

APLICACIÓN DE MÉTODOS PARA LA AUTOMATIZACIÓN DE PRUEBAS DE SOFTWARE
EN LA EMPRESA SQA S.A, MEDELLÍN ANTIOQUIA

Autor

GISSELL VALENTINA MONTAÑEZ VERA

PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURA



UNIVERSIDAD DE PAMPLONA

PAMPLONA, NOVIEMBRE de 2021

APLICACIÓN DE MÉTODOS PARA LA AUTOMATIZACIÓN DE PRUEBAS DE SOFTWARE
EN LA EMPRESA SQA S.A, MEDELLÍN ANTIOQUIA

Autor

GISSELL VALENTINA MONTAÑEZ VERA

Director

EDWIN MAURICIO SEQUEDA ARENAS

Ingeniero Electrónico

Codirector

GUSTAVO QUIJADA MACUART

Ingeniero Telecomunicaciones

PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURA



UNIVERSIDAD DE PAMPLONA

PAMPLONA, NOVIEMBRE de 2021

NOTA DE ACEPTACIÓN

Firma del presidente del Jurado.

Firma del Jurado 1.

Firma del Jurado 2.

Pamplona, Noviembre del 2021.

DEDICATORIA

Especialmente quisiera dedicarle mi trabajo de grado a mi familia, mi pareja y mi hijo porque sin su ayuda no hubiera sido posible todo este proceso, son el pilar y la motivación más importante en mi vida, y este trabajo solo es una pequeña semilla germinada que quiere brotar al máximo para hacerlos felices a ellos.

También quisiera dedicar este trabajo a Dios y mamita María, que siempre han sido mi bastón y fuerza frente a todas las dificultades que se me han presentado.

AGRADECIMIENTOS

En primera instancia quisiera agradecerle a Dios por haber colocado en mi camino cada una de estas oportunidades, por estudiar en una universidad prestigiosa y porque gracias a él nunca me faltó nada.

De igual manera a mi familia por apoyarme en todo momento, a mi mamá Mary Alejandra Vera Mantilla por ser mi apoyo incondicional y mi gran fortaleza, a mi abuela Mercedes Mantilla Espinosa por ser mi complemento y por estar ahí en cada momento que lo necesite, a mi angelito en el cielo mi papá Roger Montañez por cuidar de mí y ser luz en mi camino, a mi hermana Fabiana Vera por ser mi mayor motivación. Y a mi familia en general por siempre haber confiado en mí y por depositar todas sus esperanzas por verme triunfar. Del mismo modo a mi novio porque gracias a él logre superar cualquier dificultad que se me atravesó en el camino, y lograr darme felicidad en medio de todo.

Por otra parte, quisiera agradecerle a la Universidad de Pamplona por formarme integralmente como Ingeniera en Telecomunicaciones, del mismo modo y de manera muy especial a mi director Mauricio Sequeda y codirector Gustavo Quijada por el apoyo indispensable en todo el proceso de trabajo de grado, por depositar su confianza en mí. Así mismo a cada uno de los docentes que hicieron parte de todo mi proceso de formación profesional y a mis compañeros.

Por último a la empresa SOFTWARE QUALITY ASSURENCE por permitirme realizar mis prácticas empresariales y llenar mi mente de nuevos conocimientos, a mi Jefe inmediato Esteffany Garcés por su disposición, amabilidad y gran liderazgo.

RESUMEN

En la actualidad las empresas prestadoras de servicios de software han encontrado que un gran causante del proceso de mantenimiento sobre las aplicaciones es el proceso de pruebas. El presente anteproyecto tiene como propósito fundamental el apoyo a una aplicación de métodos para la automatización de pruebas de software en la empresa Software Quality Assurance.

Una de las principales necesidades que se puede visualizar en las solicitudes de proyectos de software es el tiempo de entrega, ya que este es un factor importante al momento de dar una solución estratégica en cada proceso. Durante este trabajo se pretende apoyar el proceso de pruebas de automatización para satisfacer las necesidades del cliente.

Palabras claves: Automatización de pruebas, calidad del servicio, pruebas de software.

ABSTRACT

Currently, software service providers have found that a major cause of the maintenance process on applications is the testing process. The main purpose of this preliminary draft is to support an application of methods for the automation of software tests in the Software Quality Assurance company.

One of the main needs that can be seen in software project requests is delivery time, since this is an important factor when providing a strategic solution in each process. During this work it is intended to support the automation testing process to meet customer needs.

