

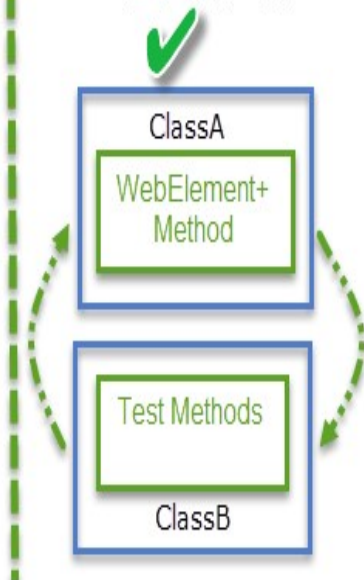
Patrones (POM, PageFactory)

POM

Non POM Structure



POM Based Structure



Es un patrón de diseño de objeto en Selenium, donde las páginas web se representan como clases, y los diversos elementos de la página se definen como variables en la clase. Todas las interacciones de usuario posibles se pueden implementar como métodos en la clase:

```
clickLoginButton();
setCredentials(user_name,user_password);
```

Dado que los métodos bien nombrados en las clases son fáciles de leer, esto funciona como una forma elegante de implementar rutinas de prueba que son legibles y más fáciles de mantener o actualizar en el futuro.

PageFactory

Page Factory es un concepto de Página Modelo de Objetos incorporado para Selenium WebDriver pero está muy optimizado.

Aquí también, seguimos el concepto de separación de Page Object Repository y Test Methods. Además, con la ayuda de la clase PageFactory, utilizamos las anotaciones **@FindBy** para encontrar WebElement. Usamos el método `initElements` para inicializar elementos web

@FindBy puede aceptar **tagName, partialLinkText, name, linkText, id, css, className, xpath** como atributos.

WebElements are identify by
@FindBy Annotation

static `initElements` method of
PageFactory class for
initializing WebElement

```
@FindBy(xpath="//table//tr[@class='heading3']")
WebElement homePageUserName;

public Guru99HomePage(WebDriver driver){
    this.driver = driver;
    //This initElements method will create all WebElements
    PageFactory.initElements(driver, this);
}
```

