
AMBIENTE ÁGIL - NPRD



TODAS AS INFORMAÇÕES AQUI CONTIDAS SÃO CONFIDENCIAIS

Somente as equipes técnicas da CAIXA estão autorizadas, pela natureza das suas funções, a receber tais informações. Nenhuma informação deve ser divulgada sem autorização. Outras pessoas ou empresas deverão ser autorizadas pela GEOTI de acordo com a política vigente sobre divulgação de informações corporativas.

Versão 1.0

REVISÕES:

Versão	Data	Propósito	Autor
1.0	17/08/2017	Criação	Flávio Túlio Côrtes e Elaine Fernandes
1.1	04/09/2017	Atualização	Leandro Oliveira Alvares
1.2	11/09/2017	Atualização	Leandro Oliveira Alvares
1.3	12/09/2017	Atualização	Elisa Sotero de Oliveira e Leandro Oliveira Alvares
1.4	13/10/2017	Atualização	Leandro Oliveira Alvares
1.5	20/10/2013	Inserção item Arquitetura do projeto	Leandro Oliveira Alvares
1.6	26/10/2017	Inserção do item PROMOÇÃO PARA PRODUÇÃO	Leandro Oliveira Alvares
1.7	27/10/2017	Atualização	Leandro Oliveira Alvares
1.8	28/12/2017	Correção de itens na página 33	Leandro Oliveira Alvares

OBJETIVO:

Descrever as ações necessárias para utilização e criação do ambiente ágil, bem como apresentar seus padrões de uso, ferramentas utilizadas e o fluxo para a criação de novos projetos nos ambientes Não-Produção (DES, TQS e HMP).

Para isso, deverão ser utilizadas diversas ferramentas e tecnologias diferentes, onde integradas são capazes de entregar um ambiente de Plataforma como Serviço, disponibilizando ao usuário final um ambiente totalmente pronto a partir de poucos cliques e poucas horas.

GLOSSÁRIO

Container – processo criado a partir da imagem contendo a ambiente virtual da aplicação.

Imagem – Arquivos e estruturas de diretório que compõem uma ambiente virtual que contém a aplicação.

Master (Origin) – Servidores com função de orquestração dos *containers*, configuração das aplicações e automações do ambiente

Nodes (*Origin*) – Servidores com função de executar os *containers*

OpenSource – Aplicações cujo código-fonte está disponível para consulta e/ou alteração, geralmente sem custo de licenciamento.

Origin – OpenShift *Origin*, versão OpenSource da ferramenta OpenShift da empresa RedHat, responsável pela administração dos containers.

Projeto – Nas ferramentas que compõem a solução é a área de cadastramento e armazenamento de configurações e arquivos associados ao sistema

Wildcard DNS – Entrada especial no DNS na forma “*.domínio”, onde o servidor traduzirá qualquer nome neste formato para o mesmo IP. Ex: a.domínio e z.domínio resolverão para o mesmo IP

Zones (*Origin*) – Segregação lógica dos Nodes por algum critério. Ao atribuir uma aplicação a uma *zone*, os *containers* daquela aplicação só executarão sobre os servidores daquela *zone*.

REQUISITOS DE CONHECIMENTO

TE191 – Descreve a concessão de acessos, inclusive para criação de contas de usuário de serviço.

GIT – Armazena e versiona os códigos-fonte e configurações, imprescindível ter conhecimento de sua operação

Docker – Tecnologia de containers utilizada pelo Origin, é importante conhecer os conceitos, vantagens e limitações da tecnologia. Também é útil conhecer sobre Kubernetes, tecnologia que agrupa “serviços” compostos de vários containers.

CI/CD – Continuous Integration / Continuous Deployment. É a forma de automação de builds e deploys de forma a executar as tarefas repetitivas de compilação, linkedição, testes, empacotamento e deploy, minimizando falhas no processo. Em especial é importante conhecer o funcionamento da ferramenta Jenkins

PREMISSAS PARA CRIAÇÃO DO AMBIENTE AGIL

- As ferramentas são *OpenSource* e sem custo de licenciamento;
 - Ferramentas utilizadas: OpenShift Origin, Jenkins, GITLab e Nexus.
- As ferramentas terão um único ambiente com gerência consolidada para todas as centralizadoras de desenvolvimento, com segregação de acesso entre os projetos;
- O ambiente onde essas ferramentas estão instaladas são de gestão das centralizadoras de operações;
- O ambiente é subdividido por DES/TQS e HMP, segregado por *zones*, contemplando as centralizadoras;
- Acesso dos Masters e Nodes do *Origin* ao PROXYDES, com autenticação por usuário de serviço, permitindo busca de Templates e Imagens dos repositórios na Internet;
- Acesso dos *containers* a internet via PROXYDES, com autenticação de usuário de serviço por aplicação, independente das regras do *Origin*;
- Domínio para aplicações será “nprd.caixa”, na forma “SIXYZ-AMB.NPRD.CAIXA”, com registro de *wildcard* DNS “*.nprd.caixa” apontando para o balanceador, que distribuirá para os “routers” na *zone* Infra. Com isso nenhum outro nome, exceto os usados pelos *containers*, poderá ter domínio NPRD.CAIXA;
- Ferramentas que estiverem no ambiente de São Paulo, ficarão no domínio “DES.CAIXA”, por exemplo: ARQUIVOS.DES.CAIXA. Quando ficarem no CTC (Serviços Comuns) ou quando se destinarem ao uso por múltiplos ambiente, deverão ficar no domínio “CAIXA”, por exemplo: CLOUDCONFIG.CAIXA;
- Cada Jenkins deverá ter uma conta de serviço para integração das ferramentas.

- Todo código fonte deverá ser armazenado no repositório GIT FONTES.

FERRAMENTAS UTILIZADAS

GITLAB

GITLAB FONTES - Repositório único de código fonte dos sistemas da CAIXA e pode ser acessado pelo endereço: <http://fontes.des.caixa>.

GITLAB CONFIG – Repositório dos arquivos de configuração de cada sistema. Endereço de acesso: <http://cloudconfig.caixa>

Avaliação de mercado

Ao avaliar as ferramentas disponíveis no mercado, observou-se que o GIT além de cumprir a premissa de ser gratuito, mostrou-se a ferramenta com mais integração com outras ferramentas, que possuía controle de acesso e relatos de estabilidade, sendo utilizada para manter a maioria dos códigos-fonte no mundo. Também foi escolhido por ser a ferramenta mais utilizada pelos desenvolvedores, conforme gráfico abaixo disponibilizado pelo Google Trends.

Interest over time. Web Search. Worldwide, 2004 - present.

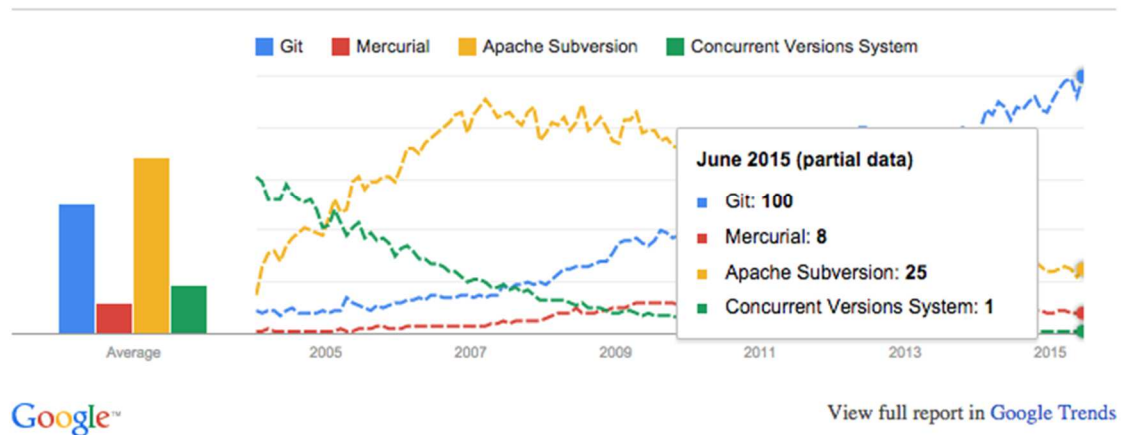


Figura 1 - Google Trends para o GIT

JENKINS

O Jenkins é a ferramenta que suportará os processos de Integração Contínua e Entrega Contínua dos sistemas desenvolvidos na CAIXA, perfazendo automação de *build*, testes unitários e globais, validações de qualidade e demais passos para facilitar e agilizar o processo de desenvolvimento.

Avaliação de mercado

Ao avaliar as ferramentas disponíveis no mercado, observou-se que a ferramenta Jenkins possui ampla utilização para realizar Entrega Contínua e além disso, possui vários plug-ins com outras plataformas, o que o torna flexível o bastante para ser utilizado nos processos da CAIXA.

Interest over time. Web Search. Worldwide, 2004 - present.

