veracode	Ff Fortify SCA

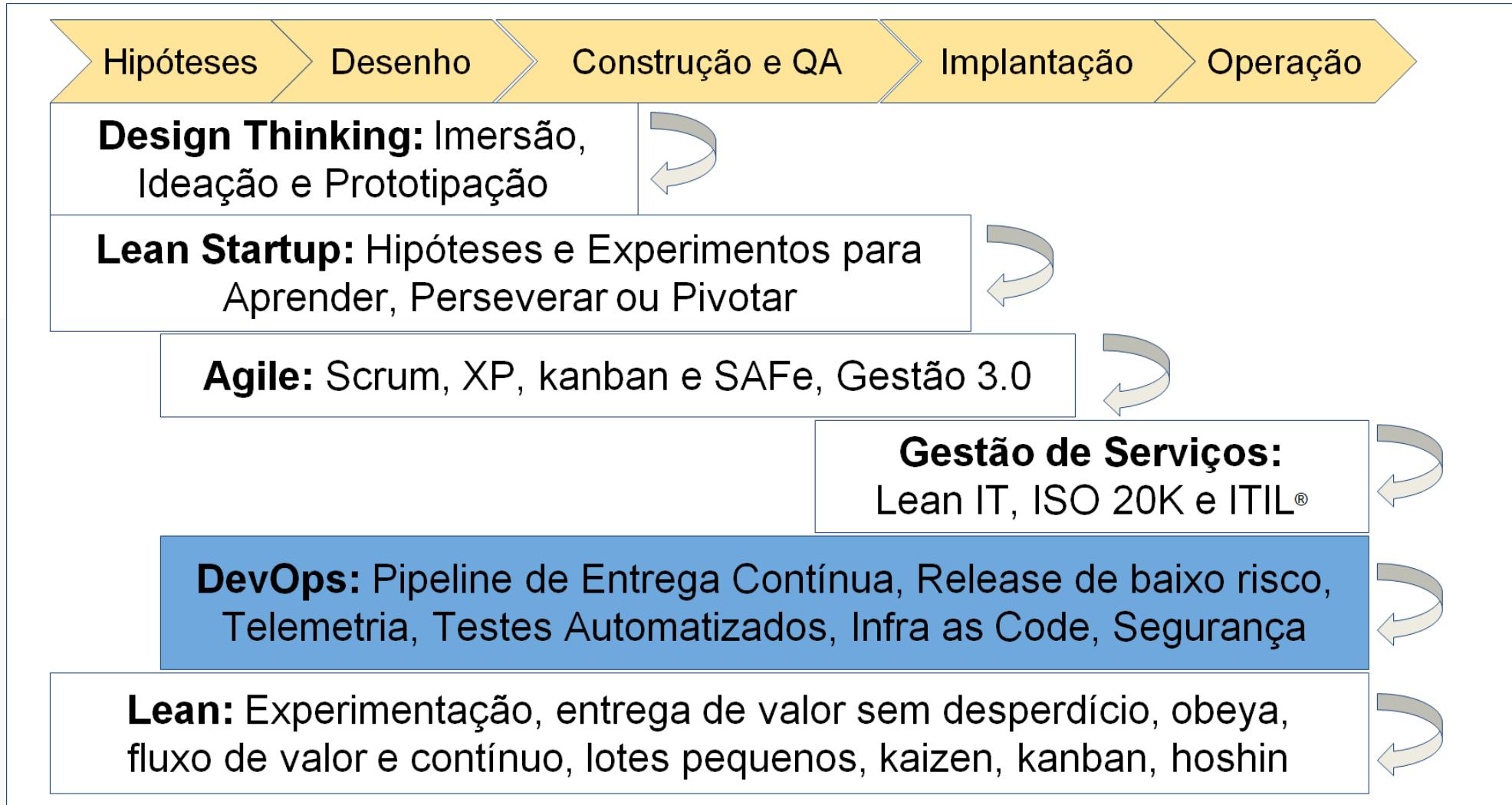
<https://softwarezen.me/campaigns/warmup/gravacao/>

A má implementação ou excesso de customização pode destruir uma ótima ferramenta...

