



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**APOIO AO DESENVOLVIMENTO DE BASE DE DADOS  
INTEGRADOS AO PORTAL WEB DO CENTRO DE CIÊNCIA  
DO SISTEMA TERRESTRE - CCST DO INPE**

Lucas Martins Oliveira

Relatório Final da Bolsa de Iniciação em  
Desenvolvimento Tecnológico e  
Inovação (PIBITI) no INPE, projeto  
orientado por Dra. Viviane Regina  
Algarve, CCST.

**INPE**  
**Cachoeira Paulista**  
**2020**



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**APOIO AO DESENVOLVIMENTO DE BASE DE DADOS  
INTEGRADOS AO PORTAL WEB DO CENTRO DE CIÊNCIA  
DO SISTEMA TERRESTRE - CCST DO INPE**

Lucas Martins Oliveira

Relatório Final da Bolsa de Iniciação em  
Desenvolvimento Tecnológico e  
Inovação (PIBITI) no INPE, projeto  
orientado por Dra. Viviane Regina  
Algarve, CCST.

**INPE**  
**Cachoeira Paulista**  
**2020**

## **RESUMO**

Com início em agosto de 2019, o projeto teve como objetivo o apoio, atualização das ferramentas e desenvolvimento de uma base de dados integrados para o portal Web do Centro de Ciências do Sistema Terrestre (CCST). Centro este que disponibiliza para sociedade diversos resultados de várias pesquisas relacionando assuntos associados a mudanças climáticas globais e outros tópicos ligados ao meio ambiente e o desenvolvimento de programas de sustentabilidade. Em razão de sua importância seus produtos e serviços tem sido cada vez mais requisitados, surgindo assim a necessidade de disponibilização das informações em uma base de dados de fácil acesso e a praticidade no acesso dos produtos e serviços disponibilizados pelo CCST. Sendo assim, se fez necessária toda a reestruturação do portal web da Pós Graduação em Ciência do Sistema Terrestre – PGCST, de forma que respeitasse e adequasse aos manuais disponibilizados pela Secretaria Especial de Comunicação Social do Governo Federal (SECOM). Além da reestruturação do banco de dados para possibilitar pesquisas e desenvolvimento de ferramentas de fácil acesso, em paralelo foram realizados monitoramentos periódicos ao Google Analytics, afim de construir gráficos e indicadores. Para que se atingisse os objetivos propostos pelo projeto, foi necessário a busca pelo aperfeiçoamento e alguns upgrades das tecnologias/ferramentas, dentre elas: Bootstrap, CSS, HTML, JavaScript, MySQL, PHP, Google Analytics e WordPress. Permitindo assim a atualização das ferramentas disponibilizadas pelo portal, integração de novas bases de dados, acompanhamento e apoio na geração de índices de atendimento dos usuários internos e externos do CCST.

Palavras-chave: Web. CCST. Reestruturação. PGCST. WordPress. Atualização.

## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> .....	1
1.1 OBJETIVOS .....	1
<b>2 DADOS E METODOLOGIA</b> .....	2
2.1 ESTUDO E PESQUISA SOBRE MUDANÇAS CLIMÁTICAS GLOBAIS, MEIO AMBIENTE, SUSTENTABILIDADE, PROJEÇÕES CLIMÁTICAS, USO E COBERTURA DA TERRA, ENERGIAS LIMPAS E OUTROS .....	2
2.2 ESTUDO DA LINGUAGEM CIENTÍFICA PARA APLICAÇÃO NO SITE WEB DA PGCST COMO APRENDIZAGEM NO WORDPRESS E AS ÁREAS DE MAIOR INTERESSE EM TI .....	2
2.2.1 CSS .....	3
2.2.2 Bootstrap.....	3
2.2.3 HTML5 .....	3
2.2.4 Javascript .....	3
2.2.5 Servidor Apache .....	4
2.2.6 MAMP .....	4
2.2.7 PHP .....	4
2.2.8 MySQL .....	4
2.2.9 Wordpress.....	5
2.2.10 Visual Studio Code.....	5
2.3 ACOMPANHAMENTO DE ATENDIMENTO AOS USUÁRIOS DO CCST .....	5
2.3.1 Google Analytics .....	5
2.4 MATERIAIS E MÉTODOS .....	6
<b>3 RESULTADOS</b> .....	7
3.1 PERÍODO DE APRENDIZAGEM .....	7
3.2 PÁGINA RESTRUTURADA .....	7
3.3 PÁGINA DA PÓS-GRADUAÇÃO DO CCST .....	8
3.3.1 Header e Navbar .....	10
3.3.2 Body e Sidebar .....	10
3.3.3 Footer.....	12
3.4 ACOMPANHAMENTO E ANÁLISES .....	12
<b>4 CONCLUSÕES</b> .....	16
4.1 SUGESTÕES PARA TRABALHOS FUTUROS .....	16
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	17
<b>APÊNDICE A - GEOSERVER</b> .....	19

## LISTA DE FIGURAS

	<u>Pág.</u>
Figura 3.1 – Tema Básico conforme o tutorial – página Sobre .....	7
Figura 3.2 – Página da PGCST antes da reestruturação – página Home .....	8
Figura 3.3 – Página da PGCST depois da reestruturação – página Home .....	9
Figura 3.4 – Header da página da PGCST antes da reestruturação .....	10
Figura 3.5 – Header da página da PGCST depois da reestruturação .....	10
Figura 3.6 – Body da página da PGCST antes da reestruturação .....	11
Figura 3.7 – Body da página da PGCST depois da reestruturação .....	11
Figura 3.8 – Footer da Página da PGCST antes da reestruturação .....	12
Figura 3.9 – Footer da Página da PGCST depois da reestruturação .....	12
Figura 3.10 – Dashboard PGCST – Referente a todo o período do projeto .....	13
Figura 3.11 – Fragmento Dashboard PGCST – Duração média da sessão .....	13
Figura 3.12 – Fragmento Dashboard PGCST – Gráfico de número de usuários e visualizações .....	14
Figura 3.13 – Dashboard PGCST – Referente ao período de 01/07/2020 a 07/07/2020 .....	14
Figura A.1.3.1 – Mapa de Fluxo do módulo GeoServer .....	22
Figura A.1.4.1 – GeoServer .....	31

# **1 INTRODUÇÃO**

Criado em 2008, o Centro de Ciências do Sistema Terrestre (CCST), possui como principal missão, fornecer informações técnico científicas, com foco no desenvolvimento nacional e para redução dos impactos ambientais no Brasil e no mundo, afim de orientar sobre às mudanças ambientais globais, de acordo com o site do CCST.

Ainda conforme o site do CCST, o centro possui como metas a modernização/atualização de seus sistemas, mantendo o foco na pesquisa e a interação entre as várias disciplinas e setores, tendo em vista solução de problemas decorrentes das mudanças ambientais globais. (CCST, 2015).

Visando também o apoio ao setor de atendimento ao usuário do CCST, foi de extrema importância que este projeto tenha se desenvolvido de forma conjunta e integrada com os pesquisadores do Centro. Tendo em vista que os mesmos estão inseridos aos objetivos estratégicos de desenvolvimento e aprimoramento de modelos do sistema terrestre, construindo e analisando cenários afim de disponibiliza-los para comunidade.

Dentro deste contexto a atualização e integração de novas bases de dados às ferramentas do portal web do CCST, bem como reestruturação da página da Pós-graduação do CCST, adequando o layout da página, seguindo sempre as diretrizes propostas pela SECOM, permitiram uma melhor integração das novas bases de dados e experiência dos usuários.

## **1.1 Objetivos**

O objetivo deste projeto consistiu no desenvolvimento de ferramentas web para análises e consulta a comunidade científica do Centro e o público externo. Sempre correspondendo as diretrizes da Secretaria Especial de Comunicação Social do Governo Federal (SECOM), sendo necessário a atualização e manutenção constantes. A construção e atualização destas ferramentas, favoreceu a cultura organizacional e implica diretamente no crescimento e valorização da imagem do CCST. Estratégias de comunicação eficazes e acompanhamentos constantes, que atendam as características singulares de cada setor, é requisito imprescindível para que públicos interno e externo conheçam e participem desta cultura.

## **2 DADOS E METODOLOGIA**

Nos tópicos seguintes serão descritas todas as atividades, que foram realizadas em todo o andamento deste projeto seguindo os parâmetros expressos no plano de trabalho.

### **2.1 Estudo e pesquisa sobre mudanças climáticas globais, meio ambiente, sustentabilidade, projeções climáticas, uso e cobertura da terra, energias limpas e outros**

Como parte importante do projeto, diversas pesquisas foram realizadas com o intuito de familiarização com os aspectos que permeiam os trabalhos realizados pelo CCST. Elas incluem a missão, visão e objetivos do centro, suas linhas de pesquisas e projetos, seu histórico, conceitos de Ciência do Sistema Terrestre, mudanças ambientais globais, mudanças climáticas, sustentabilidade e etc. Temas de pesquisas esses que foram recomendadas pelos orientadores, assim como as referências também.