Herramienta de Trabajo

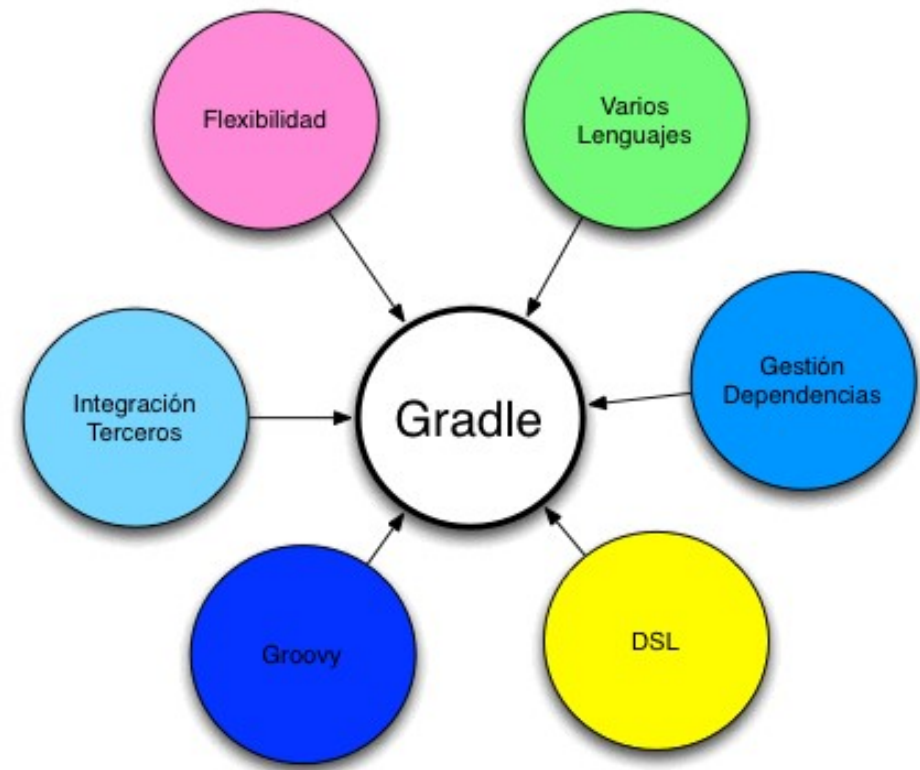
KATALON RECORDER



Katalon Recorder, una pequeña extensión para Chrome y Firefox. Esta extensión es una herramienta complementaria de Katalon Studio , una de las mejores herramientas de prueba de automatización del 2018 . A pesar de su apariencia simple, Katalon Recorder tiene muchas ventajas ocultas. La característica más importante, es su capacidad para generar locators ID, NAME y XPath, con constructores de localizadores integrados, así como constructores de localizadores personalizados.

¿Qué es GRADLE?

Gradle es una herramienta de automatización de la construcción de nuestro código que bebe de las aportaciones que han realizado herramientas como ant y maven pero intenta llevarlo todo un paso mas allá. Para empezar se apoya en Groovy y en un DSL (Domain Specific Language) para trabajar con un lenguaje sencillo y claro a la hora de construir el build comparado con Maven. Por otro lado dispone de una gran flexibilidad que permite trabajar con ella utilizando otros lenguajes y no solo Java. Dispone por otro lado de un sistema de **gestión de dependencias** sólido.



Patrón POM (Page Object Model)

SERENITY BDD con CUCUMBER

Primeros Pasos

Herramientas a Usar:

- ✓ Eclipse IDE
- ✓ Java 1.8 y JDK
- ✓ Gradle
- ✓ Cucumber
- ✓ Proyecto Base:

[Url:https://drive.google.com/open?id=1UYN3z_xHjaGRvUd0dhuLQ-6UVmfAMS6Q](https://drive.google.com/open?id=1UYN3z_xHjaGRvUd0dhuLQ-6UVmfAMS6Q)



¿Cómo Instalar Cucumber?

Ingresar a Eclipse

Menu: *Help>Install New Software*

Clic en el botón “ADD”

Escribir en Name: *Cucumber*

Escribir en Location: <http://cucumber.github.com/cucumber-eclipse/update-site>

Dar clic en el botón “OK”

Seleccionar “Cucumber Eclipse Plugin” , “Next” y “Finish”

Primeros Pasos



¿Cómo Cargar el Proyecto Base?

1. Descargar proyecto base en la ruta del workspace
2. Importar proyecto:
 - ✓ Para importar: **File > Import > Gradle > Existing Gradle Projects > Next**
 - ✓ *Seleccionamos la ruta donde se encuentra el proyecto y “Next”*
 - ✓ *Definimos la versión de Gradle que esta en nuestro PC*
 - ✓ *Dar clic en el botón “Finish”*

Nota: este proceso durará varios minutos, permita que el proceso finalice por completo. Al finalizar se cargará la siguiente estructura.



Primeros Pasos

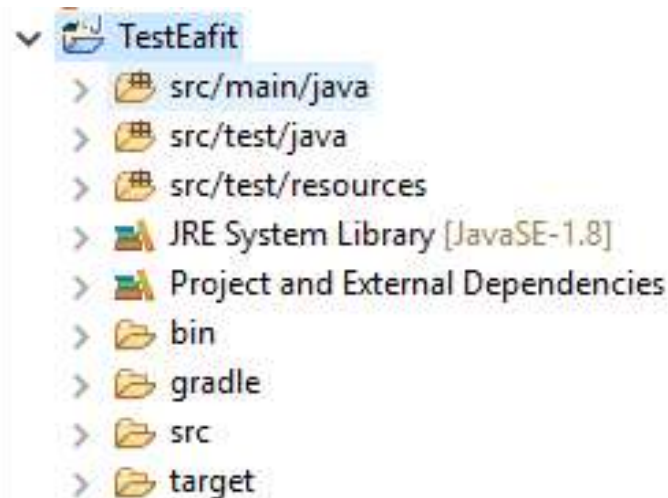


¿Cómo Cargar el Proyecto Base?

