Texto

Descripción generada automáticamente

Proyecto: Sitio Web OrangeHRM con Page Factory

Estrategia de Prueba Automatizadas

**HISTORIA DE REVISIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor(es) | Descripción | Fecha |
| 1.0 | Yeison Ortega | 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 OrangeHRM.

# **2. ALCANCE**

Se realizarán pruebas de caja negra (automatizadas) a las funcionalidades seleccionadas durante la planificación de cada sprint.

Las funcionalidades a ser automatizadas serán seleccionadas utilizando los criterios de la Lista de Chequeo “Qué casos de pruebas automatizar”.

# **3. ROLES Y RESPONSABILIDADES**

|  |  |
| --- | --- |
| Roles | Responsabilidades |
| Manager de QA | Planificación y monitoreo de las pruebas automatizadas  Reporte de Defectos  Reporte de progreso de las pruebas |
| Ingeniero QA de Automatización/ Analista QA | Diseño e implementación de las pruebas.  Ejecución de las pruebas automatizadas.  Reporte de resultados de las pruebas. |
| Product Owner/Stakeholders | Toma de decisiones |

**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 no pueden formar parte de las funcionalidades planificadas para ser automatizadas en el sprint actual | 1 | 5 | 5 | Re planificar las funcionalidades para ser automatizadas (sección 7) |
| 2 | Solicitud de cambios en aquellas funcionalidades que ya tienen casos de pruebas automatizados. Esto ocasiona re trabajo debido a que se deben actualizar estos scripts. | 3 | 3 | 9 | Estimar el tiempo del cambio y volver a priorizar la lista de funcionalidades a ser automatizadas en el sprint. |

# 

# **5. AMBIENTE Y HERRAMIENTAS DE PRUEBAS**

## **5.1 Herramientas de Pruebas**

|  |  |
| --- | --- |
| Herramienta | Función |
| Selenium WebDriver | API para automatizar sistemas Web |
| Cucumber | Realizar pruebas de aceptación y reportes |
| JUnit | Ejecución pruebas |
| Gradle | Creación de la estructura de proyectos y uso e importación de librerías |
| Chromedriver | Crea una instancia del navegador Chrome |

## **5.2 ARQUITECTURA DEL FRAMEWORK DE AUTOMATIZACIÓN**

Diagrama

Descripción generada automáticamente

Utilizaremos el patrón Page Object Model (Ver Video: Page Object Model con Selenium WebDriver 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.

La clase “Base” permite aislar todo el framework de la versión del API de Selenium WD que estemos utilizando. De esta forma si hay algún cambio en los comandos del API no tenemos que cambiar todas las clases sino solo la clase “Base”.

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 css selectors, xpath o lo que hayamos utilizado para localizar los webElements, solo tenemos que cambiarlo una sola vez en la clase “Page” y los “Tests”, que son el último nivel, no necesitan ningún cambio (a menos que haya cambiado la lógica de funcionamiento y dentro de los cambios se hayan eliminado o agregado funcionalidades al sistema).

Utilizaremos Page Factory

## **5.3 AMBIENTE DE PRUEBAS**

|  |  |
| --- | --- |
| Navegadores | Chrome |
| Sistemas Operativos | Windows |

# **6. CRITERIOS DE ENTRADA Y SALIDA**

## **6.1 Criterios de Entrada**

Las funcionalidades deben estar desplegadas en el ambiente de QA y haber sido probadas manualmente.

El framework de pruebas está instalado y listo para la ejecución

El ambiente de QA está disponible.

Los defectos críticos encontrados durante las pruebas manuales han sido resueltos y cerrados.

## **6.2 Criterios de Salida**

Ejecución de todos los casos de pruebas automatizados

Se ha logrado la suficiente cobertura de los requerimientos y funcionalidades bajo pruebas

Ningún defecto de severidad alta se encuentra abierto.