### **2.2 Estudo da linguagem científica para aplicação no site web da PGCST como aprendizagem no Wordpress e as áreas de maior interesse em TI**

A necessidade da atualização das ferramentas propostas pelo portal web PGCST, proporcionou a utilização de conceitos de engenharia de software, dentre eles a Metodologia do Desenvolvimento Ágil ou Métodos Ágeis, por meio de suas características de produção de softwares descritas no blog DevMedia, trouxe diversas vantagens do sistema antes de ele estar pronto de fato, auxiliando também na organização do trabalho e viabilizar a conclusão do sistema em um período de tempo reduzido. (DEV MEDIA, 2007)

Toda a atualização do portal da PGCST foi proposta por meio de reuniões esporádicas junto aos representantes do centro, o qual foi realizada uma análise criteriosa do portal, afim de encontrar possíveis falhas, melhor layout dos conteúdos, usabilidade, responsividade e um visual que fosse agradável, sempre respeitando as normas propostas pela Secretaria Especial de Comunicação Social (SECOM).

Visando também o público que acessa o portal por meio de dispositivos mobile, foram apontadas modificações no layout do portal, para assim garantir um visual atrativo e leve no qual a responsividade fosse empregada, bem como a organização do conteúdo de pesquisas e trabalhos realizados dispostos na página.

Posteriormente a etapa de levantamento de requisitos, iniciou-se a etapa de desenvolvimento. A reunião de todas as linguagens de programação e softwares adotados para o desenvolvimento do sistema, foi criteriosa. Deve-se levar em conta que o CCST já se utiliza de grande parte delas em suas aplicações, porém em versões que carecem de atualizações imediatas.

### **2.2.1 CSS**

O CSS é responsável pelo aspecto da página, do layout e da leitura da página web pelo navegador. Não é uma linguagem de programação ou de marcação, é uma sequência de propriedades com valores definidos para manipular o comportamento e aparência dos elementos da página, podendo estes ser herdados por outros elementos (GOMES, 2010, p. 11). No contexto deste projeto, o CSS foi utilizado na estilização do layout das páginas do portal.

### **2.2.2 Bootstrap**

De acordo com o blog HomeHost, Bootstrap é um framework Front End utilizado por milhares de desenvolvedores web pelo mundo. Ele permite que diversas etapas do desenvolvimento web se tornem mais rápidas e dinâmicas, pois já traz consigo diversos elementos prontos e estilizados. (HOMEHOST, 2014?). Essa ferramenta possibilitou a criação das páginas organizadas em linhas e colunas num sistema baseado em 12 colunas para cada linha tornando o projeto completamente responsivo.

### **2.2.3 HTML5**

Segundo o blog Hostinger, HTML significa Hypertext Markup Language. Ele permite que os usuários criem e estruturam seções, parágrafos, cabeçalhos e links para páginas da internet ou aplicações. (HOSTINGER, 2019).

Utilizado em sua versão 5, para criação das estruturas dos sistemas construídos durante o projeto.

### **2.2.4 Javascript**

Conforme o blog DevMedia, o JavaScript é uma linguagem de script interpretada pelos navegadores e encarregado atualmente de uma ampla gama de responsabilidades, que

vão desde requisições assíncronas para obtenção de dados a alterações dinâmicas de CSS. (DEVMEDIA, 2016).

### **2.2.5 Servidor Apache**

Segundo as informações do blog CanalTech, o servidor Apache ou Servidor HTTP Apache é o mais bem-sucedido servidor web livre que existe. Trata-se de um servidor web muito popular, utilizado principalmente no Linux. (CANALTECH, 2015?).

Foi utilizado durante todo o projeto, pois oferece um repositório local para a realização de testes e desenvolvimento.

### **2.2.6 MAMP**

De acordo com o blog Mirago, a plataforma MAMP foi inicialmente criada para atender os computadores da Apple provendo um pequeno servidor web e uma infraestrutura mínima para criação de websites ou blogs. Sua finalidade no projeto, foi apoiar no desenvolvimento e testes. (MIRAGO, 2016?).

### **2.2.7 PHP**

Segundo o PHP Manual (2013), PHP é uma linguagem de programação de uso livre, estando sob a licença Open Source License certificada pela Open Source Initiative e possui uma curva de desenvolvimento muito rápida. É utilizada por mais de 20 milhões de domínios da internet, e está presente em mais de 50% das instalações do servidor web Apache.

O PHP Manual (2013) informa que a linguagem possui instruções nativas para interação com o banco de dados MySQL em maior quantidade que as disponíveis para outros sistemas gerenciadores de banco de dados (SGBD).

Por ser a linguagem base do Wordpress, tecnologia utilizada para a criação dos sites, foi adotada na criação dos sistemas desenvolvidos durante o projeto.

### **2.2.8 MySQL**

Conforme as informações do blog HomeHost, o MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), gratuito e livre. Criado inicialmente em

1995, foi sofrendo evoluções com o tempo, e atualmente é a plataforma mais utilizada no mundo. Trata-se de um serviço estável, seguro e confiável. (HOMEHOST, 2016?).

O WordPress, Magento, Woocommerce e Prestashop são exemplos de aplicações que fazem uso do MySQL.

Utilizado no desenvolvimento e manipulação das bases de dados referentes ao projeto.

### **2.2.9 Wordpress**

De acordo com o blog RockContent, o Wordpress vem sendo um dos mais importantes gerenciadores de conteúdo no mundo. O Wordpress foi utilizado para o desenvolvimento do projeto. Já previamente adotado como ferramenta de desenvolvimento de sites do CCST, sua função é facilitar o desenvolvimento de sites. (ROCKCONTENT, 2019).

### **2.2.10 Visual Studio Code**

Segundo o blog DevMedia, o Visual Studio Code Trata-se de uma ferramenta, um editor de código fonte leve e multiplataforma que está disponível tanto para Windows, quanto para Mac OS e Linux e atende a uma gama enorme de aplicações inclusive a proposta por este projeto. (DEVMEDIA, 2016).

## **2.3 Acompanhamento de atendimento aos usuários do CCST**

Todo o acompanhamento dos usuários do portal foi feito pela ferramenta Google Analytics. Por meio de gráficos e tabelas, assim como a possibilidade de comparar os dados, oferecidos pela ferramenta, nos permitiu o controle e até a geração de relatórios mensais quando solicitados.

### **2.3.1 Google Analytics**

Conforme o blog MetricasBoss, o Google Analytics é um software free de Web Analytics que permite analisar o comportamento de usuários em seu site, seja ele um blog, um e-commerce ou um site institucional. Através desta ferramenta, foi possível o acompanhamento detalhado dos usuários do portal do CCST e emitir relatórios posteriormente. (METRICASBOSS, 2019).

## **2.4 MATERIAIS E MÉTODOS**

A grande importância depositada na cultura organizacional, resulta diretamente no crescimento e valorização da imagem do centro para o público interno e externo. Estratégias de comunicação eficazes, que atendam as características singulares de cada setor, é requisito imprescindível para que públicos interno e externo conheçam e participem desta cultura. Desta forma, a seguinte metodologia foi utilizada para apoio computacional e utilização de alguns softwares de web, como WordPress, Google Analytics e, também, o pacote Microsoft Office (Word, Excel e PowerPoint).

1. Estudo e pesquisa das mudanças climáticas globais, meio ambiente, sustentabilidade, projeções climáticas, uso e cobertura da terra, energias limpas e outros;
2. Estudo da linguagem científica para aplicação no site web do PGCST como aprendizagem no WordPress;
3. Pesquisa e armazenamento de informações vinculadas ao site web da PGCST; como Google Analytics;
4. Tabulação e geração de gráficos mensais de atendimento aos usuários do PGCST;
5. Planilhas para o apoio da geração dos índices para o Repositórios de indicadores de gestão do CCST e do INPE;
6. Pesquisas nas áreas de maior interesse em TI;
7. Análises de resultados mensais;
8. Desenvolvimento de ferramentas para a avaliação do grau de entendimento obtido pelo público geral que deverá utilizar as informações.

### 3 RESULTADOS

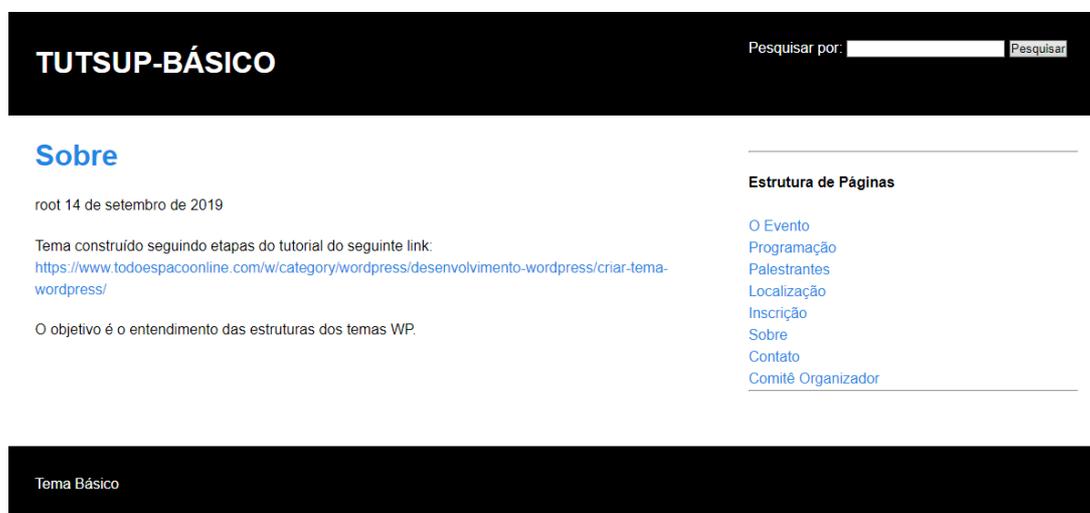
#### 3.1 Período de Aprendizagem

Antes de iniciar o desenvolvimento dos tópicos que este projeto aborda, houve a etapa de aprendizagem, cujo objetivo principal era a familiarização com a ferramenta WordPress. Para tanto foi proposto a criação de um tema básico na ferramenta, seguindo tutorial apontado pelo blog Todo Espaço Online.

