



Plano de Trabalho DevOps e Melhoria Contínua

11/12/2024

Objetivo

Fortalecer áreas técnicas estratégicas

como DevSecOps e Nuvem, aumentando a eficiência da Divisão de Desenvolvimento de Sistemas e Depto de Informática

Aprimorar processos e metodologias

assegurando maior previsibilidade, produtividade e qualidade nas entregas

Capacitar colaboradores e estagiários

por meio de treinamentos contínuos e programas de mentoria, garantindo evolução e retenção de talentos

Promover maior integração

entre equipes e setores, diminuindo silos e aprimorando a comunicação e a transparência

Fases

1. FORMAÇÃO
E MENTORIA
PARA
ESTAGIÁRIOS E
JUNIORES

2. AVALIAÇÃO
DO CENÁRIO
ATUAL

3. INTRODUÇÃO
DE
FERRAMENTAS E
PRÁTICAS
INICIAIS

4. AUTOMAÇÃO
DE TESTES E
QUALIDADE

5.
INFRAESTRUTURA
E
MONITORAMENTO

6. CULTURA DE
COLABORAÇÃO
E FEEDBACK
CONTÍNUO

7. REVISÃO E
OTIMIZAÇÃO
DO PROCESSO

8. EXPANSÃO E
MANUTENÇÃO
DA CULTURA
DEVOPS

1. FORMAÇÃO E MENTORIA PARA ESTAGIÁRIOS E JUNIORES

- 1 **Ministrar workshop de Scrum e Metodologias Ágeis** para estagiários e equipe de Infra
- 2 Implementar o **PDI (Plano de Desenvolvimento Individual)** para estagiários e funcionários
- 3 Apresentar estrutura de mentoria para orientar estagiários e **Programa de obtenção da certificação Scrum Foundation Professional Certification – SFPC** em 4 semanas.

1. FORMAÇÃO E MENTORIA PARA ESTAGIÁRIOS E JUNIORES

PDI (Plano de Desenvolvimento Individual)



MARCOS DO PROJETO

Discovery e Diagnostic

Design e Development

Delivery

Progress Control



Assembleia Legislativa do Estado da Paraíba
Departamento de Informática

Plano de Desenvolvimento Individual (PDI)

Colaborador: Lucas Andrade

Cargo: Desenvolvedor Java Júnior

Gestor: Wagner Souza

Data de Início: 15/12/2024

Validade: 10/03/2025

1. Avaliação Atual

Habilidades Técnicas

- **Fortes:** Programação em Java, Git, e conceitos de OOP.
- **Melhorar:** SQL (consultas complexas e otimização de consultas) e tasks relacionadas a APIs RESTful.

Habilidades Comportamentais

- **Fortes:** Pontualidade, boa relação com colegas de equipe.
- **Melhorar:** Comunicação em reuniões.

2. Metas de Desenvolvimento

Metas de Curto Prazo (3-6 meses)

1. Melhorar habilidades em SQL:
 - **Objetivo:** Aprender a criar consultas SQL mais complexas e identificar gargalos de performance.
2. Participar de um projeto relacionado à APIs RESTful:
 - **Objetivo:** Desenvolver conhecimento prático em APIs RESTful.
 - **Indicador:** Assistir a o video "Entendo APIs RESTful" no youtube (42 minutos).

Metas de Longo Prazo (1 ano ou mais)

3. Tornar-se especialista em API Restful:
 - **Objetivo:** Concluir curso da PluraSight API Restfuls.
4. Melhorar soft skills:
 - **Objetivo:** Comunicar-se de forma mais clara e assertiva em dailies e reuniões.
 - **Indicador:** Feedback positivo do gestor e equipe em duas avaliações trimestrais.
 - **Prazo:** Até 10/12/2025.

3. Ações Necessárias

- Para SQL:
 - Participar do curso 'SQL Performance Tuning' na plataforma LinkedIn.
 - Revisar consultas SQL atuais com o apoio de um desenvolvedor pleno da equipe.
- Para API Restful:
 - Assistir ao video "Entendendo APIs RESTful" no youtube (42 minutos).
 - Ser alocado em algum projeto que utilize essa tecnologia
- Para Soft Skills:
 - Participar de workshops internos sobre comunicação e colaboração.
 - Assistir ao curso 'Comunicação Assertiva' no Pluralsight.

4. Cronograma

Ação	Prazo
Concluir curso 'SQL Performance Tuning'	31/03/2025
Revisar consultas SQL com mentor	15/04/2025
Entendendo APIs RESTful'	30/04/2025
Participar de workshop de comunicação	30/04/2025
Concluir curso da PluraSight Comunicação Assertiva	10/12/2025

5. Registro de Feedbacks

10/01/2025: Lucas demonstrou progresso significativo no aprendizado de SQL, mas ainda precisa melhorar na performance de consultas.

10/03/2025: Feedback positivo do mentor técnico sobre aplicação prática de conhecimentos de SQL. Lucas também iniciou com êxito tarefas relacionadas à API Restful.

Colaborador: _____

Gestor: _____

Data: ____/____/____

PDI (Plano de Desenvolvimento Individual)

1. FORMAÇÃO E MENTORIA PARA ESTAGIÁRIOS E JUNIORES

- 1 Ministrar workshop de Scrum e Metodologias Ágeis para estagiários e equipe de Infra
- 2 Implementar o PDI (Plano de Desenvolvimento Individual) para estagiários e funcionários
- 3 Apresentar estrutura de mentoria para orientar estagiários e **Programa de obtenção da certificação Scrum Foundation Professional Certification – SFPC** em 4 semanas.



Scrum Foundation Professional Certification - SFPC™

Issued by [Certiprof](#)

Scrum Foundation Professional Certification holders have developed entry-level skills in Scrum that endorse their fundamental knowledge in this framework. They have demonstrated an understanding of the empirical Scrum pillars of transparency, inspection, and adaptation. Their primary focus is on the work of Sprint to make the best possible progress toward these goals.

[Learn more](#)

Certification

Foundational

Free

Skills

Agile Agile Mindset Agile Practices Agile Teams Scrum Scrum Artefacts
Scrum Events Scrum Roles Scrum Theory

Projetos / DevOps - Melhoria Co... / DOPS-3 / DOPS-16

Programa de obtenção da certificação Scrum Foundation Professional Certification – SFPC

[+ Adicionar](#)

[@ Apps](#)

Descrição

Scrum Foundation Professional Certification - SFPC™ (v2020)

- Formato:** Perguntas de múltipla escolha
- Perguntas:** 40
- Idioma:** Inglês / Espanhol / Português /
- Pontuação de aprovação:** 32/40 80% Ver.2020
- Duração:** 60 minutos

Projetos / DevOps - Melhoria Co... / DOPS-3 / DOPS-16

Programa de obtenção da certificação Scrum Foundation Professional Certification – SFPC

[+ Adicionar](#) [@ Apps](#)

Descrição

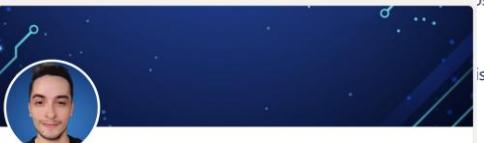
Scrum Foundation Professional Certification - SFPC™ (v2020)

- Formato:** Perguntas de múltipla escolha
- Perguntas:** 40
- Idioma:** Inglês / Espanhol / Português /
- Pontuação de aprovação:** 32/40 80% Ver.2020
- Duração:** 60 minutos

O exame é baseado no Guia oficial do Scrum.

The Scrum Guide™, The Definitive Guide to Scrum: The Rules of the Game, desenvolvido e sustentado pelos fundadores do Scrum – Ken Schwaber e Jeff Sutherland, é o guia oficial do Scrum. Ele define os conceitos, artefatos e as regras que vinculam eles juntos.

Esta certificação irá possibilitar que você navegue entre os diferentes artigos, definições e papéis. Após completa-la, você estará apto a obter a Advanced Professional Certificate (SAPC™).



Gabriel Andrade · 2º
Suporte em Tecnologia da informação | Ciência da Computação | Análise e Desenvolvimento de Sistemas
João Pessoa, Paraíba, Brasil · [Informações de contato](#)

146 conexões
 Edycarlos Rodrigues, Antonio Duarte, e mais 3 conexões em comum
[+ Conectar](#) [Enviar mensagem](#) [Mais](#)

Licenças e certificados

Scrum Foundation- Professional Certification SFPC
Certiprof
Emitido em nov de 2024 · Expira em nov de 2027

Público-alvo: Qualquer pessoa interessada em aprender sobre metodologias Agile.

Pré-requisitos: Não há pré-requisitos formais para esta certificação.

Tipo de Certificação: Fundação

Código de Certificação: SFPC™

Scrum Foundation certificação scrum Portuguese

Start date 14 de nov. de 2024

Data limite 19 de dez. de 2024

1 ...

Em andamento

Ações

Informações

Responsável

Paulo Estevar

Categorias

Nenhum

Pai

DOPS-3 FORMAÇÃO ...

Team

Nenhum

Desenvolvimento

Configurar ferramentas ...

REVISAR

2. AVALIAÇÃO DO CENÁRIO ATUAL

- 1 **Levantar todos os IT Elements disponibilizados pela Divisão de Sistemas**
- 2 **Identificar principais gargalos e problemas no processo atual**
- 3 **Mapear o fluxo de trabalho**
- 4 **Identificar ferramentas e processos críticos a serem melhorados**
- 5 **Definir KPIs iniciais para medir progresso**

A	B	C	D	E	F	
1	Nome do Sistema	Setor Principal Responsável	Descrição do Sistema	Data de Implantação	Desenvolvedor Responsável 1	Desenvolvedor Responsável 2
2	Helpdesk	Divisão de Protocolo	Sistema de suporte técnico e atendimento.	2021-09-30	Mario Rodrigues De Oliveira	Mario Rodrigues De Oliveira
3	ALPBDoc	Divisão de Licitações	Sistema para gerenciamento dos itens do Almoxarifado.	2023-05-31	Maria Elena Duarte Limeira	Jose Guimaraes Da Silva
4	LOA	Divisão de Atas	Sistema para gestão de licitações públicas.	2023-03-16	Allan Das Chagas Ugulino De Araujo	Sergio Marques De Souza
5	Painel	Plenario	Sistema para controle e organização de bibliotecas.	2022-11-26	Sergio Marques De Souza	Sergio Marques De Souza
6	Sapi	Divisão de Almoxarifado	Sistema para controle e organização de bibliotecas.	2021-03-26	Mario Rodrigues De Oliveira	Lucas Guimaraes Da Silva
7	Contratos	Divisão de Licitações	Sistema de controle de atas de reuniões.	2023-08-27	Ronaldo De Souza Silva	Ronaldo De Souza Silva
8	AAJLL	Divisão de Atas	Sistema para gerenciar protocolos de documentos.	2021-08-04	Sergio Marques De Souza	Maria Oliveira
9	Atas	Divisão de Almoxarifado	Sistema para gerenciamento de Atas	2020-06-19	Sergio Marques De Souza	Carlos Pereira
10	Protocolo	Divisão de Almoxarifado	Sistema para gerenciamento de contratos administrativos	2020-12-27	Ronaldo De Souza Silva	Ana Santos
11	ALPBDoc	Divisão de Protocolo	Sistema para gerenciamento de contratos administrativos	2023-11-19	Felipe Costa	Felipe Costa
12	Sistema de Almoxarifado	Divisão de Licitações	Sistema para gestão de licitações públicas.	2022-12-18	Ronaldo De Souza Silva	Amanda Huffman
13	Protocolo	Divisão de Licitações	Sistema para gerenciamento de contratos administrativos	2020-04-23	Clinton Nixon	Amy Mejia
14	Sistema de Compras	Divisão de Compras	Sistema para gerenciamento de contratos administrativos	2020-10-11	Sergio Marques De Souza	Ian Allen
15	Atas	Divisão de Atas	Sistema de controle de atas de reuniões.	2023-11-10	Sergio Marques De Souza	Debra Thompson

