**FACULDADE DE TECNOLOGIA SENAC GOIÁS**

ALEXANDRE COSTA SANTOS

ALESSANDRO MIGUEL MANSO

DEUSAIR WELLIGNTON PEREIRA

**PROJETO INTEGRADOR**

**MODULO V**

**Gestão da Tecnologia da Informação**

**Auditoria e Qualidade de Software**

**Professor: Elias Batista**

Goiânia

2018

**Sumario**

Entendendo as ferramentas utilizadas.................................03

Github.................................................................................03

Funcionamento do GitHub..................................................03

Testes Selenium com erro....................................................05

Testes Selenium com Sucesso...............................................06

Bibliografia.........................................................................12

**Entendendo as ferramentas utilizadas**

**GitHub**

Primeiro temos que entender o que é um Git. Este é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas. Sem um sistema de versão, não teria como duas pessoas usarem o arquivo ao mesmo tempo. Uma das aplicações do git é justamente essa, permitir que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes. Por mais complexo que isso seja, ele tenta manter tudo em ordem para evitar problemas para nós desenvolvedores.

GitHub é uma plataforma de hospedagem de código-fonte com [controle de versão](https://pt.wikipedia.org/wiki/Sistema_de_controle_de_vers%C3%B5es) usando o [Git](https://pt.wikipedia.org/wiki/Git" \o "Git). Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou [Open Source](https://pt.wikipedia.org/wiki/Open-source) de qualquer lugar do mundo. GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam [repositórios](https://pt.wikipedia.org/wiki/Reposit%C3%B3rio) remotos

**Selenium**

Selenium é uma ferramenta para testar aplicações web pelo browser de forma automatizada. Selenium se refere ao Acceptance Testing (ou functional testing) que envolve rodar testes num sistema finalizado e também automatiza as execuções dos navegadores. Primeiramente, é para automatizar aplicativos da Web para fins de teste, mas certamente não se limita apenas a isso. Tarefas de administração baseadas na web e repetitivas podem ser automatizadas também.

O Selenium tem o apoio de alguns dos maiores fornecedores de navegadores que tomaram (ou estão tomando) medidas para tornar o Selenium uma parte nativa de seu navegador. É também a principal tecnologia em inúmeras outras ferramentas de automação de navegadores, APIs e frameworks.

**Funcionamento do GitHub**

Neste trabalho o GitHub foi utilizado como repositório dos arquivos no período de edição. Através do Windows utilizando a ferramenta Git Bash. Na bibliografia está o link de download do programa, após baixar o arquivo é só executar o setup e instalar.

No momento da instalação será criado o seu Work Directory que é onde você vai trabalhar. Os arquivos ficam aí para poderem ser usados e alterados quantas vezes quiser para você. É basicamente sua pasta de arquivos dos projeto.

Sempre que abrir o Git Bash seu priemiro comando será se encaminhar ao Work Directory que terá o nome que foi definido na instalação do programa logo após entrar será necessário os seguintes comandos:

git config --global user.name "Seu nome"

git config --global user.email "seu e-mail"

Para definir seu nome de usuário e e-mail relacionados a sua conta GitHub realizar alterações no repositório. Próximo passo é clonar o repositório observação fazer somente uma vez caso já esteja clonado, é só entrar na pasta pelo guit bash

git clone (endereço do repositorio)

para listar e confirmar arquivos na pasta

Ls

(Verde arquivo liberado para uso e vermelho não liberado)

Para adicionar os arquivo no estágio de controle utilize o seguinte comando.

git add -A

Para todos

git add nomeDoarquivo.extensão

Para arquivo especifico

Para verificar situação do arquivo, se ele está na área de transferência ou estágio de controle use:

