

Anexo 2 - Mapeamento das tecnologias do Portal CNCFlora (Produto 4/ POA Ano 1).

Este documento tem a função de documentar quais foram as tecnologias empregadas no desenvolvimento do sistema CNCFlora. Justifica-se de elevada importância pois permite avaliar os pontos fortes do sistema como velocidade, independência entre os serviços utilizados e quais estruturas computacionais necessárias para uma possível instalação em um novo ambiente.

Estrutura atual

Identificamos que o sistema do CNCFlora é composto por vários repositórios independentes criados em Docker que se comunicam entre si através de API. Todos os repositórios ficam localizados nos servidores de desenvolvimento (JB048) e produção (JB049). A construção de um sistema utilizando repositórios Docker possui vantagens significativas, como por exemplo, a independência de arquitetura, redução de infra-estrutura, isolamento de uma eventual pane do sistema entre outras, mais informações em (<https://www.docker.com/why-docker>).

Para cada repositório foi utilizada uma linguagem de programação diferente, tais como: PHP, Python, Ruby e Cloujure.

Os bancos envolvidos na construção do sistema foram o [Apache Derby](#), responsável pela gestão de usuários, o [CouchDB](#) e o [ElasticSearch](#) que são bancos de dados não relacionais, responsáveis pelo armazenamento dos dados brutos do CNCFlora.

Solução Proposta

Após a identificação de alguns pontos de atenção e com o objetivo de dar continuidade ao desenvolvimento com melhorias, foi necessário a implantação de um novo ambiente de homologação para a construção de novos módulos utilizando a linguagem PHP e o Framework Zend na versão 3, como também a criação de um servidor PostgreSQL para a consolidação dos bancos de dados envolvidos.

Descrição das tecnologias

Tabela 1. Descrição da tecnologia do repositório Portal CNCFlora.

Portal CNCFlora	github: https://github.com/CNCFlora/Portal
Repositório de containers:	index.docker.io
Container docker:	cncflora/portal
Linguagem:	Python
Link:	http://cncflora.jbrj.gov.br/portal
Banco de dados:	não possui acesso direto, somente via api's
Descrição:	Página principal do sistema CNCFlora, tem como objetivo apresentar o projeto, disponibilizar informações de domínio público e disponibilizar links para todos os módulos do sistema.
Acesso a API externa ?	SIM
API's externas:	Facebook cncflora.jbrj.gov.br/services:8080 cncflora.jbrj.gov.br/assessments/taxon/ cncflora.jbrj.gov.br/assessments/2taxon/ cncflora.jbrj.gov.br/assessments/3taxon/ cncflora.jbrj.gov.br/assessments/family/ cncflora.jbrj.gov.br/assessments/families/ cncflora.jbrj.gov.br/occurrences/scientificName/

	<p>cncflora.jbrj.gov.br/occurrences/2scientific Name/</p> <p>cncflora.jbrj.gov.br/occurrences/3scientific Name/</p> <p>cncflora.jbrj.gov.br/profiles/taxon/</p> <p>cncflora.jbrj.gov.br/profiles/2taxon/</p> <p>cncflora.jbrj.gov.br/profiles/3taxon/</p>
--	---

Tabela 2. Descrição da tecnologia do repositório Entrypoint.

Entrypoint	github: https://github.com/CNCFlora/entrypoint
Repositório de containers:	index.docker.io
Container docker:	cncflora/entrypoint
Linguagem:	HTML e Javascript
Link:	http://cncflora.jbrj.gov.br/entrypoint/
Banco de dados:	Acesso remoto ao Couchdb através da api
Descrição:	Módulo que tem a função de ser o Menu do sistema, exibindo as famílias botânica no sistema.
Acesso a API externa ?	SIM
API's externas:	cncflora.jbrj.gov.br/couchdb/_all_dbs

Tabela 3. Descrição da tecnologia do repositório Connect.