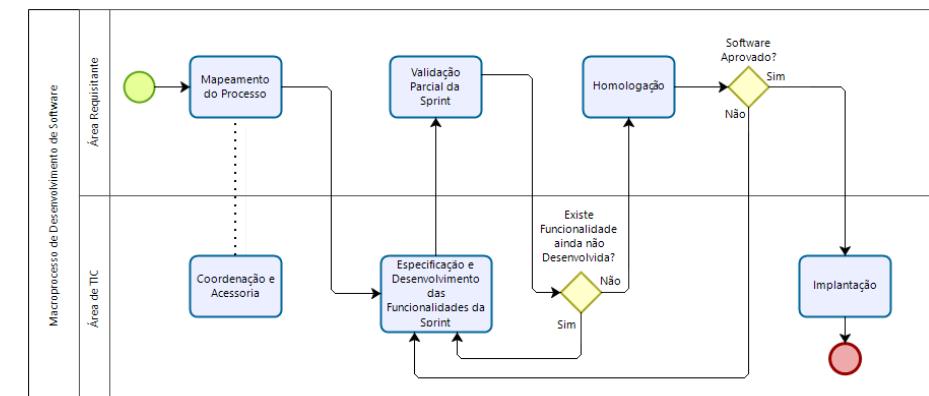
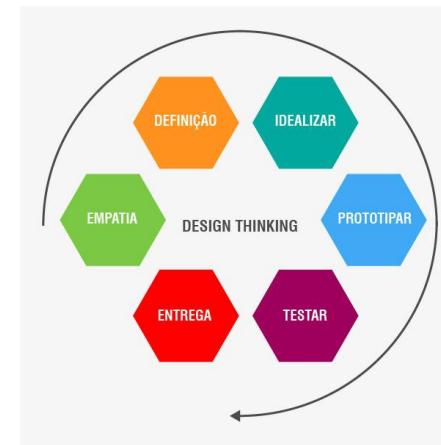
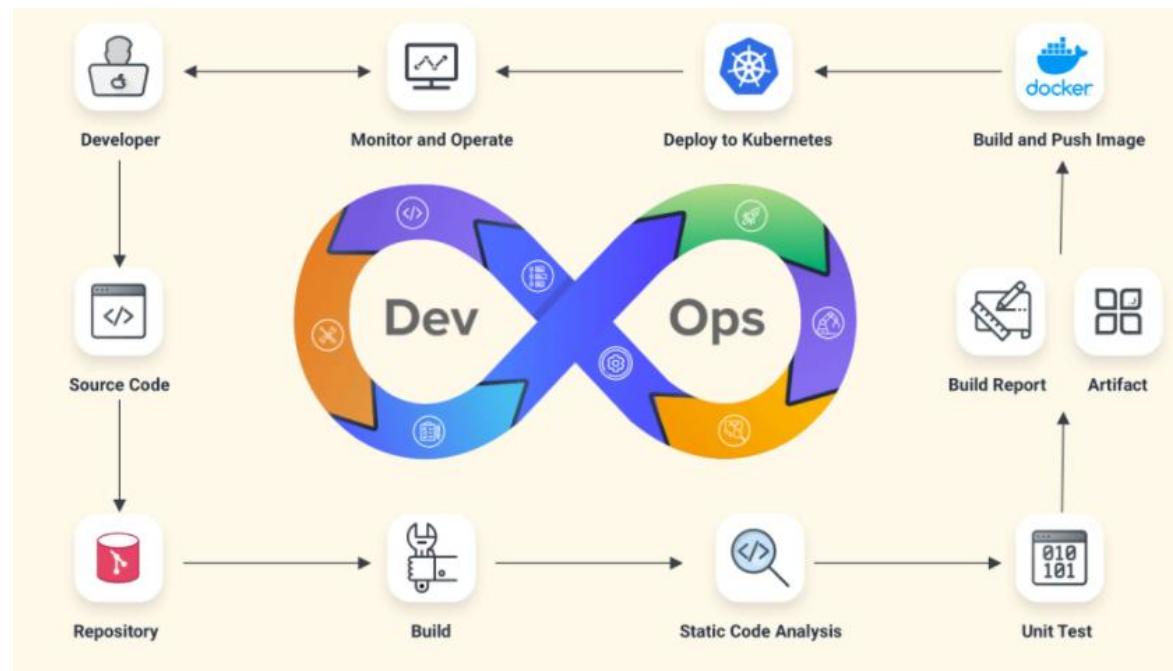
N	O	P	Q	R	S	T	U	V
1	Ca (Sim/Não)	URL do Jira	IP Dev	IP Hml	IP Prod	usuario admi	Url repositorio git	Bugs conhecido
2		https://jira.alpb.com/helpdesk	10.83.16.37	142.119.171.246	24.98.31.115	admin	https://alpbdev.git.com/helpdesk	7
3		https://jira.alpb.com/alpbdoc	10.83.16.185	110.127.101.182	7.27.191.102	principal	https://alpbdev.git.com/alpbdoc	0
4		https://jira.alpb.com/loa	10.83.16.9	107.102.120.56	124.154.86.123	principal	https://alpbdev.git.com/loa	0
5		https://jira.alpb.com/alpbdoc	10.83.16.46	46.47.106.51	177.175.33.41		https://alpbdev.git.com/alpbdoc	1
6		https://jira.alpb.com/sapl	10.83.16.161	144.182.22.84	16.119.7.64	principal	https://alpbdev.git.com/sapl	15
7		https://jira.alpb.com						
8		https://jira.alpb.com						
9		G	H	I	J	K	L	M
10		I	Dono do Produto	Tecnologias	Versão Java	Versão Spring Boot	Versão Angular	BD
11		ra	Lucas Eduardo	Spring Boot, Java, Angular	1.8	3.5.5	12.2	Postgres 9.3
12		3	Júlia Isabel	Python, Django			Zope	Não
13		4	Laura Fernanda	Delphi			Postgres 9.3	Não
14		5	Arthur Felipe	Java, Angular, Spring Boot	1.9	3.0.1	14.2	MySQL 4
15		6	Matheus Gustavo	Angular, Spring Boot, Java	1.13	3.5.0	15.0	Postgres 16
		7	Sofia Beatriz	Spring Boot, Java, Angular	1.13	3.3.2	14.3	MySQL 5
		8	Isabella Vitória	Spring Boot, Java, Angular	1.17	3.4.10	10.0	Postgres
		9	Alice Gabriela				9.3	Não
		10	Bruno Rodrigo	Java, Angular, Spring Boot	1.20	3.2.6	13.2	Postgres 16
		11	Daniel André	Angular, Spring Boot, Java	1.8	3.0.3	14.3	Postgres
		12	Marcos Vinícius			3.2.4	10.1	Postgres
		13	Laura Fernanda	Spring Boot, Angular, Java	1.9	3.5.1	12.1	MySQL 5
		14	Laura Fernanda					Postgres
		15	Laura Fernanda	Angular, Spring Boot, Java	1.14	3.5.0	13.3	Postgres

2. AVALIAÇÃO DO CENÁRIO ATUAL

- 1 Levantar todos os IT Elements disponibilizados pela Divisão de Sistemas
- 2 **Identificar principais gargalos e problemas** no processo atual
- 3 **Mapear o fluxo de trabalho**
- 4 **Identificar ferramentas e processos críticos** a serem melhorados
- 5 Definir KPIs iniciais para medir progresso

2. AVALIAÇÃO DO CENÁRIO ATUAL

Identificar principais gargalos e problemas
Mapear o fluxo de trabalho
Identificar ferramentas e processos críticos a serem melhorados



2. AVALIAÇÃO DO CENÁRIO ATUAL

- 1 Levantar todos os IT Elements disponibilizados pela Divisão de Sistemas
- 2 Identificar principais gargalos e problemas no processo atual
- 3 Mapear o fluxo de trabalho
- 4 Identificar ferramentas e processos críticos a serem melhorados
- 5 Definir KPIs iniciais para medir progresso

2. AVALIAÇÃO DO CENÁRIO ATUAL

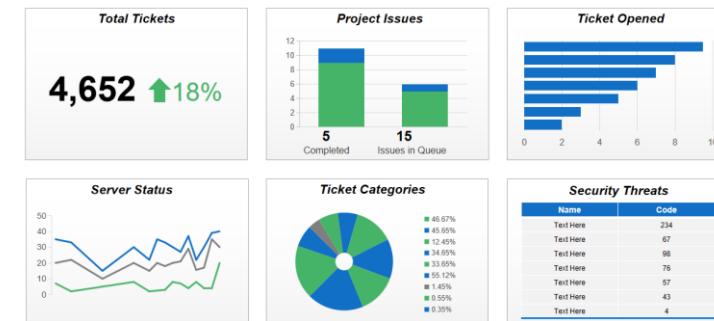
Definir KPIs iniciais para medir progresso

KPI (Key Performance Indicator), ou Indicador Chave de Desempenho, é uma métrica usada para avaliar o sucesso de uma Organização. Os KPIs ajudam a medir a eficiência, a qualidade e o impacto dos projetos de TI, facilitando a tomada de decisões informadas e o alinhamento com as metas institucionais.



Objetivo	Indicador	Meta	Iniciativas
Crescimento da receita	% de crescimento sobre o ano anterior.	Crescer mensalmente 2,21% da receita sobre o mês anterior.	Captar novos clientes.
Retorno sobre o investimento (ROI)	% de lucro líquido sobre os investimentos	Aumentar em 5% o retorno sobre o investimento	Eliminar desperdícios. Elevar a produtividade dos ativos.
Elevar a demanda	Taxa de conversão	Elevar em 10% o número de clientes efetivos	Introduzir um programa de fidelização. Melhorar o atendimento.
Aumentar a participação do mercado	Market Share	Crescer de 3% para 5% ao final do ano.	Aumentar a divulgação. Oferecer novo produto.

Information Technology KPI Dashboard Showing Project...



This graph/chart is linked to excel, and changes automatically based on data. Just left click on it and select "Edit Data".

2. AVALIAÇÃO DO CENÁRIO ATUAL

Definir KPIs iniciais para medir progresso

Satisfação do Cliente (CSAT)

Índice de satisfação dos clientes (internos ou externos) em relação às funcionalidades entregues.

Taxa de redução de vulnerabilidades OWASP

Taxa de Adoção de Novas Funcionalidades

Percentual de usuários que adotam uma nova funcionalidade após o lançamento.

Taxa de Conclusão de Tarefas no Prazo

Percentual de tarefas que foram completadas dentro do prazo estipulado.

Velocidade da Equipe (Story Points por Sprint)

Quantidade de pontos de história (story points) entregues pela equipe durante uma sprint.

Tempo Médio de Resposta da API

Média do tempo de resposta das APIs em produção, visando garantir a eficiência do sistema

Índice de Satisfação da Equipe (eNPS - Employee Net Promoter Score) e CSAT

Pontuação que mede a probabilidade de um membro da equipe recomendar o ambiente de trabalho para outras pessoas, indicando o nível de satisfação e motivação.

Número de Bugs Reportados por Mês

Quantidade de bugs identificados e reportados pelos usuários ou pela equipe interna durante o mês.

3. INTRODUÇÃO DE FERRAMENTAS E PRÁTICAS INICIAIS

- 1 Adoção de Jira na criação de projetos
- 2 Configurar pipeline CI/CD* básico
- 4 Criar scripts para automação de testes e deploy
- 5 Testar deploy contínuo em ambiente de desenvolvimento

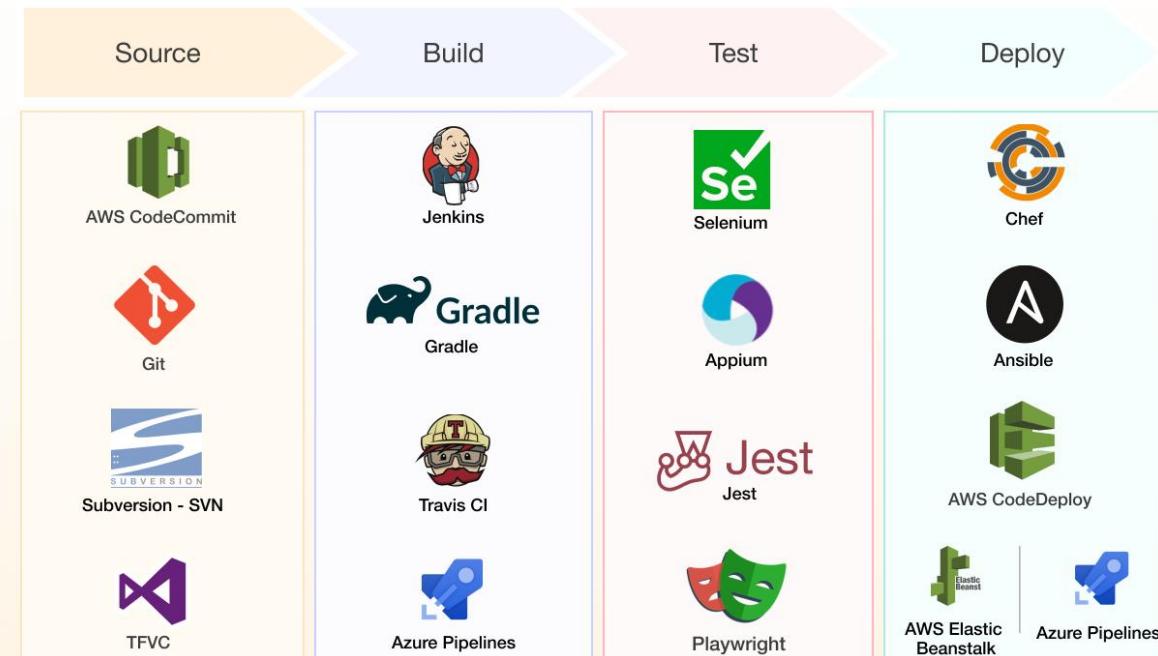
3. INTRODUÇÃO DE FERRAMENTAS E PRÁTICAS INICIAIS

Configurar pipeline CI/CD* básico
Criar scripts para automação de testes e deploy
Testar deploy contínuo em ambiente de desenvolvimento

Um **Pipeline de Desenvolvimento** é uma sequência automatizada de processos que envolve a criação, teste, integração e entrega de código. O objetivo é garantir que o código seja construído, testado e liberado de maneira rápida, eficiente e com alta qualidade.