**URL:**<https://www.todoespacoonline.com/w/2014/09/arquivos-de-um-tema-wordpress/>

Ao término do tutorial o resultado representado na figura 3.1 foi o obtido:

Figura 3.1 – Tema Básico conforme o tutorial – página Sobre



Fonte: O Autor

Ao realizar o tutorial foi possível compreender melhor como funciona a estrutura e organização de arquivos no Wordpress.

#### 3.2 Página Restruturada

Como um dos objetivos principais do projeto é a atualização e integração de novas bases de dados as ferramentas do portal web do CCST, a reestruturação da página da Pós-graduação do CCST se fez necessária. A adequação do layout da página, seguindo sempre as diretrizes propostas pela SECOM, permitiu uma melhor integração das novas bases de dados.

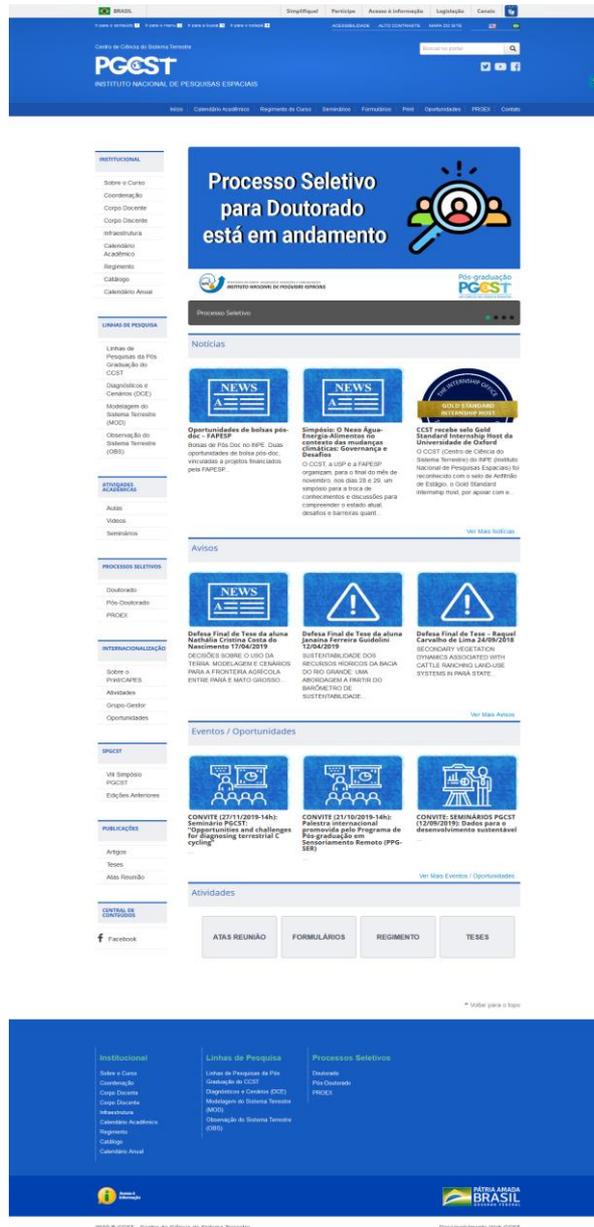
A seguir será apresentado o layout da página antes e depois da reestruturação, destacando as principais modificações e distribuição do conteúdo da página.

### 3.3 Página da Pós-Graduação do CCST

A decisão da reestruturação da página da Pós-graduação do CCST é fruto de diversas análises realizadas sobre a mesma, e foram feitas pela equipe web do CCST juntamente com a coordenação. A figura 3.2 a seguir, demonstra a página do PGCST antes da reestruturação.

URL: <http://pg.ccst.inpe.br>

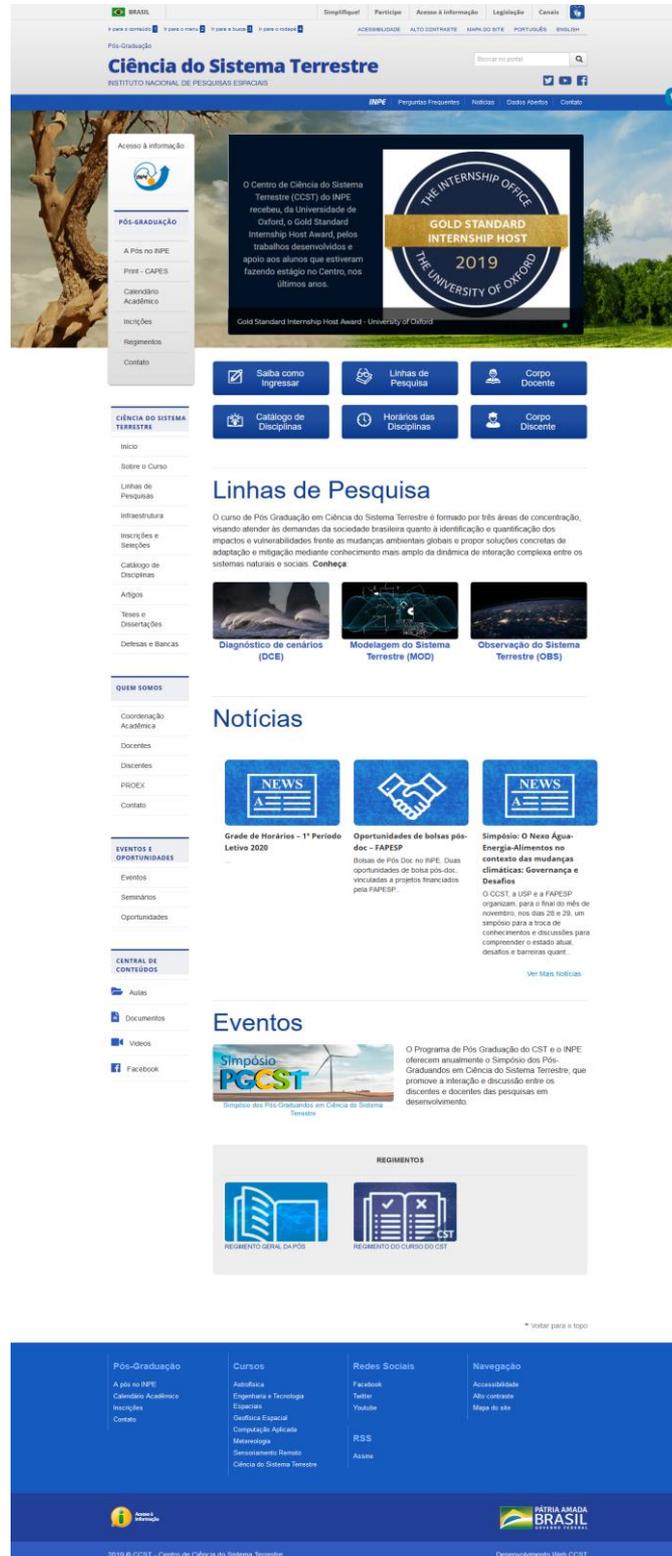
Figura 3.2 – Página da PGCST antes da reestruturação – página Home



Fonte: PGCST (2019) – Página Home

A figura 3.3 a seguir, demonstra a página do PGCST depois da reestruturação.

Figura 3.3 – Página da PGCST depois da reestruturação – página Home

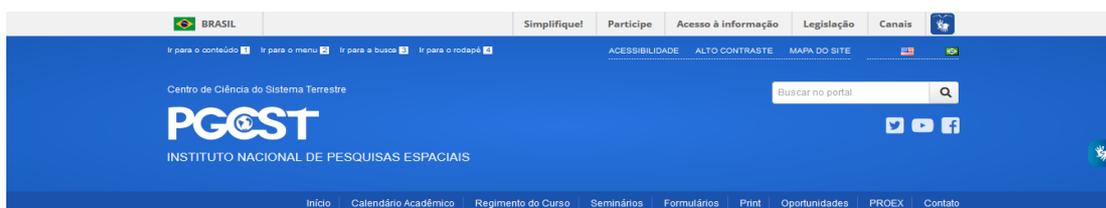


Fonte: PGCST (2020) – Página Home

### 3.3.1 Header e Navbar

As principais mudanças no *header* representadas pelas figuras 3.4 e 3.5 (cabeçalho da página) são estéticas, como a troca do logo PGCST por Pós-Graduação Ciência do Sistema Terrestre, a cor e a retirada dos ícones do plug-in de tradução.