Keywords: Test automation, quality of service, software testing.

TABLA DE CONTENIDOS

INTRODUCCIÓN	5
CAPITULO I	6
1. MARCO REFERENCIAL	6
1.1 MARCO CONTEXTUAL	6
1.1.1 INFORMACIÓN DE LA EMPRESA	6
1.1.2 LEMA	7
1.1.3 IMAGEN CORPORATIVA	7
1.1.4 UBICACIÓN GEOGRAFICA.....	7
1.1.5 ¿QUIÉN ES SQA?	7
1.1.6 ¿QUÉ HACE SQA?	7
1.1.7 ¿QUÉ SE QUIERE EN SQA?.....	8
1.1.8 VALORES DE SQA	8
1.2 MARCO TEORICO	8
1.3 ESTADO DEL ARTE.....	10
1.3.1 INVESTIGACIÓN INTERNACIONAL.....	10
1.3.2 INVESTIGACIÓN NACIONAL	11
1.3.3 INVESTIGACIÓN LOCAL.....	12
1.3.3.1 ANTECEDENTE LOCAL N°1.....	12
1.3.3.2 ANTECEDENTE LOCAL N°2.....	13
CAPITULO II	14
2. INFORMACIÓN GENERAL	14
3. PLANTEAMIENTO DEL PROBLEMA.....	16
4. OBJETIVOS	19
4.1 OBJETIVO GENERAL	19

	9
4.2 OBJETIVOS ESPECÍFICOS	19
5. JUSTIFICACIÓN	19
CAPITULO III.....	21
6. METODOLOGÍA.....	21
6.1 FUNDAMENTACION TEORICA	21
6.2 AUTOMATIZACIÓN DE PRUEBAS DE SOFTWARE	23
6.2.1 DESARROLLO DE METODOLOGÍA POR FASES, MEDIANTE EL CICLO DE VIDA DE PRUEBAS	23
6.2.1.1 CONOCIMIENTO Y ANALISIS DE REQUISITOS	24
6.2.1.2 PLANEACIÓN DE PRUEBAS	25
6.2.1.3 DISEÑO	25
6.2.1.4 EJECUCIÓN DE LA PRUEBA.....	26
6.2.1.5 CERTIFICACIÓN O CIERRE	27
6.3 PRUEBAS MANUALES	28
6.4 DESCRIPCIÓN DE ACTIVIDADES.....	33
CAPITULO IV	33
7. RESULTADOS OBTENIDOS	33
7.1 PLAN DE DESARROLLO, PROYECTO AUTOMATIZACIÓN	36
7.1.1 INFORMACIÓN DEL PROYECTO	37
7.1.2 ALCANCE DEL DESARROLLO DE LA AUTOMATIZACIÓN DE PRUEBAS	37
7.1.3 FUERA DEL ALCANCE DEL DESARROLLO DE AUTOMATIZACIÓN DE PRUEBAS	38
7.1.4 REQUISITOS Y RESTRICCIONES.....	39
7.1.5 VISTA GENERAL DEL PROYECTO	40
7.1.5.1 HERRAMIENTAS.....	40

	10
7.1.5.2 ANALISIS DE RIESGOS.....	41
7.1.5.3 ARQUITECTURA DEL DESARROLLO DE AUTOMATIZACIÓN.....	42
7.1.5.4 CRONOGRAMA DE HORAS	42
7.1.6 ORGANIZACIÓN DEL PROYECTO DE AUTOMATIZACIÓN.....	43
7.1.6.1 ROLES Y RESPONSABILIDADES.....	43
7.1.6.2 AMBIENTE.....	43
7.1.6.3 BACKUP DE LA INFORMACIÓN	44
7.2 PRUEBAS AUTOMATIZADAS	44
7.2.1 PROCEDIMIENTO.....	44
7.2.2 HERRAMIENTAS	48
CONCLUSIONES	49
RECOMENDACIONES	50
REFERENCIAS BIBLIOGRÁFICAS	50
ANEXOS.....	53

LISTA DE TABLAS

<i>Tabla 1 Descripción de actividades</i>	33
<i>Tabla 2 Listado de cursos</i>	34
<i>Tabla 3 Lista de semilleros</i>	36
<i>Tabla 4 Información general del proyecto</i>	37
<i>Tabla 5 Alcances</i>	38
<i>Tabla 6 Requisitos y restricciones</i>	39
<i>Tabla 7 Análisis de riesgos</i>	41
<i>Tabla 8 Cronograma en horas</i>	42
<i>Tabla 9 Roles y responsabilidades</i>	43
<i>Tabla 10 Ambiente</i>	43
<i>Tabla 11 Anexos</i>	53

