CHOUCAIR
Business Centric Testing

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 especifico de dominio legible por el negocio).

CHOUCAIR®
Business Centric Testing

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.

CHOUCAIR
Business Centric Testing

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.

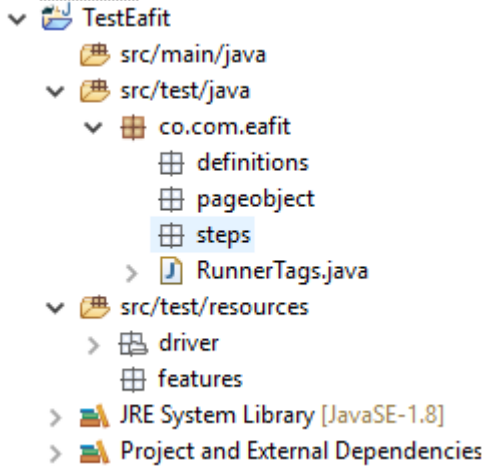
¿Qué es Serenity BDD?

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



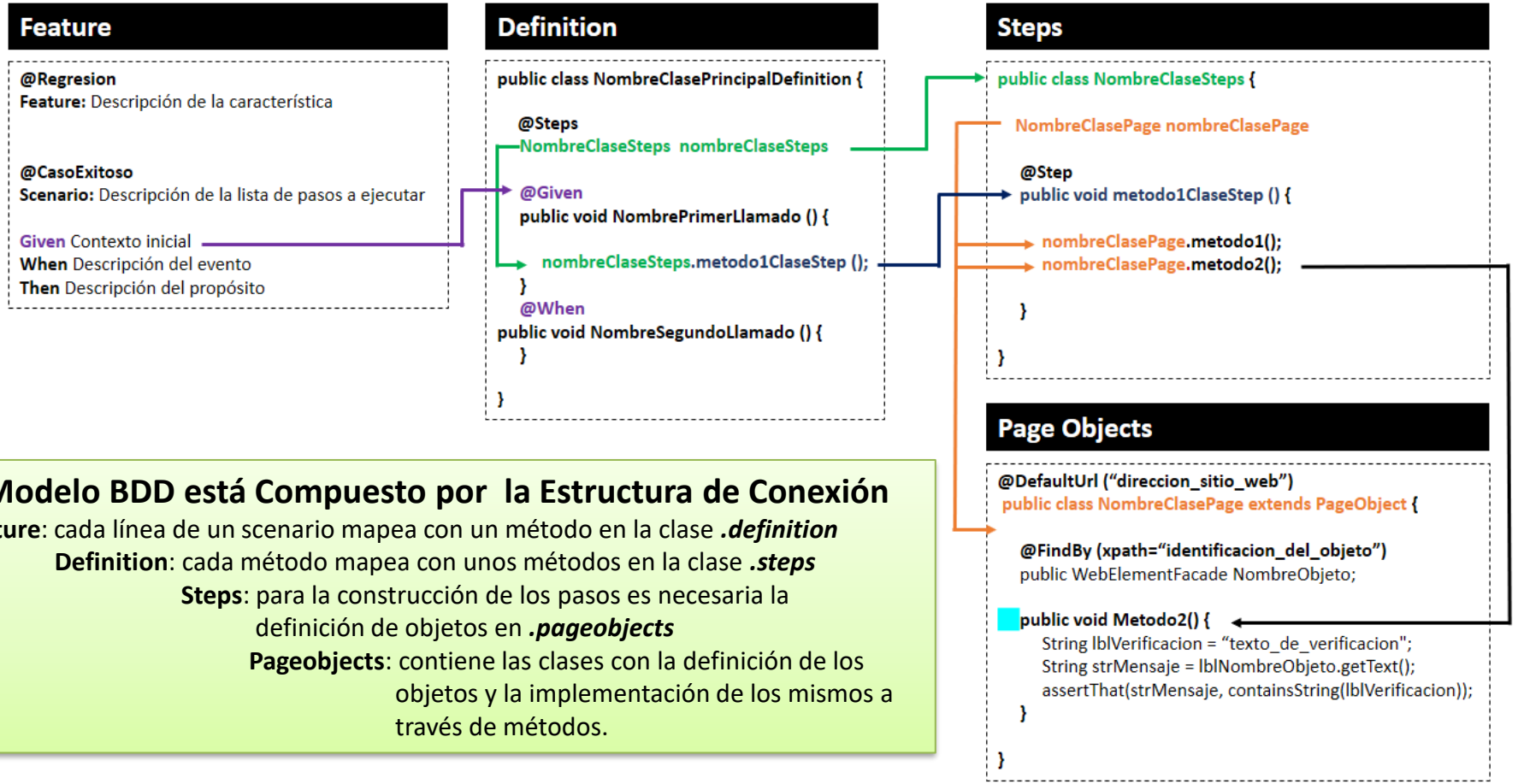
Desarrollo ejercicio Practico Serenity BDD