Figura 3.4 – Header da página da PGCST antes da reestruturação



Fonte: PGCST (2019)

Figura 3.5 – Header da página da PGCST depois da reestruturação



Fonte: PGCST (2020)

As demais modificações se atentaram a organização do conteúdo da *navbar* (barra de navegação da página), simplificando-os, deixando apenas os mais importantes para acesso.

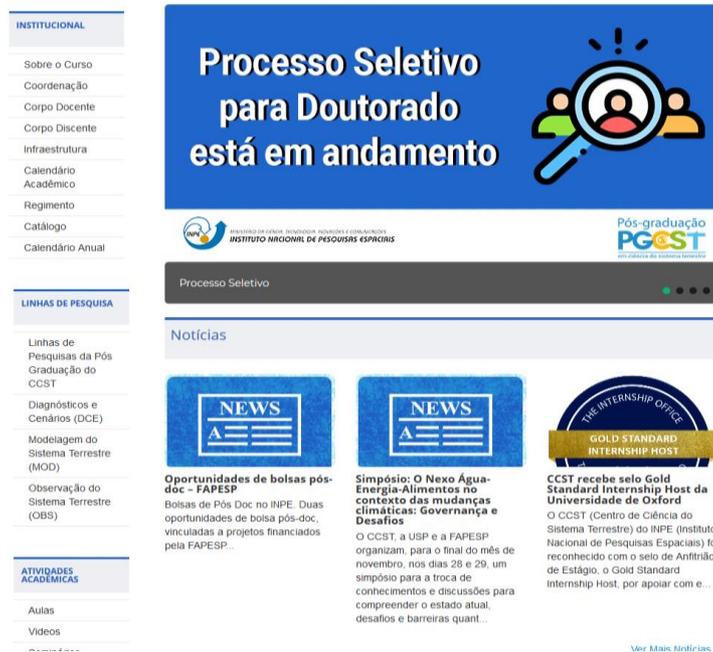
### 3.3.2 Body e Sidebar

O *body* (corpo de uma página) e a *sidebar* (barra lateral da página) são onde se concentram a maior parte do conteúdo, portanto, exigiu uma maior atenção da equipe. Além das modificações na estética da página, como imagem de fundo, cor e ícones, foi organizado todo conteúdo disposto em *sidebar*, onde parte foi redirecionado para botões logo abaixo do *slider* (carrosséis de imagens), diminuindo o tamanho da *sidebar* e assumindo uma estrutura mais organizada.

Outra importante modificação que se faz importante ser ressaltada, foi o destaque dado às Linhas de Pesquisa. Redirecionando da *sidebar*, para o *body* da página, ganhando

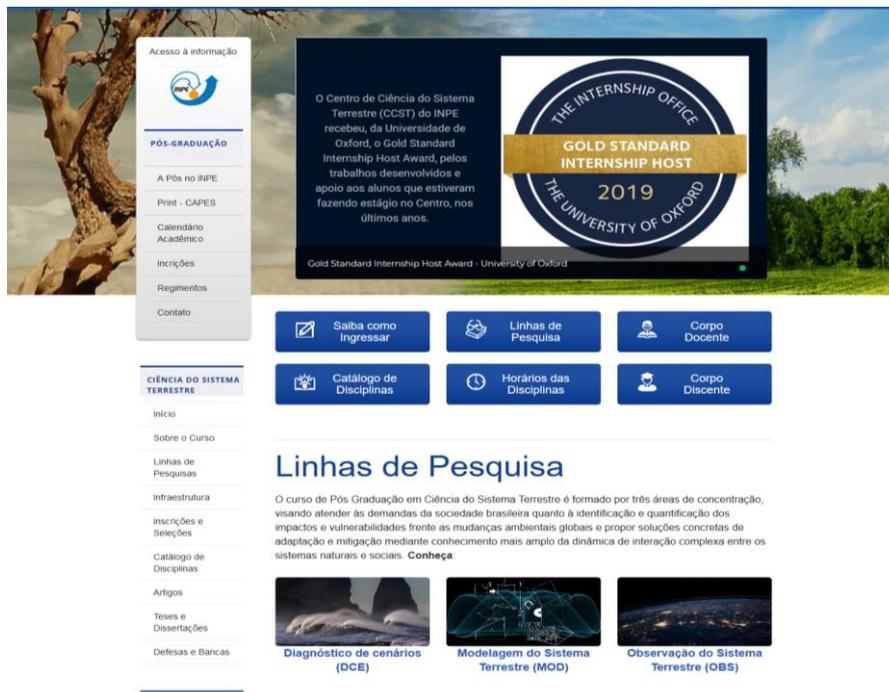
assim mais visibilidade. As figuras 3.6 e 3.7 mostram o antes e o depois da página respectivamente.

Figura 3.6 – Body da página da PGCST antes da reestruturação



Fonte: PGCST (2019)

Figura 3.7 – Body da página da PGCST depois da reestruturação

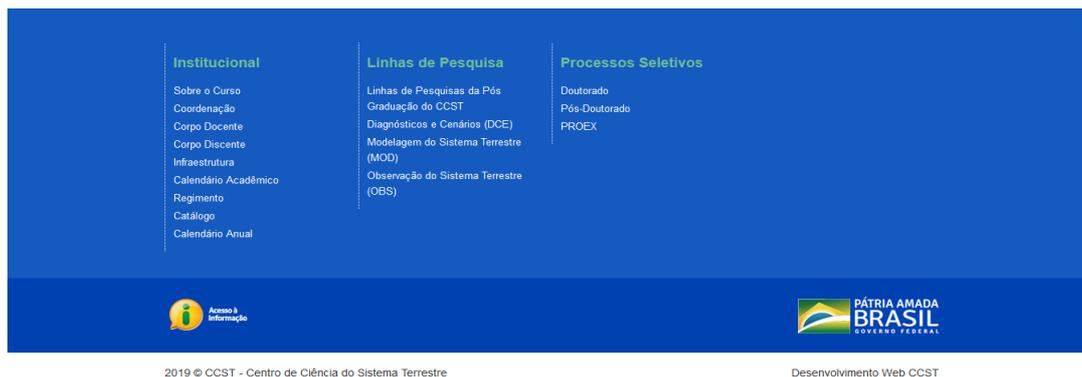


Fonte: PGCST (2020)

### 3.3.3 Footer

As mudanças no *footer* retratadas pela figura 3.8 e 3.9 (rodapé da página) se concentraram na organização das listas de conteúdos presentes nele. Elencando informações importantes sobre a Pós-Graduação, os Cursos dos quais a PGCST dispõe, redes sociais e conteúdo de auxílio a navegação.

Figura 3.8 – Footer da Página da PGCST antes da reestruturação



Fonte: PGCST (2019)

Figura 3.9 – Footer da Página da PGCST depois da reestruturação



Fonte: PGCST (2020)

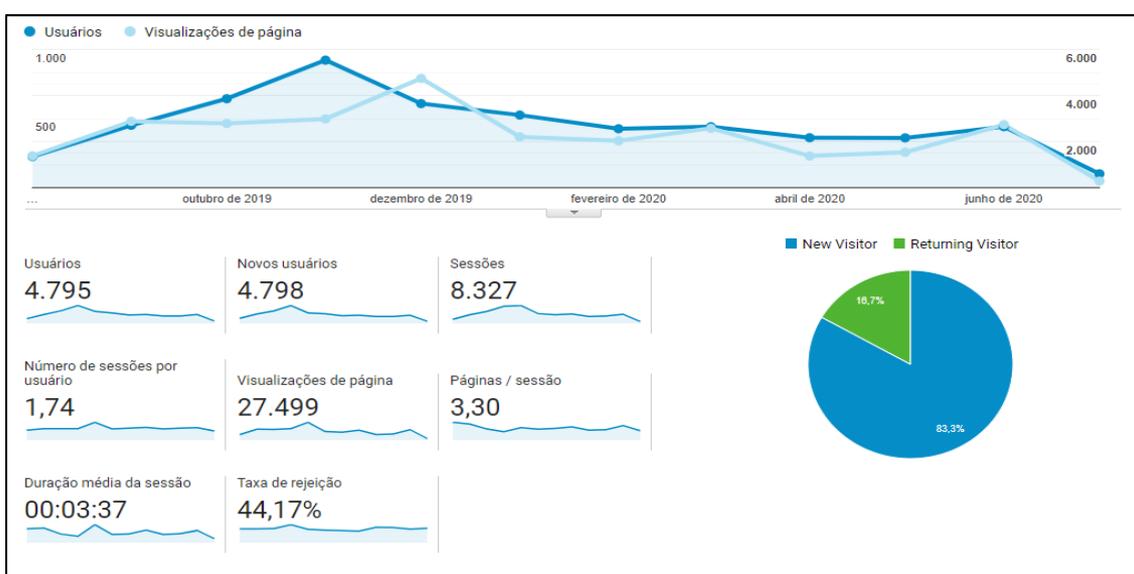
### 3.4 Acompanhamento e análises

Com a reestruturação e novo layout da página da Pós-Graduação do CCST concluída, etapas como acompanhamento dos números de acessos e taxas de rejeição foram avaliados e ajustados. Sendo assim, utilizando-se a ferramenta Google Analytics, os dados gerados após a disponibilidade de acesso da página foram coletados e manipulados para um melhor entendimento de sua aceitação perante aos usuários e a forma como é ocorrida o acesso e navegação do portal. Esses novos dados foram

devidamente analisados com as opções que o Google Analytics oferece, só assim conseguimos traçar a melhor estratégia para melhorar e aprimorar cada vez mais a estrutura da página, sempre pensando na experiência do usuário, resultando no aumento no número de acessos.