LISTA DE FIGURAS

<i>Figura 1 Imagen corporativa SQA</i>	7
<i>Figura 2 Diagrama causa-efecto</i>	18
<i>Figura 3 Esquema metodologico</i>	21
<i>Figura 4 Cursos y semilleros</i>	22
<i>Figura 5 Ingreso página automatizar</i>	29
<i>Figura 6 Login</i>	29
<i>Figura 7 Selección “BMC”</i>	29
<i>Figura 8 Click en menú</i>	30
<i>Figura 9 Click en “Motor de Registro”</i>	30
<i>Figura 10 Click en “Registro”</i>	30
<i>Figura 11 Click en “Prueba Auto-Ítem1-Subítem”</i>	31
<i>Figura 12 Click en “Registro por Demanda”</i>	31
<i>Figura 13 Llenado de Campos</i>	32
<i>Figura 14 Click en “Guardar”</i>	32
<i>Figura 15 Validación</i>	32
<i>Figura 16 BDD+Cucumber</i>	42
<i>Figura 17 Creación Nuevo Proyecto</i>	45
<i>Figura 18 Creación build.gradle</i>	45
<i>Figura 19 Creación serenity.properties</i>	46
<i>Figura 20 Versión Chrome Driver</i>	46
<i>Figura 21 Chrome Driver en la estructura del proyecto</i>	46
<i>Figura 22 Creación de carpetas y clases</i>	47
<i>Figura 23 “Datos” en la estructura del proyecto</i>	47

INTRODUCCIÓN

Actualmente la sociedad del mundo en general se ha vuelto “software-dependiente”, porque este desarrollo tecnológico tiene cada vez más un fuerte impacto en las operaciones vitales de la vida humana, tales como la medicina, la aeronáutica, la investigación espacial, las telecomunicaciones y la protección de datos, entre otras. Por eso es importante abordar los problemas de calidad relacionados tanto con el proceso del desarrollo de software como con el producto mismo. (Serna, 2012)

La automatización de pruebas reduce el esfuerzo dedicado a las pruebas en productos de software que ya están estables en la mayoría de funcionalidades y en aplicaciones que se encuentran en continuo mantenimiento. Adicionalmente, la automatización permite agilizar las pruebas que son repetitivas.

En el presente documento se pretende plasmar una minuciosa explicación de la automatización de pruebas de software ya que es la práctica que permite controlar la ejecución de un producto software de manera automática, comparando los resultados obtenidos con los resultados esperados. Esta práctica permite no solo realizar pruebas repetitivas y de regresión dentro de un proceso sino probar ejecuciones que manualmente serían difíciles de controlar. Sin embargo, la automatización de pruebas funcionales, no tiene como objetivo eliminar el testing manual, sino ayudarlo y complementarlo. Si automatizamos aquello que puede ser automatizado, entonces los testers tendrán más tiempo y podrán centrar sus esfuerzos en resolver los aspectos más complejos que se le escapan a la automatización de pruebas unitarias o repetitivas. (Andrea, Reflexionamos acerca de la automatización de pruebas funcionales de software. Qué es exactamente esta práctica, cuándo aplicarla y por qué., 2018)

CAPITULO I

Contextualización

En este primer capítulo se pretende plasmar a través de diferentes conceptos, toda la información recolectada para entender la automatización de pruebas de software, por tal motivo es de suma importancia conocer a SQA S.A ya que esta prestigiosa organización dentro de su filosofía de trabajo se destaca por ser una de las empresas número uno de Colombia en la automatización a través del software, de igual manera conceptos importantes de esta temática, y a su vez diferentes estudios citando a autores que se han enfocado en este entorno.

1. MARCO REFERENCIAL

1.1 MARCO CONTEXTUAL

1.1.1 INFORMACIÓN DE LA EMPRESA

SQA (Software Quality Assurance) es una organización moderna, experta y ágil, orientada a las personas cuyo principal objetivo es actuar frente a los contextos que se le presentan y necesidades del mismo. SQA es un conjunto de actividades planificadas y sistemáticas, donde se evalúa la calidad productos de software a los estándares, procesos y procedimientos; permitiendo así asegurar la calidad de dichos procesos y productos. (SQA, 2021)

En esta empresa se prestan servicios que ayudan a que se desarrolle mejor el software. Esto se logra mediante el aseguramiento de la calidad, buenas prácticas y metodología (propia y adaptable).