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

CHOUCAIR
Business Centric Testing



Modelo BDD está Compuesto por la Estructura de Conexión

Feature: cada línea de un scenario mapea con un método en la clase ***.definition***

Definition: cada método mapea con unos métodos en la clase ***.steps***

Steps: para la construcción de los pasos es necesaria la definición de objetos en ***.pageobjects***

Pageobjects: contiene las clases con la definición de los objetos y la implementación de los mismos a través de métodos.

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***

Desarrollo ejercicio Practico Serenity BDD

2. Realizar el ejercicio del envió de Comentarios:
URL: http://www.eafit.edu.co/institucional/contacto/Paginas/contacto-eafit.aspx

```
Feature: Contactenos Universidad EAFIT
 Como estudiante  Quiero enviar un comentario a la Universidad EAFIT A través de la pagina de Contáctenos

  @Comentario
  Scenario: Envio comentario Universidad EAFIT
    Given que Yeison quiere escribir un comentario a la Universidad EAFIT
    When el envia el comentario
    |nombre          |correo              |telefono  |ciudad    |comentario             |
    |Yeison Arias    |yarias@correo.com   |2222222   |Medellin  |Excelente universidad  |
    Then el visualiza el pantalla el mensaje Muchas gracias por su comentario.
```

Control de Versiones con Git, Bitbucket …

CHOUCAIR
Business Centric Testing

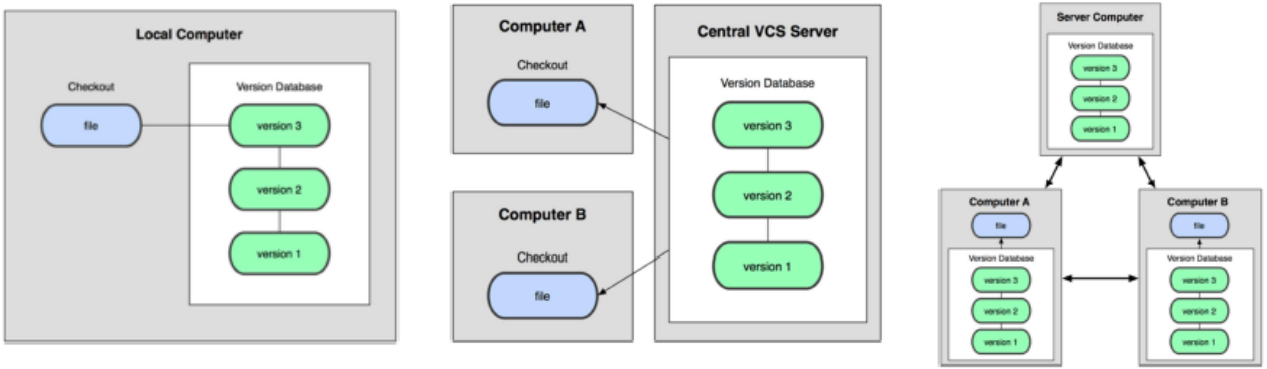
¿Qué es el control de versiones?

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**

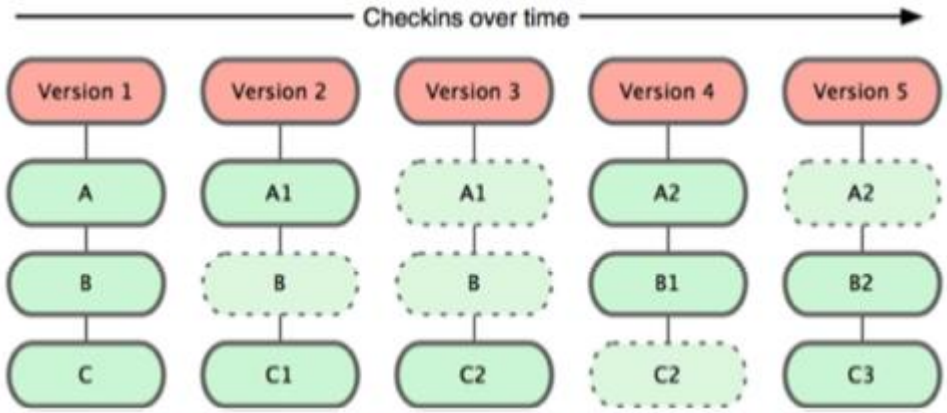


Control de Versiones con Git, Bitbucket …

CHOUCAIR
Business Centric Testing

¿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.