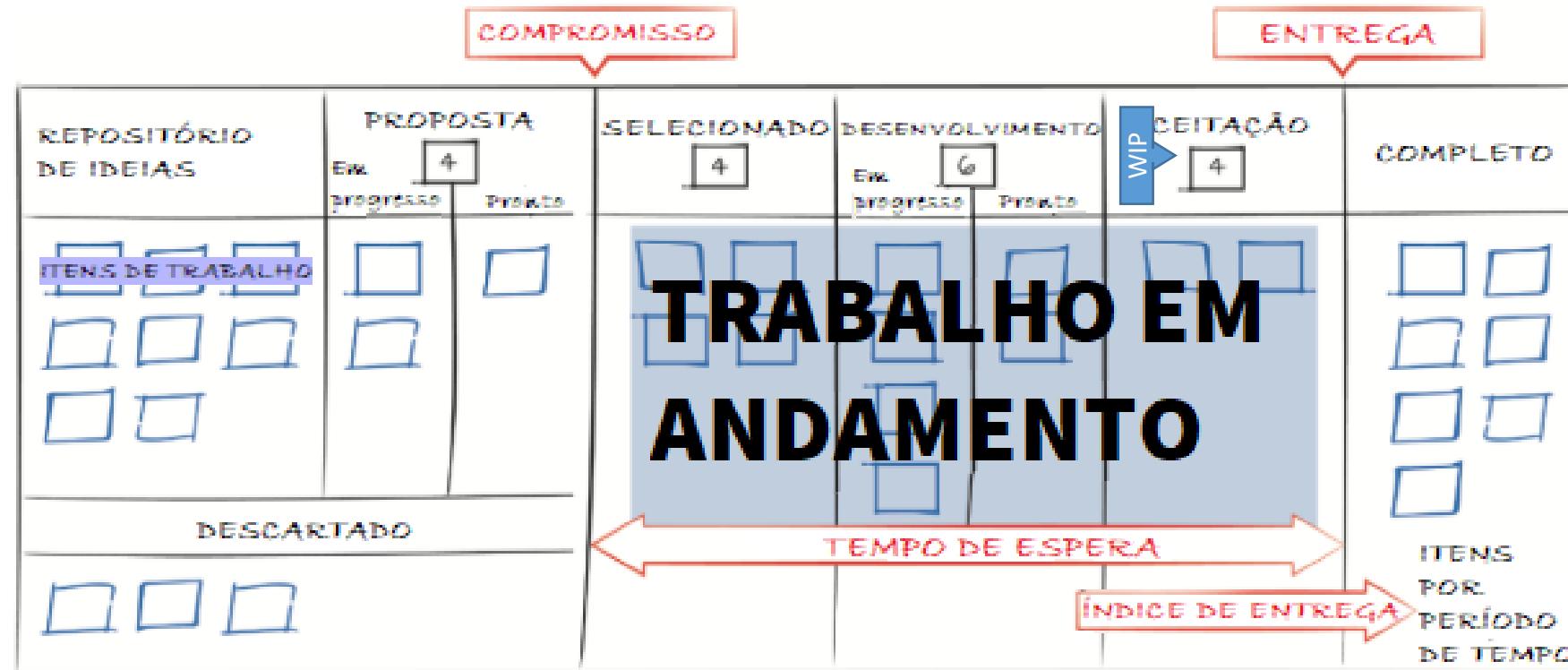
**Incentive a opinião das equipes no chão de fábrica!
Vá ao gemba !**



Introdução e adoção do Devops – Conceitos Básicos



KANBAN – Produção Puxada



Dívida técnica (Débito técnico)



- ✓ **Represents the result of short-term decisions that generate problems that become increasingly difficult to resolve over time.**
- ✓ Reduces team performance and increases TCO.
- ✓ Conflicting objectives between Dev and Ops areas contribute to increasing technical debt.

Algumas causas da dívida técnica

- ✓ Ausência de testes
- ✓ Requisitos não claros
- ✓ Documentação pobre
- ✓ Pouco foco em refatoração
- ✓ Equipes não colaborativas
- ✓ Conflitos de interesse
- ✓ Pressão de chefes e áreas de negócio

Princípios Lean



Fluxo de valor: É o processo que concretiza uma necessidade de negócio em um produto ou serviço para entrega de valor ao cliente.

Mapeamento do fluxo de valor: Visa entender como o processo funciona com foco na entrega de valor ao cliente e identifica gargalos ou desperdícios.

Gembá: É o local onde as coisas acontecem e todos deveriam ir ao gembá com frequência para conhecer o “chão de fábrica” e evitar suposições sem dados e fatos.

Obeya: Também conhecida nas organizações como “sala de guerra”, o objetivo é facilitar a gestão visual e a coordenação para solução de problemas sem os entraves das estruturas organizações clássicas.