Presta un servicio especializado en aseguramiento de la calidad del software mediante un modelo de negocio flexible y que se adapta a las necesidades de los

clientes para que así su experiencia sea mejor. SQA cuenta con aceleradores de negocio y de gestión de conocimiento para generar mayor confianza.

1.1.2 LEMA

“Que el mundo funcione mejor a través del software”.

1.1.3 IMAGEN CORPORATIVA



Figura 1 Imagen corporativa SQA

1.1.4 UBICACIÓN GEOGRAFICA

Sedes: Medellín y Bogotá.

- Carrera 30 No. 7AA-207 Torre Scaglia Barrio Poblado Medellín.
- Carrera 16 No. 97-46 Bogotá.

1.1.5 ¿QUIÉN ES SQA?

- SQA es aliado del cliente buscando ayudar a que las soluciones a sus problemas sean confiables, seguras y de calidad.
- Son expertos en testing y pretenden ser expertos en el negocio de cada uno de sus clientes.
- Entienden lo que el cliente necesita y quieren aportar ideas para que cumplan los objetivos de sus negocios.

1.1.6 ¿QUÉ HACE SQA?

- SQA hace que el mundo funcione mejor a través del software.
- Aporta a que la producción de software sea cada vez mejor.

1.1.7 ¿QUÉ SE QUIERE EN SQA?

- SQA pretende ser identificado como un generador de confianza para las soluciones que utilizan software.
- Quieren que el cliente los vea como aliados de su negocio y vean que les generan valor agregado en su servicio.
- Desean tener un buen dialogo con el cliente para lograr entender sus objetivos y riesgos.

1.1.8 VALORES DE SQA

La forma en la que prestan sus servicios está basada en 4 valores fundamentales:

- Adaptabilidad y flexibilidad.
- Compromiso.
- Honestidad.
- Confianza.

1.2 MARCO TEORICO

En el contexto actual, la AUTOMÁTICA se define como la Ciencia y Técnica de la automatización, que agrupa el conjunto de las disciplinas teóricas y tecnológicas que intervienen en la concepción, la construcción y el empleo de los sistemas automáticos. La automática constituye el aspecto teórico de la cibernética. Está estrechamente vinculada con las matemáticas, la estadística, la teoría de la información, la informática y técnicas de la ingeniería. (Moreno, 2012)

El proceso de automatización se realiza para agilizar las actividades de testeo de una aplicación, empleando un proceso selectivo de las pruebas a automatizar como las pruebas que sean más frecuentemente aplicadas o que sean más difíciles de testear de manera manual, ayudando a controlar y probar una aplicación en continuo cambio, de esta manera creas un primer criterio por el cual seleccionar una pruebas y otras no, para crear un orden sobre las pruebas a automatizar se suelen reconocer las pruebas que afecten a las zonas más importantes de la aplicación, como la facturación o el sistema de control de usuarios. (Esteve Ambrosio, 2015)

En otras palabras la automatización de pruebas es la práctica que permite controlar la ejecución de un producto software de manera automática, comparando los resultados obtenidos con los resultados esperados.

Dustin, Rashka y Paul afirman que la popularidad de la automatización de las pruebas también se debe a la aplicación de desarrollo rápido de aplicaciones (RAD) y a su popularidad. Para cumplir la mayor cantidad de actividades relacionadas con las pruebas se deben utilizar herramientas automatizadas, porque a menudo las manuales son laboriosas y propensas a errores, y simplemente no pueden competir con la calidad de la prueba automatizada, especialmente cuando hay que tener en cuenta el cronograma del proyecto. (Edgar Serna, 2021)

La integración continua es una práctica que comienza con la organización de los proyectos en una estructura de directorios adecuada para establecer el orden de ejecución de los componentes de un proyecto (incluyendo casos de prueba), y de esta manera facilitar la construcción correcta del software cuando se ejecuta el proceso de integración, logrando que el mismo sea transparente para el equipo de desarrollo. Además permite a los desarrolladores de software visualizar el estado de las construcciones que se generan en el proceso de integración identificando posibles incidencias, y de esta manera evitando que las

mismas generen errores futuros en el proyecto y se dificulte su resolución. (Alicia Salamon, 2016)

A razón de esto es importante resaltar que las pruebas funcionales, o testing de software tienen por objetivo validar cuando el comportamiento del software probado cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario. Las funciones son probadas ingresando las entradas y examinando las salidas. La estructura interna del programa raramente es considerada. A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra. (Esteve Ambrosio, 2015)

De ahí la importancia de escuchar al cliente, pues es la plataforma directa para garantizar un buen trabajo donde se lleve a cabo lo que él desea plasmar para su empresa y/o su sistema de trabajo, es por esto que el resultado final será la entrega de ese artefacto (el producto tangible resultante del proceso de desarrollo de software).

1.3 ESTADO DEL ARTE

Históricamente, la automatización surgió para reducir el esfuerzo humano requerido en actividades que podrían ser replicadas por un sistema o máquina programable. Al automatizar pruebas de software se persigue el objetivo de simplificar el trabajo dispendioso, repetitivo o complejo, haciéndolo efectivo y más productivo. De esta manera, es posible ahorrar energía, tiempo y costos, al tiempo que libera a las personas para que se concentren en otras tareas. (Rodríguez, 2018)

1.3.1 INVESTIGACIÓN INTERNACIONAL

“Automatización y Gestión de las Pruebas Funcionales usando Herramientas Open Source”.

Ignacio Esmite, Mauricio Farías, Nicolás Farías, Beatriz Pérez afirman que la automatización de las pruebas funcionales reduce significativamente el esfuerzo

dedicado a las pruebas de regresión en productos que se encuentran en continuo mantenimiento. La automatización de las pruebas debe ser considerada un proyecto en sí mismo con objetivos definidos.

En su artículo se presenta una metodología y el conjunto de herramientas open source utilizado para la automatización de las pruebas funcionales en productos con interfaz Web. Este conjunto de herramientas está compuesto por: Selenium, Eclipse y extensiones de Mozilla Firefox como son Firebug, XPath Checker y XPather. Se describe la experiencia de utilizar la metodología en un proyecto de prueba específico y se concluye la factibilidad para la automatización de las pruebas siguiendo las actividades y el conjunto de herramientas definidos. Si bien las herramientas asisten en las pruebas automatizadas, no brindan soporte para la organización de los artefactos del proyecto: scripts, documentos y reportes de ejecución. (Ignacio Esmite, 2016)

1.3.2 INVESTIGACIÓN NACIONAL

“Foro: El futuro de las pruebas automáticas de software en Colombia”

Los contextos actuales de los procesos de desarrollo han llevado a que los ciclos de entrega de productos de software sean más cortos y con mayor calidad. Factores como la presión continua por entregas rápidas, fragmentación en dispositivos, dominios de aplicación complejos, usuarios más exigentes y especializados, influyen directamente en la forma como hacemos y probamos aplicaciones de software. Por tal motivo los métodos ágiles y los procesos de automatización se han convertido en una necesidad para los equipos de desarrollo que quieren mantenerse vigentes en el mercado, sin dejar de lado la calidad, y aumentando el retorno de la inversión en sus compañías. Sin embargo, el miedo a

que los humanos seamos reemplazados por mecanismos de automatización es latente, en especial en los equipos dedicados a las pruebas de software.

Este foro busca romper con esos miedos, mostrando ejemplos reales de automatización de pruebas en industrias colombianas, presentando el estado del arte a nivel mundial, y discutiendo acerca del rol y responsabilidades del ingeniero de pruebas moderno y la necesidad de las pruebas de software como una actividad transversal a todo el proceso de desarrollo. Para esto, el foro contará con charlas gratuitas de expertos nacionales, y con talleres de bajo costo para que los asistentes tengan la oportunidad de experimentar con herramientas de automatización de pruebas existentes en el mercado (open source y comercial). (Andes, 2019)

1.3.3 INVESTIGACIÓN LOCAL

1.3.3.1 ANTECEDENTE LOCAL N°1

“Integración de Automatización y distribución de pruebas utilizando Microsoft Azure y Contenedores Docker en la empresa DIGITAL WARE”.