A figura 10, a seguir mostra uma dashboard contendo os gráficos criados a partir das informações coletadas pelo Google Analytics referente ao período de vigência deste projeto. Dentro deste período está englobado as etapas de análises, prototipação, planejamento e estruturação da página da PGCST.

Figura 3.10 – Dashboard PGCST – Referente a todo o período do projeto



Fonte: Google Analytics (2020)

Com relação aos números mostrados na figura acima, é possível realizar as seguintes análises:

Primeiro, com a reestruturação da página da PGCST, os conteúdos foram reorganizados e distribuídos pela página de forma com que as informações apresentadas fossem mais claras e facilmente identificadas, impactando diretamente na queda do tempo de exibição de cada sessão. Uma vez que o usuário conseguiu encontrar mais facilmente o que procura. O fragmento a seguir contempla o gráfico dos dados analisados em questão.

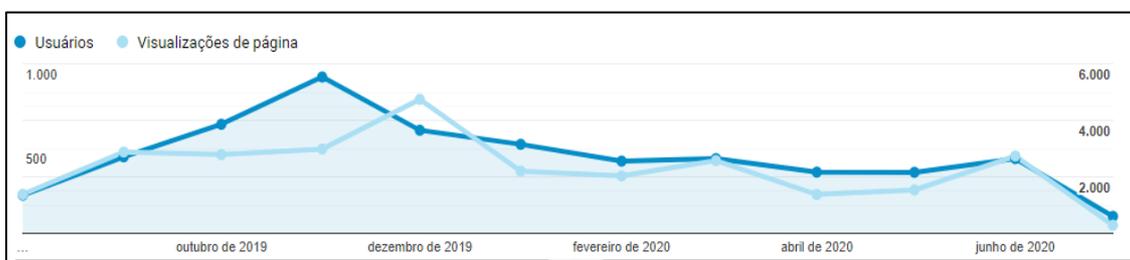
Figura 3.11 – Fragmento Dashboard PGCST – Duração média da sessão



**Fonte:** Google Analytics (2020)

Segundo, a queda no número de usuários está ligada aos calendários dos eventos referentes à página da PGCST, eventos como Simpósios, atraem uma gama de novos usuários à página, explicando assim a crescente no gráfico e a baixa logo depois do período. O fragmento a seguir contempla o gráfico de onde foram extraídas a análise citada anteriormente.

Figura 3.12 – Fragmento Dashboard PGCST – Gráfico de número de usuários e visualizações

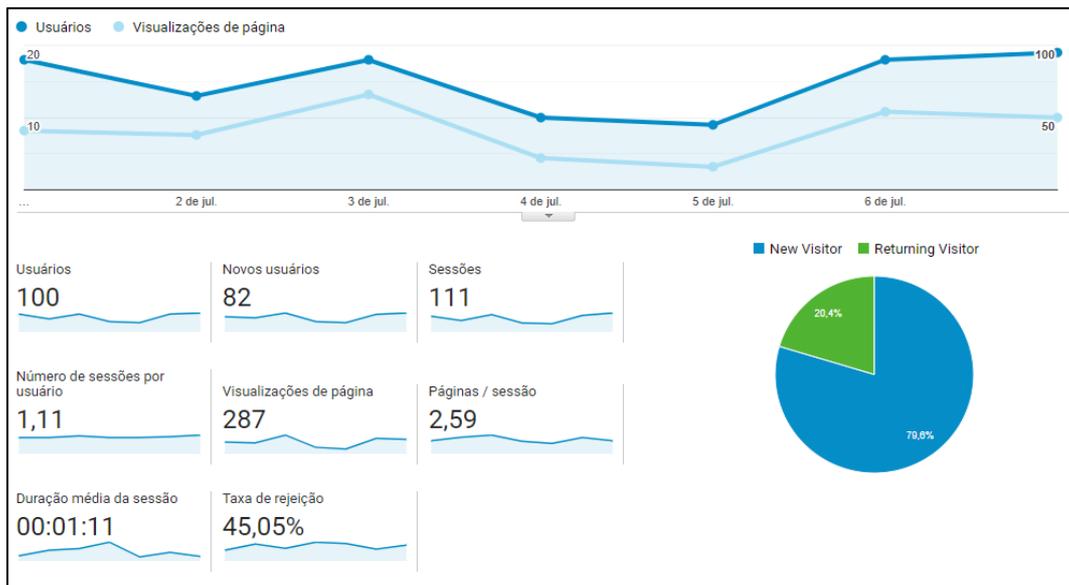


**Fonte:** Google Analytics (2020)

Por fim, a figura 11 mostra uma nova dashboard, agora contendo gráficos dos dados referentes aos últimos dias com relação a presente data.

Os dados presentes nesta dashboard só reafirmam os aspectos citados nas análises anteriores, uma vez que o tempo de exibição continua baixo e o número de acessos se manteve em uma média de crescimento.

Figura 3.13 – Dashboard PGCST – Referente ao período de 01/07/2020 a 07/07/2020



Fonte: Google Analytics (2020)

## **4 CONCLUSÕES**

O projeto desenvolvido, trouxe consigo uma série de vantagens ao centro, possibilitando a realização mais ampla de análises e consultas por parte da comunidade científica do Centro de Ciência do Sistema Terrestre e o público externo. Para tanto, mudanças e adaptações de conteúdo são inevitáveis ou adequações às futuras diretrizes impostas pela SECOM, fazendo com que sejam necessárias atualizações e manutenções constantes. Reafirmando a necessidade da continuidade do projeto.

A reestruturação do site da PGCST trouxe modificações que facilitaram a navegação dentro site, ficou mais organizado e inteligível, melhorando a experiência de seus usuários, principalmente ao acesso às informações que o portal disponibiliza.

Portanto, dentro do escopo do projeto e partindo das pesquisas e interações quanto aos conceitos que envolvem o CCST, o aprendizado das ferramentas e linguagens de programação importantes na criação das aplicações, chegando aos testes e acompanhamentos realizados pós desenvolvimento.

### **4.1 Sugestões para Trabalhos Futuros**

Como proposta para uma continuação do projeto, o Google Analytics permitirá o acompanhamento do número de acessos, localização de origem dos acessos, perfil do usuário e etc. sobre esses dados, se necessário, algumas mudanças ou adaptações serão realizadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

Como instalar o MAMP no mac. **Blog Mirago**. Disponível em: <<https://www.mirago.com.br/aula/instalar-mamp-mac/>>. Acesso em 01/02/2020.

GOMES, A. L. XHTML/CSS, Criação de Páginas Web. São Paulo, SP - Brasil: Senac, 2010. 195p.

Introdução ao Desenvolvimento Ágil. **Blog DevMedia**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-desenvolvimento-agil/5916>>. Acesso em 07/02/2020.

Introdução ao Visual Studio Code. **Blog DevMedia**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418>>. Acesso em 01/02/2020.

JavaScript tutorial. **Blog DevMedia**. Disponível em: <<https://www.devmedia.com.br/javascript-tutorial/37257>>. Acesso em 01/02/2020.

Missão, Visão e Objetivos. Site **CCST-INPE**. Disponível em: <<http://www.ccst.inpe.br/ccst-institucional/missao-visao-e-objetivos/>>. Acesso em 10/04/2020.

O que é Bootstrap: Tudo sobre este Framework. **Blog Homehost**. Disponível em: <<https://www.homehost.com.br/blog/tutoriais/o-que-e-bootstrap/>>. Acesso em 01/02/2020.

O que é Google Analytics?. **Blog MetricsBoss**. Disponível em: <<https://metricsboss.com.br/o-que-e-google-analytics.html>>. Acesso em 01/02/2020.

O Que é HTML? Guia Básico Para Iniciantes. **Blog Hostinger**. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos/>>. Acesso em 01/02/2020.

O que é MySQL?. **Blog Homehost**. Disponível em: <<https://www.homehost.com.br/blog/tutoriais/mysql/o-que-e-mysql/>>. Acesso em 01/02/2020.

O que é servidor Apache?. **Blog CanalTech**. Disponível em: <<https://canaltech.com.br/internet/o-que-e-servidor-apache/>>. Acesso em 01/02/2020.

PHP Manual. **Site PHP**. Disponível em: <[https://www.php.net/manual/pt\\_BR/preface.php](https://www.php.net/manual/pt_BR/preface.php)>. Acesso em 01/02/2020.