Connect	github: https://github.com/CNCFlora/Connect
Repositório de containers:	index.docker.io
Container docker:	cncflora/connect
Linguagem:	Clojure
Link:	http://cncflora.jbrj.gov.br/connect/index
Banco de dados:	Apache derby
Servidor Web:	Jetty
Descrição:	<p>Módulo de conexão / gerenciamento / permissões de usuários tem como objetivo autorizar os usuários, criando um token e carregando as permissões para ser compartilhado através de cookie ou sessão (a ser verificado) para outros módulos. Todos os programas dos módulos possuem uma linha para verificar se o usuário foi autenticado.</p> <p>DDL's foram encontradas no arquivo src/flora_connect/db.clj</p> <p>A gerência de dados do derby é feita exclusivamente por programas para acesso ao dados é necessário instalar o banco de dados. No container não existe o sgbd para acesso ver em manual:</p>

	https://db.apache.org/derby/papers/DerbyTut/install_software.html#derby_download
<p>Passo a passo para instalação:</p>	<ul style="list-style-type: none"> Foi realizado o download para uma pasta comum da minha máquina e logo após foi realizado uma cópia para o servidor jb048. <pre>scp db-derby-10.14.2.0-bin.tar.gz andreppinghaus@jb048:~/.</pre> Utilizando o docker é necessário pegar o id do container no servidor: <pre>sudo docker ps grep connect</pre> Realize a cópia do banco para o servidor: <pre>sudo docker cp db-derby-10.14.2.0-bin.tar.gz f5ec6d45f121:/.</pre> Conecte no servidor para o início da instalação: <pre>sudo docker exec -i -t f5ec6d45f121 /bin/bash</pre> Inicie a instalação com os comandos abaixo: <pre>mkdir -p /opt/Apache mv db-derby-10.14.2.0-bin.tar.gz /opt/Apache/ cd /opt/Apache/ tar -xzf db-derby-10.14.2.0-bin.tar.gz</pre>

	<pre> export DERBY_HOME=/opt/Apache/db-derby-1 0.14.2.0-bin export DERBY_INSTALL=/opt/Apache/db-derb y-10.14.2.0-bin export CLASSPATH=\$DERBY_INSTALL/lib/d erby.jar:\$DERBY_INSTALL/lib/derbyto ols.jar:. cd \$DERBY_INSTALL/bin . setEmbeddedCP • Testar: java org.apache.derby.tools.sysinfo • Iniciando o client IJ para verificar as tabelas no servidor: java org.apache.derby.tools.ij • Conectando no banco de dados: connect 'jdbc:derby:/var/floraconnect/users'; • Agora basta realizar uma consulta: select * from users; • Exibe as tabelas: show tables; • Exibe o conteúdo das tabelas: descreve users; </pre>
<p>Problemas encontrados</p>	<p>O banco de dados não deve ser instalado em ambientes de produção, pois não é robusto o suficiente, a comunidade pede que se utilize outros bancos como o PostgreSQL ou MySQL.</p>

	<p>O sistema de backup não é muito aprimorado, necessitando realizar programas próprios.</p> <p>O acesso às tabelas não é amigável, necessitando a instalação do banco no servidor web.</p> <p>A DDL apresentada acima possui problemas pois não existe foreign key e os nomes das tabelas são referenciadas em outras tabelas. Neste modelo, é impossível trocar a descrição dos registros pelos usuários.</p>
Solução proposta	Alterar o banco de dados para PostgreSQL, remodelar as tabelas com utilização de chaves estrangeiras.

Tabela 4. Descrição da tecnologia do repositório Assessment.

Assessment	github: https://github.com/CNCFlora/assessment
Repositório de containers:	index.docker.io
Container docker:	cncflora/assessment
Linguagem:	Ruby
Link:	http://cncflora.jbrj.gov.br/assessment/

Banco de dados:	Acesso remoto ao Couchdb e ElasticSearch através da API
Descrição:	Módulo responsável por realizar o processo de Avaliação das espécies
Acesso a API externa ?	Não

Tabela 5. Descrição da tecnologia do repositório Occurrences.