Desperdício

Existem 8 tipos de desperdícios identificados no Lean:

- Defeito e retrabalho: Desfazer, refazer algo.
- Movimentação: Caminhadas, deslocamentos, viagens.
- Espera: Pessoas aguardando informações, materiais ou outras equipes.
- Transporte: Transferências desnecessárias de materiais ou informações.
- Estoques: Informações ou materiais sem uso.
- Processamento: Etapa redundante ou desnecessária.
- Desconexão ou superprodução: Fluxo deficiente ou falta de sincronismo entre etapas (antes ou depois do necessário).
- Conhecimento: Não aproveitar as habilidades das pessoas adequadamente.

Corda de Andon: Dispositivo que existe nas fábricas da Toyota para interromper a linha de produção quando é encontrado algum defeito nos produtos. O objetivo é aglomerar imediatamente todas as equipes e líderes que podem ajudar a resolver o problema na origem, podendo mobilizar os executivos e a alta administração.

Objetivo da Primeira Maneira	Princípios e Práticas
<p>Acelerar o fluxo dos desenvolvedores (Esquerda) para operação e clientes (Direita)</p> <p>Empresa Cliente</p> <p>Dev → Ops</p> 	<ol style="list-style-type: none">1. Tornar o trabalho visível2. Reduzir o tamanho dos lotes e intervalos3. Aplicar teoria das restrições e otimizar o fluxo4. Remover desperdícios e foco no cliente5. Reduzir o número de transferências (handoff)6. Incorporar qualidade na origem7. Limitar o trabalho em andamento (WIP)8. Infraestrutura como código e self service9. Integração, entrega e implantação contínua10. Testes automatizados e TDD11. Arquitetura e releases de baixo risco