Sistema de Gestão de Conteúdos (CMS): por que implementar na sua empresa?. **Blog RockContent**. Disponível em: <<https://rockcontent.com/blog/cms/>>. Acesso em 01/02/2020.

## **APÊNDICE A - GEOSERVER**

Foi proposto como atividade complementar durante este projeto, a criação de *scripts* na Linguagem de Programação Lua para a configuração do repositório de Geo-Serviços GeoServer. Para tal, foi selecionada a biblioteca Geoserver-Shell, pois ela simplifica a adaptação da Interface de linha de comandos (CLI) para utilização nos *scripts* Lua, funcionando assim como uma interface de conexão da linguagem Lua com o GeoServer.

O objetivo desta atividade foi de conhecer a linguagem Lua e outras ferramentas que envolvem Geo-Serviços, como: GeoServer, QGis, Leaflet, Publish e TerraME.

A seguir é apresentado toda a documentação do processo de criação do módulo Lua/GeoServer.

### **A.1 Sobre GeoServer**

O GeoServer é um Software livre, mantido pelo Open Planning Project (mantenedor principal), que permite o desenvolvimento de soluções de Webmapping, integrando diversos repositórios de dados geográficos com simplicidade e alta performance. O GeoServer é um servidor de Web Map Service (WMS), 'Web Coverage Service (WCS) e de Web Feature Service-Transaction (WFS-T) completamente funcional que segue as especificações da Open Geospatial Consortium (OGC).

O foco do GeoServer é facilitar o uso e suporte para os padrões abertos, a fim de permitir qualquer um de compartilhar rapidamente suas informações geoespaciais de uma maneira Inter operável.

- Servidor de geoserviços;
- WMS;
- WFS;
- Utiliza o Tomcat;
- Serve de Base de dados para geoserviços.

#### **A.1.1 Instalação do Geoserver -Windows**

- Baixe o arquivo zippado;

- Descompactar o arquivo;
- Dentro do mesmo, selecionar a pasta webapps, onde se encontra uma pasta chamada "geoserver";
- Copie e cole está pasta na pasta webapps do Tomcat;
- Inicializar no Tomcat;
- No browser digite 'localhost:8080/geoserver/web';
- Na página do GeoServer, identifique-se digitando user: 'admin' e senha: 'geoserver';
- Pronto, agora você pode subir algum shape e estiliza-lo, além de disponibilizar o geoserviço, como se fosse uma base de dados.

#### **A.1.1.1 Tomcat**

- Baixe o Tomcat caso não o tenha instalado na máquina;
- Extraia a pasta, no local de sua preferência;
- Agora através do terminal, você deve entrar na pasta 'bin';
- Antes de executar os próximos passos, você deve conferir se sua máquina está com o Java instalado, tendo o SDK quanto o JDK.
- Agora você deve executar o seguinte comando:

*' ./startup.sh ' - para dar start no servidor local do Tomcat*

*' ./shutdown.sh ' - para desligar o servidor local do Tomcat*

#### **A.1.1.2 Instalação do Geoserver - Mac**

- Baixe o Geoserver;
- Ao baixar o GeoServer e extrair, procure a pasta 'geoserver' que consta dentro da pasta 'webapps';
- Você deve copia-la para dentro do webapps do Tomcat;
- Uma vez iniciado o servidor Tomcat, basta acessa-lo no navegador por meio da URL:

' <http://localhost:8080/geoserver/web> '

- Na página do GeoServer, identifique-se digitando user: 'admin' e senha: 'geoserver'.

### **A.1.2 GEOSERVER-SHELL - Gerencie o Geoserver usando uma CLI semelhante ao Spring Roo (opção implementada)**

Administre o Geoserver usando uma interface de linha de comando (CLI). O Geoserver Shell usa a mesma interface de Shell usada pelo Spring Roo e fornece ampla conclusão de guias, suporte ao histórico e a capacidade de executar scripts. O Geoserver Shell administra o Geoserver usando a excelente API Geoserver rest. Você pode publicar shapefiles, geotiffs e postgis, fazer upload e download de SLD's e iniciar a propagação de tile diretamente da linha de comando.

### **A.1.3 Resumo do Funcionamento das Funções GeoServer/Lua**

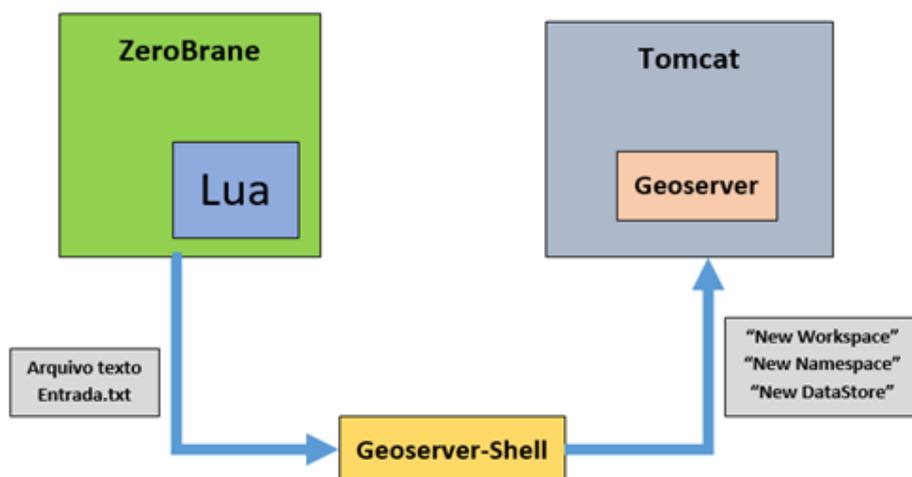
Diante da necessidade da criação de *scripts* Lua para a configuração do *GeoServer*, foi selecionada a biblioteca *Geoserver-Shell*, pois ela simplifica a adaptação da Interface de linha de comandos (CLI) para utilização nos scripts Lua, funcionando assim como uma interface de conexão da linguagem Lua com o *GeoServer*.

Basicamente as funções Lua funcionam da seguinte forma: Uma vez importadas as funções utilizando o instanciamento da classe por meio de um objeto, bastando apenas que seus métodos sejam utilizados. Os métodos podem ou não precisarem de parâmetros para que executem. A estrutura dos métodos seguem um padrão, que consiste na criação do arquivo de texto (Entrada.txt), dentro dele são escritos os comandos *Geoserver-Shell* referentes ao que o método se propõe a fazer, logo em seguida este arquivo de texto é submetido ao *Geoserver-Shell* via linha de comando, possibilitado pelo método “*os.execute()*” da linguagem Lua.

A cada comando *Geoserver-Shell* executado pelo terminal é obtida uma resposta, “*true*” se funcionou corretamente e “*false*” se falhou na execução bem como a razão desta falha.

Logo abaixo é apresentado o Mapa de Fluxo que exemplifica melhor todo o funcionamento dos scripts de forma geral.

Figura A.1.3.1 – Mapa de Fluxo do módulo GeoServer



1- O **Tomcat** é iniciado via comando **Lua** ou via terminal

2- As funções **Lua** para utilizar os comandos **Geoserver-Shell** diretamente no **ZeroBrane**, são estruturadas da seguinte forma:

2.1- As funções foram nomeadas de acordo com o comando que ela realiza;

2.2- Cada função recebe certo número de parâmetros, outras não recebem nenhum para que executem;

2.3- Ao receber os parâmetros e validar a entrada dos mesmos, eles iram compor o comando **Geoserver-Shell** que será escrito no **Arquivo texto (Entrada.txt)**;

2.4- Logo em seguida, o **Geoserver-shell** é iniciado via **Lua** ao mesmo tempo que aponta a leitura do **Entrada.txt** com os comandos;

2.5- Cada salto de linha no **Entrada.txt** é considerado a tecla “enter”, com isso os comandos são executados separadamente. Exemplo: Executar o Login, Criar a Workspace e fechar o **Geoserver-Shell**;

2.6- Para o perfeito funcionamento do **Geoserver-Shell**, é necessário que a cada função ele execute o login e ao fim feche o **Geoserver-Shell** para que outras funções possam executar;

3- Uma vez executadas no **Geoserver-Shell**, o **Geoserver** receberá as modificações.

