**Prueba Técnica Automatizador Choucair**

Se realizará la prueba asignada por la empresa choucair testing, a través del IDE Intellij IDEA implementando el gestor de proyectos Gradle, a continuación, se mencionará cada una de las herramientas implementadas para lograr el objetivo:

Se utilizará como patrón de diseño screenplay porque fomenta buenas prácticas para la escritura de pruebas y además permite dividir por pequeñas secciones el código y con esto tener un ambiente más controlado para la escritura de pruebas, cabe resaltar que al utilizar esta estrategia se mejora la mantenibilidad del código

Se utilizará la herramienta de cucumber como intérprete de lenguaje Gherkin (de negocio) a lenguaje java (Código)

Serenity: Herramienta que nos proporcionará los reportes de los resultados de las pruebas

Java como lenguaje de programación base para la escritura del código

Navegador en el que se realizó la automatización: Google Chrome versión 85.0.4183.83

**Historia de usuario 1:**

­­­­­Yo como usuario de la plataforma deseo interactuar con la sección de post para la creación y edición de este.

**Historia de usuario 2:**

Yo como usuario de la plataforma deseo interactuar con el módulo de Páginas para la creación y edición de esta.

**Criterios de aceptación:**

° Loguearse en la página wordpress.

° Crear y publicar un post.

° Editar un post ya existente.

° Modificar un post ya existente por medio de edición rápida.

° Crear una página

° Editar una página ya existente.

° Modificar por edición rápida una página ya existente

**Plan de pruebas:**

**Alcance:**

Para realizar las pruebas se hará en el navegador de Google Chrome versión 85.0.4183.83 en sistema operativo Windows 10, se probará hasta que el usuario visualice el post.

**Se verificará que:**

° Se encuentre el apartado de iniciar sesión

° Contenga el módulo de ‘Post’

° exista el botón ‘Add new’ en el módulo de post

° Contenga las opciones de ‘Edit’ y ‘Quick Edit’ en el módulo de post

° Contenga el módulo de ‘Páginas’

° exista el botón ‘Add new’ en el módulo de Páginas

° Contenga las opciones de ‘Edit’ y ‘Quick Edit’ en el módulo de Páginas

**Se validará que:**

° La plataforma permita crear publicaciones y luego validar que fue publicada correctamente

° Se permita la edición y edición rápida de una publicación

° La plataforma permita crear páginas y luego validar que fueron creadas exitosamente

° Se permita la edición y edición rápida de una página

**Fuera del alcance:**

No se probará:

º En otro navegador que no sea Google Chrome en la versión 85.0.4183.83

º En otro sistema operativo que no sea Windows 10

º Otras funcionalidades no descritas en las historias del usuario

**Estrategia:**

Para la página de wordpress inicialmente se realizará un smoke test con el fin de validar que cargue correctamente,

Primeramente, se empezará por diligenciar las credenciales proporcionadas en el apartado login de wordpress, para luego ingresar al módulo de post donde se creará una publicación, para luego validar que esta haya sido publicada correctamente, se continuará con la edición de esta publicación modificando uno de los textos, para finalmente certificar que la plataforma permita realizar ediciones rápidas modificando el título de la publicación.

**Prerrequisitos:**

° Acceso al navegador Google Chrome versión 85.0.4183.83

° Acceso al portal

° Documento de desarrollo

**Datos:**

° Credenciales de la plataforma

**Equipo de trabajo:**

Analista de certificación: Laura Mejía Serna

**Mejoras que se le pueden hacer a la plataforma:**

Debe de haber mejora en cuanto a persistencia de la información por lo que se vuelven temporales las publicaciones y las páginas.