Resumo da Integração contínua

Controle de versão

Processo de compilação

Testes e análise de código

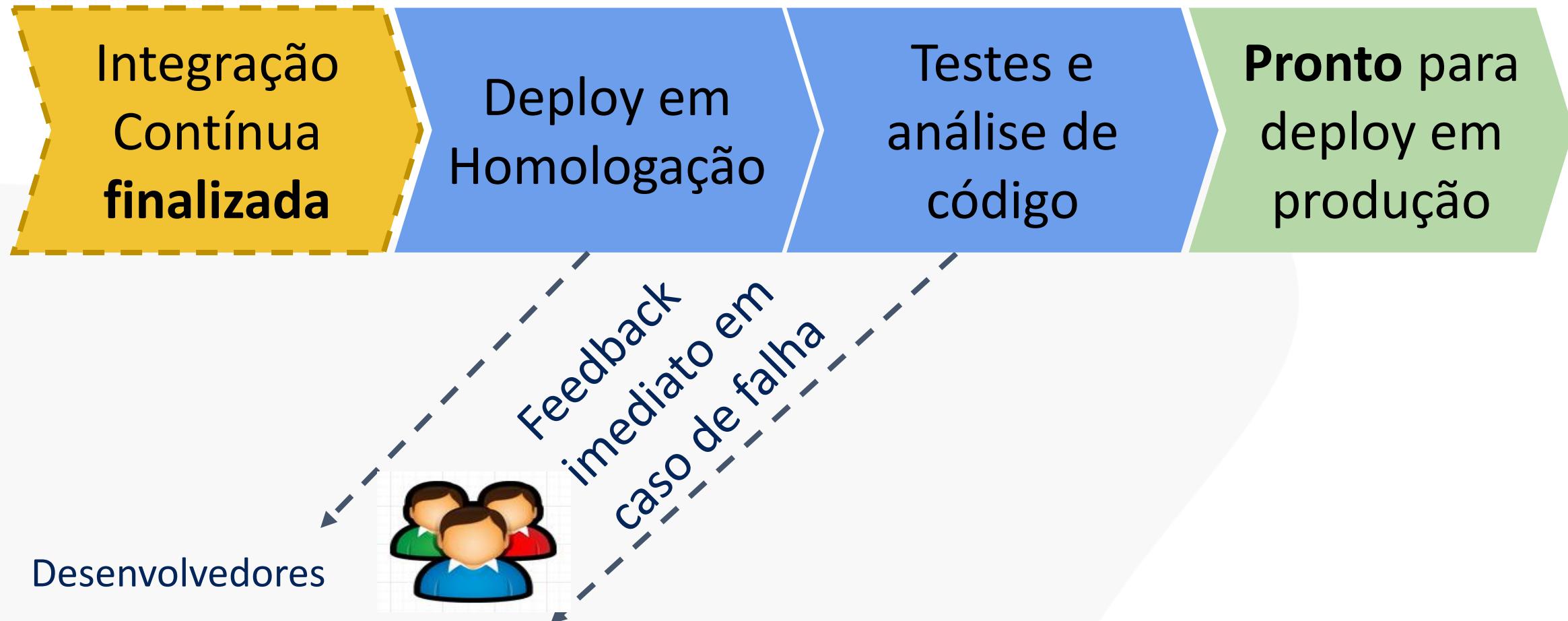
Próximos estágios



Feedback imediato
em caso de falha

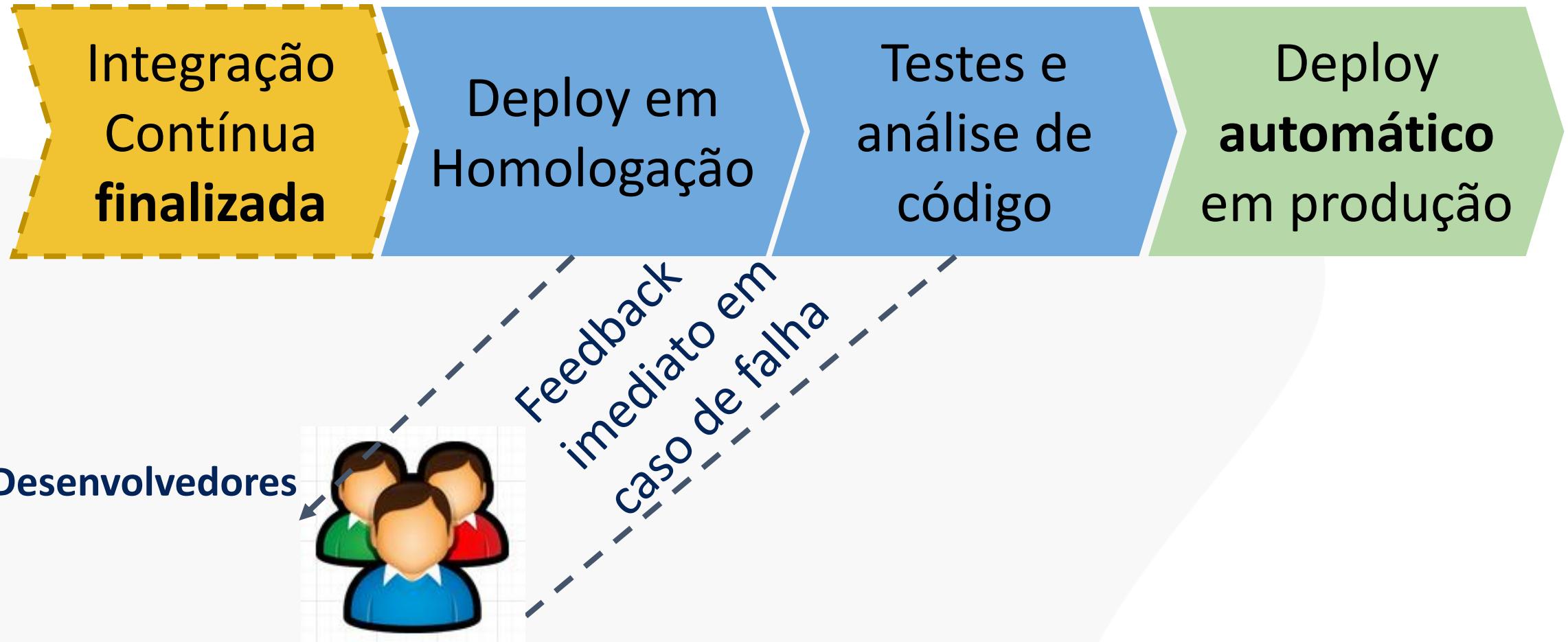
Codificação e testes no ambiente dos desenvolvedores

Resumo da Entrega contínua



Resumo da Implantação contínua

IGTI



Infraestrutura ágil



É a aplicação dos princípios ágeis na infraestrutura

Foco principal é a infraestrutura como código:

Gerência e provisão de ambientes automaticamente

Tudo fica centralizada no controle de versões

Nuvem/virtualização é a base e existem ferramentas para otimizar (Puppet, Terraform, Chef, Ansible, etc.)

Infraestrutura como código (IaC)



- ✓ Código e configurações dentro do controle de versão.
- ✓ Criação automatizada e sob demanda (self-service) em todos os ambientes, evitando trabalho manual.
- ✓ Todos os estágios do fluxo de valor com ambientes iguais ou semelhantes ao de produção.
- ✓ **Infraestrutura imutável:** Foco em recriar todo o ambiente de produção de forma rápida em vez de realizar alterações.