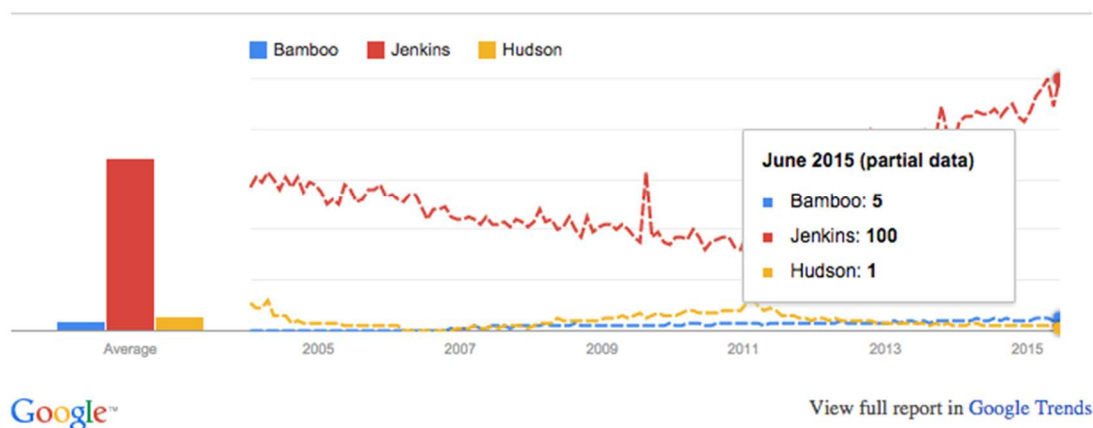


Figura 2 - Google Trens para o Jenkins

Cada centralizadora de desenvolvimento deverá possuir seu Jenkins para que sejam respeitadas as particularidades de cada.

Os servidores deverão ser disponibilizados em ambiente de containers, inicialmente à cada uma das centralizadoras pela equipe responsável pela disponibilização do ambiente, utilizando o conceito de persistência para que não ocorram perdas de dados em uma possível indisponibilidade.

Deverão ser construídos um Jenkins para CEDESBR, um para CEDESSP, um para CEDES RJ, uma para PEDeS e um para Shadow IT.

O Jenkins fará o papel de orquestrador do processo, buscando fontes a partir do GITLAB FONTES, entregando binários para o Nexus (para sistemas compiláveis) ou arquivos compactados (demais sistemas) e buscando arquivos de configurações no GITLAB CONFIG para montar o pacote que será publicado no ambiente alvo.

NEXUS

A ferramenta NEXUS será a ferramenta utilizada como repositório de bibliotecas e binários ou arquivos compactados.

Avaliação de mercado

Ao avaliar as ferramentas disponíveis no mercado, observou-se que o Nexus é a ferramenta mais utilizada como repositório de bibliotecas e binários.

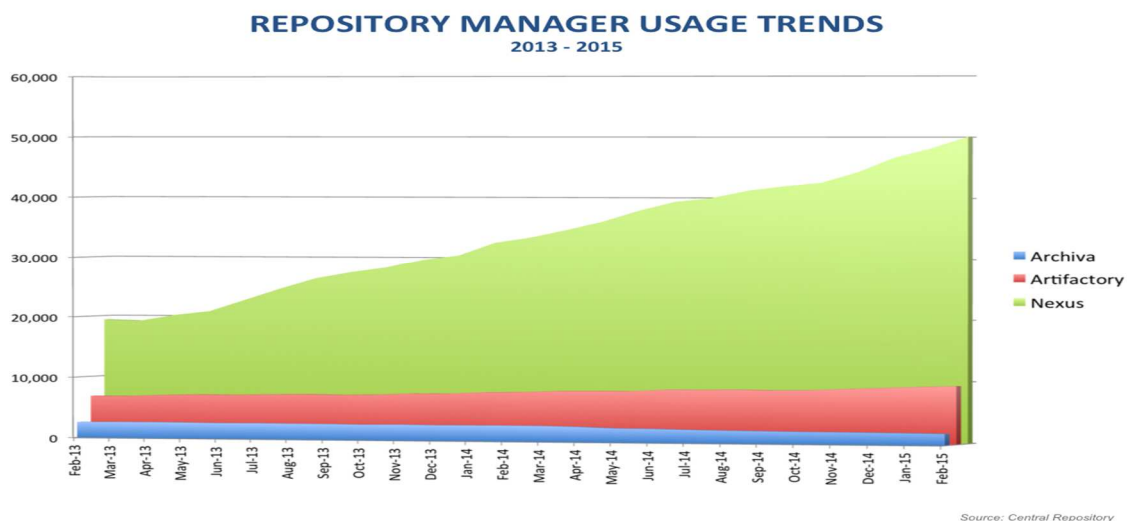


Figura 3 - Uso do Nexus no mercado

Toda *build* será armazenada no NEXUS para manter a consistência da versão entre os ambientes Não Produção e Produção.

O servidor de repositório NEXUS será único para todos os ambientes (DES, TQS, HMP e PRD) e para todas as unidades de desenvolvimento. Com a sua adoção espera-se minimizar o problema de usar pacotes errados no *deploy*, falhas na transmissão dos arquivos ou de versões diferentes em execução no ambiente, uma vez que os binários estarão centralizados.

OPENSIFT ORIGIN

É a ferramenta de gerência e controle dos containers, responsável por realizar o gerenciamento de capacidade, ciclo de vida, comunicação e controle.

Nela estará disponível um catálogo de plataformas tecnológicas (arquitetura de referência), que poderá ser solicitada por qualquer profissional de TI pelo portal <https://cloudnprd.des.caixa:8443>.

As plataformas tecnológicas disponíveis presentes nesse catálogo deverão ser testadas previamente pela equipe gestora da ferramenta (CEPTIBR21).

A ferramenta deve ser integrada com os binários do Nexus e as configurações do GITLAB CONFIG para que seja possível ter um ambiente integrado e autônomo.

A ferramenta controla o balanceamento entre os *containers* a partir do nome, as subredes de cada aplicação, configura os limites de recursos e o *auto-scaling*, tudo segregado por projeto.

Análise de mercado

Conforme pode ser observado na figura abaixo, o Origin (baseado no Openshift) é uma das ferramentas líderes do mercado e demonstrou bastante estabilidade até o presente momento, motivo pelo qual, foi a ferramenta escolhida para prover ambientes containerizados.



Figura 4 - Quadrante Mágico Openshift

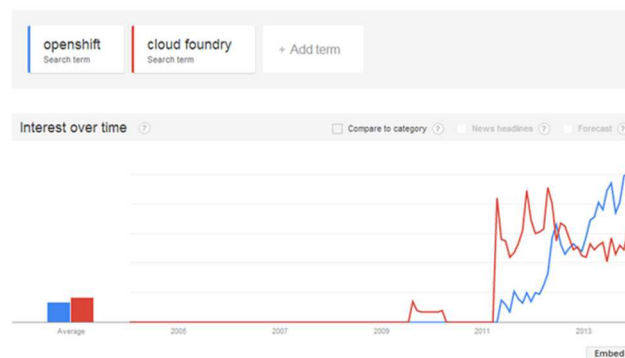


Figura 5 - Google Trends Openshift

ARQUITETURA DO PROJETO

1. O projeto é composto por 02 ambientes: 01 ambiente não produtivo e 01 ambiente produtivo, As ferramentas GITLab Fontes, GITLab CONF e NEXUS deverão ter uma única instalação. Já as ferramentas Jenkins e Origin (com todos os seus componentes) deverão possuir uma instalação no ambiente de produção.

2. A segmentação em 02 ambientes deve-se ao fato de serem instaladas em Datacenters distintos com SLAs diferentes por ambiente, além de equipamentos e administração específica.
3. Em cada uma das segmentações devem ser construídos *zones* para segmentação do ambiente.
4. O ambiente será construído de forma automática, a partir do preenchimento de um formulário disponível em um sistema.
5. Para que seja possível realizar a compilação e construção automática dos projetos, algumas premissas devem ser respeitadas:
 - O código fonte deverá estar exclusivamente no GITLab de Fontes;
 - Todas as compilações deverão ocorrer exclusivamente por um Orquestrador;
 - Todo o fonte deverá passar por testes de segurança automatizados, que devem ser definidos pela equipe de arquitetura do time de desenvolvimento;
 - Todos os fontes deverão ser compactados ou compilados automaticamente;
 - O resultado do item anterior deve ser disponibilizado no NEXUS;
 - Para que seja realizada a construção automática em Containers, deverá ser realizada a integração do orquestrador com o GITLab de CONF;
 - O GITLab de CONF deverá ser configurado com Webhook para integração com o Origin sempre que o arquivo assemble for modificado. Isso fará uma chamada de construção do novo ambiente;
 - O arquivo assemble deverá conter o caminho do resultado da compilação/compactação e o caminho dos arquivos de configuração. Isso fará um mapeamento automático e execução do arquivo assemble, fazendo com o que seja realizado o download dos requisitos do ambiente;
 - Todas as orquestrações deverão ser realizadas exclusivamente nos orquestradores, evitando-se scripts externos e processos manuais;
 - A promoção entre ambientes deverá ser realizada exclusivamente pelo orquestrador;
 - O código fonte será visível exclusivamente no ambiente de desenvolvimento. Em todos os outros ambientes deverá ser utilizado o binário/compilado e as configurações presentes no GITLab de CONF.
6. Considerando as premissas acima, o desenvolvedor deverá realizar o *upload* do código fonte no GITLab de FONTES.
7. Deverá ser configurada uma chamada no orquestrador, que realizará a validação automática do código fonte e compilação/compactação no repositório no servidor do NEXUS.

8. O orquestrador deverá comandar a construção do ambiente, seja ele em container, mobile ou servidores em geral.
9. Para a promoção entre ambientes, deverá ser criada uma área de projeto com cada um dos ambientes (DES, TQS, HMP, pré PRD e PRD).
10. Cada uma dessas áreas deverá ser capaz de realizar o deploy e construção automática da imagem(binário/compilado) disponível no NEXUS.
11. As equipes de desenvolvimento das Centralizadoras farão o Deploy até o ambiente de TQS, enquanto a equipe de operações fará o Deploy nos ambientes seguintes.
12. Sempre que um Deploy ocorrer com equipes diferentes, deverá ser criado um *step* no orquestrador com o nome pré-<ambiente>. O orquestrador deverá ser configurado para realizar a abertura de mudanças e serviços de forma automática informando a mudança.
13. O passo seguinte somente será liberado no orquestrador mediante aprovação da equipe responsável pelo ambiente, por exemplo, somente será liberado o *step* de promoção para homologação após a aprovação do *step* pré-homologação.
14. A equipe responsável pelo ambiente deverá acessar o step do orquestrador pré-<ambiente> e concluir o processo, para que seja liberado o passo de implementação no ambiente seguinte.