#### A.1.4 Funções Lua/GeoServer-Shell

```
GeoserverW = { }
```

```
-- CAPTURANDO CAMINHOS WINDOWS --
```

```
w = {
```

```
-- CAMINHO GEOSERVER-SHELL DENTRO DO TOMCAT --
```

```
caminho2 = 'C:\\xampp\\tomcat\\webapps\\geoserver\\geoserver-  
shell\\target\\geoserver-shell-0.4-SNAPSHOT-app\\gs-shell-0.4-SNAPSHOT\\bin',
```

```
-- CAMINHO Entrada.txt --
```

```
caminho4='C:\\Users\\Luk\\terrame-examples\\publish'
```

```
}
```

```
login = "geoserver set --url http://localhost:8080/geoserver"
```

```
ReadCommand = w.caminho2.."\\gs-shell.bat --cmdfile "..w.caminho4.."\\Entrada.txt"
```

```
ClearCommand = "rm "..w.caminho4.."\\Entrada.txt"
```

----- Funções GeoServer - Teste Windows -----  
-----

--[WORKSPACE]--

--- Criando uma Workspace via geoserver shell

---

Esta função cria um txt com comandos de login e criação de Workspace assim como o exit, para que o GeoserverShell execute e libere o terminal

--- Esta função recebe como parâmetro o nome da Workspace.

```
function GeoserverW.createWorkspace(nomeWorkspace)
```

```
    handler = io.open("Entrada.txt", "w")
```

```
    handler:write(login.." \n workspace create --
```

```
name "..nomeWorkspace.." \n workspace list \n exit")
```

```
    handler:close()
```

```
    print("Arquivo texto criado")
```

```
    os.execute(ReadCommand)
```

```
    os.execute(ClearCommand)
```

```
end
```

--- Listar Workspaces via geoserver shell

---

Esta função cria um txt com comandos de login e listagem das Workspaces assim como o exit, para que o GeoserverShell execute e libere o terminal

--- Esta função não recebe parâmetro.

```
function GeoserverW.listWorkspace()
```

```
    handler = io.open("Entrada.txt", "w")
```

```
    handler:write(login.." \n\n workspace list \n\n exit")
```

```
    handler:close()
```

```
    os.execute(ReadCommand)
```

```

    os.execute(ClearCommand)
end

--- Deletando uma Workspace via geoserver shell
---
Esta função cria um txt com comandos de login e deletar Workspace assim como o exit
, para que o GeoserverShell execute e libere o terminal
--- Esta função recebe como parâmetro o nome da Workspace.
function GeoserverW.deleteWorkspace(nomeWorkspace)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n workspace delete --name "..nomeWorkspace.." \n\n exit")
    handler:close()

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

--- Criando um Style via geoserver shell
---
Esta função cria um txt com comandos de login e criação do Style assim como o exit, p
ara que o GeoserverShell execute e libere o terminal
---
Esta função recebe como parâmetros o nome do Style e o caminho do arquivo de confi
guração.
function GeoserverW.createStyle(nomeStyle, caminhoArquivo )
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n style create --name "..nomeStyle.." --
file "..caminhoArquivo.." \n\n style list \n\n exit")
    handler:close()
    print("Arquivo texto criado")

    os.execute(ReadCommand)

```

```

    os.execute(ClearCommand)
end

--- Listar Styles via geoserver shell
---
Esta função cria um txt com comandos de login e listagem das Styles assim como o exit, para que o GeoserverShell execute e libere o terminal
--- Esta função não recebe parâmetro.
function GeoserverW.listStyle()
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n style list \n\n exit")
    handler:close()

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

--- Deletando uma Style via geoserver shell
---
Esta função cria um txt com comandos de login e deletar Style assim como o exit, para que o GeoserverShell execute e libere o terminal
--- Esta função recebe como parâmetro o nome da Style.
function GeoserverW.deleteStyle(nomeStyle)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n style delete --name " ..nomeStyle.." \n\n exit")
    handler:close()

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

```

```

function GeoserverW.createDatastore(nomeWorkspace, nomeDatastore, conect)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n datastore create --workspace "..nomeWorkspace.." --
name "..nomeDatastore.." --connectionParams "'..conect.'" \n\n exit")
    handler:close()
    print("Arquivo texto criado")

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

```

```

function GeoserverW.listDatastore(nomeWorkspace)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n datastore list --workspace "..nomeWorkspace.." \n\n exit")
    handler:close()

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

```

```

function GeoserverW.uploadDatastore(nomeWorkspace, nomeDatastore, tipo, arquivo)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n datastore upload --workspace "..nomeWorkspace.." --
name "..nomeDatastore.." --type "..tipo.." --file "..arquivo.." \n\n exit")
    handler:close()
    print("Arquivo texto criado")

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

```

```

function GeoserverW.deleteDatastore(nomeWorkspace, nomeDatastore)
    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n datastore delete --workspace "..nomeWorkspace.." --
name "..nomeDatastore.." --recurse false \n\n exit")
    handler:close()

    os.execute(ReadCommand)
    os.execute(ClearCommand)
end

```

---Publicando um Shape via geoserver shell

---

Essa função realiza a publicação de um Shape, onde o mesmo é zippado e depois é publicado na Workspace apontada

---

Parametros: Nome da Workspace, Caminho do arquivo .SHP e caminho do zip do .SHP

```

function GeoserverW.PublishShapefile(nomeWorkspace, caminhoShapefile, caminhoZip)

```

```

    handler = io.open("Entrada.txt", "w")
    handler:write(login.." \n\n shapefile zip --
shapefile "..caminhoShapefile.." \n\n shapefile publish --
workspace "..nomeWorkspace.." --file "..caminhoZip.." \n exit")
    handler:close()
    print("Arquivo texto criado")
    os.execute(ReadCommand)
    os.execute(ClearCommand)
    print("http://localhost:8080/geoserver/"..nomeWorkspace.."/wms")
end

```

--- Publicando um Shape via geoserver shell

---

Essa função realiza a publicação de um GeoTiff, onde o mesmo é zippado e depois é publicado na Workspace, CoverageStore apontada.

---

Parametros: Nome da Workspace, Nome da Coverage Store, Nome da nova Coverage, Caminho do arquivo .Tif e caminho do zip do .Tif

```
function GeoserverW.publishGeoTiff(nomeWorkspace, nameCoverageStore, nomeCoverage, caminhoGeoTiff, caminhoZip)
```

```
    handler = io.open("Entrada.txt", "w")
```

```
    handler:write(login.." \n\n worldimage zip --
```

```
file "..caminhoGeoTiff.." \n worldimage publish --file "..caminhoZip.." --
```

```
workspace "..nomeWorkspace.." --coveragelstore "..nameCoverageStore.." --
```

```
coverage "..nomeCoverage.." \n exit")
```

```
    handler:close()
```

```
    print("Arquivo texto criado")
```

```
    os.execute(ReadCommand)
```

```
    os.execute(ClearCommand)
```

```
    print("http://localhost:8080/geoserver/"..nomeWorkspace.."/wms")
```

```
end
```

```
return GeoserverW
```

**Fonte:** O Autor

#### **A1.4.1 Script de teste de funções Lua/GeoServer-Shell**

```
local moduleW_folder = "C:\\Users\\Luk\\terrame-examples\\publish\\publish-master\\lua\\"
```

```
package.path = moduleW_folder .. ".lua;" .. package.path
```

```
geo = require "GeoserverW"
```

-- Essa função realiza a criação de um Style, a partir do upload de um arquivo SLD pré-configurado

-- Parametros : Nome do Estilo e caminho do arquivo

```
geo.createStyle("Teste", "C://xampp//tomcat//webapps//geoserver//data//styles//default_generic.sld" )
```

--

Essa função realiza a criação de uma base de dados, onde é configurada usando o tipo de BD, a Workspace e o nome da Store

--Parametros: Nome da Workspace, Nome do Store e Tipo de BD

```
geo.createDatastore("TesteWindows", "TesteDS", "dbtype=h2 database=test.db")
```

-- Essa função realiza a criação de um Workspace

-- Parametros : Nome do Workspace

```
geo.createWorkspace("NexusTeste")
```

--

Essa função realiza a publicação de um Shape, onde o mesmo é zippado e depois é publicado na Workspace apontada

--

Parametros: Nome da Workspace, Caminho do arquivo .SHP e caminho do zip do .SHP

```
caminhoSHP = "C://Users//Luk//terrame-examples//publish//publish-master//data//br_states.shp"
```

```
caminhoZIP = "C://Users//Luk//terrame-examples//publish//publish-master//data//br_states.zip"
```

```
geo.publishShapefile("NexusTeste", caminhoSHP, caminhoZIP)
```

--

Essa função realiza a publicação de um GeoTiff, onde o mesmo é zippado e depois é publicado na Workspace, CoverageStore apontada.