O pipeline garante que as mudanças sejam implementadas de forma contínua e segura.

Stages of a CI/CD Pipeline



Configurar pipeline CI/CD* básico

Integração Contínua (CI - Continuous Integration):

- Código fonte:** O desenvolvedor escreve e envia o código para um repositório central (como Git).

- Build:** O pipeline executa uma série de etapas automáticas para compilar o código e gerar artefatos (ex: executáveis, pacotes, etc.).

- Testes:** O código gerado é automaticamente testado para garantir que não há falhas. Isso pode incluir testes unitários, de integração, entre outros.

Stage View

Average stage times:
(Average full run time: ~1min 19s)

#140 Mar 02 08:51 No Changes

#139 Mar 02 08:17 No Changes

#138 Mar 02 00:22 No Changes

#137 Mar 01 21:42 No Changes

Build docker images	Unit Test ubuntu-python27	Unit Test centos6-python26	Unit Test debian-wheezy using python-3.4.3
4s	31s	12s	31s
5s	32s failed		
4s	30s	12s	31s
4s	30s	12s	31s

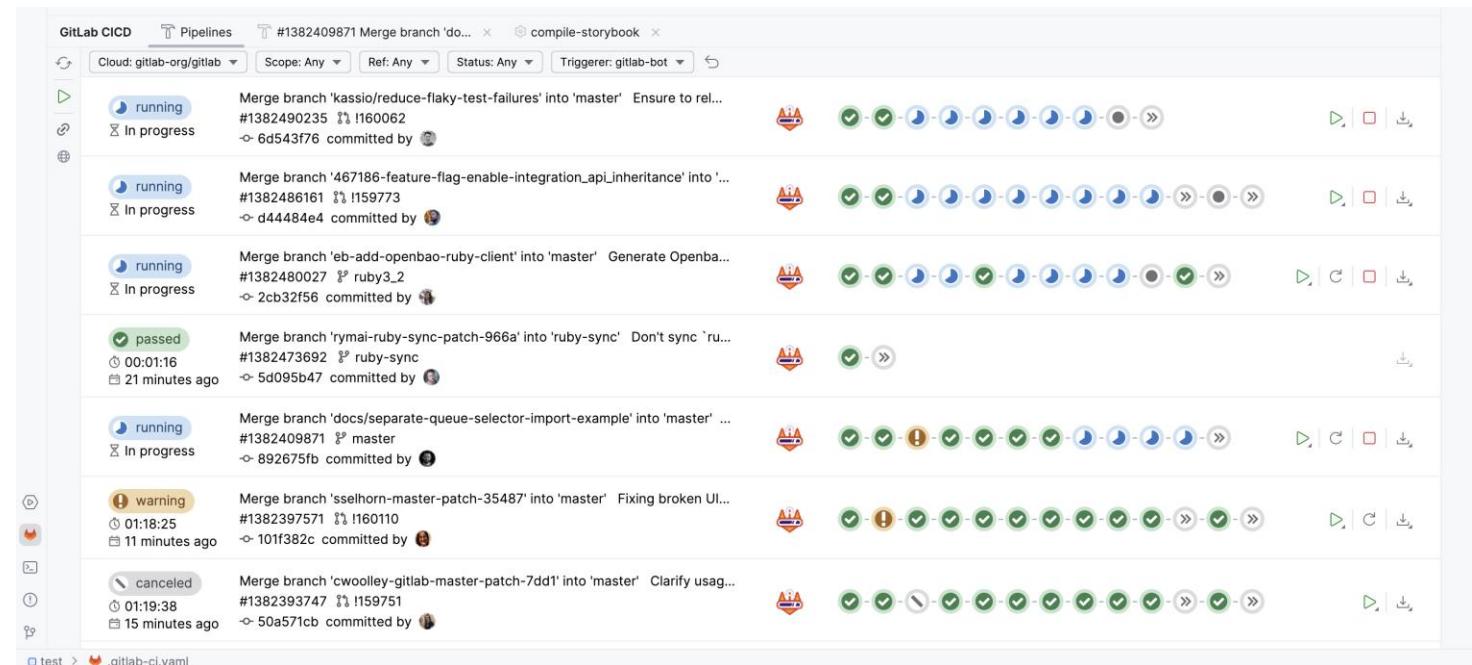
3. INTRODUÇÃO DE FERRAMENTAS E PRÁTICAS INICIAIS

Configurar pipeline CI/CD* básico

Entrega Contínua (CD - Continuous Delivery):

•Deploy de Teste: O código é implantado em ambientes de teste ou homologação, onde pode ser validado mais a fundo, por exemplo, por testers ou clientes internos.

•Testes de Aceitação: Testes adicionais, como testes de aceitação do usuário (UAT), podem ser executados para verificar se o código atende aos requisitos de negócios.



3. INTRODUÇÃO DE FERRAMENTAS E PRÁTICAS INICIAIS

Configurar pipeline CI/CD* básico

AUTOMATIZAÇÃO

Reduz a necessidade de intervenção manual, tornando o processo mais rápido e menos sujeito a erros humanos.

FEEDBACK RÁPIDO

Permite detectar e corrigir problemas de forma mais rápida e eficiente, como falhas no código ou falhas nos testes.

ENTREGA CONTÍNUA

Facilita a entrega contínua de código e atualizações para os usuários

QUALIDADE

Através de testes automatizados, reduz a chance de bugs e falhas em produção.

COLABORAÇÃO

Melhor integração entre as equipes de desenvolvimento e operações, promovendo uma colaboração mais eficaz.

MONITORAMENTO

Após a entrega, o desempenho e o comportamento do aplicativo em produção são monitorados

4. AUTOMAÇÃO DE TESTES E QUALIDADE

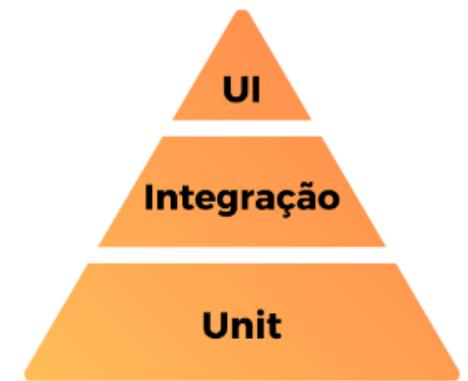
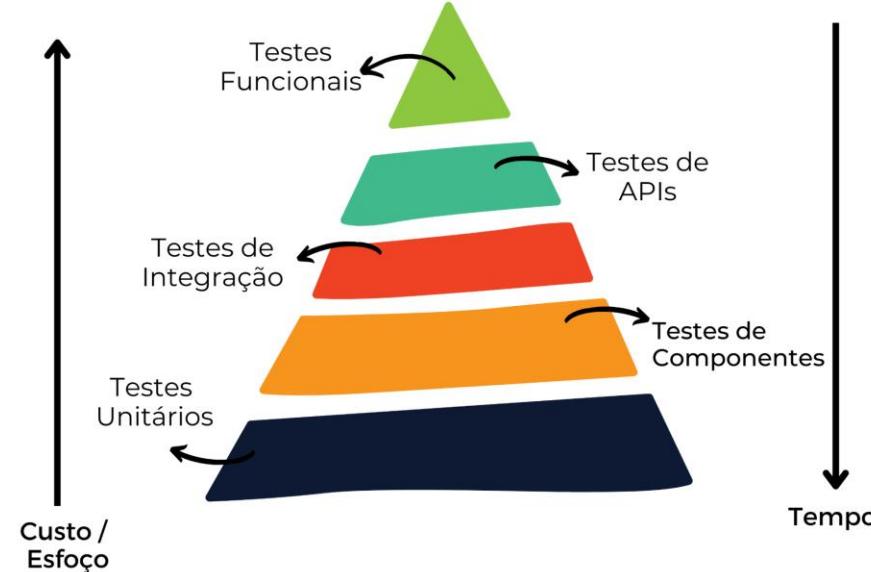
- 1 Implementar automação de **testes unitários e de integração**
- 2 Treinar equipe em **práticas de automação**
- 3 Ajustar **CI/CD** para integrar validação automatizada

4. AUTOMAÇÃO DE TESTES E QUALIDADE

Implementar automação de testes unitários e de integração

Testes unitários
são scripts
automatizados que
verificam o
funcionamento de
unidades individuais de
código, como funções
ou métodos, de forma
isolada do restante do
sistema.