TOPOLOGIA DO AMBIENTE NPRD

1. A topologia de infraestrutura do ambiente de container deverá obedecer a topologia demonstrada abaixo:

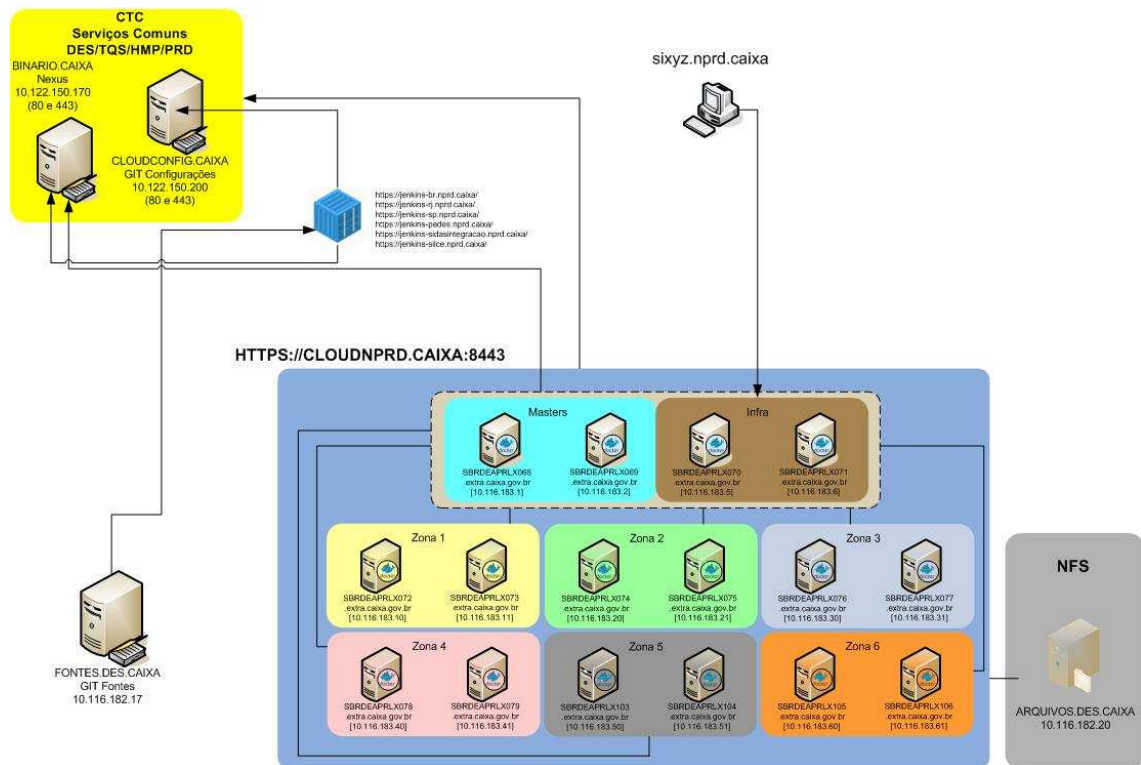


Figura 6 – Topologia do ambiente NPRD

2. Conforme pode ser observado na topologia acima, foi realizada a criação de zonas, importante no momento para prover mais segurança e segmentação do ambiente.
3. Essa topologia será o padrão também para o ambiente de produção até que seja implantada uma nova estrutura de segurança, por exemplo, API Gateway / Proxy de containers, para que as zonas sejam unificadas, seguindo uma das premissas do conceito de NUVEM.
4. Antes da disponibilização do código fonte, o usuário deverá realizar o preenchimento do DAS do sistema pelo SIDAS, acessado pelo endereço <http://sidas.caixa>.

O SIDAS deverá integrar com a camada de Middleware, no qual deverá criar os requisitos do ambiente, conforme detalhes descritos na seção PROCESSO DE CRIAÇÃO DO AMBIENTE ÁGIL.

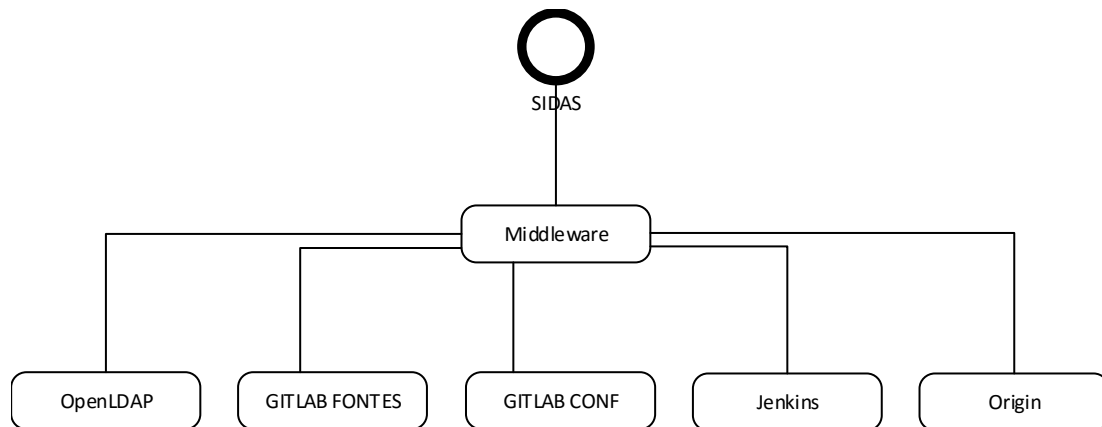


Figura 7 -Processo de criação do ambiente

5. Uma vez que os requisitos foram criados, deverão ser seguidos os próximos passos para que ocorra o processo de Integração Contínua:

a) Equipe de suporte realiza a integração do orquestrador com o repositório do GITLAB FONTES;

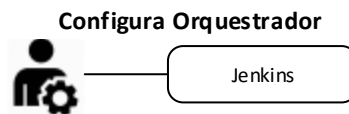


Figura 8 - Configura Orquestrador

b) Desenvolvedor entrega o código fonte no GITLAB FONTES;

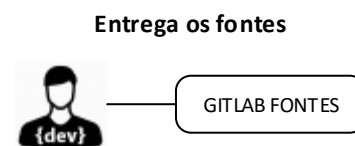


Figura 9 - Entrega de fontes

c) Para que ocorra a compilação, algumas integrações poderão ser realizadas, conforme pode ser observado na figura 10. Cada área de desenvolvimento poderá realizar a integração do jeito que se sentirem mais confortáveis;

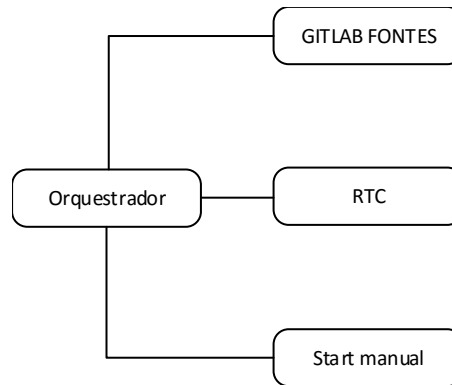


Figura 10 - Integração das ferramentas de IC

- d) Após a compilação do código ou compactação, o resultado deverá ser disponibilizado no servidor do NEXUS;

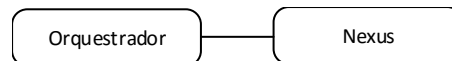


Figura 11 - Disponibilização no NEXUS

- e) O Orquestrador deverá ser a ferramenta que disponibilizará ambientes, assim, sempre que for necessário realizar a construção ou promoção do ambiente, deverá ser configurado o orquestrador para executar tais tarefas;

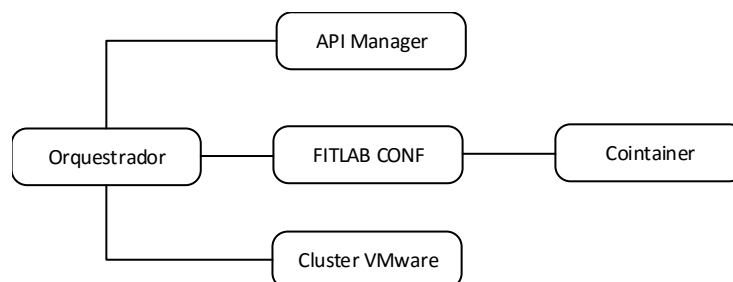


Figura 12 - Disponibilização do ambiente

REQUISITOS DE SEGURANÇA

- Deverá existir uma conta de serviço para que cada ferramenta (GITLAB FONTES, GITLAB CONFIG, Jenkins e *Origin*) valide as credenciais no OpenLDAP;

- Deverá existir uma Unidade de Organização (OU) para cada sistema no OpenLDAP, sob a OU=AGIL. Ex.: SIXYZ;
- Deverão existir três Unidades Organizacionais sob a OU anterior para separação de perfis nas ferramentas quando sincronizadas e acessados pelo SIGAL;
- Deve ser realizado um mapeamento no SIAAS vinculando os grupos do OPENLDAP com os gestores das aplicações;
- As permissões de acesso devem ser específicas por sistema e solicitadas via SIGAL (acessologico.caixa);
- Possuir um script que sincroniza os usuários pertencentes a cada perfil da Unidade Organizacional do OpenLDAP com os respectivos grupos das ferramentas (GIT's e *Origin*), conforme figuras abaixo:

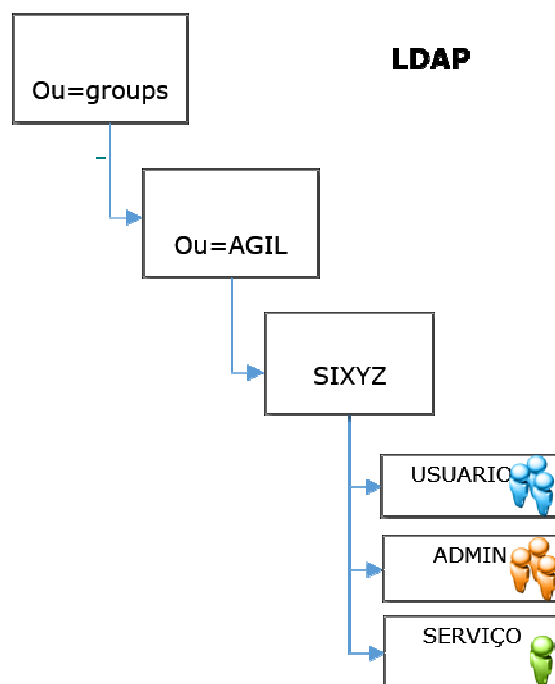


Figura 13 - Estrutura LDAP

- Deverá ser configurado um repositório único de fontes, sem distinção de unidade da CAIXA.

- Deverá ser criado um grupo no GITLAB FONTES com o nome do sistema na raiz da ferramenta.
- O GITLAB FONTES deve possuir os projetos dentro do grupo criado no item anterior. Caso possua somente um projeto, deverá ser criado o grupo e o projeto com o mesmo nome.

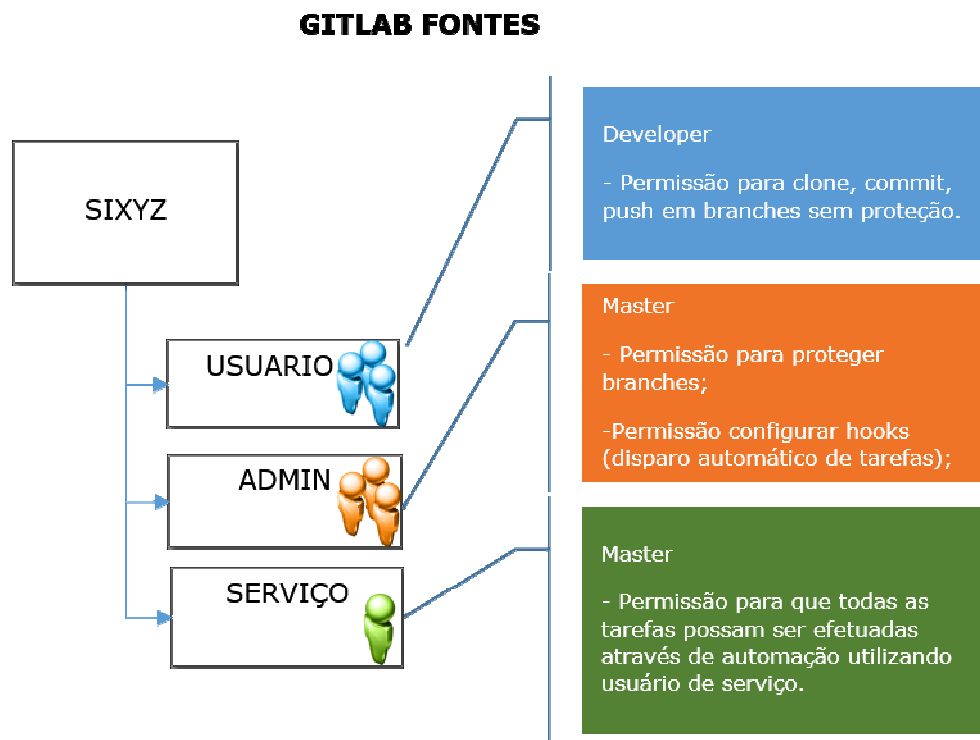


Figura 14 – Estrutura GitLab fontes

- Deverá ser configurado um repositório único de configuração, sem distinção de unidade da CAIXA.
- Deverá ser criado um grupo no GITLAB FONTES com o nome do sistema na raiz da ferramenta.
- O GITLAB CONF deve possuir os projetos dentro do grupo criado no item anterior.
- Cada projeto fará referência a um ambiente (DES, TQS, HMP e PRD), respeitando a nomenclatura presente na imagem abaixo.

GITLAB CONF

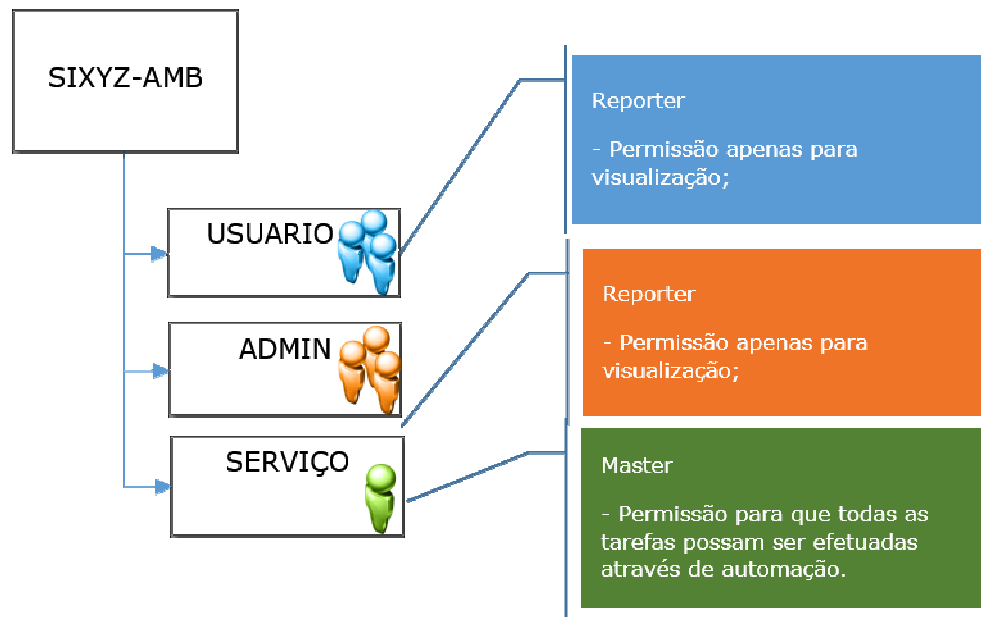


Figura 15 – Estrutura GitLab Conf

- Deverá ser configurado um projeto para cada sistema e para cada ambiente caso seja utilizado Container.
- A criação de ambientes deverá ser realizada pela camada de Middleware.
- Caso o processo seja manual, o ambiente deverá ser criado exclusivamente pelas equipes de suporte de desenvolvimento.
- Cada projeto fará referência a um ambiente (DES, TQS, HMP e PRD), respeitando a nomenclatura presente na imagem abaixo.

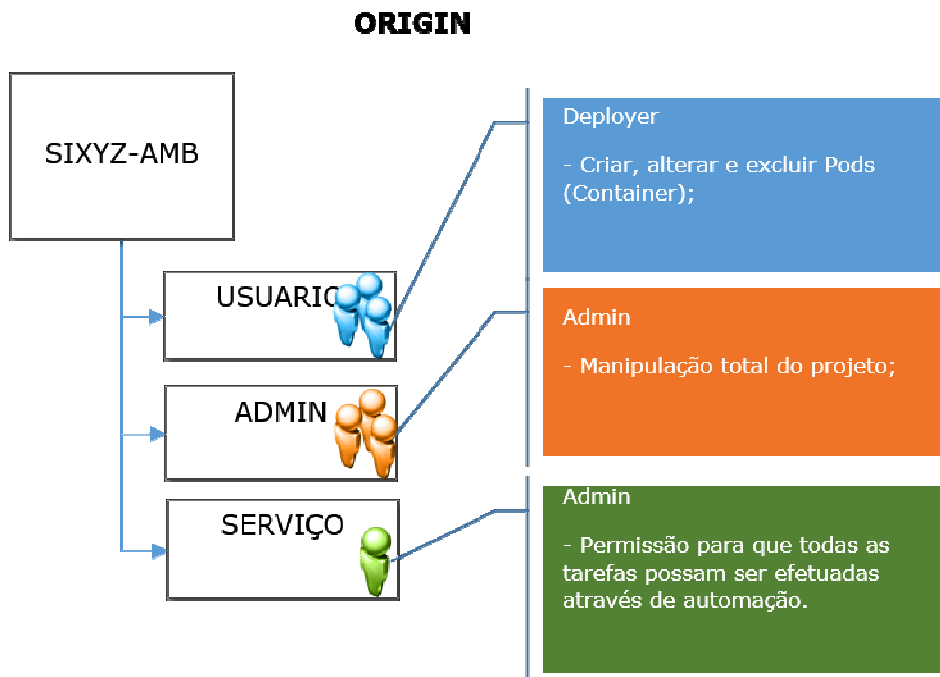


Figura 16 - Estrutura ORIGIN

- Para simplificação das regras de firewall, deverá ser criado um grupo de regras no firewall.
- O grupo do firewall deverá conter uma série de IPs, que são os servidores de Docker Host de cada uma das zonas criadas para segmentação do ambiente.
- É de responsabilidade do solicitante realizar pedido da regra de firewall à equipe da CETAD informando o grupo de firewall no qual o sistema pertence, origem, destino, protocolo, porta e direção da comunicação.
- A equipe responsável pela administração do ambiente deverá solicitar a criação dos grupos de regras no Firewall.
- Deverão ser configurados grupos de Brasília, Rio de Janeiro, São Paulo, PEDeS e Departamental. A construção das zonas do Origin deverá obedecer a mesma lógica.
- É de responsabilidade da equipe que realiza a administração do ambiente manter atualizadas as informações das regras de firewall referentes aos grupos no firewall,

adicionando-se ou subtraindo-se informações sempre que houver alguma modificação nos servidores que compõem as zonas.