En el presente trabajo se expone el desarrollo de un sistema de integración de automatización y distribución de pruebas que permite ejecutar pruebas utilizando las tecnologías Microsoft Azure y Docker, dando solución a dos de las principales problemáticas existentes actualmente en el área de automatización de Digital Ware y estas son el tiempo de ejecución de las pruebas y el requerimiento de una pantalla de escritorio activa para ejecutarlas. Lo que se plantea es el despliegue de varios contenedores Docker que posibiliten la ejecución de pruebas aplicando el paralelismo dentro de un mismo dispositivo físico, cada uno de los contenedores Docker contiene una máquina

virtual que será configurada con su respectivo agente de prueba y herramientas necesarias para la ejecución de los test de manera remota, liberando las máquinas de desarrollo del área de automatización de la compañía y permitiendo que corran todas las pruebas dentro de un solo equipo servidor en los diferentes contenedores, siendo el medio de integración la plataforma Azure. Al finalizar el proyecto se logró la ejecución de pruebas desplegadas desde Azure dentro de los contenedores Docker, llevando a cabo una ejecución sin la renderización de la capa de aplicación y directamente dentro de un mismo equipo de cómputo. (Díaz, 2021)

1.3.3.2 ANTECEDENTE LOCAL N°2

“Desarrollo de demo portable y Automatizado para las aplicaciones KACTUS-HCM fundamentado en contenedores sobre Docker para la empresa Digital Ware”

El presente trabajo se llevó a cabo en la empresa DigitalWare, con el fin de ofrecer una alternativa de la elaboración de los Demo de las aplicaciones Kactus-HCM que ofrece la compañía. La elaboración del Demo actual se Basa en el uso de máquinas virtuales para la ejecución de dos servidores, el primero para el despliegue de las aplicaciones web y el segundo para almacenar y ofrecer el servicio de Base de Datos, esto hace que no se aprovechen de manera óptima los recursos de la máquina anfitrión al verse obligada a ejecutar varios sistemas operativos y sus servicios simultáneamente, además de su instalación tan tediosa y demorada. Con este trabajo se pretende optimizar el tiempo de la creación de las aplicaciones Demo, aprovechar al máximo los recursos de la máquina anfitrión y que el Demo sea portable, lo que permitiría su ejecución en cualquier computadora con Windows 10 y Docker Desktop instalado. (Castillo, 2020)

CAPITULO II

Problemática

En este segundo capítulo se desglosa todo el planteamiento del problema partiendo de una explicación fundamentada en la automatización de pruebas de software para entender de manera detallada las dificultades que de allí se desglosan, para poder tratarlas y mejorarlas, buscando así conocer las razones por las que se realiza este proyecto.

2. INFORMACIÓN GENERAL

La automatización de proceso es un instrumento tecnológico para la optimización de las pruebas de un negocio, permitiendo a los analistas de calidad interactuar con los diferentes procesos a través de las pruebas de regresión. Además, según Ernst & Young (2018) “La automatización robótica de procesos (RPA) técnica que nos va a que permitir configurar “Robot Software” es decir, que todas aquellas tareas manuales podrán ser automatizadas o semi-automatizadas de tal forma que se puede integrar con los sistemas actuales de la organización.” (Haro, 2021)

Por esto Software Quality Assurance (SQA) tiene la capacidad de automatizar en diferentes arquitecturas y herramientas que tengan la adherencia suficiente para operar con la aplicación objeto de la automatización. El servicio de automatización ofrecido por SQA incluye la automatización de pruebas o procesos mediante herramientas comerciales y el desarrollo propio basado en herramientas Open Source. Es por esto que esta organización ofrece a todos sus clientes, una alternativa altamente eficiente que le permitirá ejecutar pruebas de regresión, teniendo un mayor cubrimiento de pruebas en menor tiempo. (SQA, 2021)

En la práctica de las pruebas de software se definen una serie de conceptos claves que permiten distinguir claramente los elementos y fases dentro de su proceso. Han sido propuestos por instituciones como la IEEE o el Departamento de Defensa de Estados Unidos, los que se han encargado de formalizar la adopción de las pruebas como parte de los procesos

que permiten medir la calidad de los productos de software. Se destacan los siguientes conceptos:

- **Validación:** Es el proceso de evaluación del software al término de su desarrollo para asegurar que cumple con el uso previsto que motivó su construcción.
- **Verificación:** Es el proceso que busca determinar si los productos de software cumplen en cada fase del desarrollo con los requisitos establecidos en la fase anterior.
- **Defecto de software:** Corresponde a un defecto estático en el software e introducido durante su construcción.
- **Error de software:** Es un estado interno incorrecto que es causado por un defecto en el software.
- **Falla de software:** Corresponde a un comportamiento externo incorrecto respecto a los requerimientos u otra descripción del comportamiento esperado.
- **Pruebas:** Evaluación del software mediante la observación de su ejecución.
- **Depuración:** Proceso que permite encontrar un defecto producto de una falla.
- **Valores del caso de prueba:** Valores de entrada necesarios para la ejecución del software sobre el que se esté aplicando la prueba.
- **Resultado esperado:** Corresponde al resultado que genera la ejecución de la prueba si el software cumple con el comportamiento buscado.
- **Caso de prueba:** Está compuesto por los valores del caso de prueba, los resultados esperados y todos los valores necesarios para la ejecución y evaluación completa del software que se está probando.