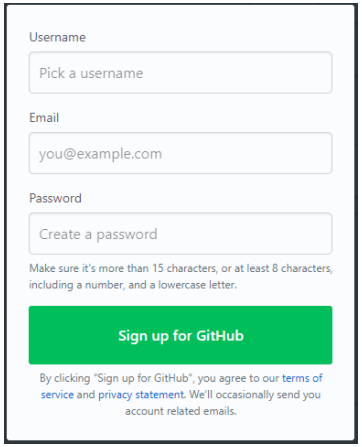


Git almacena la información como instantáneas del proyecto a lo largo del tiempo, básicamente hace una foto del aspecto de todos tus archivos en ese momento, y guarda una referencia a esa instantánea.

CHOUCAIR
Business Centric Testing

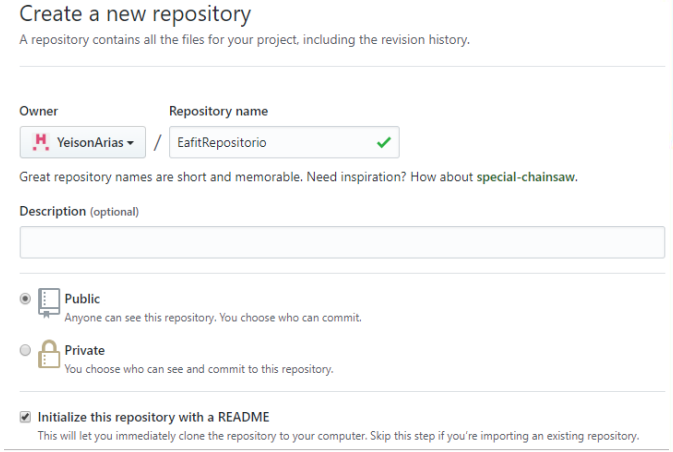
Crear una cuenta en GitHub …

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.*



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***)

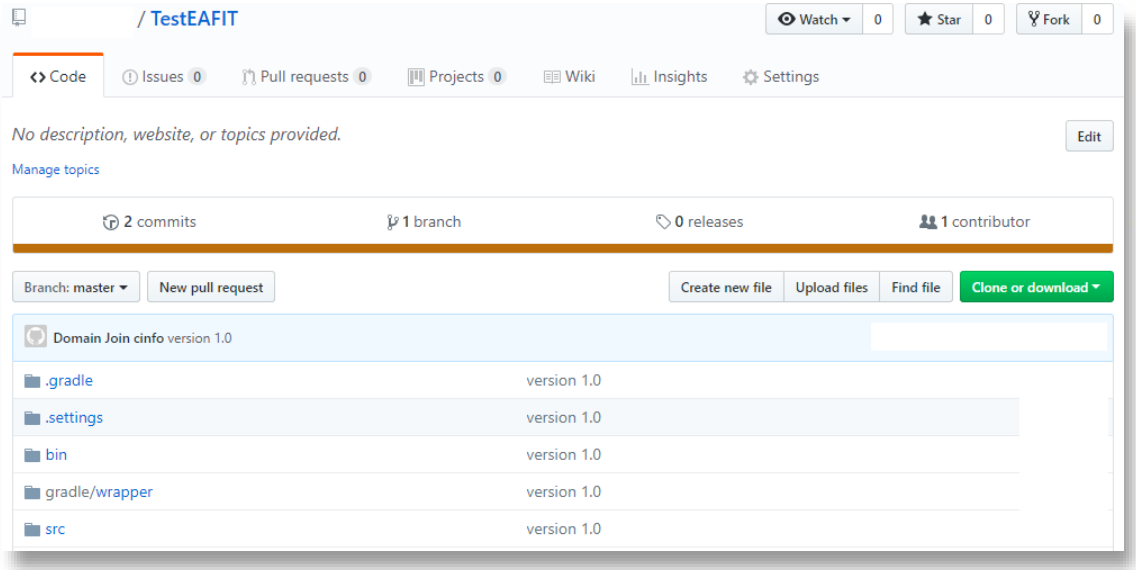


Nota: Se debe acceder al archivo ***JenkinsFile*** del proyecto y actualizar la URL del repositorio y usuario del mismo.