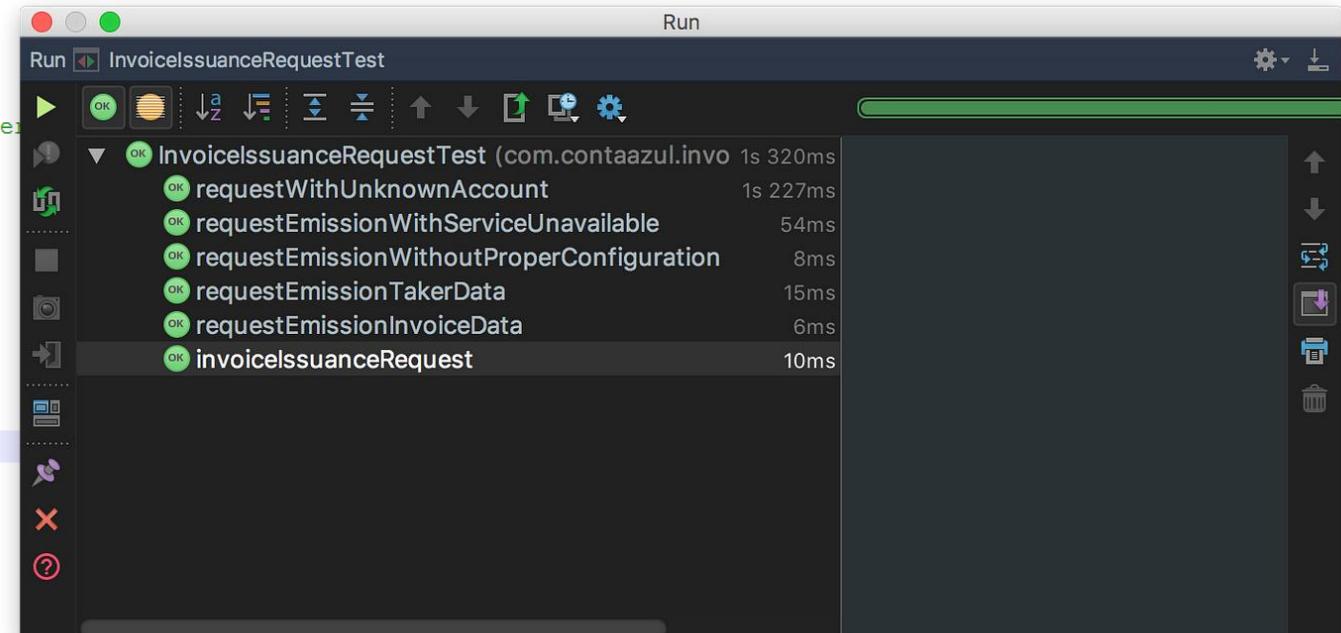
Pirâmide de Testes



4. AUTOMAÇÃO DE TESTES E QUALIDADE

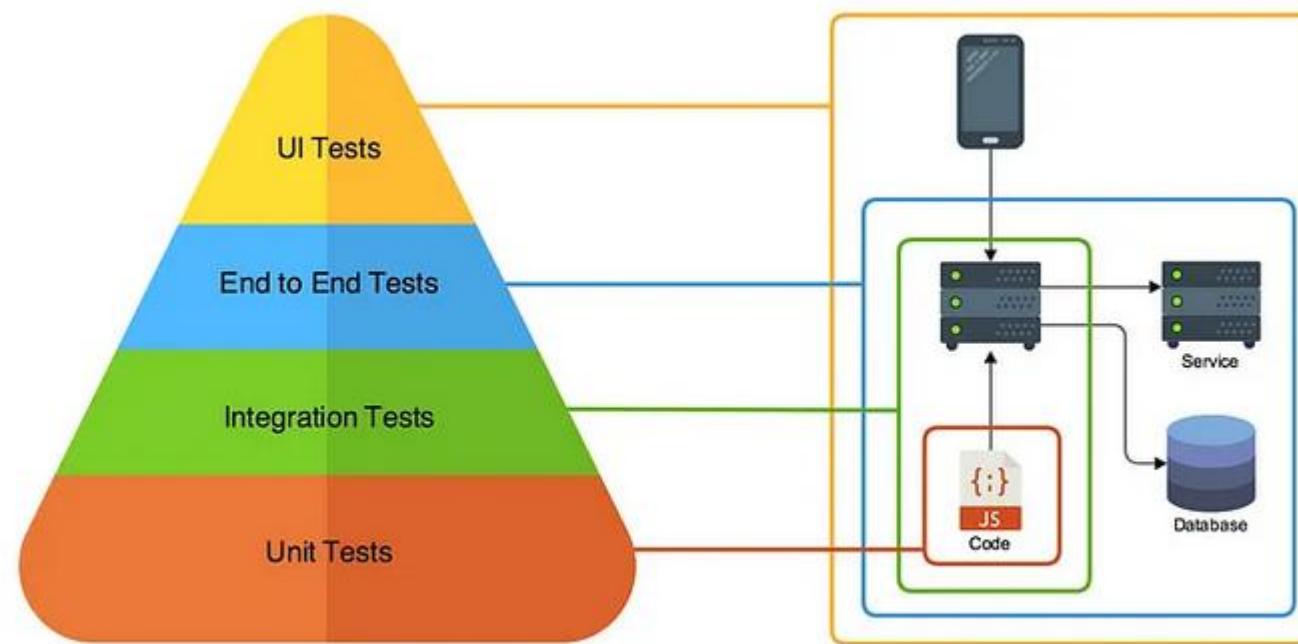
Implementar automação de testes unitários e de integração

```
1 import org.junit.*;
2 public class WriteAUnitTest {
3     // JUnit calls this method one time before all tests
4     @BeforeClass
5     public static void setUp() {
6         // Place code here for any set up required prior to tests
7     }
8     @AfterClass
9     public static void finished() {
10        // Place code here for any clean up that should be done after all tests
11    }
12    @Test
13    public void testFirstName() {
14        Person p=new Person();
15        p.setFirstName("Stephen");
16        Assert.assertEquals("Stephen", p.getFirstName());
17    }
18    @Test
19    public void testLastName() {
20        Person p=new Person();
21        Assert.assertNotNull(p.getLastName());
22    }
23 }
```



4. AUTOMAÇÃO DE TESTES E QUALIDADE

Implementar automação de testes unitários e de integração



4. AUTOMAÇÃO DE TESTES E QUALIDADE

1 Implementar automação de **testes unitários e de integração**

3 **Ajustar CI/CD** para integrar validação automatizada

2 Treinar equipe em **práticas de automação**

5. AUTOMAÇÃO DE TESTES E QUALIDADE

Ajustar CI/CD para integrar validação automatizada

Jenkins > nightwatch-sample > blog/xunit > #7 > Test Results ENABLE AUTO REFRESH

Test Result
4 failures (+4), 16 skipped (+16) 40 tests (+20)
Took 3 min 43 sec.

All Failed Tests

Test Name	Duration	Age
CHROME_48_0_2564_97_Linux_.guineaPig_1.Guinea Pig Assert Title 1 - A	4.3 sec	1
Error Details Testing if the page title equals "I am a page title - Sauce Labs - A".		
Stack Trace		
FIREFOX_45_0_LINUX_.guineaPig_1.Guinea Pig Assert Title 1 - A	7.1 sec	1
Error Details Testing if the page title equals "I am a page title - Sauce Labs - A".		
Stack Trace		
INTERNET EXPLORER_11_WINDOWS_.guineaPig_1.Guinea Pig Assert Title 1 - A	7.9 sec	1
MICROSOFTEDGE_undefined_ANY_.guineaPig_1.Guinea Pig Assert Title 1 - A	16 sec	1

All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
CHROME_48_0_2564_97_Linux_	24 sec	1	+1	4	+4	5		10	+5
FIREFOX_45_0_LINUX_	42 sec	1	+1	4	+4	5		10	+5
INTERNET_EXPLORER_11_WINDOWS_	46 sec	1	+1	4	+4	5		10	+5
MICROSOFTEDGE_undefined_ANY_	1 min 49 sec	1	+1	4	+4	5		10	+5

New Failures	Chart	See children	Build Number => Package-Class-Testmethod names ↓	11	10	9	8	7
	■	-	(root)	2983.539	74.351	74.435	2995.979	3005.566
	■	+	001-002_Create_and_Edit	2983.539	74.351	74.435	2995.979	3005.566
	■	+	010_Creating_a_Form_206	N/A	N/A	N/A	N/A	3005.566
	■	+	010_Creating_a_Product_revision_2	N/A	N/A	N/A	N/A	3005.566
	■	+	010_Creating_a_Project_14	N/A	N/A	N/A	N/A	3005.566
	■	+	010_Creating_a_Request_212	N/A	N/A	N/A	N/A	3005.566
	■	+	010_Creating_a_new_Resource_78	N/A	N/A	N/A	N/A	3005.566
	■	+	020_Editing_a_Form_203	N/A	N/A	N/A	N/A	3005.566
	■	+	020_Editing_a_Resource_79	N/A	N/A	N/A	N/A	3005.566
	■	+	030_Copying_a_Form_310	N/A	N/A	N/A	N/A	3005.566
	■	+	030_Copying_a_Product_revision_6	N/A	N/A	N/A	N/A	3005.566
	■	+	030_Copying_a_Resource_89	N/A	N/A	N/A	N/A	3005.566
	■	+	030_Create_User_without_resource_151	N/A	N/A	N/A	N/A	3005.566
	■	+	030_Deleting_a_Project_18	N/A	N/A	N/A	N/A	3005.566
	■	+	031_Creating_a_new_Resource_78	2983.539	74.351	74.435	2995.979	N/A
	■	+	032_Editing_a_Resource_79	2983.539	74.351	74.435	2995.979	N/A
	■	+	033_Copying_a_Resource_89	2983.539	N/A	N/A	2995.979	N/A

5. INFRAESTRUTURA E MONITORAMENTO

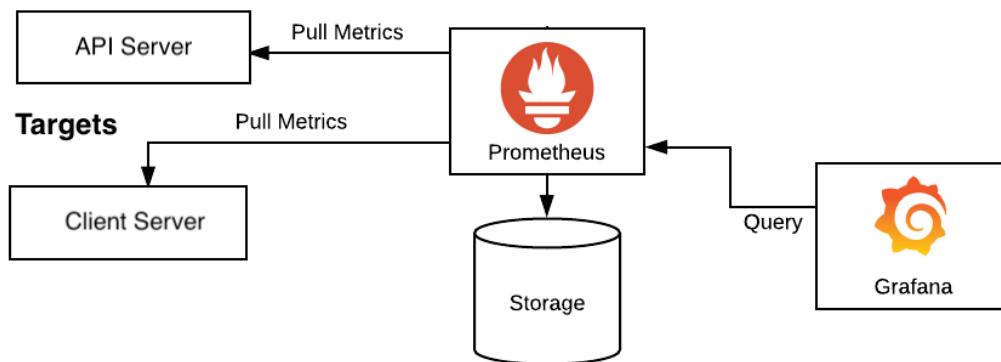
- 1 Configurar **ambiente de monitoramento** dashboard (Prometheus, Grafana)
- 2 Implementar **Infrastructure as Code (IaC)**
- 3 Criar **alertas automáticos** para incidentes críticos
- 4 Padronizar ambiente para **desenvolvimento, homologação e produção**

5. INFRAESTRUTURA E MONITORAMENTO

Configurar **ambiente de monitoramento** dashboard (Prometheus, Grafana) Criar **alertas automáticos** para incidentes críticos

Prometheus coleta métricas de sistemas e aplicativos por meio de scraping (coleta automática) de endpoints HTTP expostos, que oferecem dados em um formato específico. Ele armazena essas métricas em uma base de dados de séries temporais, permitindo consultas eficientes.

Grafana se conecta a diversas fontes de dados (como Prometheus, Elasticsearch, InfluxDB, etc.) e permite que você crie visualizações dinâmicas para acompanhar métricas e logs.

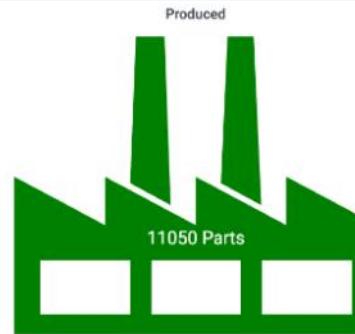


Country All City All Plant All Machine All Maintenance



Number of plants

4

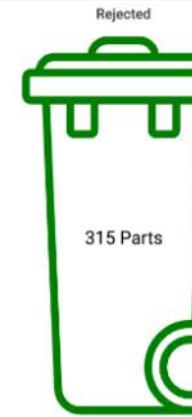


Next maintenance (Days)

5

Number of machines

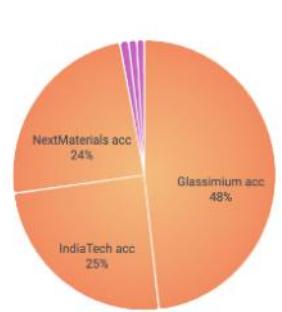
32



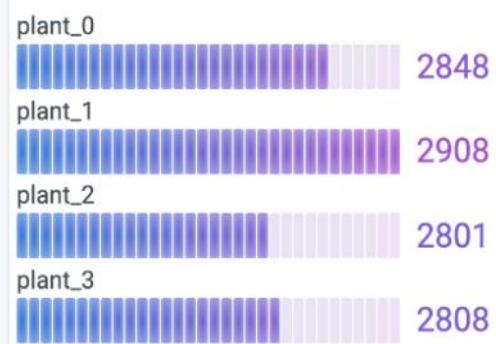
Productivity by country



Material Providers



Productivity by plant



Yield by plant



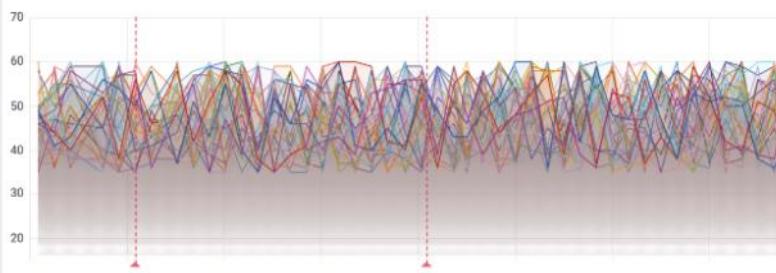
Yield by material



Yield by machine



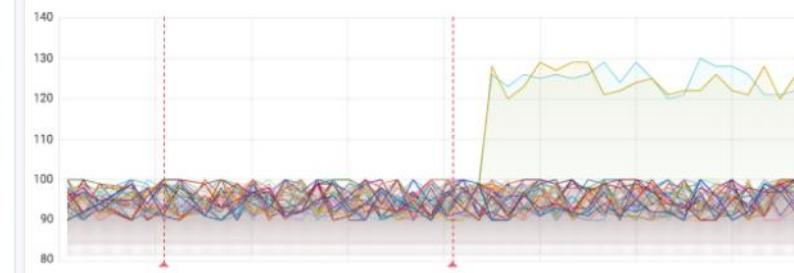
Pressure



Legend for Pressure:

- pressure machine_0_0
- pressure machine_0_1
- pressure machine_0_2
- pressure machine_0_3
- pressure machine_0_4
- pressure machine_0_5
- pressure machine_0_6
- pressure machine_0_7
- pressure machine_1_0
- pressure machine_1_1
- pressure machine_1_2
- pressure machine_1_3
- pressure machine_1_4
- pressure machine_1_5
- pressure machine_1_6
- pressure machine_1_7
- pressure machine_2_0
- pressure machine_2_1
- pressure machine_2_2
- pressure machine_2_3
- pressure machine_2_4
- pressure machine_2_5
- pressure machine_2_6
- pressure machine_2_7
- pressure machine_3_0
- pressure machine_3_1
- pressure machine_3_2
- pressure machine_3_3
- pressure machine_3_4
- pressure machine_3_5
- pressure machine_3_6
- pressure machine_3_7