PADRÃO DE NOMECLATURA NAS FERRAMENTAS

Cada sistema deverá obedecer um padrão de nomenclatura conforme a ferramenta e ambiente. Supondo que o nome do sistema seja SIXYZ, deverá ser criada a estrutura conforme abaixo:

- GIT Fontes – **SIXYZ**;
- GIT Configurações – **SIXYZ-AMBIENTE**. Ex: SIXYZ-DES; SIXYZ-TQS; SIXYZ-HMP; SIXYZ-PRD;
- Nexus – **SIXYZ**;
- Jenkins – **SIXYZ-AMBIENTE**;
- *Origin* – **SIXYZ-AMBIENTE**, como no GIT Configurações. O acesso ao sistema se dará por **SIXYZ-AMBIENTE.NPRD.CAIXA** (ex: SIXYZ-DES.NPRD.CAIXA).

PROCESSO DE CRIAÇÃO DO AMBIENTE ÁGIL

Após a instalação das ferramentas, deverão ser criados grupos locais com os respectivos administradores, por exemplo, no GIT FONTES, deverão ser criados grupos QUALIDADE_BR, QUALIDADE_RJ, QUALIDADE_SP, QUALIDADE_PEDES, QUALIDADE_CETEC.

Cada um desses grupos deve conter os funcionários responsáveis pela administração da ferramenta.

O processo de criação de ambiente se dará em duas formas distintas e deverá ocorrer para cada ambiente (DES, TQS, HMP e PRD):

PRIMEIRA – Ambiente automatizado;

SEGUNDA – Criação manual do ambiente.

- 1) Para realizar a criação completamente automatizada do ambiente:
 - a) Deverá ser utilizado o sistema de DAS, pelo endereço <http://sidas.caixa>.
 - a. No SIDAS será preenchido todas as informações necessárias sobre a arquitetura do sistema, por exemplo, comunicações com banco, filas MQ, comunicação com outros sistemas, etc.
 - b. Com essas informações, serão gerados insumos necessários e suficientes para realizar o Deploy no servidor e ter o sistema ativo.
 - c. As configurações do sistema serão armazenadas no GITLab de configurações para uso posterior.
 - d. Caso não possua acesso ao SIDAS, deverá ser preenchida FICUS e encaminhada à equipe da CEDESJRJ041.
 - b) Com as informações preenchidas no SIDAS, o sistema enviará as informações, como nome, ambiente, tecnologia e configurações para um Middleware, construído pelo PEDeS, sob supervisão da SUCTI06.
 - c) O Middleware receberá as informações e deverá realizar as seguintes ações:
 - a. Realizar a criação de grupo de LDAP conforme nome do sistema e subgrupos com 03 níveis de permissionamento.

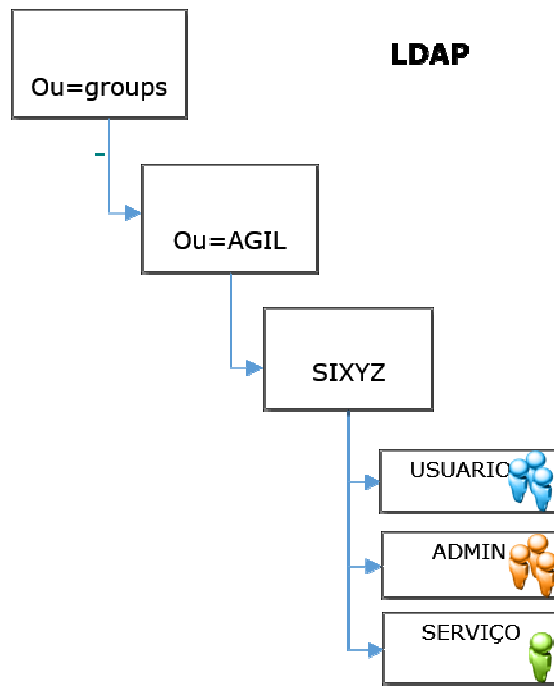


Figura 17 – Modelo LDAP

- b. Caso o sistema tenha que acessar a internet, deverá ser realizado a criação de um usuário de serviço e adicioná-lo ao grupo serviço (domínio corp.caixa.gov.br).
- c. Diante das informações repassadas pelo SIDAS, deverá ser realizada a criação do grupo e do projeto no GIT de fontes, com visibilidade *private*.
- d. O projeto deverá ser compartilhado com o grupo local dos administradores, que devem possuir perfil Master.

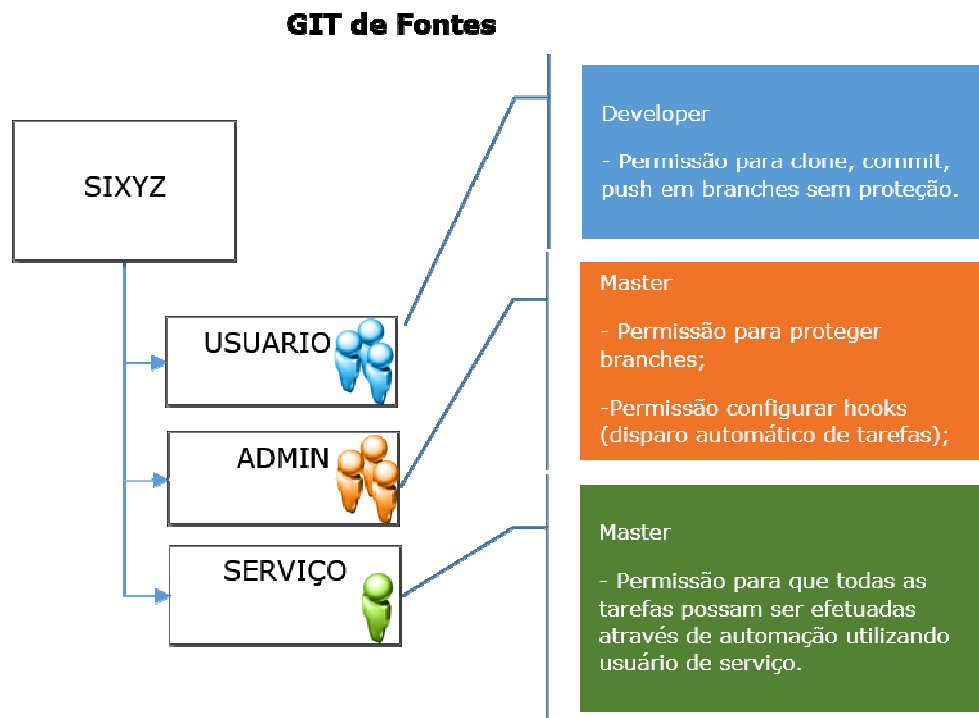


Figura 18 - Modelo GitLab Fontes

- e. Após a criação do repositório no GIT de fontes, deverá ser realizada a criação do repositório no GIT de Configurações com visibilidade *private*.
- f. O projeto deverá ser compartilhado com o grupo local dos administradores, que devem possuir perfil Master.

GIT de Configurações

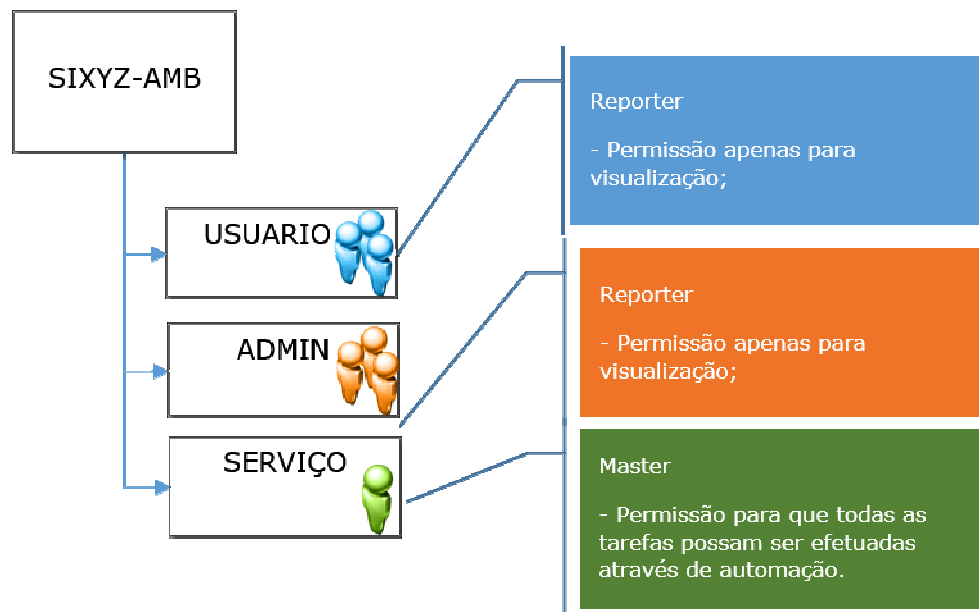


Figura 19 - Modelo GitLab CONF

- g. Após a criação do repositório, deverá ser armazenado no mesmo, o arquivo de configuração necessário para realizar a execução posterior, por exemplo, standalone.xml no caso de aplicações JEE.
- h. Realizar a criação do projeto do Jenkins, para que as equipes de desenvolvimento realizem as configurações posteriormente.