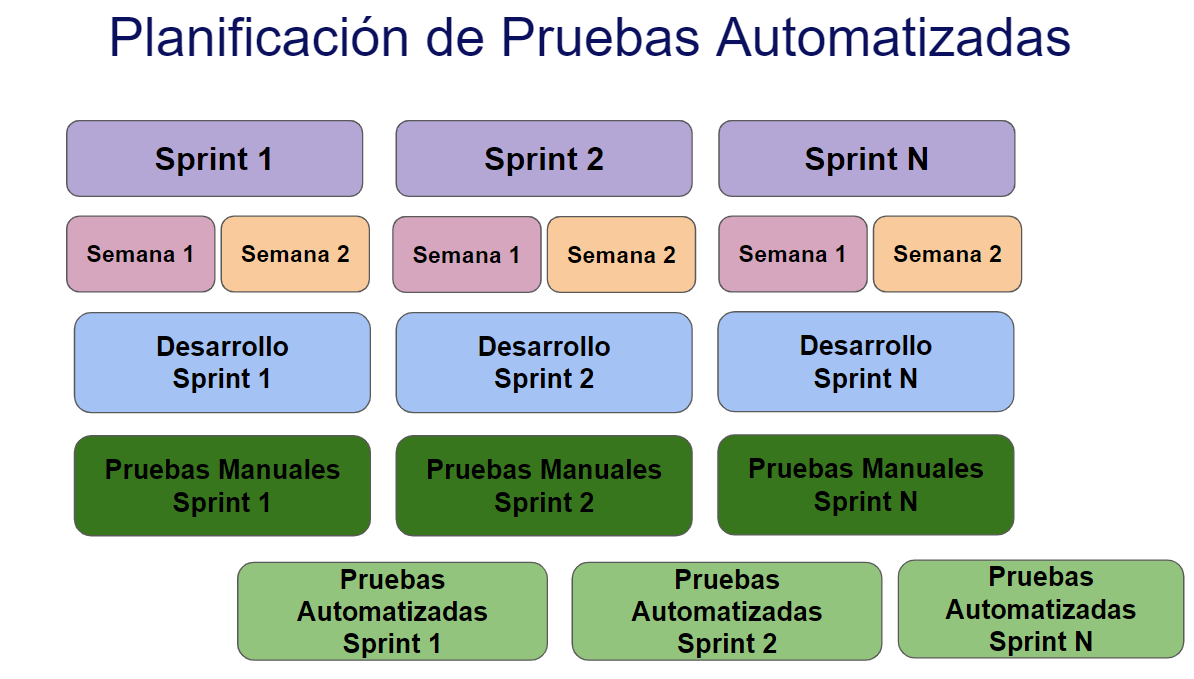
# **7. PLANIFICACIÓN DE EJECUCIÓN DE LAS PRUEBAS**

Lista de funcionalidades a ser automatizadas por Sprint

|  |  |  |
| --- | --- | --- |
| Sprint número | Funcionalidades | Comentarios |
| 1 | 1. Módulo My Info   * 1. sección Add Dependent      1. Assigned Dependents   1.1.1.1 Añadir un dependents con todos los datos   * + 1. Add Attachments   1.1.2.1 Añadir un attachement cargando un archivo y agregando comentario. | * + - 1. **CA**: Al dar clic en el botón sabe, debe mostrase el mensaje:   **Successfully Saved**   * + - 1. **CA:** Al dar clic en el botón Upload, se debe mostrar el mensaje:   **Successfully Saved** |
| 2 |  |  |

Las pruebas de automatización normalmente comenzarán en la segunda semana del Sprint (de 2 semanas).

Es necesario que las funcionalidades a automatizar se desarrollen, implementen y prueben manualmente para que tengan un nivel determinado de estabilidad cuando comienzan las tareas de automatización.



## **7.1 PLANIFICACIÓN DE LAS PRUEBAS DE REGRESIÓN**

Las suites de regresión se ejecutarán al final de cada Sprint (antes de la Revisión del Sprint), al realizarse un cambio o por solicitud de los Clientes, Product Owner y Project Manager.

# **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.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente