

Proyecto: Reto Sitio Web Orange HRM OPEN SOURCE HR MANAGEMENT

Estrategia de Prueba Automatizadas

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | David Zaraza | Creación del documento | Octubre 2021 |

Tabla de Contenidos

[1.Introducción](#_knb62690yxb3)

[2. Alcance](#_qlpgkz6qysqk)

[3. Roles y Responsabilidades](#_7b58fkv4on7k)

[5. Ambiente y Herramientas de Pruebas](#_d7hpcbc6w3za)

[5.1 Herramientas de Pruebas](#_3dy6vkm)

[5.2 Arquitectura del framework de automatización](#_4d34og8)

[5.3 Ambiente de Pruebas](#_17dp8vu)

[6. Criterios de Entrada y Salida](#_ye1dj1oecscb)

[6.1 Criterios de Entrada](#_26in1rg)

[6.2 Criterios de Salida](#_lnxbz9)

[7. Planificación de ejecución de las pruebas](#_453gv79mohyv)

[7.1 Planificación de las Pruebas de Regresión](#_1ksv4uv)

[8. Reporte de Pruebas](#_44sinio)

# 1.Introducción

En esta Estrategia para la realización de pruebas automatizadas se describe el alcance de las pruebas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar, los riesgos, planes de contingencia y el calendario de ejecución de las pruebas del proyecto reto orange HRM.

# 2. Alcance

Se realizarán pruebas automatizadas a las funcionalidades asignadas para el desarrollo del reto. Las funcionalidades a ser automatizadas estan dentro de los criterios vistos en cada sesión de mentorías.

# 3. Roles y Responsabilidades

|  |  |
| --- | --- |
| Roles | Responsabilidades |
| Automatizador de pruebas | Planificación de estrategia de pruebas  Diseño e implementación de las pruebas.  Ejecución de las pruebas automatizadas.  Reporte de resultados de las pruebas. |

4. Riesgos y Planes de Contingencia

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Riesgos | Probabilidad de Ocurrencia  (1-5) | Impacto  (1-5) | Severidad  (Prob\*Impacto) | Plan de Contingencia |
| 1 | Funcionalidades no terminadas en el tiempo estimado | 1 | 5 | 10 | Evaluar las funcionalidas automatizar y estimar el tiempo necesario de acuerdo a los recursos. |
| 2 | Daño en la energía del hogar, servicios de internet o equipos de computo donde se desarrolla el proyecto | 1 | 5 | 10 | Se cuenta con un equipo adicional (personal) diferente al suministrado por Sofka. En caso de que el internet se caiga se cuenta con plan de datos para compartir los mismos.  Ante el servicio de energía mantener el equipo cargado en todo momento. |