1. Descargar proyecto base en la ruta del workspace
2. Importar proyecto:
 - ✓ Para importar: **File > Import > Gradle > Existing Gradle Projects > Next**
 - ✓ Seleccionamos la ruta donde se encuentra el proyecto y “Next”
 - ✓ Definimos la versión de Gradle que esta en nuestro PC
 - ✓ Dar clic en el botón “Finish”

Nota: este proceso durará varios minutos, permita que el proceso finalice por completo.

Al finalizar se cargará la siguiente estructura.



Tip: Una vez finalizada la carga del proyecto crear un archivo `.feature` para verificar la carga correcta del mismo.

Mi Primera Historia de Usuario

Introducción a BDD, gherkin y Cucumber

BDD (Behavior Driven Development): Desarrollo guiado por el comportamiento, consiste básicamente en una estrategia de desarrollo y lo que plantea es la definición de los requisitos desde el punto de vista del comportamiento de la aplicación, desde el negocio, creado en un lenguaje común para el negocio y para los técnicos.

Gherkin, es un lenguaje común, que lo puede escribir alguien sin conocimientos en programación, pero que lo puede comprender un programa, de forma tal que se pueda utilizar como especificación de pruebas.

Estas pruebas se almacenan en archivos “.feature” los cuales deberían ser versionados junto al código fuente que se está probando.

Gherkin es considerado un lenguaje Business Readable DSL (Lenguaje específico de dominio legible por el negocio).

Mi Primera Historia de Usuario

Para empezar a hacer BDD, sólo se necesita conocer 5 palabras con lo que vamos a describir las funcionalidades:

Feature: Nombre de la funcionalidad que vamos a probar.

Scenario: habrá uno por cada prueba que quiera especificar para esta funcionalidad

Given: precondiciones

When: acciones que se van a ejecutar

Then: Se especifica el resultado esperado, las verificaciones a realizar.

Nota: Una **feature** puede contener varios escenarios de prueba.

Mi Primera Historia de Usuario

Ejemplo:

Feature : Acceso Aplicativo Metis

Como usuario

Quiero autenticarme en Metis

A través de la pagina de acceso a la aplicación.

Scenario: Realizar la Autenticación en Metis.

Given que Yeison quiere acceder a Metis

When en escribe el usuario demo y la clave demo

Then el ve el mensaje de Bootstrap-Admin-Template



Cucumber, es una de las herramientas framework que podemos utilizar para automatizar nuestras pruebas BDD, permite ejecutar descripciones funcionales en texto plano como pruebas de software automatizadas.

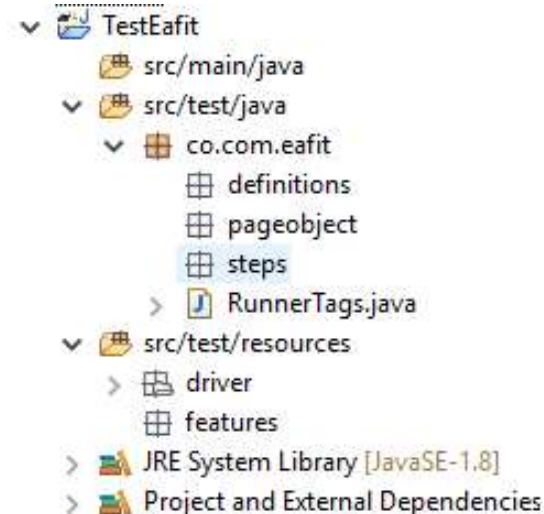
¿Qué es Serenity BDD?

Serenity es una biblioteca de código abierto que le ayuda a redactar pruebas de aceptación automatizadas de mayor calidad de forma más rápida.

Serenity te ayuda a:

- ✓ Escribir pruebas que sean más flexibles y fáciles de mantener
- ✓ Producir informes ilustrados y narrativos sobre sus pruebas
- ✓ Asigne sus pruebas automatizadas a sus requisitos
- ✓ Vea cuánto de su aplicación se está probando realmente
- ✓ Y controle el progreso del proyecto

Conozcamos Nuestro Proyecto Base



Desarrollo ejercicio Practico Serenity BDD

1. Realizar la Autenticación en la aplicación Metis.

URL: <https://colorlib.com/polygon/metis/login.html>

Feature : Acceder al Aplicativo Metis

Como usuario

Quiero autenticarme en Metis

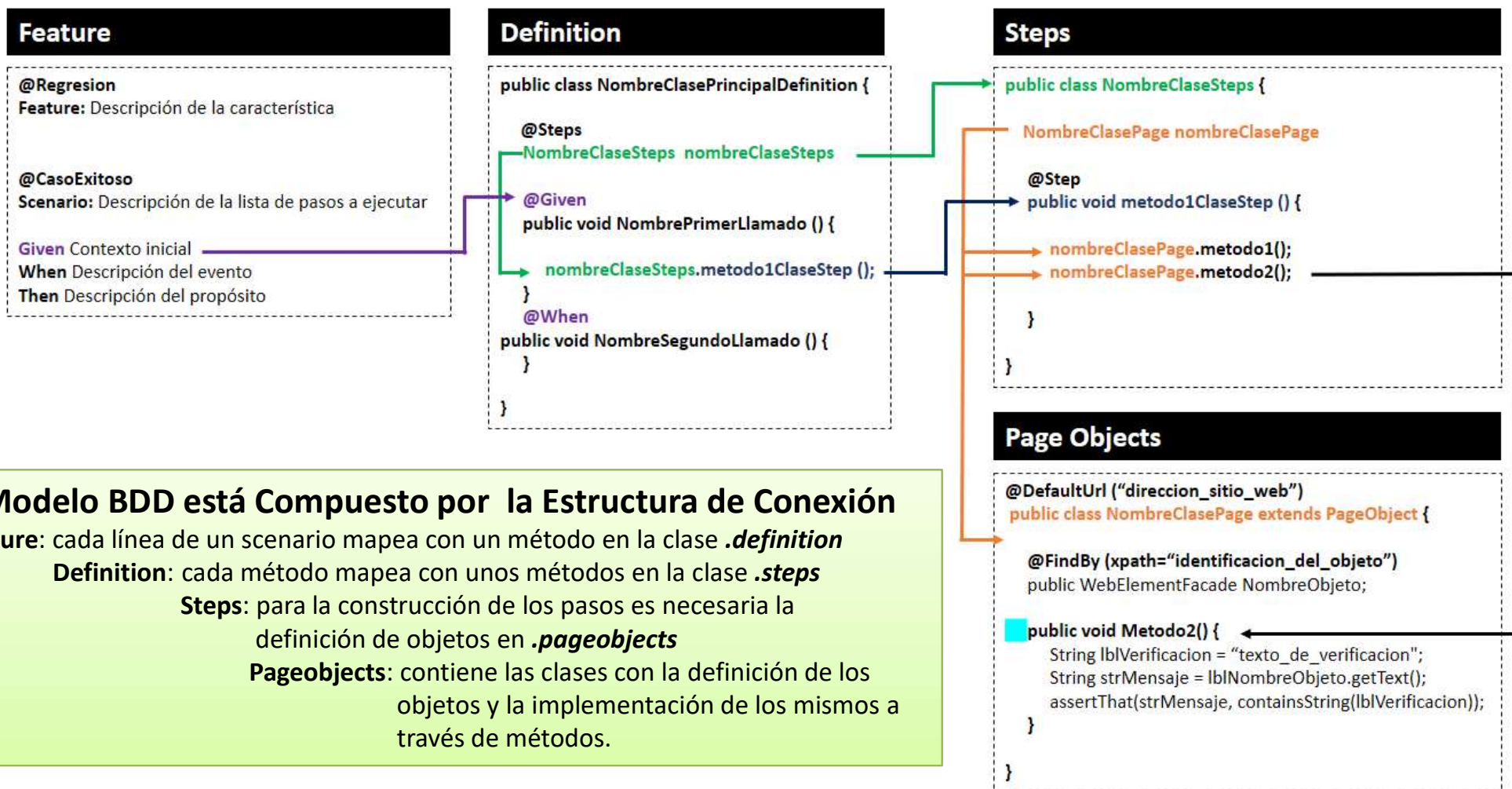
A través de la pagina de acceso a la aplicación.

Scenario: Realizar la Autenticación en Metis.

Given que Yeison quiere acceder a Metis

When en escribe el usuario demo y la clave demo

Then el ve el mensaje de Bootstrap-Admin-Template



Desarrollo ejercicio Practico Serenity BDD

2. Realizar la ejecución por consola

- ☐ Limpiar evidencias

 - ***gradle clean***

- ☐ Ejecución con evidencias de serenity

 - ***gradle -Dtest.single=RunnerTags test aggregate***

Control de Versiones con Git, Bitbucket ...



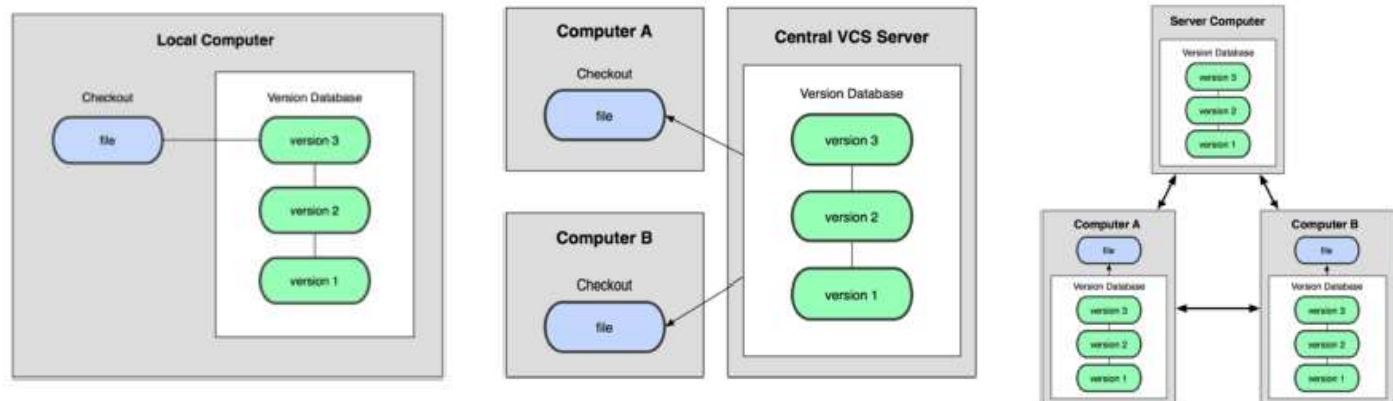
¿Qué es el control de versiones?

El control de versiones es un sistema (Version Control System o VCS) que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

Además, permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más.

Tipos de CVS

- Locales
- Centralizados
- Distribuidos

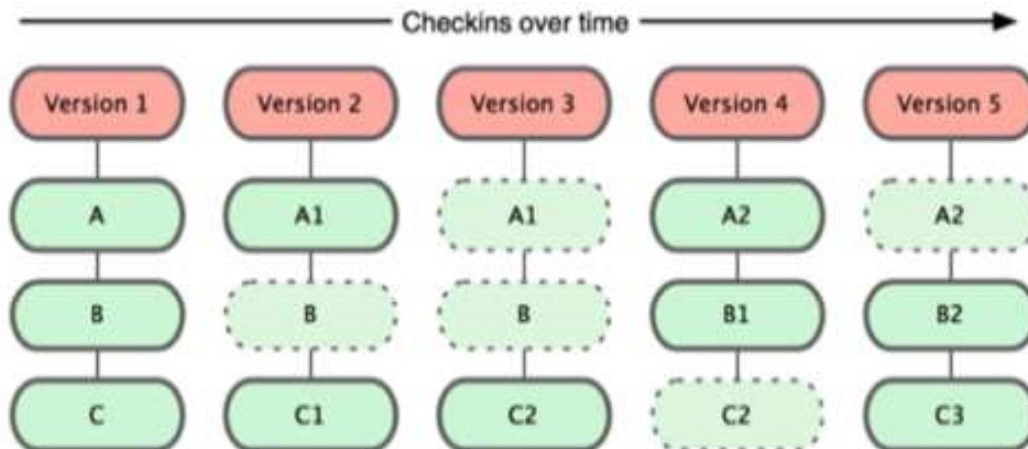


Control de Versiones con Git, Bitbucket ...