Carga nuestro proyecto a Git (Se debe instalar GIT)

CHOUCAIR
Business Centric Testing

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 traces 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*



Pipeline en Jenkins (Instalar GRADLE)

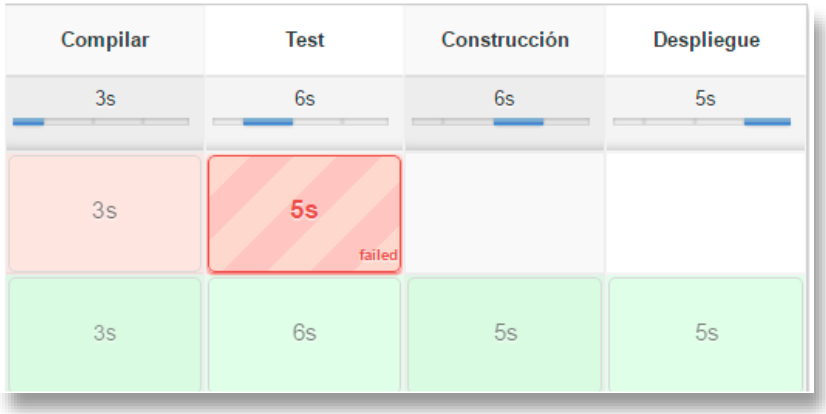
CHOUCAIR
Business Centric Testing

¿Qué son los pipelines de Jenkins?

Definen el ciclo de vida de la aplicación de nuestro flujo de integración/entrega continua utilizando un lenguaje basado en Groovy. Debido a su gran flexibilidad y compatibilidad con numerosos plugins, permiten crear flujos complejos completos capaces de implementar sistemas de entrega continua (Continuos Delivery – CD). Además los pipelines pueden sobrevivir a reinicios del servidor y pueden ser pausados a la espera de que una persona realice una acción antes de que continúe la ejecución del flujo.

¿Qué elementos componen un pipeline?

Un pipeline se compone de distintas etapas (***stages***) secuenciales que ejecutan tareas (***steps***) que son lanzadas en nodos de trabajo (***nodes***).



Creación del Pipeline en Jenkins

Jenkins

CHOUCAIR
Business Centric Testing

1. Creación del repositorio en Github
2. Definición del Pipeline, esta la realizaremos en un archivo ***Jenkinsfile*** que se encuentra en el directorio principal de nuestro proyecto. Este archivo es un script basado en Groovy que permite definir un pipeline de Jenkins.
3. Creación del pipeline en Jenkins
    1. Acceder a la interfaz web de Jenkins y realizar la autenticación
    2. Una vez allí creamos un nueva tarea (***New Item***)
    3. Ingresamos el nombre de la tarea, seleccionamos el tipo (***Pipeline***) y clic en el botón "***OK***"
    4. Dar clic en la opción "**This project is parameterized**" , luego dar clic en el botón "***Add Parameters***" y seleccionar la opción "***String Parameter***".
    5. Colocar un nombre "***RUNNER***" y en el campo **Default Value** colocar el texto "***RunnerTags***"
    6. El la sección **Build Triggers** seleccionamos la opción "**Build periodically**" y adicionamos en el campo **Schedule** el valor "***H/15 * * * ****" con lo cual estamos definiendo una ejecución cada 15 minutos.

Creación del Pipeline en Jenkins

Creación del pipeline en Jenkins

7. Es la sección **Pipeline** para el campo **Definition** seleccionamos el valor "**Pipeline script from SCM**".
8. En el campo **SCM** seleccionamos la opción "**Git**".
9. En la sección **Repositories,** para el campo "***Repository URL***" agregamos la Url del repositorio donde se encuentra nuestro proyecto en Github.
10. En el campo **Script Path** colocamos el nombre de nuestro archivo **JenkinsFile** que para nuestro caso se llama "***JenkinsfileLocal***"
11. Finalmente damos clic en el botón "***SAVE***".

CHOUCAIR®
Business Centric Testing

¡Muchas gracias!

¡En Choucair somos más que pruebas!