IaaS, SaaS e PaaS

IGTI



Fonte: <https://azure.microsoft.com/>

PaaS AzureDevOps + Azure Services



O Azure DevOps gerencia o processo de desenvolvimento.

[Azure DevOps](#)

Ciclo de Desenvolvimento



Os agentes de build e lançamento do Microsoft Release Management implantam o modelo do Azure Resource Manager e o código associado nos vários ambientes.

[Microsoft
Release
Management](#)

Criação de Releases



Os grupos de recursos do AzureDevOps são usados para definir todos os serviços necessários para implantar a solução em um ambiente de desenvolvimento e teste ou de produção.

[Grupos de
recursos](#)

Criação de Pipelines

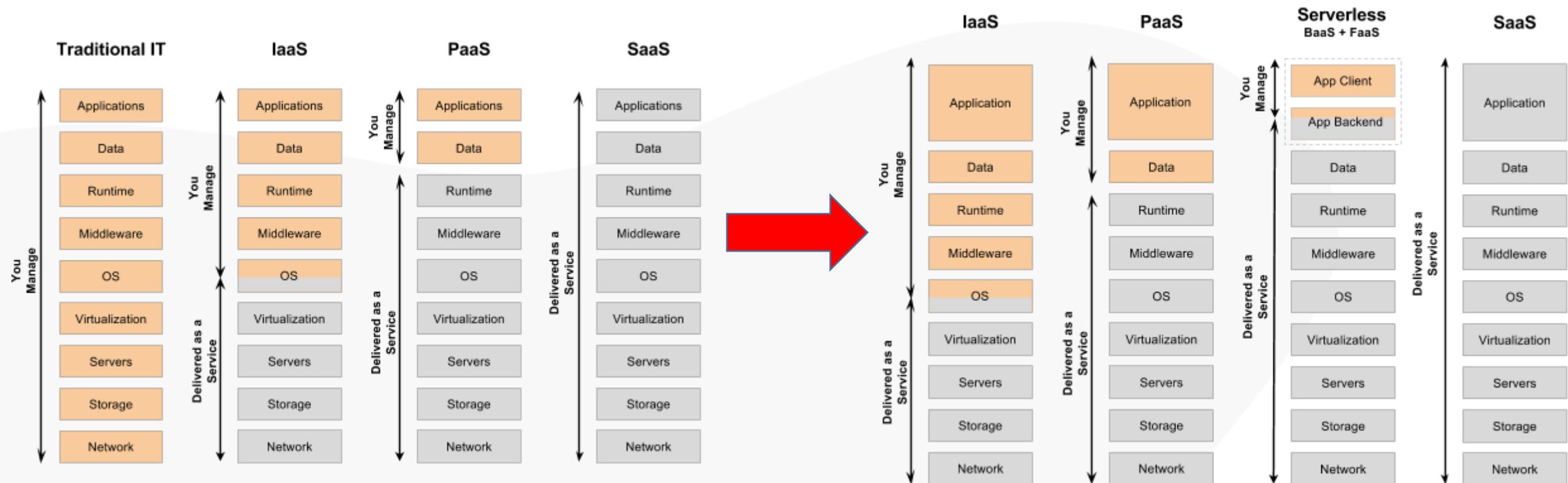


Um aplicativo Web executa o site e é implantado em todos os ambientes. Slots de preparo são usados para alternar entre versões de pré-produção e de produção.

[Aplicativos Web](#)

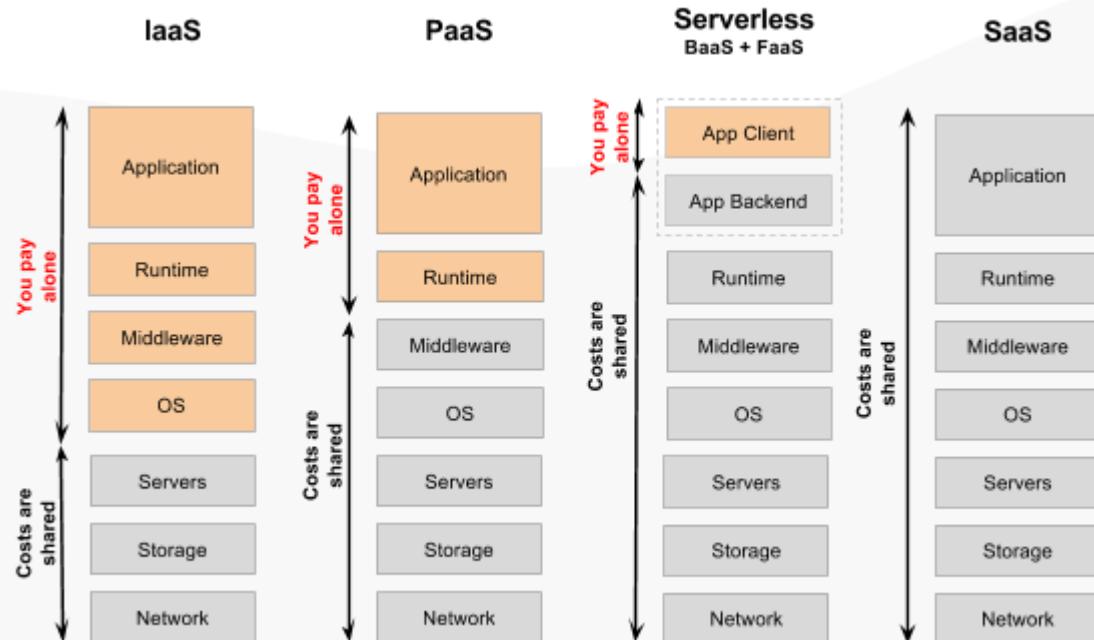
Criação de Ambiente
para hospedar um site