--

Parametros: Nome da Workspace, Nome da Coverage Store, Nome da nova Coverage, Caminho do arquivo .Tif e caminho do zip do .Tif

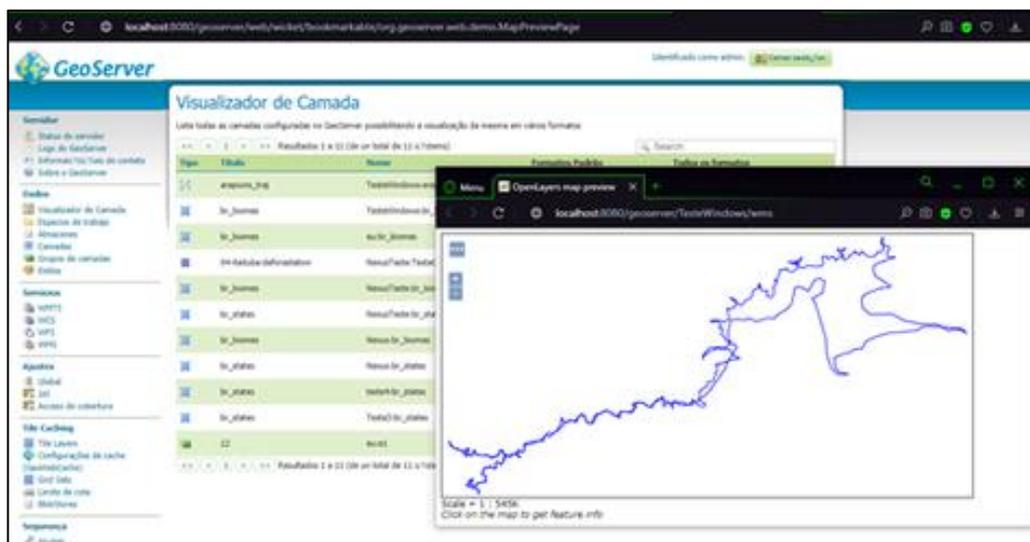
```
caminhoGeoTiff = "C://Users//Luk//terrame-examples//gis-projects//04-itaituba-deforestation.tif"
```

```
caminhoZIP = "C://Users//Luk//terrame-examples//gis-projects//04-itaituba-deforestation.zip"
```

```
geo.publishGeoTiff("NexusTeste", "TesteCovStore", "TesteCov", caminhoGeoTiff, caminhoZIP)
```

**Fonte:** O Autor

Figura A.1.4.1 – GeoServer



**Fonte:** localhost:8080/geoserver/web/

## A.1.5 Comandos Geoserver-Shell

- **Built ins**
  - exit = exit the shell
  - quit = exit the shell
  - help = list all commands and their usages
  - date = displays local date and time
  - version = display current cli version
  
- **Geoserver**
  - geoserver set --url <http://localhost:8080/geoserver> --user admin --password geoserver
  - geoserver show
  - geoserver reload
  - geoserver reset
  - geoserver backup --directory backup1 --includedata false --includegwc false --includelog true
  - geoserver restore --directory backup1
  - geoserver getmap --layers states
  - geoserver getlegend --layer states
  - geoserver getfeature --typeName topp:states
  
- **About**
  - version list
  
- **Workspace**
  - workspace list
  - workspace create --name test
  - workspace delete --name test
  
- **Namespace**

- namespace list
- namespace create --prefix test --uri [test.com](http://test.com)
- namespace delete --prefix test --recurse true

### • **Style**

- style list
- style list --workspace topp
- style create --name new\_line --file line.sld
- style delete --name new\_line

### • **Datastore**

- datastore list --workspace topp
- datastore create --workspace topp --name h2test --connectionParams "dbtype=h2 database=test.db"
- datastore delete --workspace topp --name h2test --recurse false
- datastore upload --workspace topp --name states\_convexhull --type shp --file states\_convexhull.zip

### • **Shapefile**

- shapefile zip --shapefile states\_voronoi.shp --zipfile states\_voronoi.zip
- shapefile publish --workspace topp --datastore states\_voronoi --layer states\_voronoi --file states\_voronoi.zip

### • **Postgis**

- postgis datastore create --workspace topp --datastore postgis --host localhost --port 5432 --database postgis --schema public --user user --password pass
- postgis featuretype publish --workspace topp --datastore postgis --table world\_boundaries

### • **Coverage stores**

- coverage store list --workspace nurc

- coverage store upload --workspace nurc --coveragestore test --file alki.tif --type geotiff
- coverage store delete --workspace nurc --coveragestore test --recurse true
- coverage store create --workspace nurc --name raster --type GeoTiff --url file:coverages/raster/raster.tif

#### · **Coverage**

- coverage create --workspace nurc --coveragestore worldImageSample --coverage test
- coverage delete --workspace nurc --coveragestore worldImageSample --coverage test --recurse true

#### · **Worldimage**

- worldimage zip --file NaturalEarth/MediumScale/GRAY\_50M\_SR\_OB/GRAY\_50M\_SR\_OB.tif
- worldimage publish --file GRAY\_50M\_SR\_OB/GRAY\_50M\_SR\_OB.zip --workspace naturalearth --coveragestore myworld --coverage test

### **A.1.6 Termos comuns do GeoServer/Geoserver-Shell**

#### **Workspace**

No GeoServer, um **workspace** é frequentemente usado para agrupar camadas semelhantes. As camadas podem ser referidas pelo nome do **workspace**, dois pontos, nome da camada (por exemplo topp:states). Duas camadas diferentes podem ter o mesmo nome, desde que pertençam a diferentes **workspace's** (por exemplo, sf:statese topp:states).

#### **Namespace**

Um **Namespace** é um agrupamento exclusivamente identificável de tipos de features. É identificado por um prefixo e um URI.

#### **Layer**

No GeoServer, o termo "**layer**" refere-se a um conjunto de dados **raster** ou vetorial que representa uma coleção de recursos geográficos. Camadas de vetor são análogas a "**featureTypes**" e camadas de Raster são análogas a "**coverages**". Todas as camadas têm uma fonte de dados, conhecida como **Store**. A camada está associada ao **workspace** no qual a Store está definida.

## Style

Os **Styles** são usados para controlar a aparência dos dados geoespaciais. Os estilos para o GeoServer são escritos em vários formatos diferentes:

- **SLD** (Styled Layer Descriptor): um padrão OGC para estilo geoespacial. Disponível por padrão.
- **CSS** (Cascading Style Sheets): uma sintaxe semelhante ao CSS. Disponível através de uma extensão.
- **YSLD**: um equivalente SLD baseado em YAML para criação aprimorada. Disponível através da extensão yslid.
- **MBStyle**: uma sintaxe baseada em JSON para melhor interoperabilidade. Disponível através da extensão mbstyle

## Store

Um **Store** se conecta a uma fonte de dados que contém dados de **Raster** ou **Vector**. Uma fonte de dados pode ser um arquivo ou grupo de arquivos, uma tabela em um banco de dados, um único arquivo de **Raster** ou um diretório (por exemplo, uma biblioteca de formato de produto vetorial). A construção de armazenamento permite que os parâmetros de conexão sejam definidos uma vez, em vez de para cada conjunto de dados em uma fonte. Como tal, é necessário registrar uma **Store** antes de configurar os conjuntos de dados nela.

## Coverage

Uma **Coverage** é um conjunto de dados **Raster** que se origina de um **Coverage Store**.

## Coverage Store

Um **Coverage Store** contém dados espaciais no formato **Raster**.

### **A.1.7 GeoServer API via cURL (Opção)**

O GeoServer fornece uma interface RESTful através da qual podemos configurar uma instância usando chamadas HTTP simples. Usando a interface REST, podemos configurar o GeoServer sem a necessidade de usar a interface de administração Web.

Para realizar as operações utilizaremos o cURL (uma ferramenta de linha de comando para executar solicitações HTTP e transferência de arquivos), porém elas podem ser adaptadas para qualquer ferramenta ou biblioteca HTTP.

```
curl -v -u admin:geoserver -XPOST -H "Content-type: text/xml"  
-d "<workspace><name>acme</name></workspace>"  
http://localhost:8080/geoserver/rest/workspaces
```

Modulo Geoserver API via cURL foi efetivo!

#### **Download cURL**

Link 1: <https://curl.haxx.se/windows/>

#### **cURL Install**

Link 2: <https://develop.zendesk.com/hc/en-us/articles/360001068567-Installing-and-using-cURL#install>

#### **Exemplos Geoserver API via cURL**

Link 3: <http://www.fernandoquadro.com.br/html/2016/01/15/configurando-o-geoserver-via-rest-api/>

### **A.1.8 Geoserver Rest Client PHP (Opção)**

Usando o PHP com classes cURL e FTP sobre a API REST do GeoServer para criar um serviço da Web que copia arquivos do servidor FTP remoto e configura uma camada no GeoServer com base no conjunto de GeoTiffs e em um documento no estilo SLD.

**Link:** <https://github.com/andre-carvalho/geoserver-rest-client>

### **A.1.9 Links importantes**

#### **Controlar Geoserver via comandos Shell**

Link 1: <https://github.com/jericks/geoserver-shell>

#### **Automatizar funções Geoserver**

Link 2: <http://www.fernandoquadro.com.br/html/2016/05/06/configuracao-automatica-de-estilos-e-camadas-no-geoserver-via-script-bash/>

#### **Manual lua - os.execute()**

Link 3: <http://www.lua.org/manual/5.3/manual.html#pdf-os.execute>

#### **Exemplos lua - os.execute()**

link 4:

<https://books.google.com.br/books?id=BWSTv8Q48YcC&pg=PA203&lpg=PA203&dq=os.execute+lua+exemplos&source=bl&ots=dQAhxXYIAE&sig=ACfU3U0SFPc5IbBJ-s22Lu0Rw7tTO-8OfQ&hl=pt-BR&sa=X&ved=2ahUKEwishavQjuHnAhXVFbkGHWiwAO4Q6AEwBHoECAkQAQ#v=onepage&q=os.execute%20lua%20exemplos&f=false>

#### **Testes Geoserver**

Link 5: <http://www.fernandoquadro.com.br/html/2016/09/29/foss4g-o-estado-do-geoserver/>

#### **Tutorial Geoserver-shell**

Link 6: <https://www.slideshare.net/JaredErickson/geo-servershell>