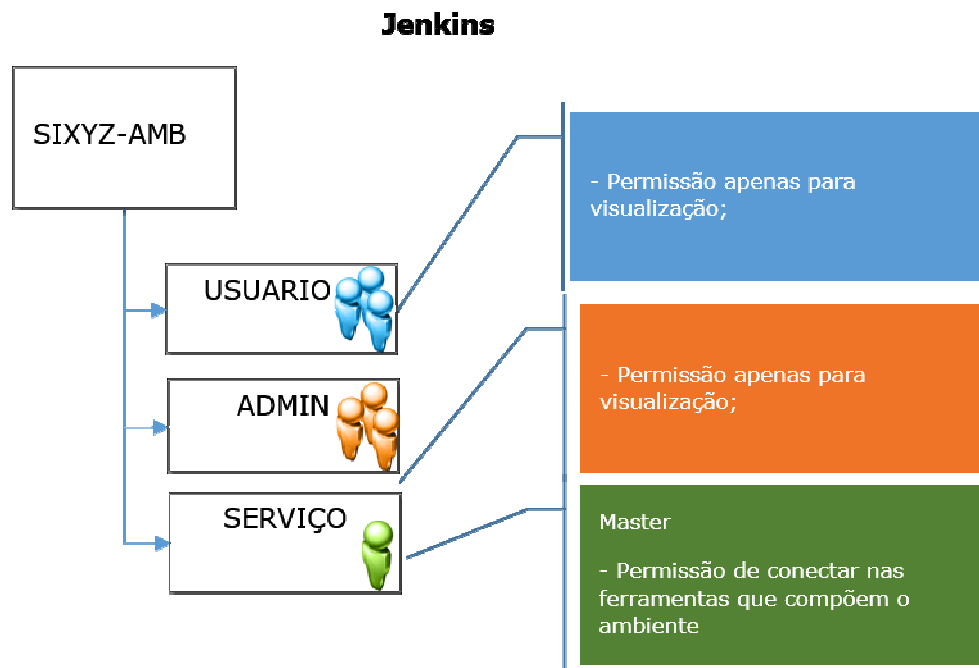


Figura 20 - Modelo Jenkins

- i. Realizar a criação do projeto no Origin

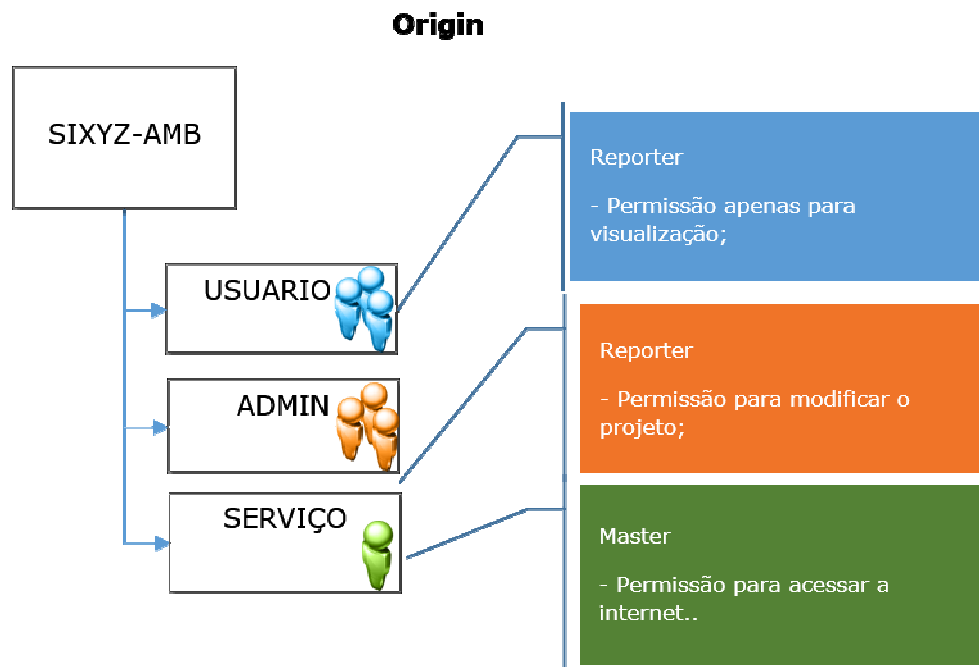


Figura 21 - Modelo ORIGIN

- j. Caso o sistema necessite acesso à Internet (informação passada pelo SIDAS), deverá ser configurado o Proxy do projeto no Origin.

- k. Configurar o Proxy para liberar acesso de internet ao usuário.
- d) O solicitante deverá solicitar à sua respectiva equipe de segurança, via serviços.caixa a criação de um MPAS pelo sistema siaas.caixa, informando:
- a. Criação de um MPAS com o nome do sistema e perfil de usuário (conforme estrutura criada no LDAP). Nesse perfil deverão estar todos os desenvolvedores.
 - b. Criação de um MPAS com o nome do sistema e perfil de administração (conforme estrutura criada no LDAP). Nesse perfil deverão estar todos os integrantes do SAI no ambiente de desenvolvimento, por exemplo.
- e) Após a criação, acessar o SIGAL (acessologico.caixa) e dar permissão para cada colaborador que participa do projeto, seja com permissão de desenvolvedor ou administrador, conforme construção de perfil de acessos realizada no item anterior.
- f) CEPTIBR desenvolveu um script que verifica o LDAP periodicamente e concede as devidas permissões nas ferramentas GITLab FONTES e CONFIG, Origin e NEXUS de acordo com o nível de acesso demonstrado na figura acima.
- g) Automaticamente deverão ser criados todos os permissionamentos nas ferramentas que compõem a solução de disponibilização ágil de ambiente.
- 2) A segunda maneira corresponde ao processo de criação do ambiente de forma manual, e, para isso, alguns passos deverão ser executados, conforme abaixo:
- a) A área de desenvolvimento solicita a criação de grupos no OpenLDAP com 03 subgrupos, conforme demonstrado em “REQUISITOS DE SEGURANÇA – Imagem 01” para cada um dos sistemas que deseja realizar a configuração.
 - a. A solicitação deve ser encaminhada via FICUS para a equipe de segurança das Centralizadoras (CEPTIXX055), conforme nomes padrão definido no tópico REQUISITOS DE SEGURANÇA.
 - b) Caso o sistema necessite acesso à Internet (informação que consta no DAS), deverá ser solicitado um usuário de serviço à equipe de segurança das equipes de Operações, preenchendo o MO15128 e enviando anexo, conforme TE191.

- c) A área de desenvolvimento solicita à CEDES/RJ via item de trabalho na área de projeto CEDESJRJ040 – Coordenação Arquitetura na ferramenta RTC a criação do projeto no Nexus.
- d) A equipe de desenvolvimento solicita à equipe de gerenciamento de configurações da CEDES ou à equipe de suporte do PEDeS para que seja realizado a criação de um novo projeto no GitLab de fontes e de configuração, seguindo os padrões de nomenclaturas definidos nas imagens 13 e 14.
- e) A equipe de desenvolvimento solicita à equipe da CEPTIBR21 a criação de um novo projeto no ORIGIN via ITSM. Caso o sistema utilize Internet, deverá ser passado o usuário de serviço e senha para configuração do Proxy no Origin.
- f) Caso o sistema utilize Internet, solicitar à CETAD via ITSM a configuração de acesso à Internet, no proxy, informando o usuário e senha.
- g) O solicitante deverá solicitar à sua respectiva equipe de segurança, via serviços.caixa a criação de um MPAS pelo sistema siaas.caixa, informando:
 - a. Criação de um MPAS com o nome do sistema e perfil de usuário (conforme estrutura criada no LDAP). Nesse perfil deverão estar todos os desenvolvedores.
 - b. Criação de um MPAS com o nome do sistema e perfil de administração (conforme estrutura criada no LDAP). Nesse perfil deverão estar todos os integrantes do SAI no ambiente de desenvolvimento, por exemplo.
- h) Solicitante, via acessologico.caixa, deverá adicionar/vincular os usuários ao SISTEMA obedecendo os níveis de permissionamento que cada um deve possuir (PROCESSO DE CRIAÇÃO DO AMBIENTE, item “c”, subitem “a”).

O processo é completamente manual, devendo ser substituído pelo processo automatizado utilizando a integração SIDAS/Camada de Middleware/demais ferramentas.

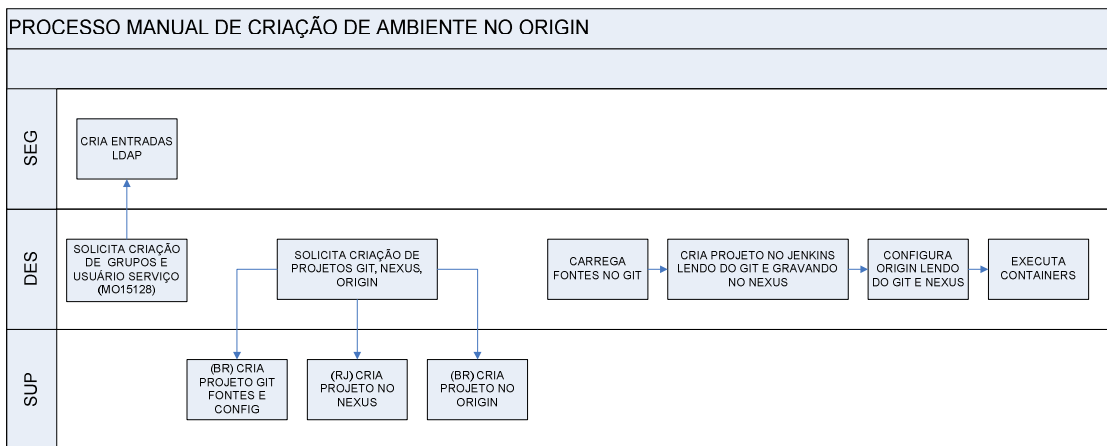


Figura 22 - Processo criação ambiente

PROCESSO DE INTEGRAÇÃO CONTÍNUA

Após a entrega do ambiente, os passos a seguir deverão ser executados para que haja a compilação automática do fonte e a disponibilização automática no ambiente:

1. Disponibilização do código fonte no GitLab de fontes. Uma vez que os acessos já foram realizados através da montagem do ambiente e da disponibilização e preenchimento da MPAS.
 - a. É imprescindível que o código fonte esteja no repositório do GitLab.
 - i. Uma vez que a ferramenta TFS, orquestrador utilizado pela equipe do PDeS já possui uma estrutura completa de Integração Contínua, a mesma deve ser configurada para realizar integração com as ferramentas aqui descritas e não utilizar seus repositórios internos.
2. A equipe de arquitetura de desenvolvimento deverá acessar o orquestrador, e realizar as configurações necessárias para a compilação e testes do código fonte:
 - a. Integrar com a ferramenta de teste RQM, que tem como objetivo orquestrar a execução dos testes planejados. O RQM retornará os registros de execução dos testes realizados, o qual deverá ser avaliado com a equipe de qualidade de testes de cada site, que definirá se o pacote poderá ser promovido para o próximo ambiente.

- b. Integrar com as ferramentas de qualidade de código SONAR e EQJ, que tem como objetivo realizar a análise estática de código fonte. Os registros de saída possuirão uma pontuação, o qual deverá ser avaliado com a equipe de qualidade de arquitetura de cada site, que definirá se o fonte será compilado e se pacote poderá ser promovido para o próximo ambiente.
 - c. Integrar com a ferramenta de qualidade de segurança FORTIFY, que tem como objetivo realizar a análise estática de código seguro. Os registros de saída possuirão uma pontuação, o qual deverá ser avaliado com a equipe de qualidade de testes de cada site, que definirá se o fonte será compilado e se pacote poderá ser promovido para o próximo ambiente.
- 3. Para realizar a compilação, é necessário que o time de desenvolvimento realize suas configurações do NEXUS utilizando o repositório “caixa-group”, que possui os apontamentos para repositórios internos e externos.
- 4. Caso não exista um projeto criado no NEXUS, é necessário que seja realizada a solicitação de inclusão de novo repositório via RTC para a equipe de arquitetura da CEDESJ.
- 5. Com o código compilado, o Orquestrador deverá integrar com o repositório de binários NEXUS, onde deverá ser disponibilizado o binário no caso de linguagens compiladas, e arquivo compactado no caso de linguagens não compiladas, no repositório caixa-group (<http://binario.caixa:8081/repository/caixa-group/>)
 - a. A estrutura foi criada para ser agnóstica quanto à localidade, por esse motivo, não há segmentação de Brasília, São Paulo e Rio de Janeiro.
 - b. Todas as bibliotecas desenvolvidas internamente por cada unidade, deverá ser disponibilizada no repositório próprio destinado às elas (br, sp, rj, PEDeS, departamental).
 - c. Os repositórios internos deverão ser acrescentados ao repositório caixa-group (<http://binario.caixa:8081/repository/caixa-group/>) nas configurações do NEXUS.

6. Após a disponibilização do binário, o Orquestrador deverá guardar quaisquer configurações adicionais no repositório GitLab de configuração, no repositório referente ao projeto compilado.

PROCESSO DE DISPONIBILIZAÇÃO DE AMBIENTE

Uma vez que o ambiente está criado e o código fonte foi compilado, testado e as configurações estão disponibilizadas no GitLab de configurações, pode-se realizar a integração com algumas formas diferentes de provimento de ambiente.

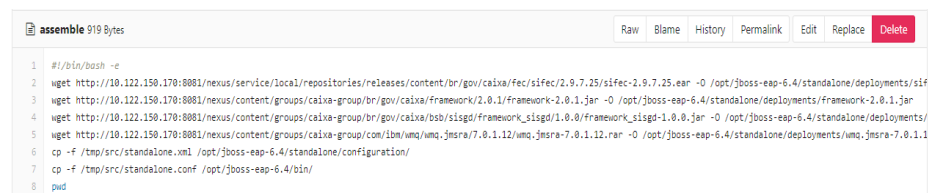
1) Containers

- a) Na primeira vez, deverá ser acessado o projeto pelo administrador (processo de acesso descrito anteriormente) o qual deverá realizar a adição da sua plataforma (produtos) utilizando o catálogo de serviços disponibilizado.
- b) Para realizar a criação automatizada do ambiente de Containers, deve-se criar uma estrutura no GitLab de Configurações conforme abaixo:

i) Criar uma pasta oculta chamada **.s2** e uma pasta chamada **bin**.

ii) Nessa estrutura, realizar a criação de dois arquivos. **assemble** e **run**.

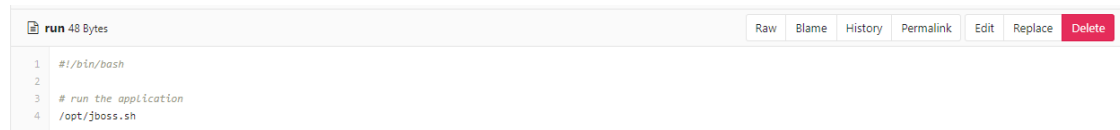
- (1) O arquivo **assemble**, deverá possuir como conteúdo um scripts desenvolvidos na linguagem shell scrip para realizar o download das bibliotecas que serão utilizadas pelo sistema e do arquivo binário que foi previamente gravado no servidor de repositório NEXUS. Também será necessário realizar a cópia dos arquivos de configuração para as pastas de destino, por exemplo:



```
1 #!/bin/bash -e
2 wget http://10.122.150.170:8081/nexus/service/local/repositories/releases/content/br/gov/caixa/fec/sifec/2.9.7.25/sifec-2.9.7.25.ear -O /opt/jboss-eap-6.4/standalone/deployments/sifec-2.9.7.25.ear
3 wget http://10.122.150.170:8081/nexus/content/groups/caixa-group/br/gov/caixa/framework/2.0.1/framework-2.0.1.jar -O /opt/jboss-eap-6.4/standalone/deployments/framework-2.0.1.jar
4 wget http://10.122.150.170:8081/nexus/content/groups/caixa-group/br/gov/caixa/bsb/sisgd/framework_k_sisgd/1.0.0/framework_k_sisgd-1.0.0.jar -O /opt/jboss-eap-6.4/standalone/deployments/framework_k_sisgd-1.0.0.jar
5 wget http://10.122.150.170:8081/nexus/content/groups/caixa-group/com/ibm/wmq/jmsra/7.0.1.12/wmq_jmsra-7.0.1.12.rar -O /opt/jboss-eap-6.4/standalone/deployments/wmq_jmsra-7.0.1.12.rar
6 cp -f /tmp/src/standalone.xml /opt/jboss-eap-6.4/standalone/configuration/
7 cp -f /tmp/src/standalone.conf /opt/jboss-eap-6.4/bin/
8 pwd
```

Figura 23 - Arquivo *assemble*

- (2) Já o arquivo **run** deverá possuir o conteúdo para iniciar o jboss quando o Container for iniciado.



```
run 48 Bytes
1 #!/bin/bash
2 #
3 # run the application
4 /opt/jboss.sh
```

Figura 24 - Arquivo run

- c) Deverá ser criada uma integração do Origin com o GitLab utilizando o parâmetro **GitHub Webhook URL** do Origin com o *Webhook* do GitLab. A finalidade desse passo é realizar a criação automática do ambiente utilizando containers sempre que um novo arquivo de configuração for adicionado ao GitLab, por exemplo, a modificação da versão presente no arquivo assemble.
- d) Configurar o Orquestrador para forçar a criação de um novo arquivo no GitLab fará com o que o arquivo seja sobrescrito, realizando a criação automática do Container da aplicação.

2) Servidores Virtuais

- a) Caso a instância ainda não esteja criada, deverá ser configurado o Orquestrador para que realize a interação com o ambiente e realize a criação de uma nova instância do cluster, por exemplo, JBoss EAP.
- b) Para servidores virtuais que já existe o ambiente criado, deverá ser realizado a integração do Jenkins com o cluster que realiza o provimento da aplicação.
- c) Após a disponibilização do arquivo no NEXUS (arquivos compilados /ou compactados), o Orquestrador deverá buscar as configurações no GitLab de configurações e os arquivos no NEXUS e realizar o Deploy no cluster previamente configurado.
- d) É facultado aos administradores dos orquestradores utilizar plug-ins que lhes convenham, por exemplo, WildFly para Jenkins+JBoss.

PROCESSO DE MONITORAÇÃO DE IAAS

1. Toda a monitoração das zonas do ambiente deverá ser realizada pela equipe responsável pela manutenção do ambiente, utilizando a ferramenta vROPS, podendo esta ser complementada com outras ferramentas de gerenciamento.
2. A CEPTIBR será a equipe responsável por realizar o gerenciamento de capacidade das zonas, assim, quando uma zona começar a ficar sem recurso, a equipe da CEPTIBR deverá

realizar a construção de um novo servidor e adicioná-lo à zona de forma tempestiva e proativa.

3. Deverá ser configurado um portal que seja capaz de realizar a configuração/monitoração do ambiente de servidores virtuais e containers.
4. Por esse portal, os administradores das zonas poderão realizar solicitações de novos servidores para adicioná-los às suas zonas de Containers quando for percebido que os recursos computacionais das zonas estiverem acabando.
5. A ferramenta deverá se integrar em ambientes de nuvens distintos, bem como fornecer informações de custos aos solicitantes.

PROCESSO DO PROMOÇÃO DO AMBIENTE/ENTREGA CONTÍNUA

1. Toda a promoção da pacotes entre ambientes deverá ser realizada unicamente pelos orquestradores, eliminando-se trabalhos manuais, e, por consequência, diminuindo-se os riscos operacionais.
2. Os orquestradores deverão ser integrados com a ferramenta NEXUS para que seja possível realizar *deploys* de ambiente, realizando um apontamento para a versão que se deseja utilizar, como por exemplo: selecionando-se a versão.
3. O orquestrador, deverá ser único para ambientes não produtivos, podendo-se utilizar um orquestrador complementar no ambiente de produção.
4. Deve-se integrar os orquestradores com a ferramenta ITSM, responsável pelo gerenciamento de mudanças e CMDB, registrando-se todas as mudanças no ambiente.

1) PROMOÇÃO PARA TQS

a) Ambientes centralizados:

- i) O time de desenvolvimento será responsável por realizar a promoção do sistema do ambiente de desenvolvimento para o ambiente de testes.
- ii) O solicitante deverá acessar o projeto no ambiente necessário no Orquestrador e solicitar a construção do ambiente, apontando-se para a versão no qual se deseja realizar o *deploy* (ex: SIXYZ-TQS).
- iii) Caso o sistema não possua ambiente de testes (TQS), deverá ser acessado o ambiente subsequente, onde o time de operações deverá realizar a promoção do ambiente de desenvolvimento para a homologação ou para produção.

- iv) Somente para o PEDeS, o time de desenvolvimento poderá realizar, por meio do orquestrador, a promoção do binário do ambiente de desenvolvimento até o ambiente de produção.
 - v) Deverão ser criadas áreas ou projetos diferentes para que cada equipe visualize somente o seu ambiente (DES, TQS, HMP e PRD) e somente os seus projetos.
 - vi) O orquestrador deverá ser integrado com o GITLAB CONF para que seja possível visualizar as versões que foram construídas, uma vez que podem haver casos em que há arquivos binários no NEXUS que não foram construídos nos ambientes.
 - vii) Ao confirmar a construção, o Orquestrador deverá realizar o download do binário ou dos arquivos compactados na ferramenta Nexus (binario.caixa) e as configurações no GITLAB CONF (cloudconfig.caixa).
 - viii) O orquestrador deverá ser configurado para interagir com o ambiente de destino para que seja realizado o Deploy.
 - ix) Os passos para a construção do ambiente devem ser os mesmos, independentemente do tipo de ambiente, seguindo os passos descritos na seção “PROCESSO DE CONSTRUÇÃO DO AMBIENTE”.
 - x) Se o sistema utilizar Containers:
 - (1) O servidor GITLAB CONF deve ser configurado e integrado com o ORIGIN
 - (2) Deverão ser seguidos os passos presentes em PROCESSO DE DISPONIBILIZAÇÃO DO AMBIENTE.
- b) Ambientes descentralizados (PEDeS):
- i) No ambiente descentralizado (PEDeS), o time de desenvolvimento será responsável por realizar a promoção do ambiente desde o ambiente de desenvolvimento até a produção.
 - ii) Os passos necessários para a construção e disponibilização do ambiente devem ser os mesmos, conforme descrito no item a).
 - iii) Somente para o PEDeS, é permitida a utilização do Orquestrador Microsoft TFS, uma vez que as equipes já utilizam a ferramenta e já estão habituadas.
- a) Ambientes corporativos (*shadow* IT):
- i) O time de desenvolvimento será responsável por realizar a promoção do sistema do ambiente de desenvolvimento para os demais ambientes.
 - ii) Os passos necessários para a construção e disponibilização do ambiente devem ser os mesmos, conforme descrito no item 1.

Caso o sistema não possua ambiente de TQS ou HMP, a promoção do ambiente deverá ser realizada no ambiente subsequente obedecendo as mesmas características de construção já descritas anteriormente.

2) PROMOÇÃO PARA HMP

- a) O time de operações será responsável por realizar a promoção do sistema do ambiente de desenvolvimento para o ambiente de homologação.
- b) O solicitante deverá acessar o projeto no ambiente necessário no Orquestrador e solicitar a construção do ambiente, apontando-se para a versão no qual se deseja realizar o *deploy* (ex: SIXYZ-HMP).
- c) Caso o sistema não possua ambiente de homologação (HMP), deverá ser acessado o orquestrador do ambiente não produção, onde o time de operações deverá realizar a promoção do ambiente de desenvolvimento para a homologação ou para produção.
- d) Somente para o PEDeS, o time de desenvolvimento poderá realizar, por meio do orquestrador, a promoção do binário do ambiente de desenvolvimento até o ambiente de produção.
- e) Deverão ser criadas áreas ou projetos diferentes para que cada equipe visualize somente o seu ambiente (DES, TQS, HMP e PRD) e somente os seus projetos.
- f) O orquestrador deverá ser integrado com o GITLAB CONF para que seja possível visualizar as versões que foram construídas, uma vez que podem haver casos em que há arquivos binários no NEXUS que não devem ser construídos nos ambientes.
- g) Ao confirmar a construção, o Orquestrador deverá realizar o download do binário ou dos arquivos compactados na ferramenta Nexus (binario.caixa) e as configurações no GITLAB CONF (cloudconfig.caixa).
- h) Os passos para a construção do ambiente devem ser os mesmos, independentemente do ambiente, seguindo os passos descritos na seção “PROCESSO DE CONSTRUÇÃO DO AMBIENTE”.
- i) Se o sistema utilizar Containers:
 - (1) O servidor GITLAB CONF deve ser configurado e integrado com o ORIGIN;
 - (2) Deverão ser seguidos os passos presentes em PROCESSO DE DISPONIBILIZAÇÃO DO AMBIENTE.

3) PROMOÇÃO PARA PRODUÇÃO

- a) Deverá ser criada uma estrutura de *timeline* pré-produção/produção.

- b) Na estrutura pré-produção, deverão ser criados dois *steps*, um que realizará a abertura automática de mudança para que seja avaliada a mudança junto ao colegiado e outro que liberará o *step* de produção.
 - c) Após a aprovação do colegiado, deverá ser acessado o *step* pré-produção e adicionar o número da mudança que foi aprovada.
 - d) O orquestrador deve ser configurado para liberar o *step* de produção (que pode ser realizado via API) após a inserção do número da mudança, que deve ser realizado pela equipe de produção.
 - e) O time de operações será responsável por realizar a promoção do sistema do ambiente de homologação para o ambiente de produção.
 - f) O solicitante deverá acessar o projeto no ambiente necessário no Orquestrador e solicitar a construção do ambiente, apontando-se para a versão no qual se deseja realizar o *deploy* (ex: SIXYZ-PRD).
 - g) Somente para o PEDeS, o time de desenvolvimento poderá realizar, por meio do orquestrador, a promoção do binário do ambiente de desenvolvimento até o ambiente de produção.
 - h) O orquestrador deverá ser integrado com o GITLAB CONF para que seja possível visualizar as versões que foram construídas, uma vez que podem haver casos em que há arquivos binários no NEXUS que não devem ser construídos nos ambientes.
 - i) Ao confirmar a construção, o Orquestrador deverá realizar o download do binário ou dos arquivos compactados na ferramenta Nexus (binario.caixa) e as configurações no GITLAB CONF (cloudconfig.caixa).
 - j) Os passos para a construção do ambiente devem ser os mesmos, independentemente do ambiente, seguindo os passos descritos na seção “PROCESSO DE CONSTRUÇÃO DO AMBIENTE”.
 - a. Se o sistema utilizar Containers:
 - i. O servidor GITLAB CONF deve ser configurado e integrado com o ORIGIN;
- Deverão ser seguidos os passos presentes em PROCESSO DE DISPONIBILIZAÇÃO DO AMBIENTE.

SOLICITAÇÃO DE NOVAS IMAGENS NO CATÁLOGO DE SERVIÇOS DE CONTAINERS

1. Deverá ser construído um *template* na ferramenta ITSM para solicitação de novas imagens que não estão disponibilizadas no Catálogo de Serviços.
2. A solicitação deverá ser direcionada à CEPTIBR21 para que seja construída uma nova imagem.
3. A CEPTIBR21 deverá construir a imagem e informar à GEARQ sobre a disponibilização da nova imagem.
4. A GEARQ deverá, juntamente com algum time técnico da CEPTIBR21, realizar a homologação da imagem.
5. Após a homologação, a CEPTIBR21 deverá disponibilizar a imagem para utilização por meio do Catálogo de Serviços e liberação no Docker Registry.
6. Será de responsabilidade da CEPTIBR21, juntamente com o time de desenvolvimento e com as outras CEPTI realizar a criação de uma imagem padrão com todos os componentes necessários para que a aplicação funcione corretamente. Após a construção, a CEPTIBR21 deverá disponibilizar a imagem criada no Catálogo de Serviços.

REFERÊNCIAS

- Gitlab.com
- <https://www.openshift.org/>
- <https://jenkins.io/>
- <http://www.sonatype.org/nexus/>