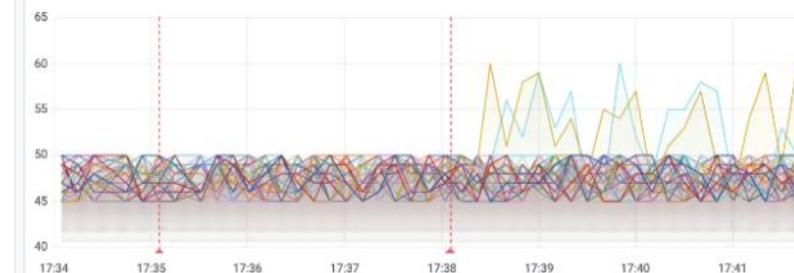
Vibration



Legend for Vibration:

- vibration machine_0_0
- vibration machine_0_1
- vibration machine_0_2
- vibration machine_0_3
- vibration machine_0_4
- vibration machine_0_5
- vibration machine_0_6
- vibration machine_0_7
- vibration machine_1_0
- vibration machine_1_1
- vibration machine_1_2
- vibration machine_1_3
- vibration machine_1_4
- vibration machine_1_5
- vibration machine_1_6
- vibration machine_1_7
- vibration machine_2_0
- vibration machine_2_1
- vibration machine_2_2
- vibration machine_2_3
- vibration machine_2_4
- vibration machine_2_5
- vibration machine_2_6
- vibration machine_2_7
- vibration machine_3_0
- vibration machine_3_1
- vibration machine_3_2
- vibration machine_3_3
- vibration machine_3_4
- vibration machine_3_5
- vibration machine_3_6
- vibration machine_3_7

Temperature



Legend for Temperature:

- temperature machine_0_0
- temperature machine_0_1
- temperature machine_0_2
- temperature machine_0_3
- temperature machine_0_4
- temperature machine_0_5
- temperature machine_0_6
- temperature machine_0_7
- temperature machine_1_0
- temperature machine_1_1
- temperature machine_1_2
- temperature machine_1_3
- temperature machine_1_4
- temperature machine_1_5
- temperature machine_1_6
- temperature machine_1_7
- temperature machine_2_0
- temperature machine_2_1
- temperature machine_2_2
- temperature machine_2_3
- temperature machine_2_4
- temperature machine_2_5
- temperature machine_2_6
- temperature machine_2_7
- temperature machine_3_0
- temperature machine_3_1
- temperature machine_3_2
- temperature machine_3_3
- temperature machine_3_4
- temperature machine_3_5
- temperature machine_3_6
- temperature machine_3_7

5. INFRAESTRUTURA E MONITORAMENTO

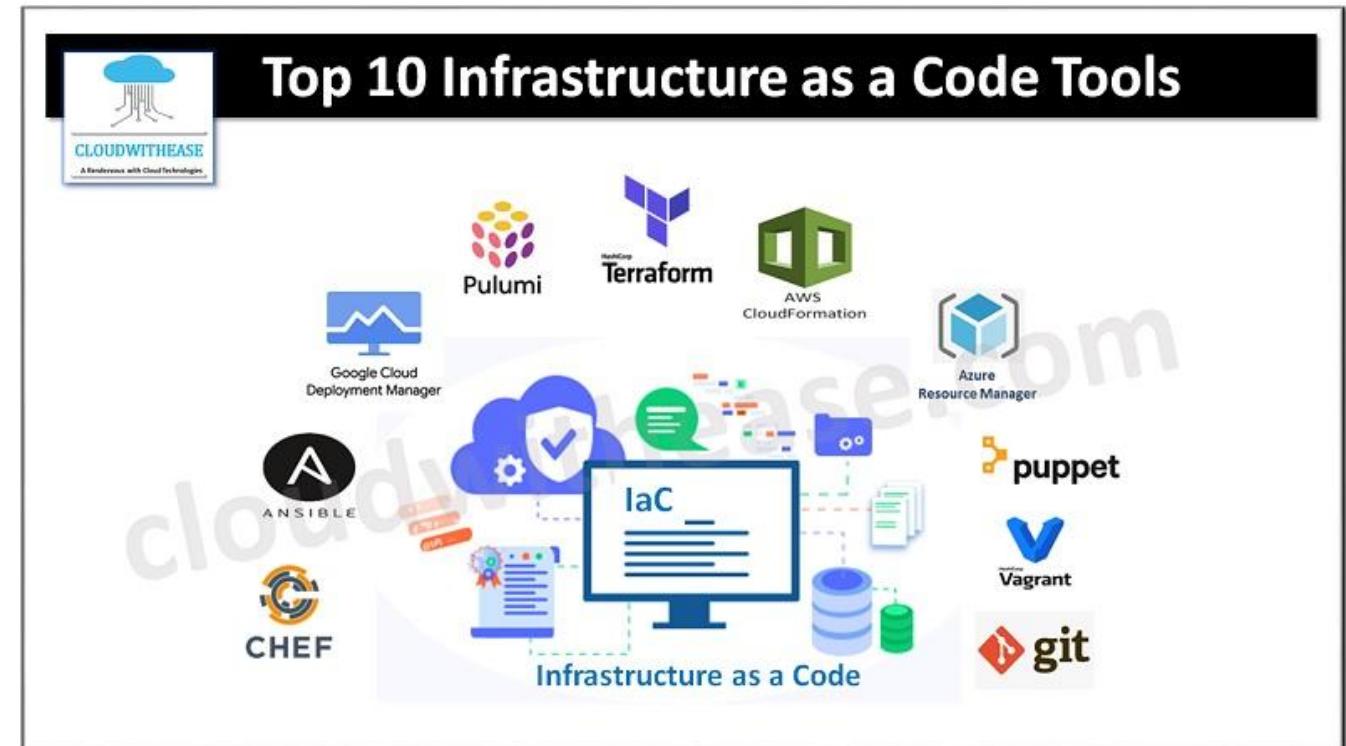
- 1 Configurar **ambiente de monitoramento** dashboard (Prometheus, Grafana)
- 2 Implementar **Infrastructure as Code (IaC)**
- 3 Criar **alertas automáticos** para incidentes críticos
- 4 Padronizar ambiente para desenvolvimento, homologação e produção

5. INFRAESTRUTURA E MONITORAMENTO

Implementar Infrastructure as Code (IaC)

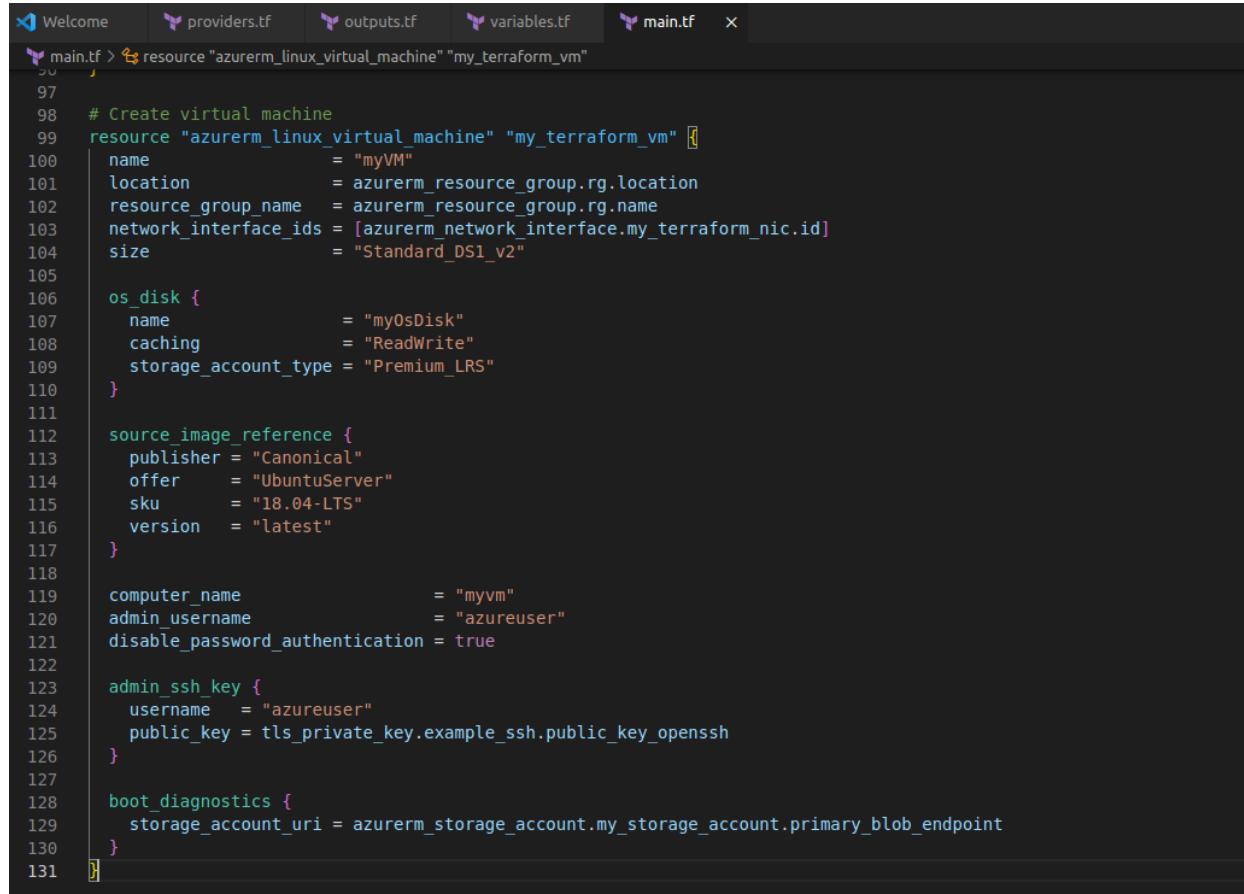
Infrastructure as Code (IaC) é uma prática de **DevOps** que permite gerenciar e provisionar recursos de infraestrutura (como servidores, redes, armazenamento) usando código e scripts, em vez de configuração manual ou ferramentas de gerenciamento tradicionais.

Em resumo, IaC permite que a infraestrutura seja tratada da mesma forma que o código de um aplicativo, proporcionando maior controle, automação e consistência no gerenciamento dos recursos.



5. INFRAESTRUTURA E MONITORAMENTO

Implementar Infrastructure as Code (IaC)

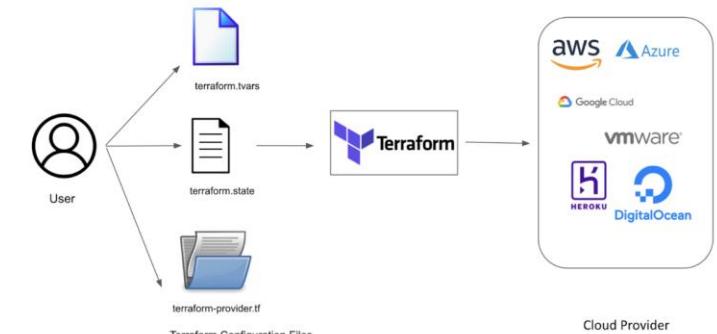


```

  Welcome providers.tf outputs.tf variables.tf main.tf ×

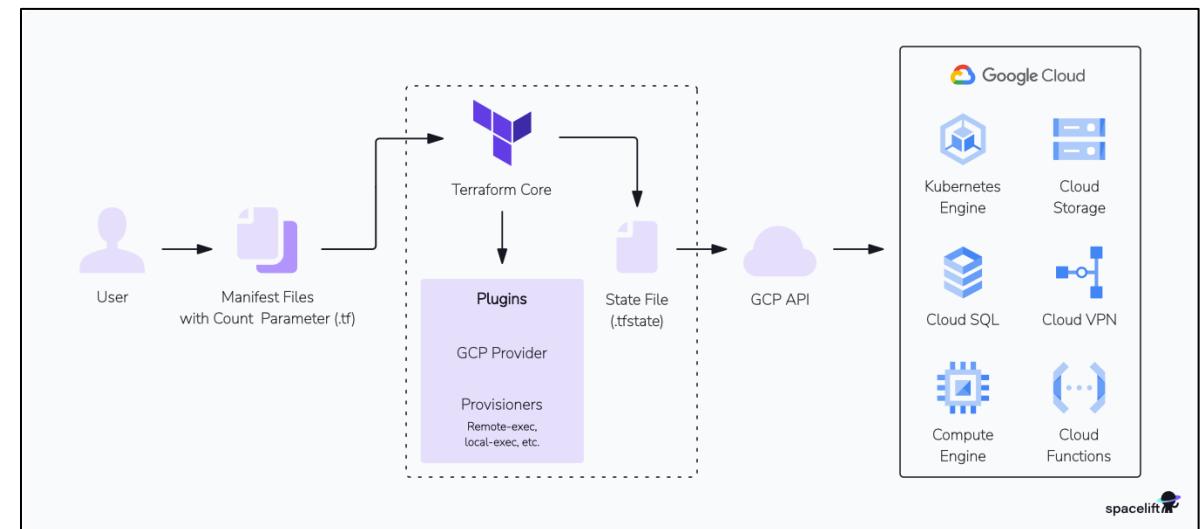
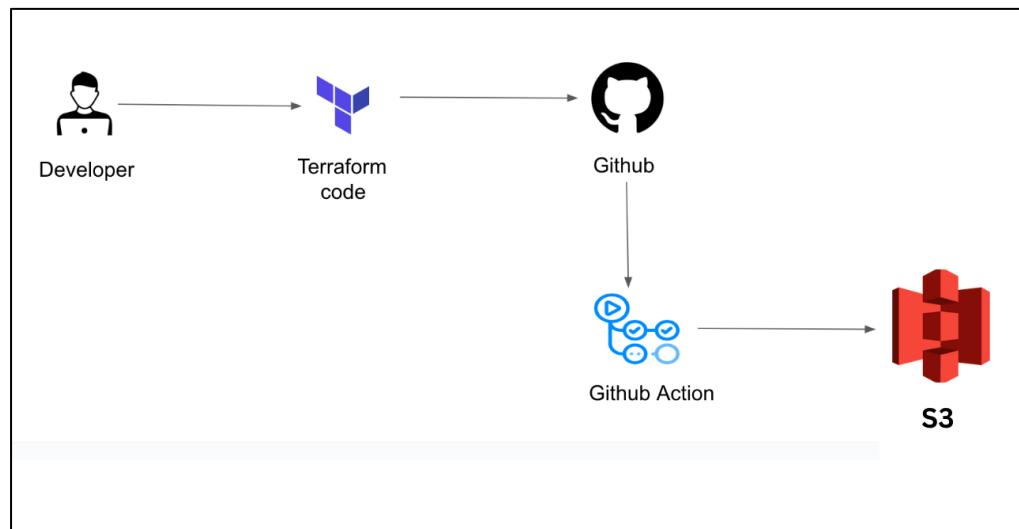
  main.tf > resource "azurerm_linux_virtual_machine" "my_terraform_vm"
  50
  97
  98 # Create virtual machine
  99 resource "azurerm_linux_virtual_machine" "my_terraform_vm" {
  100   name          = "myVM"
  101   location       = azurerm_resource_group.rg.location
  102   resource_group_name = azurerm_resource_group.rg.name
  103   network_interface_ids = [azurerm_network_interface.my_terraform_nic.id]
  104   size           = "Standard_DS1_v2"
  105
  106   os_disk {
  107     name          = "myOsDisk"
  108     caching        = "ReadWrite"
  109     storage_account_type = "Premium_LRS"
  110   }
  111
  112   source_image_reference {
  113     publisher      = "Canonical"
  114     offer          = "UbuntuServer"
  115     sku            = "18.04-LTS"
  116     version        = "latest"
  117   }
  118
  119   computer_name      = "myvm"
  120   admin_username     = "azureuser"
  121   disable_password_authentication = true
  122
  123   admin_ssh_key {
  124     username      = "azureuser"
  125     public_key    = tls_private_key.example_ssh.public_key_openssh
  126   }
  127
  128   boot_diagnostics {
  129     storage_account_uri = azurerm_storage_account.my_storage_account.primary_blob_endpoint
  130   }
  131 }

```



5. INFRAESTRUTURA E MONITORAMENTO

Implementar Infrastructure as Code (IaC)



Terraform (v0.14): Automatizando sua infraestrutura na AWS

Aprenda a automatizar sua infraestrutura na AWS com Terraform codificando 3 exemplos práticos do mundo real

Classificação mais alta

4,6 ★★★★★ (1.604 classificações) 7.795 alunos

Criado por Cleber Gasparoto

Última atualização em 07/2024 Português Português [Automático]

O que você aprenderá

- ✓ Será capaz de automatizar a criação e manutenção da infraestrutura de suas aplicações na AWS
- ✓ Criar um site estático com React utilizando AWS S3, Cloudfront, certificado SSL e domínio personalizado
- ✓ Criar uma aplicação serverless utilizando AWS Lambda, DynamoDB, API Gateway, SNS e Cognito
- ✓ Aprender conceitos básicos e avançados como módulos, interpolação, arquivos de estado do Terraform
- ✓ Criar um aplicação auto-escalável com VPC, Internet Gateway, Subnets, Security Groups, EC2, Load Balancer e Auto Scaling Group

v0.14

Pré-visualizar este curso

Pessoal

Equipes

R\$34,90 R\$209,90

83% de desconto

Só mais 5 horas por este preço!

Adicionar ao carrinho

Garantia de devolução do dinheiro em 30 dias

Acesso vitalício total

Compartilhar

Presentear este curso

Aplicar cupom

O código de cupom **2021PM25** foi aplicado
Cupom da Udemy

Inserir cupom

Aplicar

5. INFRAESTRUTURA E MONITORAMENTO

Implementar Infrastructure as Code (IaC)

AUTOMATIZAÇÃO

Desenvolvedores podem rapidamente provisionar e configurar infraestrutura sem intervenção manual, economizando tempo e esforço em tarefas repetitivas. Redução de erros humanos, como configurações inconsistentes ou esquecidas, é minimizada.

CONSISTÊNCIA ENTRE AMBIENTES

ELIMINA O PROBLEMA DE "FUNCIONA NA MINHA MÁQUINA" Desenvolvedores podem criar e testar a infraestrutura de maneira idêntica em diferentes ambientes (desenvolvimento, teste, produção), garantindo que o que funciona localmente também funcione em produção.

VERSIONAMENTO E CONTROLE DE MUDANÇAS

Com IaC, as configurações de infraestrutura podem ser versionadas e auditadas, permitindo que os desenvolvedores acompanhem as mudanças ao longo do tempo. Rollback Simples: Se um erro ocorre após uma alteração na infraestrutura, os desenvolvedores podem facilmente reverter.

ESCALABILIDADE E AGILIDADE

IaC permite que os desenvolvedores criem e escalem recursos de infraestrutura de maneira dinâmica, de acordo com as necessidades do projeto. Com isso, eles podem se concentrar mais no desenvolvimento de funcionalidades.

COLABORAÇÃO E FLEXIBILIDADE

É mais fácil para os desenvolvedores colaborar, pois todos podem acessar, revisar e modificar o código de infraestrutura. IaC facilita o uso de práticas ágeis e DevOps, proporcionando maior colaboração entre equipes de desenvolvimento e operações.

DOCUMENTAÇÃO AUTOMÁTICA

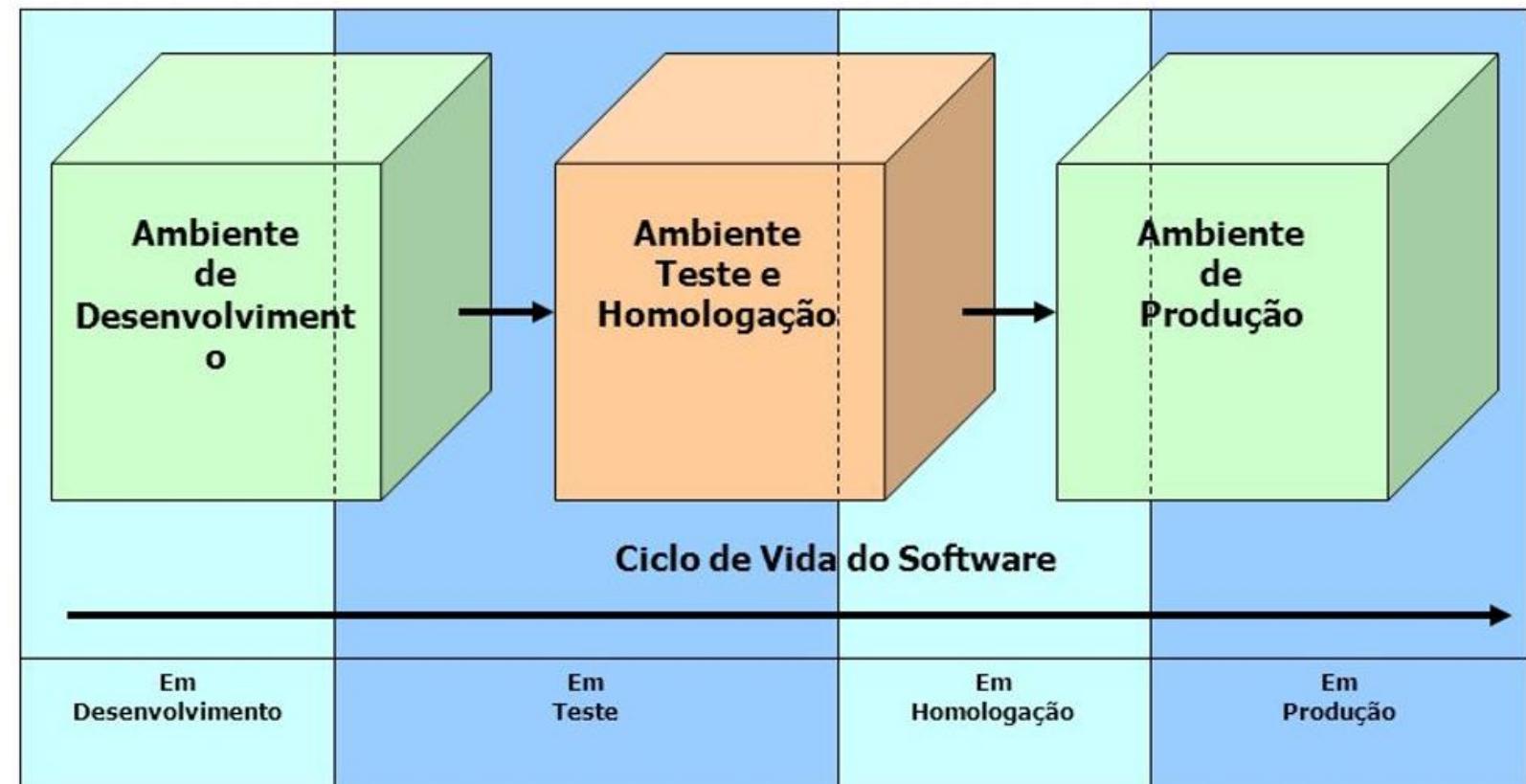
O próprio código serve como documentação da infraestrutura. Não é necessário escrever documentação separada sobre como a infraestrutura foi configurada, pois tudo está codificado.

5. INFRAESTRUTURA E MONITORAMENTO

- 1 Configurar **ambiente de monitoramento** dashboard (Prometheus, Grafana)
- 2 Implementar **Infrastructure as Code (IaC)**
- 3 Criar **alertas automáticos** para incidentes críticos
- 4 Padronizar ambiente para **desenvolvimento, homologação e produção**

Padronizar ambiente para **desenvolvimento, homologação e produção**

- 1 Projeto**
- 1 repositório git**
- 1 pipeline**
- 3 servidores idênticos**
- 3 bancos de dados**
- 3 urls**



6. CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO

- 1 Realizar **treinamentos internos** sobre **ferramentas e práticas DevOps**
- 2 Introduzir ritos ágeis de **feedback**
- 3 Reuniões de retrospectiva para **ajustes no fluxo de trabalho**

6. CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO

Realizar treinamentos internos sobre ferramentas e práticas DevOps

Oficina	Descrição
POSTMAN API	Ferramenta essencial para testar, documentar e automatizar APIs, facilitando a comunicação entre desenvolvedores e integradores.
CULTURA DEVOPS - TURMA 1 E TURMA 2	Promove práticas que integram desenvolvimento e operações, acelerando entregas e melhorando a qualidade de software.
GIT	Sistema de controle de versão distribuído que possibilita o rastreamento de mudanças em projetos, essencial para colaboração e versionamento.
CYBER SECURITY - TURMA 1 E TURMA 2	Foco em práticas e ferramentas para proteger sistemas e dados contra ameaças digitais, imprescindível no cenário atual.
DEBUGANDO APLICAÇÃO JAVA COM INTELLIJ	Demonstra técnicas para identificar e corrigir erros em código Java, otimizando o processo de desenvolvimento.
FRAMEWORKS FRONTEND	Exploração de ferramentas modernas como Angular, React e Vue.js para criar interfaces de usuário dinâmicas e responsivas.
SCRUM EM PROFUNDIDADE PARA TIMES ÁGEIS	Treinamento detalhado para aprimorar a implementação do Scrum, maximizando a eficiência e colaboração em projetos ágeis.
CHATGPT PARA DESENVOLVEDORES	Introdução ao uso do ChatGPT como assistente para acelerar o desenvolvimento e resolver problemas complexos.
SQL BÁSICO	Fundamentos de bancos de dados relacionais, ensinando consultas básicas para manipular e extraír dados.
MÁQUINAS VIRTUAIS	Explica o uso de VMs para isolamento, testes e deploy, otimizando recursos computacionais.
TESTES UNITÁRIOS	Prática fundamental para garantir a qualidade de código, detectando problemas em pequenos componentes do sistema.
INTEGRAÇÃO CONTÍNUA / ENTREGA CONTÍNUA (CI/CD)	Implementação de pipelines automáticos para melhorar o fluxo de desenvolvimento e deploy.
TERRAFORM	Ferramenta de IaC (Infraestrutura como Código) para provisionamento e gerenciamento de infraestruturas de forma eficiente.
SCRUM E METODOLOGIAS ÁGEIS - TURMA 3	Capacitação prática para implementar Scrum e outras metodologias ágeis em diferentes contextos de projetos.
DOCKER E KUBERNETES	Capacita no uso de containers e orquestração, simplificando o deploy e gerenciamento de aplicações.
ARQUITETURA DE MICROSERVIÇOS	Explora padrões e melhores práticas para criar sistemas escaláveis, independentes e resilientes.
CLOUD - TURMA 1 E TURMA 2	Aborda fundamentos e ferramentas para utilizar computação em nuvem, essencial para escalabilidade e disponibilidade.
PROMETHEUS / GRAFANA	Monitoramento e visualização de métricas de sistemas, ajudando na identificação de problemas e na tomada de decisões.
SONARQUBE	Ferramenta para análise estática de código, garantindo qualidade, segurança e redução de débito técnico.
SCRUM E METODOLOGIAS ÁGEIS	
JIRA	
PLANNING POKER E ESTIMATIVAS	

6. CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO

Realizar treinamentos internos sobre ferramentas e práticas DevOps

OFICINA	CH	INSTRUTOR	DATA
SCRUM E METODOLOGIAS ÁGEIS - TURMA 1	1h	Scrum Master	14/11/24
JIRA - TURMA 1	1h		28/11/24
SCRUM E METODOLOGIAS ÁGEIS - TURMA 2	1h		28/11/24
PLANNING POKER E ESTIMATIVAS	30min		05/12/24
JIRA - TURMA 2	1h	Scrum Master	12/12/24
POSTMAN API	1h	Dev Pleno	13/02/25
CULTURA DEVOPS - TURMA 1	30min	Engenheiro DevOps	20/02/25
CULTURA DEVOPS - TURMA 2	30min		27/02/25
GIT	1h	Dev Pleno	06/03/25
CYBER SECURITY - TURMA 1	30min	Especialista em Segurança	06/03/25
CYBER SECURITY - TURMA 2	30min		13/03/25
DEBUGANDO APLICAÇÃO JAVA COM INTELLIJ	1h	Dev Pleno	
FRAMEWORKS FRONTEND	1h	Dev Frontend Pleno	20/03/25
SCRUM EM PROFUNDIDADE PARA TIMES ÁGEIS	1h	Scrum Master	20/03/25
CHATGPT PARA DESENVOLVEDORES	30min	Dev Pleno	27/03/25
SQL BÁSICO	1h	Dev Senior	03/04/25
MÁQUINAS VIRTUAIS	1h	Engenheiro DevOps	10/04/25
TESTES UNITÁRIOS	1h	Dev Senior	17/04/25
INTEGRAÇÃO CONTÍNUA / ENTREGA CONTÍNUA (CI/CD)	2h	Engenheiro DevOps	24/04/25
TERRAFORM	1h	Engenheiro DevOps	25/04/25
SCRUM E METODOLOGIAS ÁGEIS - TURMA 3	1h	Scrum Master	08/05/25
DOCKER E KUBERNETES	2h	Cloud Engineer	15/05/25
ARQUITETURA DE MICROSERVIÇOS	4h		22/05/25
CLOUD - TURMA 1	4h		29/05/25
CLOUD - TURMA 2	4h		05/06/25
PROMETHEUS / GRAFANA	1h	Engenheiro DevOps	12/06/25
SONARQUBE	1h	Engenheiro DevOps	19/06/25

BENEFÍCIOS DAS OFICINAS

Aumento da eficiência: Com mais conhecimento, menos erros e retrabalhos.

Redução do tempo de aprendizado: Novos membros se integram mais rapidamente.

Melhoria na retenção de talentos: Funcionários se sentem mais valorizados e engajados.

Maior inovação: Equipes bem treinadas são mais ágeis e criativas ao resolver problemas.

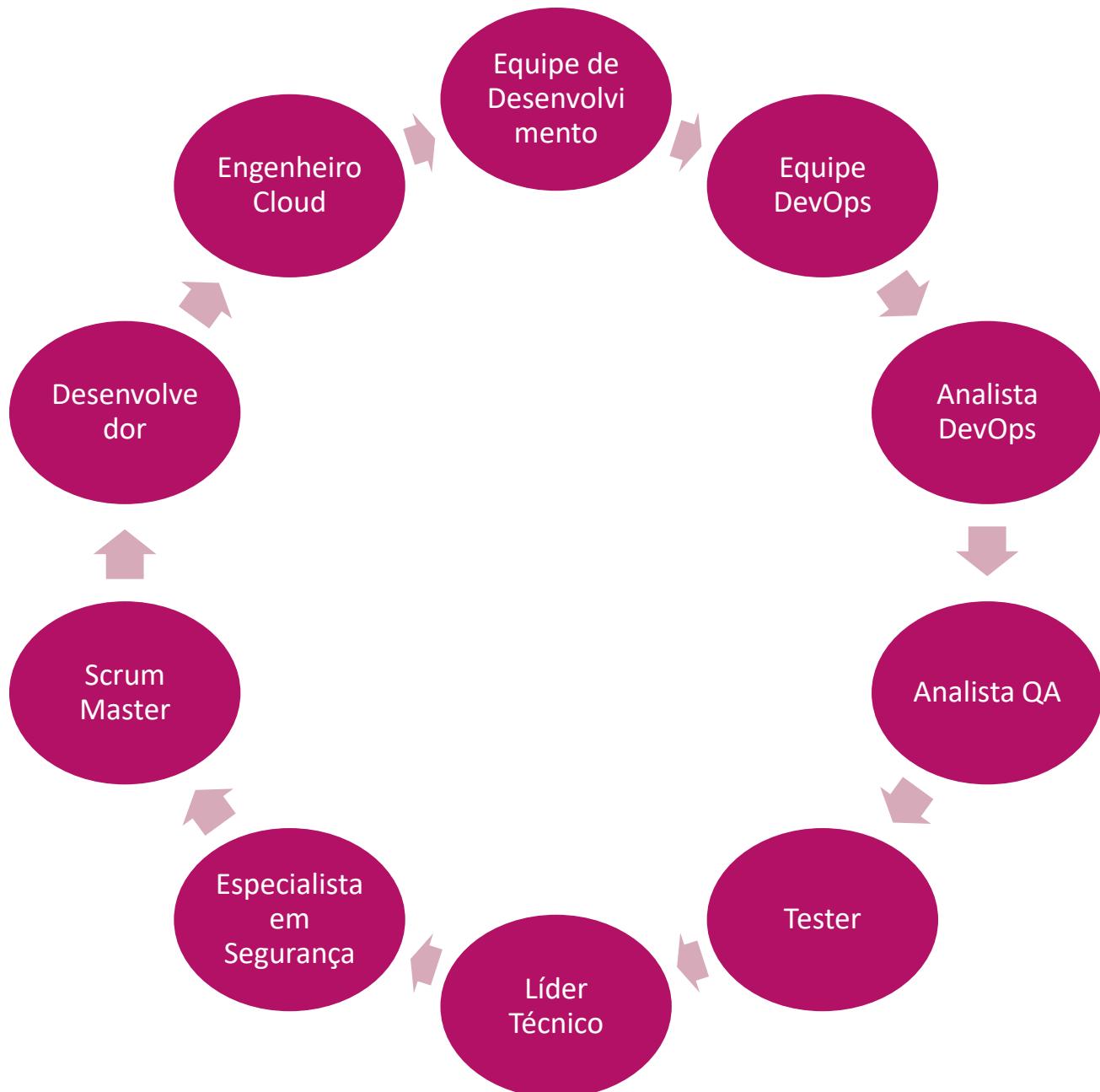
6. CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO

Reuniões de retrospectiva para ajustes no fluxo de trabalho



6. CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO

Realizar **treinamentos internos** sobre ferramentas e práticas **DevOps**



7. REVISÃO E OTIMIZAÇÃO DO PROCESSO

- 1 **Avaliar os KPIs** definidos na Etapa 2
- 2 **Identificar melhorias** adicionais para automação
- 3 **Revisar** práticas e **ajustar** processos conforme necessário

8. EXPANSÃO E MANUTENÇÃO DA CULTURA DEVOPS

1 **Estabelecer** práticas de DevSecOps

3 Implementar processos de **governança e compliance**

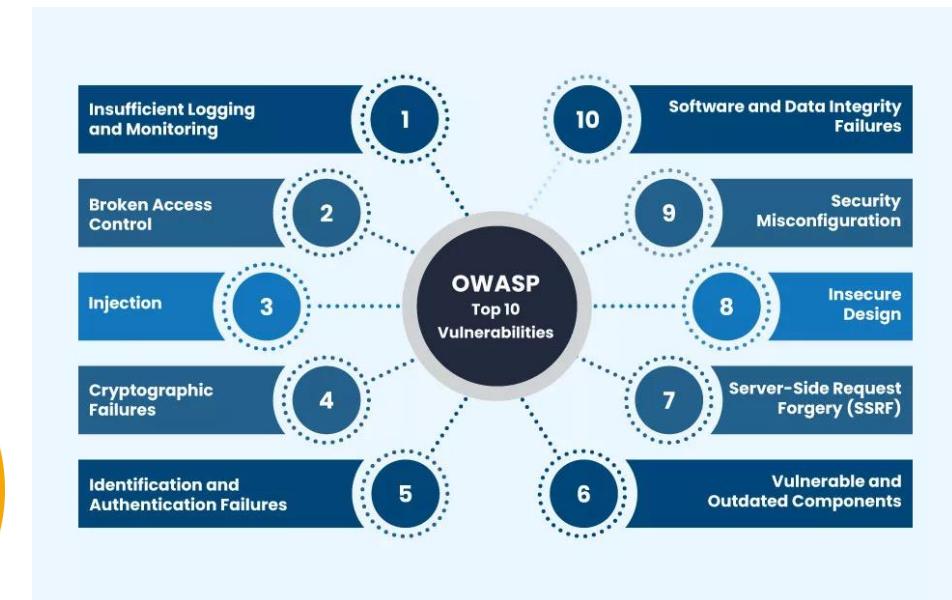
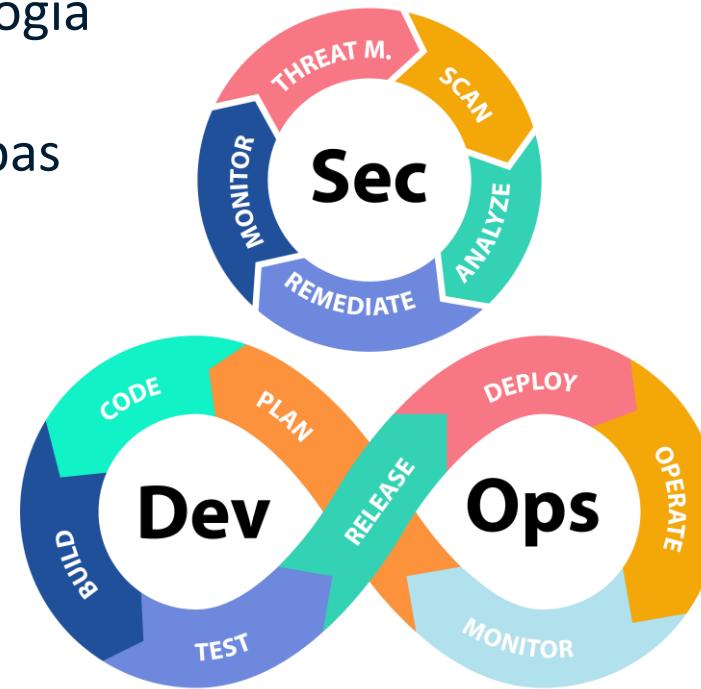
2 **Monitorar** evolução dos KPIs

4 **Atualização contínua** de treinamentos e documentação

8. EXPANSÃO E MANUTENÇÃO DA CULTURA DEVOPS

Estabelecer práticas de DevSecOps

- ▶ DevSecOps é uma metodologia que integra testes de segurança em todas as etapas do desenvolvimento de software.
- ▶ O termo é uma junção das palavras **development**, **security** e **operations**;



8. EXPANSÃO E MANUTENÇÃO DA CULTURA DEVOPS

Estabelecer práticas de DevSecOps



SonarQube Findbugs Plugin

Issues Measures Code Administration ▾

November 28, 2016 2:00 PM Version 3.4.4

Bugs & Vulnerabilities

Leak Period: last 30 days started 2 months ago

Bugs	Vulnerabilities	New Bugs	New Vulnerabilities
0 A	0 A	0	0

FindBugs is a program that uses static analysis to look for bugs in Java code. It can detect a variety of common coding mistakes, including thread synchronization problems, misuse of API methods.

Code Smells

Debt	Code Smells	New Debt	New Code Smells
2d A	73	0	0

started a year ago

Duplications

Duplications	Duplicated Blocks	Duplications on New Code
0.0%	0	—

Version: 3.4.4 November 28, 2016

Quality Gate Passed

(Default) SonarQube way

Quality Profiles

- (Java) Sonar way
- (XML) Sonar way

Filters

Key

org.sonarsource.sonar-findbugs-plugin:sonar

Events

All

Quality Gate: Green (was Red)

November 28, 2016

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Your Project master April 28, 2020, 3:30 PM Version 3.3-SNAPSHOT

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

QUALITY GATE STATUS Passed

MEASURES

New Code	Overall Code
1 Bugs	Reliability
0 Vulnerabilities	Security
6 Security Hotspots	0.0% Reviewed
4d 4h Debt	259 Code Smells
0.0% Coverage on 12k Lines to cover	659 Unit Tests
1.4% Duplications on 30k Lines	32 Duplicated Blocks

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Perspective: Overall Status Sort by: Name Last analysis: January 31, 2019, 4:21 PM

My Favorites All 3 projects

Filters

Quality Gate

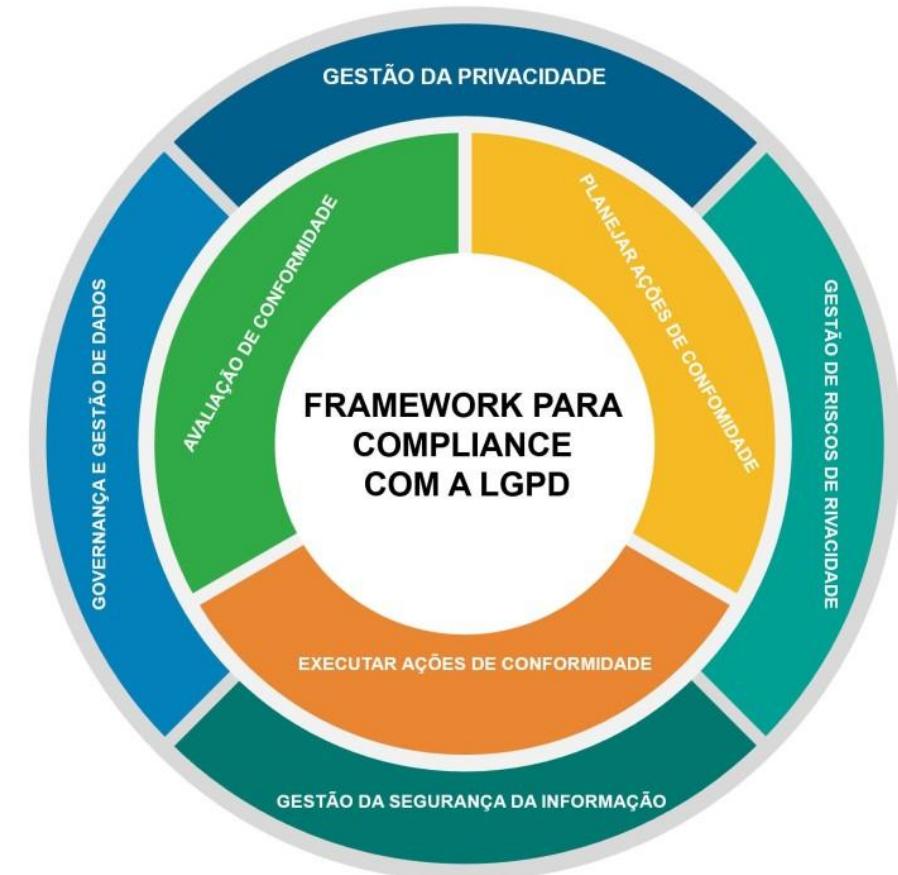
DemoProject	Passed
5 Bugs	6 Vulnerabilities
97 Code Smells	12.7% Coverage
0.6% Duplications	3.8k Lines of Code

Reliability

katalon-api	Passed
0 Bugs	0 Vulnerabilities
5 Code Smells	0.0% Coverage
0.0% Duplications	242 Lines of Code

8. EXPANSÃO E MANUTENÇÃO DA CULTURA DEVOPS

Implementar processos de **governança e compliance**



	SET	OUT – DEZ	JAN – MAR '25	ABR – JUN '25
✓ DOPS-3 FORMAÇÃO E MENTORIA PARA ESTAGIÁRIOS E JUNIOR...				
✓ DOPS-2 Oficina - JIRA CONCLUÍDO PAULLO ES...				
✓ DOPS-11 Ministrar workshop de Scrum e Metod... CONCLUÍDO PAULLO ES...				
✓ DOPS-15 Oficina - SCRUM E METODOLOGIAS... CONCLUÍDO PAULLO ES...				
✓ DOPS-1 Oficina - SCRUM E METODOLOGIAS... CONCLUÍDO PAULLO ES...				
✓ DOPS-14 Oficina - PLANNING POKER E ESTI... CONCLUÍDO PAULLO ES...				
✓ DOPS-16 Programa de obtenção da certificação... EM ANDA... PAULLO ES...				
✓ DOPS-12 Implementar o PDI* para estagiários e... TAREFAS P... PAULLO ES...				
✓ DOPS-4 AVALIAÇÃO DO CENÁRIO ATU...				
✓ DOPS-17 Levantar todos os IT Elements disponibilizados pel... TAREFAS P...				
✓ DOPS-18 Identificar principais gargalos e problemas no proce... TAREFAS P...				
✓ DOPS-19 Mapear o fluxo de trabalho TAREFAS P...				
✓ DOPS-20 Identificar ferramentas e processos críticos a sere... TAREFAS P...				
✓ DOPS-21 Definir KPIs iniciais para medir progresso TAREFAS P...				
✓ DOPS-5 INTRODUÇÃO DE FERRAMENTAS E PRÁTICAS INICIAIS				
✓ DOPS-22 Adoção de Jira na criação de projet... TAREFAS P...				
✓ DOPS-23 Configurar pipeline CI/CD* bási... TAREFAS P...				
✓ DOPS-24 Criar scripts para automação de testes e depl... TAREFAS P...				
✓ DOPS-25 Testar deploy contínuo em ambiente de desenvolvi... TAREFAS P...				
✓ DOPS-6 AUTOMAÇÃO DE TESTES E QUALIDA...				
✓ DOPS-26 Implementar automação de testes unitários e de int... TAREFAS P...				
✓ DOPS-27 Treinar equipe em práticas de automação TAREFAS P...				
✓ DOPS-28 Ajustar CI/CD para integrar validação automatizada TAREFAS P...				
✓ DOPS-7 INFRAESTRUTURA E MONITORAMEN...				
✓ DOPS-29 Configurar ambiente de monitoramento dashboard... TAREFAS P...				
✓ DOPS-30 Implementar Infrastructure as Code (Ia... TAREFAS P...				
✓ DOPS-31 Criar alertas automáticos para incidentes críticos TAREFAS P...				
✓ DOPS-32 Padronizar ambiente para desenvolvimento, homol... TAREFAS P...				

- ✓ DOPS-8 CULTURA DE COLABORAÇÃO E FEEDBACK CONTÍNUO
 - ✓ DOPS-33 Realizar treinamentos internos sobre ferramentas e... [TAREFAS P...](#)
 - ✓ DOPS-34 Introduzir ritos ágeis de feedba... [TAREFAS P...](#)
 - ✓ DOPS-35 Reuniões de retrospectiva para ajustes no fluxo de... [TAREFAS P...](#)
- ✓ DOPS-9 REVISÃO E OTIMIZAÇÃO DO PROCESSO
 - ✓ DOPS-36 Avaliar os KPIs definidos na Etapa... [TAREFAS P...](#)
 - ✓ DOPS-37 Identificar melhorias adicionais para automação [TAREFAS P...](#)
 - ✓ DOPS-38 Revisar práticas e ajustar processos conforme nec... [TAREFAS P...](#)
- ✓ DOPS-10 EXPANSÃO E MANUTENÇÃO DA CULTURA DEVO...
 - ✓ DOPS-39 Estabelecer práticas de DevSecOps [TAREFAS P...](#)
 - ✓ DOPS-40 Monitorar evolução dos KPIs [TAREFAS P...](#)
 - ✓ DOPS-41 Implementar processos de governança e complian... [TAREFAS P...](#)
 - ✓ DOPS-42 Atualização contínua de treinamentos e documenta... [TAREFAS P...](#)
- ✓ DOPS-43 OFICINAS
 - ✓ DOPS-44 Oficina - POSTMAN A... [TAREFAS P...](#)
 - ✓ DOPS-45 Oficina - CULTURA DEVOPS - TURM... [TAREFAS P...](#)
 - ✓ DOPS-46 Oficina - CULTURA DEVOPS - TURM... [TAREFAS P...](#)
 - ✓ DOPS-47 Oficina - GIT [TAREFAS P...](#)
 - ✓ DOPS-48 Oficina - CYBER SECURITY - TURMA... [TAREFAS P...](#)
 - ✓ DOPS-49 Oficina - CYBER SECURITY - TURMA... [TAREFAS P...](#)
 - ✓ DOPS-50 Oficina - DEBUGANDO APLICAÇÃO JAVA COM I... [TAREFAS P...](#)
 - ✓ DOPS-51 Oficina - FRAMEWORKS FRONTEND [TAREFAS P...](#)
 - ✓ DOPS-52 Oficina - SCRUM EM PROFUNDIDADE PARA TIM... [TAREFAS P...](#)
 - ✓ DOPS-53 Oficina - CHATGPT PARA DESENVOLVEDOR... [TAREFAS P...](#)
 - ✓ DOPS-54 Oficina - SQL BÁSIC... [TAREFAS P...](#)
 - ✓ DOPS-55 Oficina - MÁQUINAS VIRTUAIS [TAREFAS P...](#)
 - ✓ DOPS-56 Oficina - TESTES UNITÁRI... [TAREFAS P...](#)
 - ✓ DOPS-57 Oficina - INTEGRAÇÃO CONTÍNUA / ENTREGA C... [TAREFAS P...](#)
 - ✓ DOPS-59 Oficina - TERRAFOR... [TAREFAS P...](#)
 - ✓ DOPS-60 Oficina - SCRUM E METODOLOGIAS ÁGEIS - TU... [TAREFAS P...](#)
 - ✓ DOPS-61 Oficina - DOCKER E KUBERNETES [TAREFAS P...](#)
 - ✓ DOPS-62 Oficina - ARQUITETURA DE MICROSERVIÇ... [TAREFAS P...](#)
 - ✓ DOPS-63 Oficina - CLOUD - TURMA... [TAREFAS P...](#)
 - ✓ DOPS-64 Oficina - CLOUD - TURMA... [TAREFAS P...](#)
 - ✓ DOPS-65 Oficina - PROMETHEUS / GRAFA... [TAREFAS P...](#)
 - ✓ DOPS-66 Oficina - SONARQU... [TAREFAS P...](#)

Áreas de atuação



GERÊNCIA DE
PROJETO



INTEGRAÇÃO E
COMUNICAÇÃO
INTERNA



DEVSECOPS E
AUTOMAÇÃO



TREINAMENTO E
CAPACITAÇÃO

Por que a implantação de DevOps precisa de apoio estratégico?

Decisões sobre processos precisam ser apoiadas por todos os níveis da organização

Implantar DevOps requer mudanças culturais

Ferramentas de DevOps podem ter custos

Aprovação para contratar mais 1 ou 2 pessoas para o time

E-mail: paulloestevam@gmail.com

Linkedin: <https://www.linkedin.com/in/paulloestevam>