Serverless – Backend as a Service (BaaS) and Function as a Service (FaaS).



Fonte: <https://specify.io/concepts/serverless-baas-faas>

Serverless – Backend as a Service (BaaS) and Function as a Service (FaaS).

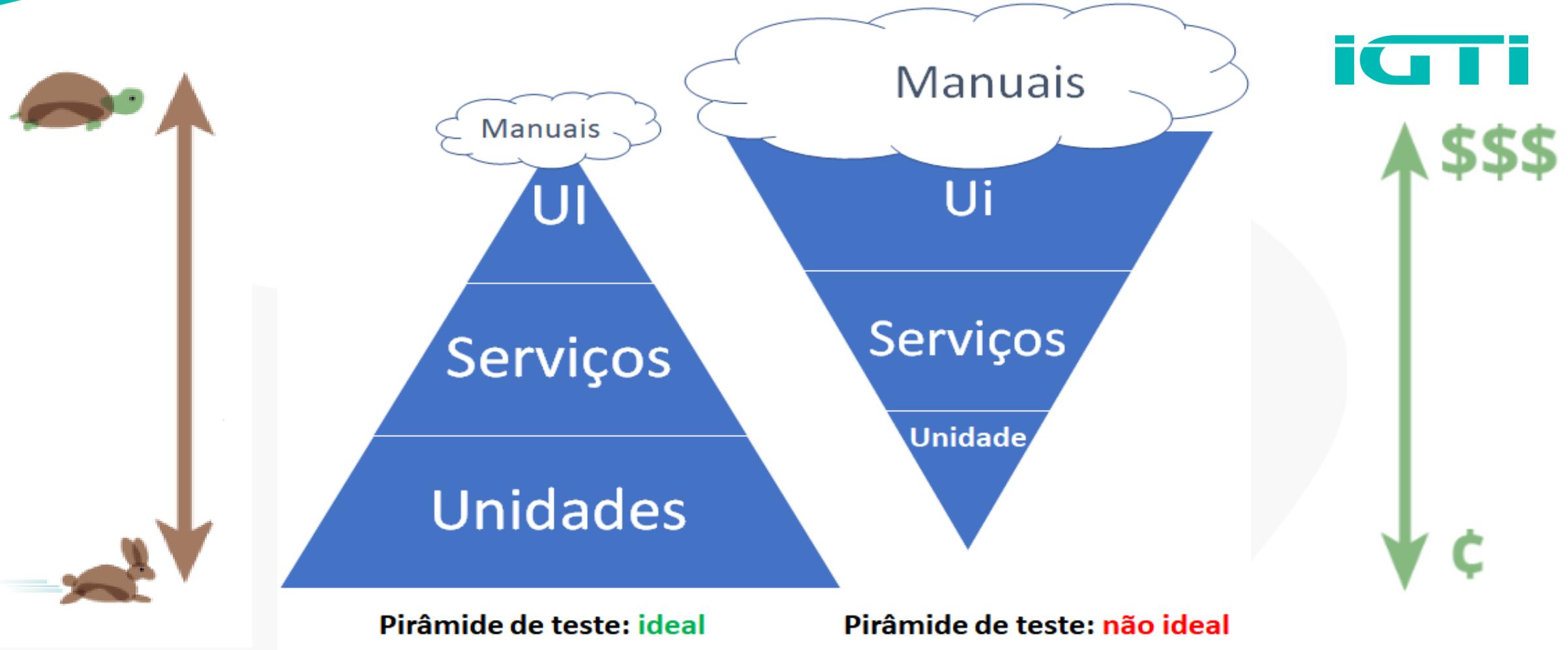


Quando você está usando FaaS, você compartilha o tempo de execução com outras pessoas. Por exemplo: quando sua "função" é escrita em JavaScript, esta parte do código será executada no mesmo servidor node.js que as "funções" de outros usuários deste FaaS (isso pode ser diferente para alguns fornecedores).

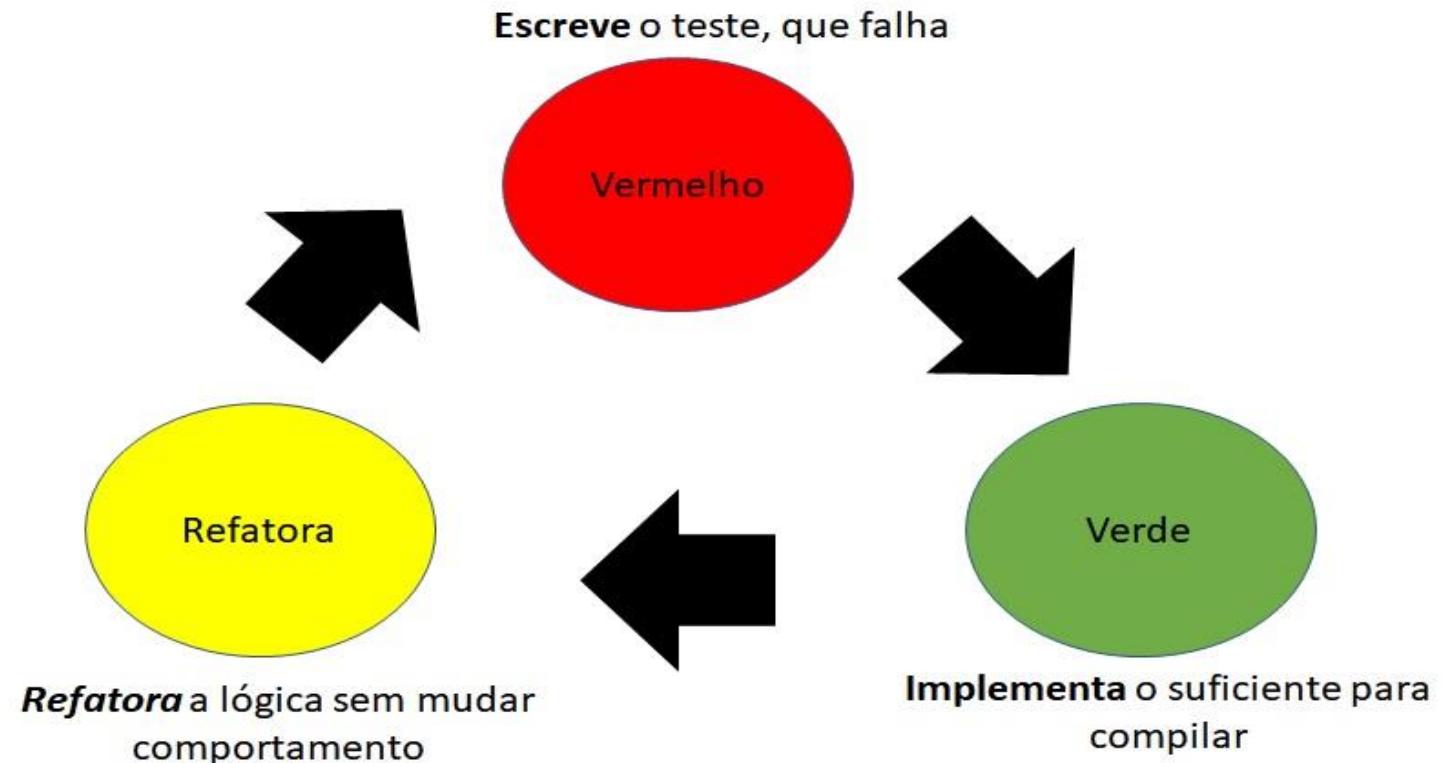
Ao usar o BaaS, você compartilha o mesmo BaaS com outros usuários deste BaaS (isso pode ser diferente para alguns fornecedores).

Fonte: <https://specify.io/concepts/serverless-baas-faas>

AGENDA DEVOPS (1ª Maneira: Fluxo)



Sequência básica do TDD



Estratégias de branching (ramificação)



Estratégia	Vantagens	Desvantagens
Produtividade individual	Projeto privado que não atrapalha outras equipes	Merge do código ocorre no final do projeto e gera muitos problemas
Produtividade da equipe (Desenv. Baseado no trunk)	Fila única com todos trabalhando no trunk e commit frequente Não há o estresse de merge no final do projeto	Difícil de implantar e cada commit pode quebrar o projeto inteiro Deve-se puxar a corda de andon para corrigir

Categorias de liberação (Release)



Baseado no Ambiente

Há 2 ou mais ambientes e apenas um fica ativo para os clientes (Ex.: balanceadores)

Azul verde (Blue-green) e Canário (Canary)

Baseado no aplicativo

Novas funcionalidades de forma seletiva usando configurações simples (não precisa fazer deploy)

Alternância de recursos (Feature toggles) e Lançamento escuro

AGENDA DEVOPS (1ª Maneira: Fluxo)

Arquitetura	Vantagens	Desvantagens
Monolítica	<p>Inicialmente simples</p> <p>Baixa latência entre processos</p> <p>Eficiente para recursos em pequena escala</p>	<p>Fraca escalabilidade e redundância</p> <p>Implantação big bang</p> <p>Longo período de build</p>
Microserviço	<p>Cada unidade é simples</p> <p>Escalonamento independente</p> <p>Teste e implantação independente</p> <p>Facilita visão por produto</p>	<p>Latência de rede</p> <p>Precisa de ferramentas para gerenciar dependências</p>

Objetivo da Segunda Maneira	Princípios e Práticas
<p>Rápido feedback em todos os estágios do fluxo de valor (Direita para a Esquerda)</p>  <p>The diagram illustrates a flow from right to left. A thick blue arrow points from the word 'Ops' on the right towards the word 'Dev' on the left. Above this main arrow, a curved blue arrow loops back from 'Dev' back towards 'Ops', representing a feedback mechanism.</p>	<ol style="list-style-type: none">1. Ver problemas quando ocorrem (“ir ao gemba”)2. Aglomerar quando problema aparece (Andon)3. Qualidade próxima da fonte (menos aprovações)4. Telemetria self service e irradiadores de informação disponível para todos5. Desenvolvimento por hipóteses e Testes A/B6. Equipes Dev e Ops compartilham o trabalho diário e plantões de suporte 24 x77. Revisão de código usando as técnicas: Programação em pares, sobre os ombros, divulgação por email, assistida por ferramentas

Telemetria

✓ Coleta remota e automática de dados para monitorar a saúde das aplicações

✓ Visão de todo o fluxo de valor permite otimizar e comunicar melhor



Fornece feedback efetivo:

- ✓ **Experiência do cliente**
- ✓ Erros no pipeline de implantação e após deploy
- ✓ Eventos proativos
- ✓ **Funcionalidades mais usadas pelos clientes**

Feedback

Visão clara do fluxo de valor em todos os estágios do ciclo de vida do serviço, produto, desenvolvimento e implementação, entrada em operação e retirada

Dessa forma, todos os participantes do fluxo de valor tem a oportunidade de aprender desde os primeiros estágios do projeto

Opções para resolver problemas



Correção: Fix Forward

É uma mudança no código e deve ser executada em ambientes com grande maturidade:
Teste automatizado, implantação rápida e telemetria

Reversão: Rollback

Usuário deixa de receber a implantação e há 2 formas:

1. Implantar a versão anterior (Ex.: Canário)
1. Alternância de recursos:
Mais fácil de reverter e menos arriscada

Suporte compartilhado



A implantação de um pipeline com testes automatizado não garante zero erro, em função da complexidade dos sistemas

Erro de software depende que a equipe Dev priorize a solução para não impactar a operação por muito tempo

Todos os participantes do Fluxo de Valor devem compartilhar as responsabilidades para resolver os incidentes em produção

Eficácia do Pull Request



Recomendação
é que a revisão
seja realizada
pelo colega
antes de efetivar
o código no trunk

Revisão deve ser em pequenos lotes.
Shoup diz: “Quando aumentamos o
código de 10 linhas para 100 linhas, a
chance de erro aumenta 10 vezes”

Mudanças críticas podem ter a
revisão do especialista no assunto
(segurança, BD, redes)

Programação em pares



O código é criado por duas pessoas trabalhando juntas em um único computador

A melhor maneira de parear o programa é sentar lado a lado na frente do monitor para revezar as atividades de codificação (piloto) e revisão (navegador)

Tempo gasto a mais é 15% e aumenta qualidade do software de 70% para 85%