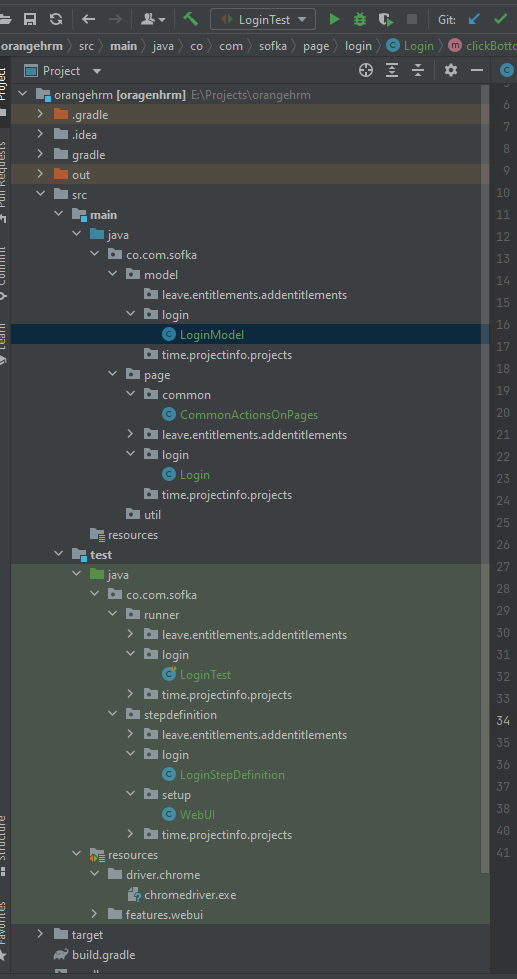
# 5. Ambiente y Herramientas de Pruebas

## 5.1 Herramientas de Pruebas

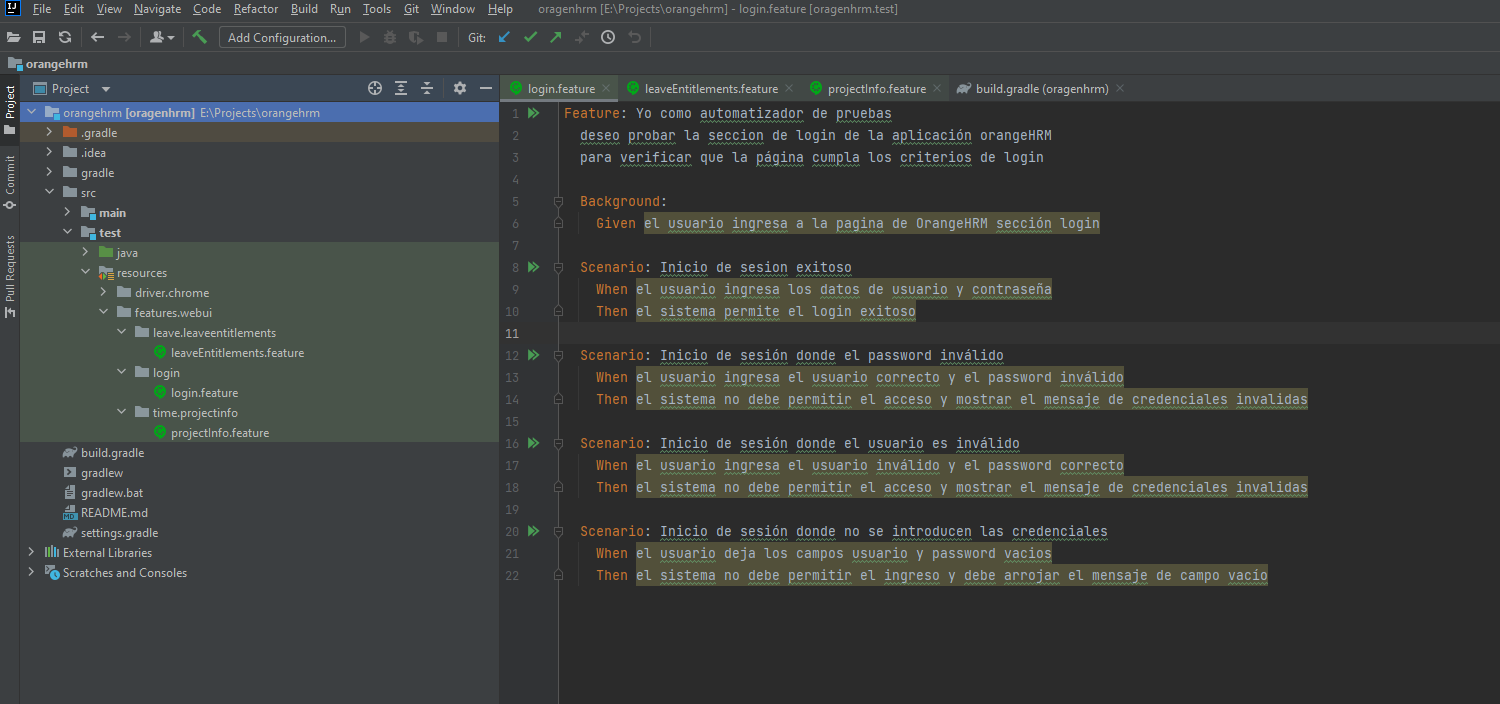
|  |  |
| --- | --- |
| Herramienta | Función |
| Selenium WebDriver | API para automatizar sistemas Web |
| JUnit testing framework | Ejecución y Reporte de las pruebas |
| Gradle | Creación de la estructura de proyectos y uso e importación de librerías |
| Chromedriver | Crea una instancia del navegador Chrome |
| IntelliJ | IDE para la codificación y compilación del proyecto |
| Java JDK y JSE | V 1.8 como lenguaje para trabajar el framework Selenium y el DSL Cucumber y Gherkin |
| Cucumber y Gherkin | Herramienta para crear los escenarios con orientación al BDD |