Mejorar el diseño del front para mantener unas buenas practicas dentro de los nodos

**Solución Preguntas:**

**Conceptos programación:**

1. Principio Solid:

Es un conjunto de principios y buenas practicas que ayudan al desarrollo de las aplicaciones, se aplican con la finalidad de hacer el código más limpio, mantenible y escalable al futuro

Single Responsability = ‘cada clase y método hace una única función’,

Open / Close = ‘abierto a extensiones y usos y cerrado a modificaciones’,

Liskov Sustitution = ‘herencia, una clase hijo se debe comportar como padre’,

Interface Segregation = ‘implementar un solo método por interfaz’,

Dependency Inversion / Inyection = ‘recibir objetos ya creados en una tarea, no poner new dentro de la tarea’

¿Cómo aplicamos Solid en el reto?:

En el patrón de diseño implementado (Screenplay) evidenciamos los principios solid de la siguiente manera

ºSingle Responsability: Gracias a la estructura que el patrón nos ofrece se puede evidenciar que cada una de las carpetas y clases están definidas para cumplir una única funcionalidad teniendo como beneficio un código más limpio, la mantenibilidad de este y la escabilidad a futuro

ºOpen/Close: Podemos evidenciarlo en las clases creadas para Tasks y Questions, cuando se implementa la clase Task y Questions aplicamos el principio por que estas clases están abiertas a extenderse, pero cerrada a la modificación por lo que podemos hacer uso de los métodos de estas

ºLiskov sustitution: Se evidencia también en las clases mencionadas anteriormente, gracias a las clases implementadas para crear las Task o Questions podemos aplicar el principio sobrescribiendo uno de los métodos implementados y a la hora de llamar la clase en el step va a saber diferenciar el método con su lógica.

2.Patron Singleton

Es un patrón que se implementa con el fin de evitar una instanciación excesiva de objetos de una misma clase, lo que busca es que solamente exista un objeto propio de la clase junto con un método estático que nos permitirá llamar la instancia cada que lo necesitemos.

3.Patron FIRST

Es un principio que contiene las características que deben de tener los test unitarios y cumplir con la calidad de este:

Rápido: Indica la posibilidad de ejecutar una cantidad de pruebas en cuestión de segundos y detectar bugs de manera rápida y sencilla

Independiente: Cada prueba unitaria debe ser totalmente independiente de la otra, el resultado de una de ellas no debe alterar ninguna de las otras pruebas, nos ayuda a ganar tiempo en las regresiones

Repetible: Las pruebas no pueden tener dependencia de un servidor por lo que el resultado debe ser el mismo para todos estos

Auto evaluable: Pueden ejecutarse de forma automática, mientras se esté realizando otra tarea sin preocuparnos por dicha ejecución

Oportuno: Las pruebas deben de realizar en la etapa más temprana del software, en caso de que hallan pruebas de regresión y no dejarlo para lo último cuando ya se va a subir a producción

4.Patron AAA

Es un patrón diseñado para pruebas unitarias, se organiza de la siguiente manera:

º Organizar/Iniciativa: Se define con que objetos y el valor de los datos se realizarán el Test

º Actuar: Se hace la llamada al método para probar con los parámetros asignados anteriormente en la iniciativa

º Confirmar/Comprobar: Comprueba que el Test se comporte como se tenía previsto anteriormente, sino cumple el framework marcara con rojo cada falsa expectativa

5.Pull Request

como lo dice la palabra es una petición de validación, consiste en la integración de cambios a un repositorio principal, realizado por el resto del equipo desarrollador generando aprobaciones o retroalimentaciones

6.Release Train

Es un orden que se le da a las entregas de software, consiste en un calendario al cual tiene acceso todos los miembros de un equipo, con anterioridad se comprometen con tareas a entregar representando un compromiso para estos, el calendario debe ser cumplido y se entrega lo que se esta listo para ser integrado

7.Quality Gates:

Es una solución que sirve para el seguimiento de la calidad de datos arrojando Estadísticas con el fin de arreglar pendientes dentro de un reporte, que se basan en un análisis diseñado

8.Diferencias entre servicios SOAP/REST

**Soap:** Es un protocolo que permite la comunicación de aplicaciones diseñadas en diferentes lenguajes y plataformas, su tecnología es orientada a sectores como

Financiero, telecomunicación y vías de pago

**Rest:** Es un conjunto de principios arquitectónicos que se ajusta a las necesidades de servicios web y las aplicaciones móviles ligeros, su tecnología es orientada a sectores como

Medios sociales, chat web, Mobile

la diferencia entre estos dos protocolos es que REST se considera más rápido para servicios web, mientras que SOAP tiene requisitos más específicos