 <i>Uso Interno</i>	Tipo de Documento: Procedimento
	Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
	Título do Documento:

Sumário

1. OBJETIVO1

2. ÂMBITO DE APLICAÇÃO1

3. DEFINIÇÕES2

4. DOCUMENTOS DE REFERÊNCIA9

5. RESPONSABILIDADES.....10

6. REGRAS BÁSICAS.....12

7. CONTROLE DE REGISTROS31

8. ANEXOS31

9. REGISTRO DE ALTERAÇÕES31

1. OBJETIVO

+

Essa norma tem como objetivo definir o que é e como funciona o ciclo de vida de Desenvolvimento Seguro de Software (SDLC) para o ambiente da CPFL, bem como estabelecer os parâmetros mínimos aceitáveis para a que um projeto de desenvolvimento de software se enquadre nesse ciclo.


Ela deve ser utilizada um guia para todos os projetos de software que estejam iniciando ou legados que se adaptem as metodologias ágeis de desenvolvimento de aplicações de forma a promover uma maior integração entre os times de desenvolvimento, segurança e operações permitindo entregas de melhor qualidade, mais seguras e mais rápidas.

2. ÂMBITO DE APLICAÇÃO

2.1. Empresa

O **Grupo CPFL Energia**, em seu contexto organizacional, possui um grupo econômico composto por diversas empresas. As distintas regras de negócios, aplicadas aos objetivos de cada empresa do grupo e respectivas áreas organizacionais, ora traduzidas e processadas a

N.Documento:	Categoria:	Versão:	Aprovado por:		
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	1 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Nm, l.ç;] através de seus (i) sistemas críticos/não críticos, (ii) plataformas e (iii) infraestrutura de tecnologia da informação, são efetivos geradores de consumo tecnológico.

Esta prática abrange todo o serviço de Tecnologia, pertencente ao Grupo CPFL Energia, devidamente migrado, identificado no console de gerenciamento, desenvolvido ou adquirido para operação em estrutura Cloud e/ou Multicloud, podendo estar alocado em um contexto de Cloud Pública, Privada ou Híbrida.

2.2. Área

Todo o **Grupo CPFL Energia**.

3. DEFINIÇÕES

3.1. Gerenciamento do Ciclo de Vida de Aplicação (ALM):


O gerenciamento do ciclo de vida de aplicação (ou **ALM**, do inglês *Application Lifecycle Management*), compreende as quatro atividades básicas da engenharia de software que se combinam para formar uma abordagem sistemática que orienta uma aplicação desde a sua especificação até a sua aposentadoria e tem como finalidade melhorar a qualidade das aplicações produzidas, otimizar a produtividade dos times, e facilitar o gerenciamento e a manutenção dos produtos e serviços relacionados.

Segundo (Sommerville, 2019) essas quatro atividades são as seguintes:

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	2 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 <i>Uso Interno</i>	Tipo de Documento: Norma	
	Tecnologia de Informação	
	Área de Aplicação: Tecnologia de Informação	
	Boas Práticas DevSecOps	
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software	

Especificação: É a fase em que clientes e engenheiros definem a aplicação que será produzida, suas funcionalidades e restrições impostas à sua operação.

Desenvolvimento: Nessa etapa as especificações geradas na fase anterior são implementadas gerando uma aplicação funcional ao seu final.

Validação: Na validação a aplicação é analisada para garantir que as funcionalidades implementadas no desenvolvimento atendem as necessidades dos clientes.

Evolução: A fase de evolução serve para que mudanças nas necessidades tanto dos clientes como do mercado sejam refletidas na aplicação.




3.2 Ciclo de Vida de Desenvolvimento Seguro de Software:

O ciclo de vida de desenvolvimento de software (ou SDLC, do inglês *Software Development Life Cycle*) é uma representação simplificada de um modelo de software (Sommerville, 2019) que é criada a partir de perspectiva de modo a fornecer informações parciais sobre o processo de desenvolvimento de softwares. Genericamente, esses modelos são divididos em três categorias, como descrito a seguir.

- 1. **Modelo em cascata:** é o modelo mais tradicional de desenvolvimento de softwares, nele as atividades de especificação, desenvolvimento, validação e evolução são

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	3 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

representadas como fases distintas executadas sequencialmente onde a fase posterior depende da finalização da fase anterior para realizar as suas atividades, tendo como resultado o software pronto para uso.

2. **Desenvolvimento Incremental:** nesse modelo, as atividades de especificação, desenvolvimento e validação são intercaladas de forma que ao final de cada ciclo temos uma versão funcional da aplicação cujas especificações são incrementadas ciclo após ciclo.
3. **Integração e Configuração:** esse processo se baseia na utilização de componentes e sistemas reusáveis, focando na configuração desses componentes de modo que eles possam ser utilizados em novos contextos e na sua integração em um sistema.

Não existe um modelo universal que atenda a todos os projetos de software e a escolha do modelo mais adequado depende de diversos fatores que vão desde o tipo do software desenvolvido até para quem esse software será criado. Contudo devido as suas características entendemos que o modelo mais adequado para a maioria dos projetos conduzidos na CPFL se enquadra no processo de DevOps, que é um modelo de desenvolvimento que incorpora tanto características tanto modelo de desenvolvimento incremental como do modelo de integração e configuração.


3.3 DevOps:

De acordo com (Freeman, 2021), não existe uma definição exata do que é o DevOps, contudo ela entende que o DevOps é uma filosofia que prioriza pessoas a processo e processos a ferramentas, que constrói uma cultura de confiança, colaboração e melhoria contínua, que vê o processo de desenvolvimento de softwares de modo holístico, levando em consideração todos os envolvidos no processo como o pessoal das áreas de negócio, times de desenvolvimento, times de operações e infraestrutura de TI, segurança da informação etc. Como especificado na sessão anterior, entendemos que o DevOps é a melhor abordagem para os projetos de desenvolvimento de softwares executados na CPFL pelos seguintes pontos:

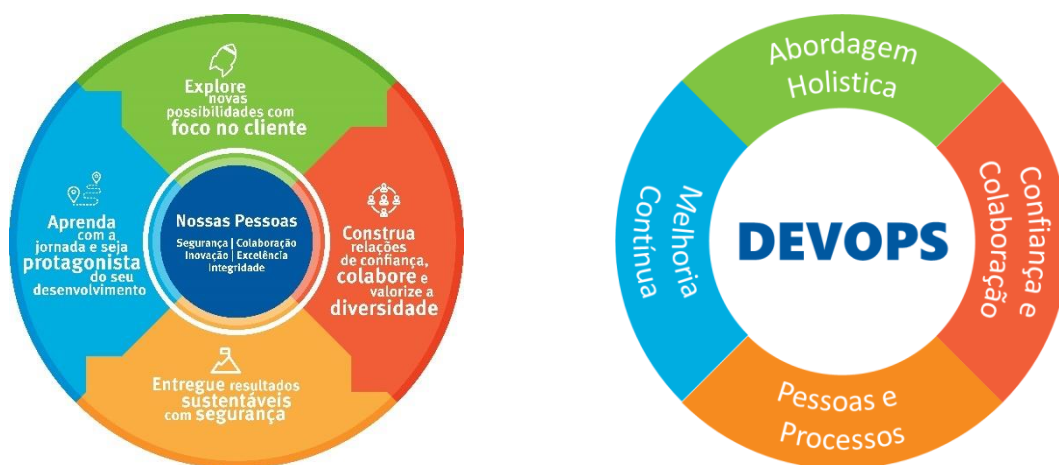
N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	4 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Nome: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Boas Práticas DevSecOps
	Ciclo de Vida de Desenvolvimento Seguro de Software

1. A filosofia do DevOps conversa diretamente com a cultura empresarial da CPFL de acordo com é especificado no nosso jeito de ser. Essa conformidade, permite a rápida assimilação do processo entre os times operacionais e facilita a compreensão da atividade por parte do negócio.




Nossos valores x Filosofia DevOps

2. O DevOps permite na maioria dos casos a entrega de código com mais qualidade, segurança e com um ciclo de tempo inferior quando comparado aos outros processos de SDLC, através da automação dos processos de integração e entrega (CI / CD).
3. O DevOps possibilita a recuperação rápida no caso de incidentes, pois sua metodologia de trabalho permite aos times uma maior integração e maturidade através do aprendizado compartilhado e do monitoramento de todos os processos, performance e riscos.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	5 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

3.4 DevSecOps

O DevSecOps é uma extensão do conceito de DevOps de modo a efetivamente incorporar o mindset de segurança através de todo o processo. O DevSecOps surgiu devido a dificuldade em se criar aplicações e ambientes que atendam às necessidades de segurança exigidas pelos negócios e seus clientes devido ao impacto trazido pela forma que os controles de segurança são implementados atualmente.


Hoje a maioria das empresas possuem times e processos de segurança da informação apartados do pipeline DevOps, fazendo com que as avaliações sobre a segurança das aplicações e dos ambientes de implantação sejam realizadas tardiamente fazendo com que sistemas que já estavam em seus estágios finais de entrega sejam forçados a retroceder para o início do processo para a recodificação ou em casos mais extremos para o replanejamento. Esse tipo de situação trás grandes desafios para o DevOps pois vai de encontro aos conceitos de agilidade por ele pregado fazendo com que a pressão por novas funcionalidades acabe por superar a necessidade de aplicações seguras tendo como resultado os grandes incidentes de segurança que são frequentemente noticiados pela mídia.

A implantação do DevSecOps tem a intenção de trazer a segurança da informação acompanhe a aplicação desde o seu planejamento até a sua implantação, fazendo com que os riscos à segurança da informação e do ambiente sejam mapeados desde o primeiro momento permitindo que eles sejam mitigados durante o processo de criação dos sistemas. Com isso o DevSecOps minimiza o impacto dos controles de segurança não só nos custos e tempo de desenvolvimento como também diminui o risco não mapeados nas funcionalidades da aplicação.

3.5 Pipeline:

É um processo iterativo amplamente automatizado. Você pode considerar um pipeline de DevSecOps como uma linha de montagem para um projeto em que cada estágio, algo é adicionado, removido, balanceados ou testado para garantir um produto de alta qualidade. Os componentes de um Pipeline bem construído consistem em ferramentas que automatizam etapas e permitem os processos de iteração contínua que são:

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	6 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
Título do Documento:	
Ciclo de Vida de Desenvolvimento Seguro de Software	

Integração Contínua (CI, do inglês *Continuous Integration*): A integração contínua é uma prática que permite que o desenvolvedor integre continuamente código alterado ou desenvolvido ao projeto principal. O CI permite que novas funcionalidades sejam adicionadas a aplicação, além de possibilitar que problemas sejam detectados rapidamente e resolvidos rapidamente ainda nos estágios iniciais do desenvolvimento.

Entrega Contínua (CD, do inglês *Continuous Delivery*): A entrega contínua é uma extensão do CI e é uma prática que garante a entrega automatizada do código produzido pela equipe de desenvolvedores para o ambiente de produção em um processo confiável e previsível, ela acelera o mecanismo de feedback, permitindo aos desenvolvedores resolverem problemas rapidamente e com precisão.


Implantação Contínua (*Continuous Deployment*): A implantação contínua é a prática que procura automatizar o processo de liberação do produzido pelos desenvolvedores, eliminando a necessidade de liberações agendadas e minimizando o impacto na performance da aplicação em produção. A implantação contínua é uma prática essencial para se estabelecer processos de integração continua e entrega contínua eficientes.

Feedback Contínuo (*Continuous Feedback*): É a prática de avaliar as atualizações de software com base do feedback do cliente e usuário final. Ela impacta diretamente no resultado final da aplicação e determina o tom para as melhorias da versão atual e a liberação de novas versões.

Monitoramento Contínuo (*Continuous Monitoring*): A prática do monitoramento contínuo tem como objetivo observar e detectar qualquer ameaça de segurança ou problemas de conformidade a tempo e melhorar a eficiência global da aplicação. Essa prática garante que o projeto seja executado sem problemas e com segurança, atendendo a todos os requisitos de conformidade.