¿Qué es GIT?

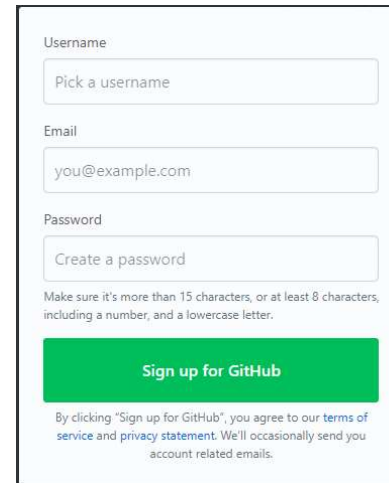
Es un software de control de versiones diseñado por Linus Torvalds. Permite la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, proporcionando las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.



Git almacena la información como instantáneas del proyecto a lo largo del tiempo, básicamente hace una foto del aspecto de todos tus archivos en ese momento, y guarda una referencia a esa instantánea.

Crear una cuenta en GitHub ...

1. Ingresar a la URL: <https://github.com/>
2. Registrarse con 3 simples datos:
 - ✓ Username
 - ✓ Email
 - ✓ Password
3. Seleccionar Plan Free y clic en Continuar
4. Finalizar el registro.



Username
Pick a username

Email
you@example.com

Password
Create a password

Make sure it's more than 15 characters, or at least 8 characters, including a number, and a lowercase letter.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

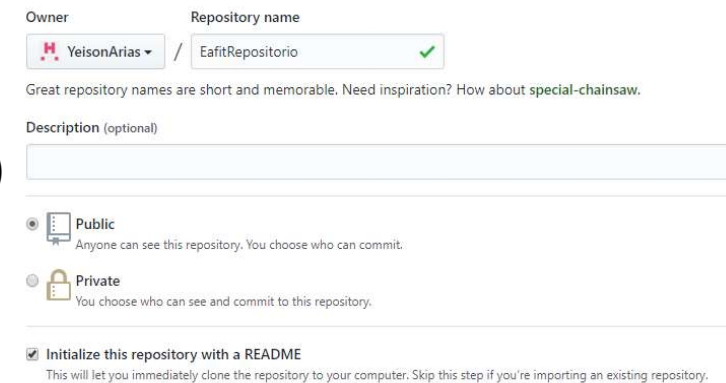
Crear un nuevo Repositorio...

1. Clic en Nuevo repositorio (**New repository**)
2. Escribir el nombre del repositorio (**Repository name**)
3. Dejar seleccionada la opción Publico (**Public**)
4. Seleccionar el check (**Initialize this repository with a README**)
5. Dar clic en el botón Crear repositorio (**Create repository**)

Nota: Se debe acceder al archivo **JenkinsFile** del proyecto y actualizar la URL del repositorio y usuario del mismo.

Create a new repository

A repository contains all the files for your project, including the revision history.



Owner: YeisonArias / Repository name: EafitRepositorio ✓

Great repository names are short and memorable. Need inspiration? How about [special-chainsaw](#).

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Carga nuestro proyecto a Git (Se debe instalar GIT)

1. Una vez creado el repositorio, da clic en el botón “Clone and Download” y copiar la Url del repositorio que allí se carga.
2. Ejecutar la consola de Windows a traves del comando “CMD” desde la raíz del proyecto.
3. Ejecutar los siguientes comandos en la consola:
 1. `git init`
 2. `git remote add origin <<url del repositorio>>`
 3. `git remote -v`
 4. `git pull origin master`
 5. `git add -A`
 6. `git status`
 7. `git commit -m “versión 1.0”`
 8. `git push origin master`