La comprensión apropiada de estos conceptos facilitará la correcta aplicación de las pruebas en cada ámbito del proceso por parte de los ingenieros de pruebas, ingenieros de software y todo el equipo involucrado en el desarrollo del software. (Martín, 2014)

En algunos casos, el esfuerzo requerido en las pruebas de software es sobre un 50% del costo del desarrollo del software e incluso más si el desarrollo está orientado a actividades de misión crítica, por ejemplo, aplicaciones aeronáuticas o del área de la medicina. Debido a esto, uno de los objetivos de las pruebas de software es implementar su automatización tanto como sea posible, lo que permite disminuir costos, reducir errores humanos y ejecutar las pruebas de regresión de manera más fácil. La estrategia adoptada para automatizar las pruebas debe mantener el equilibrio entre los costos que reducirá y los asociados a la automatización. (Luciano Baresi, 2006)

3. PLANTEAMIENTO DEL PROBLEMA

A nivel mundial se puede evidenciar que la automatización de pruebas de software surgió como una alternativa para agilizar su ejecución, pues es el reemplazo de las pruebas manuales y a la vez se utilizan para mejorar la fiabilidad del producto y su calidad. Pero, aunque su estudio e investigación se inició hace casi 50 años, su progreso todavía no satisface la demanda por el mejoramiento de la calidad.

Muchas organizaciones que desarrollan software de manera ágil buscan adoptar el enfoque de entrega continua, el cual permite la liberación del software a producción en cualquier momento. Sin embargo, uno de los principales desafíos es mantener la calidad del mismo gestionando adecuadamente el tiempo que se invierte en las pruebas. En este sentido, se han reportado numerosos problemas relacionados con la calidad del software al adoptar este enfoque. (Maximiliano A. Mascheron, 2018)

La Latinoamérica se ha visto afectada con el pasar de los años por el bajo avance tecnológico pues esto se debe a que, en su mayoría está constituida por países

tercermundistas, la globalización va avanzando a medida que pasa el tiempo, sin embargo, actualmente en nuestro entorno tenemos ciertos atrasos que no permite el paso al mundo tecnológico en un 100%. Es por esto que se recurre más al recurso humano ya que los contextos actuales de los procesos de desarrollo han llevado a que los ciclos de entrega de productos de software sean más cortos y con mayor calidad. Factores como la presión continua por entregas rápidas, fragmentación en dispositivos, dominios de aplicación complejos, usuarios más exigentes y especializados, influyen directamente en la forma como hacemos y probamos aplicaciones de software. Por tal motivo los métodos ágiles y los procesos de automatización se han convertido en una necesidad para los equipos de desarrollo que quieren mantenerse vigentes en el mercado, sin dejar de lado la calidad, y aumentando el retorno de la inversión en sus compañías. Sin embargo, el miedo a que los humanos son reemplazados por mecanismos de automatización es latente, en especial en los equipos dedicados a las pruebas de software.

En Colombia algunas organizaciones aún no han podido adoptar al CI (Integración continua) y CD (entrega continua) completamente, y otras han encontrado muchos obstáculos. La mayor parte de los problemas reportados están relacionados principalmente con pruebas e integración. Como se mencionó anteriormente, esta disciplina soporta prácticas ágiles y reduce los tiempos de lanzamiento de nuevas versiones del producto al mercado, de semanas a solo horas. Sin embargo, de acuerdo a Prusak “la industria aún no ha cerrado el círculo cuando se trata de implementar un proceso completo de entrega continua”. Si bien la literatura contiene instrucciones y buenas prácticas de cómo adoptar CD, su implementación ha sido un desafío en la práctica. (Maximiliano A. Mascheron, 2018)

Mediante el siguiente diagrama Causa-Efecto se ilustrará una lista de problemas relacionados con la calidad de software, que han sido reportados en revisiones de la literatura, casos de estudio y artículos empíricos, cuyo objetivo era el estudio y la implementación de automatización de pruebas.