Operação Contínua (*Continuous Operations*): Para qualquer sistema, é importante ter uma estratégia de operações para manter a disponibilidade máxima. Objetivo é limitar o tempo de inatividade planejado e evitando paralisações ou incidentes não planejados. A operação contínua permite as organizações publicar constantemente e acelerar a disponibilização de novos produtos.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	7 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

3.6. Escopo

Implementar e operacionalizar a cultura *DevSecOps*, tendo como escopo utilizar Ferramentas, Governança e Pessoas:

FERRAMENTAS:


Aderente e resiliente ao modelo de negócio Grupo CPFL Energia, os processos do *DevSecOps* devem implementar o seguinte ferramental:

- Ferramentas para a abertura de chamados e demandas de modo que todas as atividades possam ser registradas, acompanhadas, medidas e auditadas.
- Ferramentas de controle dos processos de governança como a liberação de mudanças, o controle de ativos, gestão do catálogo de serviços dentre outros, de forma que os processos de negócios e o e as estratégias de serviços possam se manter atualizados na mesma velocidade com que as novas funcionalidades ou novos sistemas são liberados para os usuários.
- Ferramentas para automatização dos processos relativos ao pipeline *DevSecOps*, como sistemas de controle de versão, ferramentas de build e frameworks de testes automatizados, gerenciadores de pacotes, rastreadores de problemas e plataformas de virtualização e containers. De modo que o processo de produção da aplicação desde a criação do código até a sua liberação em produção possa ser ágil e consistente com o mínimo de interação manual.
- Ferramentas de análise de segurança, como aplicações para detecção de vulnerabilidades, sistemas de análise estática de código e dependências, ferramentas de análise dinâmica de aplicação, aplicações para proteção dos endpoints, plataformas de detecção e alerta de comportamentos maliciosos no ambiente, sistemas de controle de tráfego de rede de registro de atividades. De modo que o código produzido livre de erros e atenda as melhores práticas de programação segura e que o processo de publicação da aplicação aconteça dentro das regras de segurança da CPFL e não introduza vulnerabilidades desconhecidas no ambiente.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	8 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

GOVERNANÇA

A governança no ciclo de vida de desenvolvimento seguro de software tem o papel de implementar Cultura DevSecOps, através da criação de processos e metodologias ágeis que possam garantir as entregas dentro dos padrões definidos pelas práticas do DevSecOps, orientando os times envolvidos no processo em como proceder, além de estabelecer métricas que auxiliem a mensurar evolução e o amadurecimento da prática na CPFL.

PESSOAS

As pessoas são os agentes diretos para a execução com sucesso das práticas do DevSecOps na CPFL, são elas que atuam nos Squads para modernização de aplicações legadas e na implantação de novos projetos, participando ativamente na tomada de decisões e na realização das tarefas existentes. Por isso é muito importante que as pessoas compreendam de forma clara a cultura DevSecOps e entendam como participar dela de modo a contribuir de maneira ativa para a disseminação e adoção dos processos do DevSecOps por toda a CPFL.


4. DOCUMENTOS DE REFERÊNCIA

- CPFL. Diretrizes de Segurança da Informação. (GED 14369)
- CPFL Procedimento para Abertura, Priorização e Atendimento de Demandas (GED 17425)
- CPFL Gestão de Logs e Eventos (GED 18758)
- CPFL. Procedimento para o Desenvolvimento Seguro. (GED 18872)
- CPFL. Norma de Desenvolvimento Seguro. (GED 18883)
- CPFL Norma de Uso de Criptografia (GED 18892)
- CPFL. Requisitos de Segurança para Projetos. (GED 19271)
- NIST.SP.800-2180 - Secure Software Development Framework (SSDF) Version 1.1
- DoD Enterprise DevSecOps Reference Design v1.0_Public Release
- Sommerville, I. (2019). Engenharia de Software. São Paulo: Pearson Universidades.
- Deogun, D. (2019). Secure by Design. Shelter Island: Manning.
- Verona, J. (2016). Practical DevOps. Birmingham: Packet.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	9 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

- Wilson, G. (2020). DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement. Londres: Rethink Press.
- Freeman, E. (2021). DevOps para Leigos. Rio de Janeiro: Alta Books.
- Shostack, Adam (2014). Threat Modeling: Designing for Security. Boulevard: Wiley.
- Jet Brains. (n.d.). Um guia para ferramentas de CI/CD. Guia de CI/CD do TeamCity: <https://www.jetbrains.com/pt-br/teamcity/ci-cd-guide/ci-cd-tools/>
- Kim, G., Humble, M., Debois, P., Willis, J., & Forsgren, N. (2021). The DevOps Handbook. Portland: IT Revolution.
- Mohanan, R. (2022, Março 16). What Is DevOps Lifecycle? Definition, Key Components, and Management Best Practices. Retrieved from Spiceworks: <https://www.spiceworks.com/tech/devops/articles/what-is-devops-lifecycle/>
- Ragan, T. (2017, Junho 29). Continuous Delivery vs. Continuous Deployment. Retrieved from DevOps.com: <https://devops.com/continuous-delivery-vs-continuous-deployment/>
- Red Hat. (2022, Maio 11). What is a CI/CD pipeline? Retrieved from Red Hat: <https://www.redhat.com/en/topics/devops/what-cicd-pipeline>

5. RESPONSABILIDADES

Gestores e Executivos de Tecnologia

Os líderes dos times de tecnologia têm como responsabilidade promover a cultura do DevOps através da divulgação dos seus benefícios e da eliminação das barreiras para a sua adoção, além disso, também cabe a estes identificar pessoas chave que possam atuar como evangelizadores da cultura através da corporação e garantir o alinhamento entre os times que atuam no processo.


Time Ágil

Os times ágeis são equipes multifuncionais que tem habilidade e a responsabilidade de definir, construir, testar e manter softwares que entregam valor ao negócio, garantindo a estabilidade

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	10 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software
	Boas Práticas DevSecOps

e a disponibilidade dentro dos padrões alinhados com a companhia. Os times ágeis são compostos por três papéis básicos:



SCRUM MASTER

É um facilitador, visa fazer com que os valores e os princípios do Ágil sejam seguidos no dia a dia.

Auxilia na resolução dos impedimentos e facilita a integração das equipes.

Seu foco é fazer com que o time trabalhe cada vez mais rápido e melhor.



PRODUCT OWNER

É o representante do negócio, tem contato com a liderança e concentra as demandas de sua área.

Define as prioridades com foco em maximizar a entrega de valor.

É responsável elaborar o Product Backlog.



TIME DE DESENVOLVIMENTO

Equipe que desenvolve e suporta o software.

Cada integrante do time possui diferentes habilidades técnicas que em conjunto viabilizam a entrega de valor para o negócio dentro de uma iteração.

É responsável por estimar o trabalho e definir o escopo de cada iteração.


Infraestrutura e Cloud

Equipe responsável sustentação da infraestrutura dos ambientes tanto on-premisses como nas nuvens contratadas pela CPFL. Esse time atua na execução dos projetos de infraestrutura, sejam eles para a expansão do ambiente existente ou para atualização tecnológica do parque, no desenho da arquitetura de infraestrutura e na definição das diretrizes de infraestrutura para os ambientes produtivos e não produtivos, além de atuar com o time de DevOps e Segurança da Informação nos alinhamentos para a construção dos pipelines do ambiente e na entrega de novos projetos.

Time DevOps

Está envolvido no início do ciclo de uma aplicação, tanto no planejamento, acompanhando o desenvolvimento, automação e serviços. Gerencia ferramentas de CI/CD, integração contínua, entrega contínua e implantação contínua. O time de DevOps tem como atribuições, administrar as ferramentas de automação do pipeline, participar em conjunto com o time ágil,

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 111 de 311
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	------------------------------

 <i>Uso Interno</i>	Tipo de Documento: Procedimento
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Infraestrutura e Cloud e de segurança da informação das definições dos projetos, padrões e modelos de branch, construção dos pipelines e integração das diversas ferramentas de monitoração e avaliação de segurança nos ambientes.

Segurança da Informação

Composto por arquitetos e especialistas em segurança, esse grupo tem a responsabilidade de interagir com os times de desenvolvimento e operação para garantir que tanto ao código produzido como a infraestrutura criada atendam as melhores práticas de segurança. Criar e executar testes manuais e automatizados que permitam a detecção de vulnerabilidades no código, na lógica e na infraestrutura da aplicação durante todo o ciclo de vida do sistema. Disseminar entre os times as melhores práticas de segurança. Ampliar o conhecimento técnico dos times sobre as principais ameaças à segurança das aplicações existentes, seus riscos e mitigações e habilitar os times de DevOps, infraestrutura e Cloud, Ágil e de Gestores e Executivos a tomarem decisões educadas sobre os riscos de segurança versus as necessidades de novas funcionalidade no processo de criação de novos sistemas e na manutenção dos sistemas já existentes.


6. REGRAS BÁSICAS

6.1 Introdução

Apresentar todas as etapas do DevSecOps e como elas se integram com o ciclo de vida de desenvolvimento seguro de software com a finalidade de orientar não só os times envolvidos diretamente no processo como também toda a CPFL em como atuar nos projetos internos e externos para a disponibilização novas soluções de software e de melhorias dos sistemas legados para que possam atender as exigências de funcionalidades do mercado com prontidão e segurança.

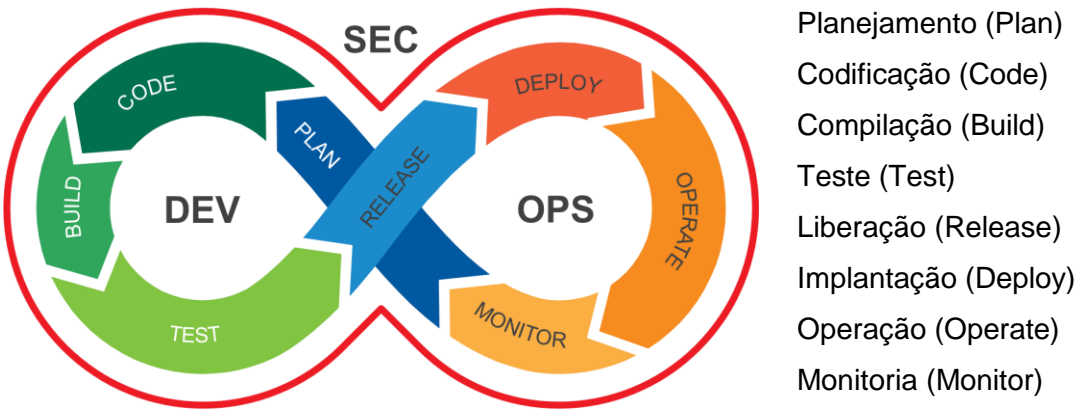
Como definido anteriormente, o ciclo de vida se dará em forma de pipeline permitindo uma visualização clara de cada uma dessas etapas e como elas interagem entre si, além de definir os requisitos mínimos de segurança para cada uma delas.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	12 de 31

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software	

6.2 O Pipeline DevSecOps

O pipeline DevSecOps é composto pelas seguintes fases:



Cada uma delas cumprindo com uma missão específica durante o ciclo de vida da aplicação, contando com a segurança da informação como parte integral do processo, garantindo as todas as funcionalidades requeridas pelo negócio sejam atendidas e que os riscos de segurança atrelados a estas funcionalidades sejam mapeados e mitigados ao longo do processo, evitando assim retrocessos indesejados ou liberações inseguras devido a necessidades imediatas do negócio.

6.2.1. Planejamento (Plan).

O primeiro estágio do pipeline do ciclo de vida do desenvolvimento seguro de software, o planejamento tem como objetivo produzir o roadmap irá orientar o desenvolvimento da aplicação. Nessa fase o time ágil na pessoa do dono do produto trás as necessidades do negócio para os times de desenvolvimento, infraestrutura e Cloud, DevOps e segurança da informação para que juntos possam analisar os requerimentos e criar uma lista de tarefas (Backlog) a serem executadas para a criação do sistema, sendo este backlog utilizado para planejar os ciclos de desenvolvimento (Sprints).

É durante essa fase é imprescindível que no mínimo as seguintes definições sejam elaboradas pelos times dentro de suas responsabilidades.

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 13 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

Tipo de Documento:	Norma
Área de Aplicação:	Tecnologia de Informação
Título do Documento:	Ciclo de Vida de Desenvolvimento Seguro de Software

Quais são os requerimentos que serão priorizados para desenvolvimento?


Como será a experiência do usuário na interação com as funcionalidades a serem implementadas?

Quais são os serviços necessários para sustentar as funcionalidades dos requerimentos que serão criados?

A aplicação será hospedada on-premises ou na Cloud?

Como os serviços necessários para a sustentação do ambiente serão disponibilizados?

Quais serão os produtos e ferramentas utilizadas para disponibilizar esses serviços?

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Nome: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação Boas Práticas DevSecOps
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Quais são os recursos de infraestrutura necessários para a disponibilização desses serviços?

Como esses serviços serão expostos?

Time de DevOps

Como serão criados os pipelines de integração, implantação e entrega do produto?

Quais são as ferramentas necessárias para realizar a automação e a integração dos processos do pipeline?

Como serão implementados os processos de automação e integração do pipeline?


Segurança da Informação

As principais práticas de desenvolvimento seguro e segurança do ambiente estão presentes nas definições do produto?

Quais são os riscos de segurança da informação e de infraestrutura existentes nesse projeto?

As equipes envolvidas no projeto compreendem quais são os riscos mapeados e as suas consequências para o negócio?

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 15 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

O negócio compreende e aceita os riscos mapeados para a implementação do projeto?

Quais são os controles necessários para minimizar esses riscos?


Quais são as possíveis ameaças existentes nas funcionalidades desejadas e como mitigá-las?

As equipes envolvidas no projeto entendem os conceitos de segurança necessários para implementar os controles necessários para mitigar as possíveis ameaças mapeadas?

É responsabilidade de todos os times participantes a criação de definições compartilhadas como por exemplo nomenclaturas, estruturas de repositórios para dados e artefatos, estruturas de dados para o registro de atividades (Logs), definição dos perfis de acessos para os membros do projeto, planos de testes etc. Além da definição de métricas qualitativas e quantitativas que serão utilizadas para medir a qualidade do que foi produzido em cada uma das fases do pipeline e deverão ser utilizadas para avaliar se o resultado de cada uma das fases atende as necessidades mínimas para a progressão para a próxima fase do pipeline. Os seguintes documentos devem ser considerados durante essa fase para orientar os trabalhos e os questionamentos realizados de modo a garantir que os resultados atinjam os requisitos mínimos especificados para os sistemas CPFL:

- CPFL. Diretrizes de Segurança da Informação. (GED 14369)
- CPFL Gestão de Logs e Eventos (GED 18758)
- CPFL. Procedimento para o Desenvolvimento Seguro. (GED 18872)
- CPFL. Norma de Desenvolvimento Seguro. (GED 18883)
- CPFL Norma de Uso de Criptografia (GED 18892)
- CPFL. Requisitos de Segurança para Projetos. (GED 19271)

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 16 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

 <i>Uso Interno</i>	Tipo de Documento: Norma	
	Tecnologia de Informação	
	Área de Aplicação: Tecnologia de Informação	
	Boas Práticas DevSecOps	
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software	

O produto da fase de planejamento é um documento contendo todas as definições necessárias para a realização das atividades definidas que servirá como base e orientará todas as decisões tomadas e ações realizadas durante os outros estágios do pipeline. Este documento deverá ser periodicamente revisado, rediscutido e aprovado por todos os times participantes durante todo o tempo de vida do projeto.

6.2.2. Codificação (Code).

O estágio de codificação é considerado por muitas vezes o mais importante de todo o pipeline e por muitas vezes é o que recebe o maior nível de atenção, já que é nesse estágio que os sistemas se materializam através da criação do código de programação, contudo, uma vez que temos definições claras de quais são as necessidades do projeto em termos de planejamento, arquitetura e segurança o processo de codificação se torna relativamente mais simples e direto visto o direcionamento do que deve ser feito e como devemos fazê-lo já foram feitos e os times podem se concentrar em como criar o melhor código para atender as funcionalidades desejadas.


Durante esse estágio os seguintes tópicos devem ser endereçados pelos times de acordo com a sua responsabilidade.

Time Ágil

Todas as tarefas priorizadas para implementação pelo time de desenvolvimento devem ser registradas no sistema de abertura e controle de demandas e devem ser acompanhadas periodicamente pelo Product Owner e pelo Scrum Master, de acordo com o procedimento de abertura, priorização e atendimento de demandas (GED 17425).

Situações que possam trazer dificuldades na implementação ou que venham a impedir a implementação de parte ou todas as funcionalidades planejadas devem ser reportadas pelo time de desenvolvimento para o Scrum Master ou

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	177 de 311

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Product Owner, assim que detectadas, de modo a permitir a análise e o replanejamento das ações.

Todo código produzido pelos desenvolvedores deve seguir as melhores práticas de programação e as convenções de estilo definidas pelo projeto, sendo essa prática avaliada através do uso de ferramentas automatizadas de revisão de código.

É de responsabilidade do time de desenvolvimento garantir que o código criado esteja livre de erros e falhas de segurança que já tenham sido mapeadas na fase de planejamento ou que eventualmente venham a aparecer durante o projeto sendo essa prática avaliada através do uso de ferramentas automatizadas de revisão de código e pelo desenvolvimento de scripts de testes automatizados para as validações das funcionalidades desenvolvidas.


Todo código produzido deve ser obrigatoriamente armazenado em repositório centralizado com controle de versão e assinatura digital ativos, além de acesso restrito aos participantes do projeto. Sendo que todas as operações realizadas nesses repositórios devem ser registradas através de um sistema de registro de acessos.

O time ágil é responsável também por catalogar todos os frameworks e bibliotecas de terceiros utilizadas como auxílio na produção das funcionalidades implementadas, além de manter os arquivos pertencentes e estes frameworks armazenados em um repositório centralizado com controle de versão e assinatura digital ativos além de acesso restrito aos participantes do projeto designados a gestão desse catálogo e aos processos de automatização do pipeline. Sendo que todas as operações realizadas nesses repositórios devem ser registradas através de um sistema de registro de acessos.

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 18 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

O time de desenvolvimento é responsável por priorizar frameworks de bibliotecas de terceiros que sejam reconhecidas e amplamente adotadas pelo mercado além de garantir que a versão selecionada para o projeto seja a mais atualizada e não possua vulnerabilidades de segurança, sendo sua responsabilidade também garantir que bibliotecas e frameworks inseguros sejam de priorizados e substituídos conforme o avanço do projeto.

O time de desenvolvimento é responsável por garantir que todo os serviços publicados nos ambientes da aplicação como APIs, repositórios de códigos e artefatos, entre outros tenham o acesso controlado e que somente as pessoas envolvidas nos projetos ou os serviços necessários possam utilizá-los. Além disso é de responsabilidade do time de desenvolvimento manter todos os segredos utilizados para acessar esses serviços seguros através de uma ferramenta de cofre de senhas.

Infraestrutura e Cloud


Criar e configurar os ambientes necessários para a sustentação da aplicação desenvolvida sendo que todas as tarefas necessárias para essas atividades devem ser registradas no sistema de abertura de chamados e demandas e acompanhadas periodicamente pelo Product Owner e pelo Scrum Master, de acordo com o procedimento de abertura, priorização e atendimento de demandas (GED 17425).

Situações que possam trazer dificuldades na implementação ou que venham a impedir a criação de parte ou toda a infraestrutura conforme o planejado devem ser reportadas pelo time de infraestrutura e Cloud para o Scrum Master ou

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 19 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
Título do Documento:	
Ciclo de Vida de Desenvolvimento Seguro de Software	

Product Owner, assim que detectadas, de modo a permitir a análise e o replanejamento das ações.

Dar preferência ao uso de tecnologias que permitam a automatização processo de criação dos componentes de infraestrutura através do uso de código (IaC - Infrastructure as Code) e que permitam o escalonamento tanto vertical como horizontal da capacidade de processamento na demanda.

Garantir que todos os artefatos produzidos atendam aos requisitos funcionais e de segurança conforme o estipulado no projeto, além de mantê-los atualizados com as últimas correções do fabricante, livres de erros e de vulnerabilidades de segurança. Sendo essa atividade monitorada através do uso de ferramentas automatizadas para a avaliação de compliance e detecção de vulnerabilidades.


Todos os artefatos produzidos para a sustentação do ambiente, como imagens de servidores, scripts de criação de infraestrutura e arquivos de configuração devem ser armazenados em um repositório centralizado com controle de versão e assinatura digital ativos além de acesso restrito aos participantes do projeto designados a gestão desses artefatos e aos processos de automatização do pipeline. Sendo que todas a operações realizadas nesses repositórios devem ser registradas através de um sistema de registro de acessos.

Garantir que os acesso privilegiados aos artefatos de infraestrutura como servidores, componentes de rede e serviços publicados sejam controlados e que somente as pessoas responsáveis por essa atividade tenham esse acesso. Sendo que todas a operações realizadas nesses artefatos devem ser registradas através de um sistema de registro de acessos.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	220 de 311

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

O time de infraestrutura e Cloud é responsável por garantir que todos os serviços publicados nos ambientes da aplicação como repositórios de artefatos, servidores, componentes de rede entre outros tenham o acesso controlado e que somente as pessoas envolvidas nos projetos ou os serviços necessários possam utilizá-los. Além disso é de responsabilidade do time de infraestrutura e Cloud manter todos os segredos utilizados para acessar esses serviços seguros através de uma ferramenta de cofre de senhas.

DevOps

O time de DevOps é responsável criar os pipelines dos ambientes, além de elaborar os scripts de automatização e testes desses pipelines, garantindo que estes estejam livres de erros e dentro das melhores práticas de segurança de acordo com o definido no projeto. Sendo que todas as atividades necessárias para esta atividade devem ser registradas no sistema de abertura de chamados e demandas e acompanhadas periodicamente pelo Product Owner e pelo Scrum Master, de acordo com o procedimento de abertura, priorização e atendimento de demandas (GED 17425).


Situações que possam trazer dificuldades na implementação ou que venham a impedir a criação de parte ou toda a estrutura dos pipelines conforme o planejado, devem ser reportadas pelo time de DevOps para o Scrum Master ou Product Owner, assim que detectadas, de modo a permitir a análise e o replanejamento das ações.

Todos os scripts e artefatos produzidos devem ser armazenado obrigatoriamente em repositório centralizado com controle de versão e assinatura digital ativos, além de acesso restrito aos participantes do projeto.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	21 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Procedimento
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Sendo que todas as operações realizadas nesses repositórios devem ser registradas através de um sistema de registro de acessos.

O time de DevOps deve garantir que todas as atividades realizadas tanto pelo pipeline como pelos scripts de integração tenham as suas ações registradas através de um sistema de registro de acessos.

O time de DevOps é responsável por garantir que todos os serviços publicados através dos pipelines tenham o acesso controlado e que somente as pessoas envolvidas nos projetos ou os serviços necessários possam utilizá-los. Além disso é de responsabilidade do time de DevOps manter todos os segredos utilizados para acessar esses serviços seguros através de uma ferramenta de cofre de senhas.

Segurança da Informação


O time de segurança da informação tem como atribuição a criação de testes estáticos, aplicados nos códigos fonte do projeto (Aplicação, Infraestrutura, Automação) ou em suas dependências, bem como testes dinâmicos e interativos, aplicados no ambiente em execução, com a finalidade de identificar fraquezas ou vulnerabilidades que possam ter sido criadas durante as fases do projeto. Todas as atividades necessárias para realização dessa tarefa devem ser registradas no sistema de abertura de chamados e demandas e acompanhadas periodicamente pelo Product Owner e pelo Scrum Master, de acordo com o procedimento de abertura, priorização e atendimento de demandas (GED 17425).

Analisar situações que possam trazer riscos à segurança da aplicação ou do ambiente ou que venham a impedir a criação de parte ou toda infraestrutura ou

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 222 de 311
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	------------------------------

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

aplicação e reportá-las ao Scrum Master ou Product Owner, assim que detectadas, de modo a permitir a análise e o replanejamento das ações.

Acompanhar os outros times em suas atividades de modo a auxiliá-los na aplicação das melhores práticas de segurança tanto na criação do código como na configuração do ambiente de infraestrutura e de pipeline.

A fase de codificação é provavelmente a parte mais longa e substancial do processo, pois é onde o projeto real é de fato criado

6.2.3 Compilação (Build).

Nesse estágio o código fonte criado pelos times é testado para garantir que ele se encontra livre de erros e transformado em código objeto através do uso de compiladores, ferramentas de vinculação (linking editor), etc.

Esta fase é altamente automatizada com pouca ou nenhuma intervenção humana, contudo ela é responsável por garantir que os binários gerados estejam em conformidade com as especificações do projeto.

É na fase do Build que os testes estáticos de segurança de aplicação são executados, validando se o código criado pelo time de desenvolvimento atende as práticas de desenvolvimento seguro e não contém vulnerabilidades conhecidas, além disso é realizada a análise de vulnerabilidades nas dependências impedindo que vulnerabilidades em frameworks e pacotes de terceiros sejam herdadas pela aplicação ou que frameworks e pacotes que não foram autorizados para uso sejam introduzidos no ambiente.


É nesse momento também que realizado o empacotamento, versionamento e o armazenamento dos artefatos criados.

Nessa etapa é o time de DevOps que o responsável pelos tópicos endereçados de acordo com as suas responsabilidades.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	23 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software	

DevOps

O time de DevOps tem a responsabilidade de garantir que os processos automatizados para a geração do código binário da aplicação, testes de segurança, empacotamento e armazenagem dos artefatos aconteçam sem impedimentos e atuar em caso de problemas em qualquer um desses estágios.

Como produto dessa fase temos um relatório com os resultados dos testes de análise estática da aplicação e de verificação de dependências que serão utilizados como base para decidir se os artefatos gerados podem ou não seguir para a próxima fase.


6.2.4. Teste (Test).

O teste de software é a fase do pipeline que visa garantir que os binários gerados para a aplicação se comportam de acordo com o especificado e atendem as funcionalidades solicitadas pelo negócio. Existem inúmeros testes que podem ser aplicados nessa fase, como por exemplo, testes unitários, testes funcionais, testes de integração, testes de regressão, testes de segurança, testes de performance etc.

Nessa fase a responsabilidade de todos os times é executar o plano de testes definido no planejamento do projeto para garantir que a aplicação atenda as especificações e funcionalidades desejadas.

A seguir apresentamos uma série de testes sugeridos para a implementação no pipeline da CPFL, contudo é necessário salientar esta lista não apresenta todos os testes possíveis e que nem todos esses testes são aplicáveis a todos os cenários. Contudo ela assinala os testes que são considerados mínimos obrigatórios para os processos.

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 224 de 311
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	------------------------------


 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação
	Área de Aplicação: Boas Práticas DevSecOps
Título do Documento:	
Ciclo de Vida de Desenvolvimento Seguro de Software	

Teste	Descrição	Mandatário
Teste Unitário	Verifica se uma unidade de software individual executa sua função conforme o planejado.	Sim
Teste dinâmico de segurança de aplicação (DAST)	Verifica a aplicação procurando por vulnerabilidades e fraquezas existentes e sugere possíveis mitigações para os problemas reportados.	Sim
Teste de integração	Verifica se as integrações entre as unidades de software implementadas executam conforme o planejado.	Sim
Teste de sistema	Verifica se os sistemas existentes performam como desejado.	?
Teste manual de segurança	Procura por problemas de segurança específicos em um ambiente ou aplicação e indica possíveis correções para as vulnerabilidades encontradas.	Não
Teste de performance	Garante que a aplicação irá performar conforme o esperado sob uma determinada carga de trabalho.	?
Teste de regressão	Testa se as alterações introduzidas no código não afetam negativamente as funcionalidades já existentes.	?
Testes de aceitação	Os testes de aceitação são uma série de testes que são executados com a finalidade de avaliar se um sistema está pronto para a operação.	?
Teste de políticas de containers	Verifica se os containers disponibilizados atendem as políticas estabelecidas no ambiente.	?
Scan de compliance	Realiza a auditoria do ambiente para as políticas aplicadas e reporta os pontos que estão fora das especificações.	Não
Testes funcionais de banco de dados	Executa testes funcionais e unitários das bases de dados para validar se a definição dos dados, gatilhos e restrições foram implementados adequadamente.	?

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	25 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 <i>Uso Interno</i>	Tipo de Documento: Norma	
	Tecnologia de Informação	
	Área de Aplicação: Tecnologia de Informação Boas Práticas DevSecOps	
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software	

Teste	Descrição	Mandatário
Testes não funcionais de banco de dados	Executa testes de performance, carga, stress e failover para garantir a confiabilidade e disponibilidade dos bancos de dados.	?

Como resultado final dessa fase temos uma série de relatórios que apontam o quão pronta a aplicação se encontra, sendo essas informações utilizadas para a tomada de decisão para que a aplicação prossiga para a próxima fase e para a elaboração de ações de correção dos problemas encontrados.

6.2.5 Liberação (Release).

A fase de liberação é o ponto em que os times do projeto consideram uma compilação pronta para implantação no ambiente de produção. Nesse estágio, o código criado pelo time de desenvolvimento e o ambiente disponibilizado pelo time de infraestrutura e cloud já foi testado e os problemas encontrados, corrigidos a um ponto que a ocorrência de algum tipo incidente por causado por alguma falha no projeto é pouco provável.


Nessa etapa a atividade para todos os times envolvidos no projeto é a análise dos relatórios de auditoria criados pela fase de testes e a aprovação da liberação do ambiente para implantação.

Como resultado dessa fase temos a liberação do pacote de implantação do projeto.

6.2.6 Implantação (Deploy).

A fase de implantação é tem como objetivo disponibilizar a aplicação para os usuários. Esta é uma fase que nos ambientes mais convencionais fica ao encargo do time de operações de TI, contudo de ponto de vista do DevSecOps, essa atividade deve ser simplificada e automatizada de modo que todo desenvolvedor do time de projetos seja capaz de implantar o código.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	26 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Nessa etapa o pacote contendo os artefatos de produção aprovados no processo de release para a produção é replicado para o repositório de produção que por sua vez disponibiliza a aplicação para os usuários. O time responsável por executar essa atividade depende de como o projeto foi desenhado.

6.2.7 Operação.

A fase de operações tem como objetivo manter a disponível para os usuários dentro dos parâmetros definidos no projeto através do uso de ferramentas para o escalonamento de sistemas, balanceamento de carga e backup sendo que essas atividades devem automatizadas através da configuração de políticas.


As tarefas da etapa de operação são realizadas pelo time de infraestrutura e Cloud uma vez que na CPFL eles são responsáveis por manter a infraestrutura tanto on-premises como na nuvem.

Como suas atribuições nessa fase, o time de infraestrutura dar preferência por ferramentas que automatizem os processos de escalonamento de sistemas, balanceamento de carga e backup do ambiente.

Deve manter e atualizar as políticas de balanceamento de carga e de escalonamento de sistemas do ambiente de modo a conferir a aplicação uma performance estável e disponibilidade dentro do estabelecido pelo projeto.

O time também deve criar e manter uma ferramenta de dashboard contendo informações sobre o status operacional do ambiente, alertas e recomendações.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	27 de 31

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação Boas Práticas DevSecOps
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

6.2.8 Monitoria (Monitor).

A fase de monitoração é a última fase do ciclo de DevSecOps, nela a aplicação já está operacional e disponível para os usuários e agora se faz necessário acompanhar seu comportamento durante todo o resto do ciclo de vida. Esta etapa é muito importante para a saúde do sistema pois os dados produzidos nessa fase serão utilizados como retorno para os times do projeto, que por sua vez, os utilizarão como insumos para o planejamento do próximo ciclo de melhorias.

O processo de coleta é análise dos registros gerados pelas atividades da aplicação é uma tarefa compartilhada por todos os times envolvidos no projeto que dentro de suas atribuições utiliza as informações extraídas desses logs para gera relatórios sobre a saúde do ambiente, investigar eventos de segurança ou gerar alertas sobre possíveis problemas que venham a ocorrer com na operação do ambiente.


A quantidade de registros que podem ser gerados por uma aplicação é imensa e varia de acordo com o projeto, contudo as regras a seguir são fundamentais para que esses registros sejam uteis para os times:

- Sempre que possível deve ser adotado um repositório centralizado para armazenar os dados coletados do ambiente.
- Todos os logs gerados pelo ambiente devem seguir um formato padrão para a representação dos dados neles armazenados a ser adotado por todos os projetos da CPFL.
- Todos os logs gerados pelo ambiente devem seguir um conteúdo padrão por atividade registrada, por exemplo, todos os logs de performance devem registrar a hora do evento, o nome do dispositivo, o percentual de CPU utilizado etc. E esse padrão deve ser seguido por todos os projetos da CPFL.

N.Documento: 19370	Categoria: Instrução	Versão: 1.1	Aprovado por: Alexandre Alves Barrias	Data Publicação: 26/04/2023	Página: 28 de 31
------------------------------	--------------------------------	-----------------------	---	---------------------------------------	----------------------------

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 Uso Interno	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

- Para os casos em que não seja possível seguir os padrões de log determinados pela CPFL, um processo automatizado para a conversão dos logs incompatíveis para o padrão definido deverá ser criado e os registros deverão ser convertidos antes de serem armazenados no repositório de dados.
- É imprescindível que todos os logs registrem a data e hora do evento e que os relógios de todos os servidores do ambiente estejam sincronizados de acordo com uma fonte única de tempo para todo o ambiente, normalmente um servidor de NTP corporativo determinado pelo time de Infraestrutura e Cloud.


Além disso existem uma infinidade de registros que podem ser capturados do ambiente, a seguir serão apresentados alguns desse registro e apontados os que são considerados como o mínimo mandatório para a CPFL.

Tipo de Registro	Dados capturados	Mandatório
Monitoria de performance dos sistemas.	Informações sobre o uso dos recursos de hardware dos servidores, nível de carga dos serviços do ambiente, taxa de ocupação da rede etc.	Sim
Monitoria de segurança dos sistemas.	Eventos de segurança de todos os componentes do ambiente, reporte de vulnerabilidades encontradas, registro de conformidade do ambiente com as políticas de segurança.	Sim
Inventário de ativos	Registro de todos os ativos do ambiente.	Sim


N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	29 de 31

IMPRESSÃO NÃO CONTROLADA

IMPRESSÃO NÃO CONTROLADA

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Área de Aplicação: Tecnologia de Informação
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

Monitoração da configuração dos sistemas.	Registro de conformidade dos sistemas com as políticas do ambiente.	Sim
Monitoria de performance e segurança de banco de dados.	Registro da carga de trabalho dos servidores de banco de dados, registo dos níveis de ocupação dos recursos de hardware dos servidores que hospedam o SGDB, eventos de auditoria de acesso aos dados.	Sim

 <i>Uso Interno</i>	Tipo de Documento: Norma
	Nome: Tecnologia de Informação
	Área de Aplicação: Tecnologia de Informação Boas Práticas DevSecOps
	Título do Documento: Ciclo de Vida de Desenvolvimento Seguro de Software

7. CONTROLE DE REGISTROS

Identificação	Armazenamento e Preservação	Proteção (acesso)	Recuperação e uso	Retenção	Disposição
Norma de DevSecOps	Eletrônico (GED)	Restrição de Acesso	Por tema ou título	Até a próxima atualização do documento	Substituição

8. ANEXOS

NA

9. REGISTRO DE ALTERAÇÕES

9.1. Colaboradores

Empresa	Área	Nome
CPFL Energia	EIDN – Gerência de Operações em Nuvem	Luiz Henrique Paulista Toledo
CPFL Energia	EIDN – Gerência de Operações em Nuvem	Leandro Aurelio Martinelli
CPFL Energia	EIDN – Gerência de Operações em Nuvem	Samuel Flores
CPFL Piratininga	EIS – Gerencia de Segurança da Informação	Alexandre Fukaya

9.2. Alterações

Versão Anterior	Data da Versão Anterior	Alterações em relação à Versão Anterior
V0	-	Criação do documento.
1.0	25/11/2022	Revisão geral do documento.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19370	Instrução	1.1	Alexandre Alves Barrias	26/04/2023	31 de 31