## 5.2 Arquitectura del framework de automatización

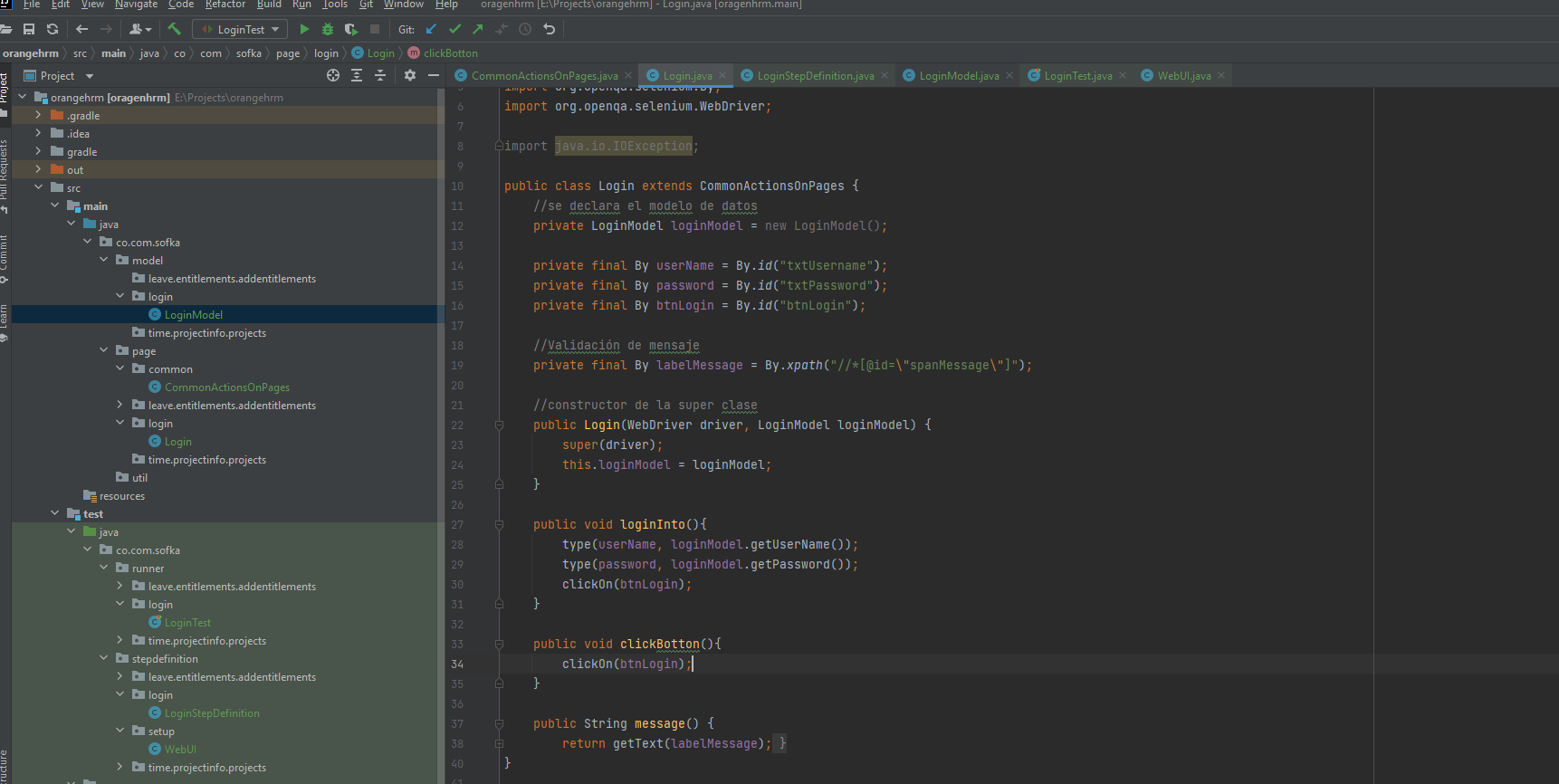
El proyecto estará desarrollado bajo el patrón Page Object Model y POM - Page Factory para “mapear” las páginas del sistema a clases “Page” que permitan aislar las acciones de las diferentes páginas y a la vez agrupar todos los webElements de una página y las acciones que se pueden llevar a cabo, en una misma clase.