git status

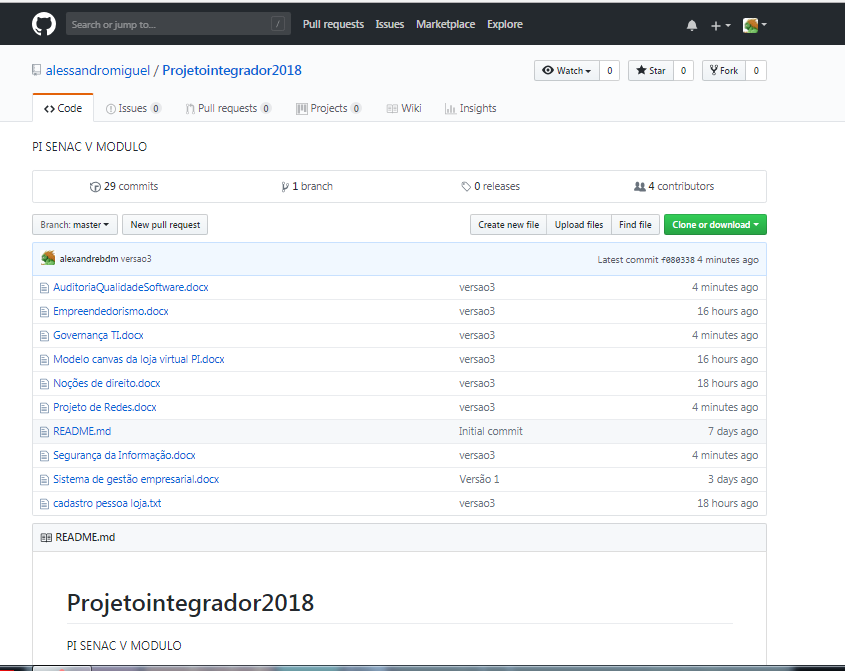
Para enviar digite:

git commit –m “pode-se por alguma observação entre aspas”

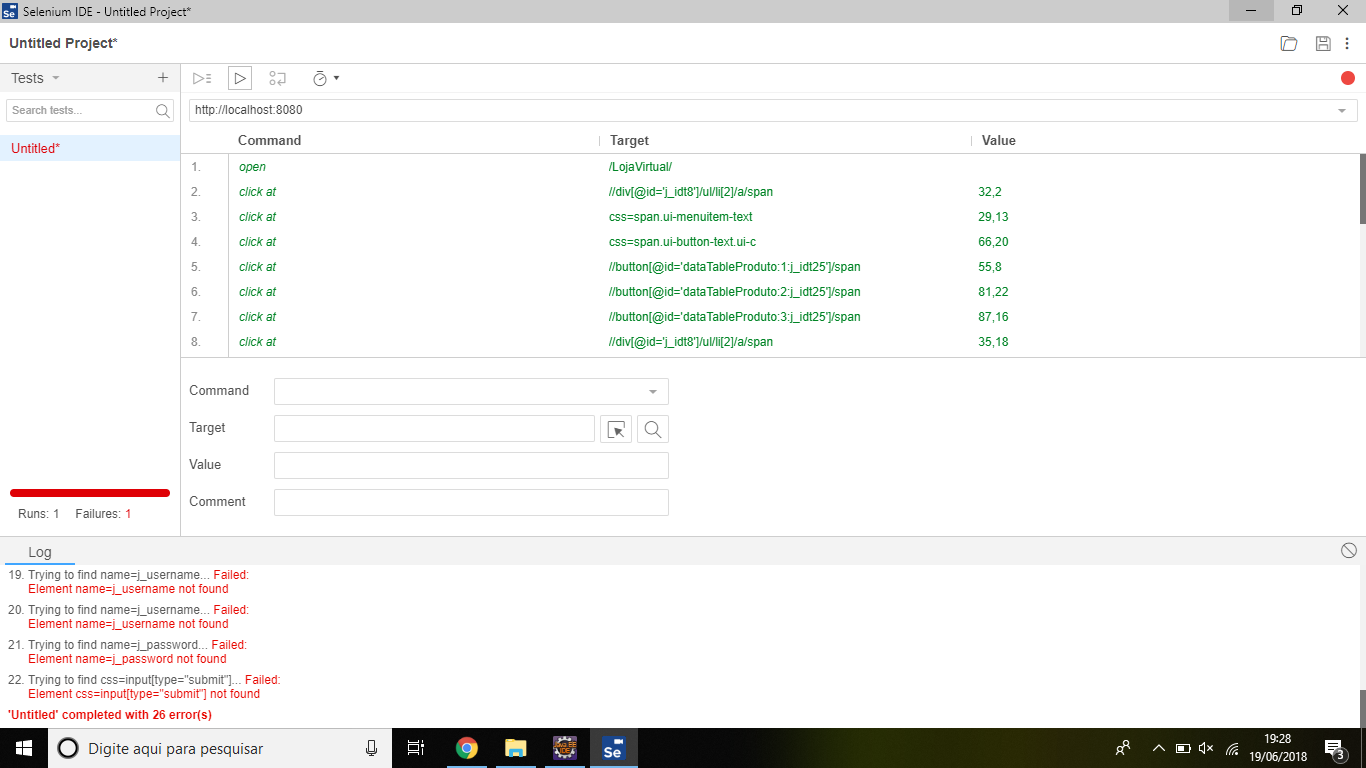
git push

Envia os commits locais para o repositório remoto

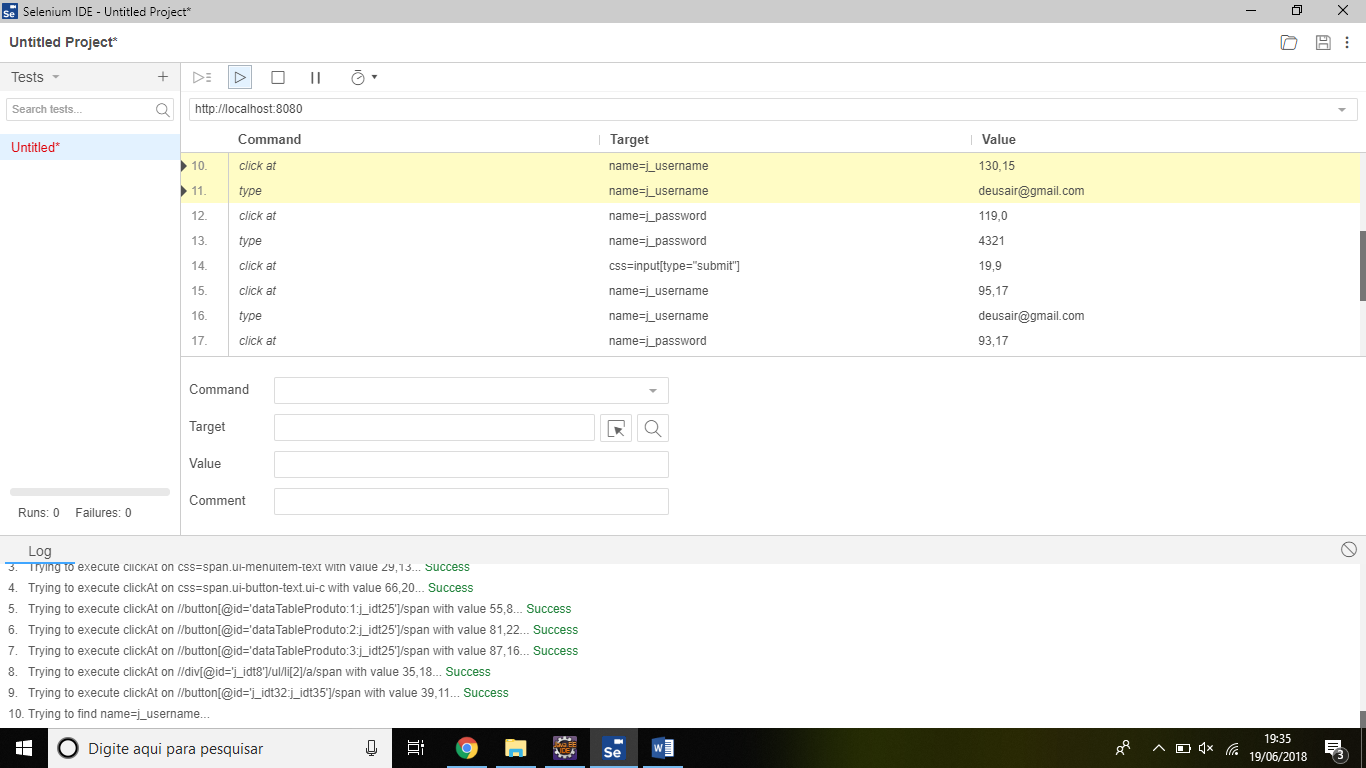
Como exemplo utilizaremos o repositório deste trabalho



Testes com erros de login para acesso de usuário para a loja usa músico.