Occurrences	github: https://github.com/CNCFlora/occurrences
Repositório de containers:	index.docker.io
Container docker:	cncflora/occurrences
Linguagem:	PHP
Link:	http://cncflora.jbrj.gov.br/occurrences
Banco de dados:	Acesso remoto ao Couchdb e ElasticSearch através da API.
Descrição:	Módulo responsável por realizar o processo de Ocorrência e geolocalização das espécies.
Acesso a API externa ?	Não

Tabela 6. Descrição da tecnologia do repositório Checklist.

Checklist	github: https://github.com/CNCFlora/checklist
Repositório de containers:	index.docker.io
Container docker:	cncflora/checklist
Linguagem:	Ruby
Link:	http://cncflora.jbrj.gov.br/checklist
Banco de dados:	Acesso remoto ao Couchdb e ElasticSearch através da API.
Descrição:	Módulo responsável por realizar o processo de Taxonomia e Perfil das espécies.
Acesso a API externa ?	Não

Tabela 7. Descrição da tecnologia do repositório Report.

Reports	github: https://github.com/CNCFlora/reports2
Repositório de containers:	index.docker.io
Container docker:	cncflora/reports2

Linguagem:	PHP
Link:	http://cncflora.jbrj.gov.br/reports2
Banco de dados:	Acesso remoto ao Couchdb e ElasticSearch através da api
Descrição:	Módulo responsável pela geração de vários relatórios no sistema.
Acesso a API externa ?	Não

Tabela 8. Descrição da tecnologia do repositório Services.

Services	github: https://github.com/CNCFlora/services
Repositório de containers:	index.docker.io
Container docker:	cncflora/services
Linguagem:	Ruby
Link:	http://cncflora.jbrj.gov.br/services
Banco de dados:	Acesso remoto ao Couchdb e ElasticSearch através da api
Descrição:	Módulo que concentra todas as API's do sistema.
Acesso a API externa ?	Não

Tabela 9. Descrição da tecnologia do banco Elasticsearch.

Elasticsearch	
Descrição	Banco de dados não relacional
Verificação de versão:	<pre>andreepinghaus@jb048:~\$ curl -XGET localhost:9200 { "status" : 200, "name" : "Midnight Sun", "cluster_name" : "docker_es", "version" : { "number" : "1.4.4", "build_hash" : "c88f77ffc81301dfa9dfd81ca2232f09588bd 512", "build_timestamp" : "2015-02-19T13:05:36Z", "build_snapshot" : false, "lucene_version" : "4.10.3" }, "tagline" : "You Know, for Search" }</pre> <p>Nota: A última versão do elasticsearch é a 6.5 e o último manual é a versão 5. Necessidade de atualização constante.</p>
Verificando os nodes existentes:	<pre>curl -X GET "localhost:9200/_cat/nodes?v" host ip heap.percent ram.percent load node.role master name</pre>

	<pre>02ec0c074e7c 172.19.0.12 13 28 0.20 d * Midnight Sun</pre>
<p>Verificando os índices:</p>	<pre>andreppinghaus@jb048:~\$ curl -X GET "localhost:9200/_cat/indices?v" health status index pri rep docs.count docs.deleted store.size pri.store.size yellow open teste_2015_08_28_11:05 5 1 17 0 82.9kb 82.9kb yellow open outros_2015_10_22_16:55 5 1 654 0 1.5mb 1.5mb yellow open endemicas_rio_de_janeiro_2016_01_15_13: 59 5 1 24188 4 39.8mb 39.8mb yellow open reavaliacoes 5 1 9 0 121.9kb 121.9kb yellow open novas_especies_2015_10_22_16:56 5 1 228 0 868.4kb 868.4kb yellow open public_2015_08_28_10:59 5 1 335076 9 296.4mb 296.4mb yellow open especies_indicadas_2015_10_22_16:53 5 1 1122 0 2.7mb 2.7mb yellow open pan_grao_mogol_2015_10_22_15:00 5 1 283 0 586.9kb</pre>