El proyetco contará con un estructura por paquetes por pagina para mantener un orden en el códigoy hacer su mantenimiento más asequible. Al utilizar cucumber y Gherkin se creará un feature con el respectivo escenario por cada funcionalidad a automatizar.



El Page Object Model también nos ayuda a concentrar los localizadores en estas clases “Page”, de forma que cuando el sistema cambia y es necesario actualizar el código de los localizadores que se hayan mapeado para cada elemento.



## 5.3 Ambiente de Pruebas

|  |  |
| --- | --- |
| Navegadores | Chrome |
| Sistemas Operativos | Windows 10 de 64 bits |

# 6. Criterios de Entrada y Salida

## 6.1 Criterios de Entrada

El framework de pruebas está instalado con la dependencias necesario y listo para la ejecución

El ambiente de QA está disponible.

Realizar pruebas manules antes de ejecutar las pruebas automatizadas para recorrer los caminos a automatizar.

Archivo Excel suministrado por Ivan Arroyave conl las funcionalidades a automatizar por cada analsita.

## 6.2 Criterios de Salida

Ejecución de todos los escenarios de pruebas automatizados

Reporte en cucumber y/o IDE por cada escenario automatizado

Codigo limpio y refactorizable siguiendo las buenas prácticas de programación

# 7. Planificación de ejecución de las pruebas

Lista de funcionalidades a ser automatizadas por el reto

|  |  |  |
| --- | --- | --- |
| Reto | Funcionalidad | Comentarios |
| 1 | POM - Inicio de sesión | Contiene 4 escenarios   * Inicio de sesión exitoso. * Inicio de sesión donde el password es inválido. * Inicio de sesión donde el usuario es inválido. * Inicio de sesión donde no se introducen credenciales. |
| 1.1 | POM - Modulo Leave  Para el menú: Entitlement - Add Leave Entitlement | Consta de un escenario   * Para un empleado existente, programarle un periodo de vacaciones. |
| 2 | POM - Page Factory  Para el menú: Project Info - Projects - Add | Consta de un escenario:   * Añadir un nuevo proyecto usando todos los campos. Se debe añadir un nuevo cliente   (llenar todos los campos requeridos). Añadir mínimo 2 administradores de proyecto.  Además, agregar una actividad existente de otro proyecto(Copy From) con todos sus ítems. |

# 8. Reporte de Pruebas

El Reporte automático de pruebas se obtendrá a través de Cucumber. Este Reporte informará sobre los resultados de la ejecución de cada caso de prueba. Incluirá las pruebas que pasaron y las que fallaron, los errores encontrados, la tasa de éxito y el tiempo transcurrido.