Testes com Sucesso de login para acesso de usuário para a loja usa músico.



**JAVA**

**PACKAGE PERSISTENCIA**

**import java.io.Serializable;**

**import java.util.List;**

**import org.hibernate.Query;**

**import org.hibernate.Session;**

**import org.hibernate.Transaction;**

**import beans.\*;**

**import negocio.PessoaCtrl;**

**public class PessoaDAO implements Serializable{**

**private static final long serialVersionUID = 1L;**

**public static void inserir(Pessoa pessoa) {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Transaction t = sessao.beginTransaction();**

**sessao.save(pessoa);**

**t.commit();**

**sessao.close();**

**}**

**public static void alterar(Pessoa pessoa) {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Transaction t = sessao.beginTransaction();**

**sessao.update(pessoa);**

**t.commit();**

**sessao.close();**

**}**

**public static void excluir(Pessoa pessoa) {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Transaction t = sessao.beginTransaction();**

**sessao.delete(pessoa);**

**t.commit();**

**sessao.close();**

**}**

**public static List<Pessoa> listagem(String filtro) {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Query consulta;**

**if(filtro.trim().length() == 0){**

**consulta = sessao.createQuery("from Pessoa order by pes\_nome");**

**} else {**

**consulta = sessao.createQuery("from Produto "**

**+ "where pes\_nome like :parametro order by pes\_nome");**

**consulta.setString("parametro", "%" + filtro + "%");**

**}**

**List<Pessoa> lista = consulta.list();**

**sessao.close();**

**return lista;**

**}**

**public static void excluir(Fone fones) {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Transaction t = sessao.beginTransaction();**

**sessao.delete(fones);**

**t.commit();**

**sessao.close();**

**}**

**public static Pessoa retornaUsuario() {**

**Session sessao = HibernateUtil.getSessionfactory().openSession();**

**Query consulta = sessao.createQuery("from Pessoa where pes\_email ='" + PessoaCtrl.getUsuarioLogado()+"'");**

**Pessoa pessoa = (Pessoa) consulta.uniqueResult();**

**sessao.close();**

**return pessoa;**

**}**

**}**

**HTML**

**<?xml version="1.0" encoding="UTF-8"?>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">**

**<ui:decorate xmlns="http://www.w3.org/1999/xhtml"**

**xmlns:h="http://java.sun.com/jsf/html"**

**xmlns:ui="http://java.sun.com/jsf/facelets"**

**xmlns:f="http://java.sun.com/jsf/core"**

**xmlns:p= "http://primefaces.org/ui"**

**template="/template/tempcliente.xhtml">**

**<h1>LOGIN</h1>**

**<h:body>**

**<div align="center">**

**<h:panelGroup rendered="#{!empty aram.login\_error}">**

**Erro ao efetuar o login.<br />**

**Motivo: e-mail ou Password incorreto**

**<!-- #{SPRING\_SECURITY\_EXCEPTION.message}-->**

**</h:panelGroup>**

**<form id="login" method="post"**

**action="#{request.contextPath}/j\_spring\_security\_check"**

**align="center">**

**<table>**

**<tr>**

**<td>Login</td>**

**<td><input type='text' name='j\_username' /></td>**

**</tr>**

**<tr>**

**<td>Senha</td>**

**<td><input type='password' name='j\_password' /></td>**

**</tr>**

**<tr>**

**<td align="right"><input type="checkbox"**

**name="\_spring\_security\_remember\_me" /></td>**

**<td>Logar automaticamente</td>**

**</tr>**

**<tr>**

**<td></td>**

**<td><input type="submit" value="Logar" /></td>**

**</tr>**

**</table>**

**<script>**

**document.getElementById("login").j\_username.value = "#{SPRING\_SECURITY\_LAST\_USERNAME}";**

**</script>**

**</form>**

**</div>**

**</h:body>**

**</ui:decorate>**

**Bibliografia**

<https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>

https://pt.wikipedia.org/wiki/GitHub

<https://www.seleniumhq.org/>

<https://git-scm.com/downloads> (Git Bash)