	586.9kb
	yellow open test_suite_db
	5 1 3 0 10.4kb 10.4kb
	yellow open
	pan_bacia_alto_tocantins_2015_10_22_15:
	00 5 1 839 0 1.4mb
	1.4mb
	yellow open
	plantas_raras_do_cerrado_2015_10_22_15:
	01 5 1 16185 0 31.4mb
	31.4mb
	yellow open
	livro_vermelho_2013_2015_08_28_10:49
	5 1 335076 0 287.9mb
	287.9mb
	yellow open
	test_suite_db2_2015_08_28_11:05 5
	1 3 0 10.4kb 10.4kb
	yellow open id-test-index
	5 1 2 0 5.4kb 5.4kb
	yellow open .kibana
	1 1 19 3 174.1kb
	174.1kb
	yellow open
	pan_espinhaco_meridional_2015_10_22_16
	:54 5 1 8499 0 9mb
	9mb
	yellow open reavaliacao_portaria_443
	5 1 7468 1220 9.4mb
	9.4mb
	yellow open

	<pre>plantas_raras_do_cerrado_revisao 5 1 16185 0 29.8mb 29.8mb</pre>
Realizando uma busca com todos os registros e exibindo em json:	<pre>curl -X GET "localhost:9200/endemicas_rio_de_janeiro_ 2016_01_15_13:59/_search?pretty" less</pre>
Realizando uma busca de um registro	<p>Foi criado um arquivo chamado busca1_elastic.sh no servidor jb048 pasta raiz do andreppinghaus com o seguinte conteúdo:</p> <pre>curl -X GET "localhost:9200/endemicas_rio_de_janeiro_ 2016_01_15_13:59/_search?pretty" -H 'Content-Type: application/json' -d' { "query": { "bool": { "must": [{ "match": { "scientificNameAuthorship": "(Hiern) Leonard" } }] } } }</pre>

Problemas encontrados	<ul style="list-style-type: none"> • Este banco de dados é muito útil quando não há necessidade de relacionamento entre os dados, o que não acontece no sistema do CNCFlora pois os dados são dependentes de relacionamento para a geração de relatórios. • Para recuperar e atualizar um registro é necessário empregar linhas de código, dificultando muito o trabalho do desenvolvedor.
Solução proposta	Alterar o banco de dados para PostgreSQL, remodelar as tabelas com utilização de chaves estrangeiras.

Tabela 10. Descrição da tecnologia do banco CouchDB.

CouchDB	
Descrição	Banco de dados não relacional
Verificação de versão:	http://jb048:5984/_utils/ Versão utilizada: 1.6.1 Nota: A última versão é 2.3.0
Verificando os nodes existentes:	http://jb048:5984/_utils/
Verificando os índices:	http://jb048:5984/_utils/
Realizando uma busca com todos os registros e exibindo em json:	http://jb048:5984/_utils/

Realizando uma busca de um registro	Foi criado um arquivo chamado couch.php jb048 pasta raiz do andreppinghaus.
Problemas encontrados	<ul style="list-style-type: none"> • Este banco de dados é muito útil quando não há necessidade de relacionamento entre os dados, o que não acontece no sistema do CNCFlora pois os dados são dependentes de relacionamento para a geração de relatórios. • Para recuperar e atualizar um registro é necessário empregar linhas de código, dificultando muito o trabalho do desenvolvedor. • Todos os dados são replicados no couchdb e elastic.
Solução proposta	Alterar o banco de dados para PostgreSQL, remodelar as tabelas com utilização de chaves estrangeiras.