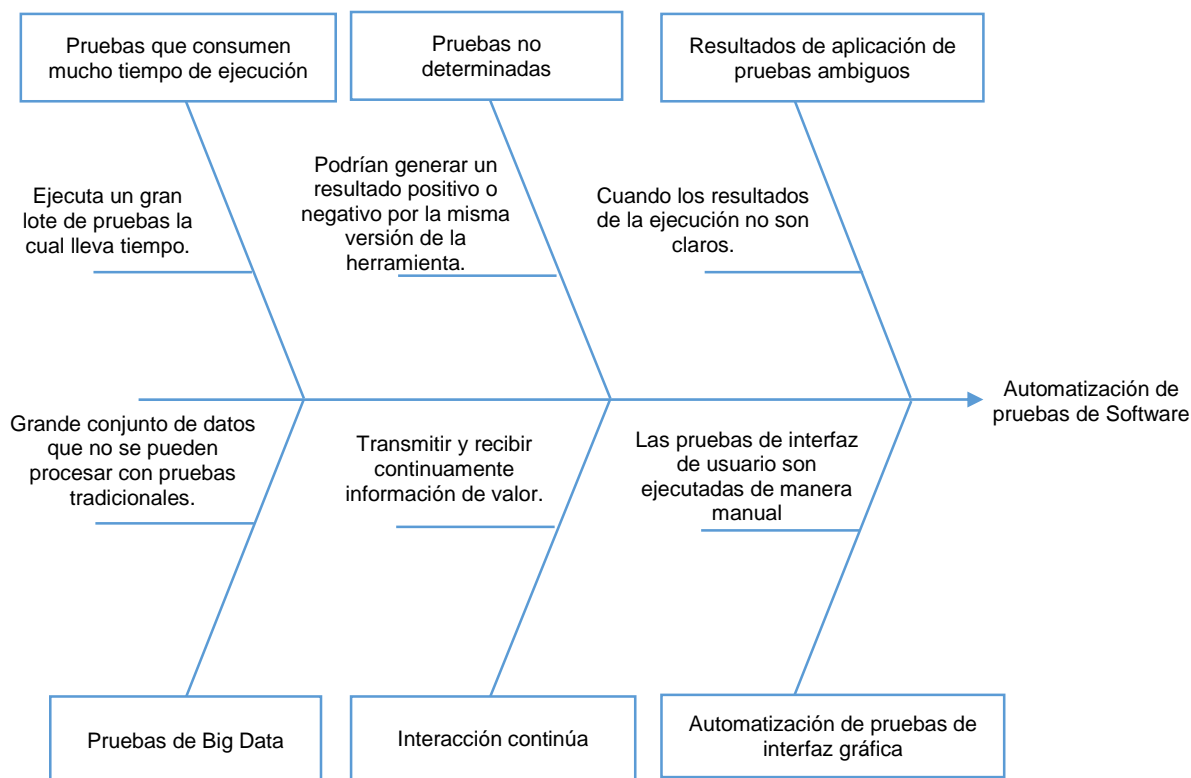


Figura 2 Diagrama causa-efecto

Las problemáticas anteriores que se presentan en el diagrama en cuanto a la automatización de pruebas de software, permite centrar su mirada en las dificultades que pueden perjudicar el desarrollo del proyecto. Cabe resaltar que existen diferentes problemas que van surgiendo con la ejecución del trabajo, pues con las constantes pruebas puede que allí se deriven más de las anteriormente nombradas.

Finalmente es importante mencionar que en los proyectos de automatización de pruebas en la empresa SQA S.A se trabaja en el mejoramiento continuo, velando por la calidad del servicio, el bienestar de sus usuarios, y de sus colaboradores. Contribuyendo en el cuidado del medio ambiente, siendo una empresa socialmente sostenible, garantizándoles a sus clientes calidad de servicio y buscando así la mejor relación de beneficio-costos.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar un compendio de pruebas funcionales aplicando métodos de automatización con la finalidad de agilizar los procesos de pruebas de software para un cliente de la empresa SQA S.A en Medellín.

4.2 OBJETIVOS ESPECÍFICOS

- Fundamentar la base de conocimiento en conformidad con las características del proyecto.
- Desarrollar las pruebas funcionales como parte fundamental en el aseguramiento de la calidad del producto de software del cliente a través de la aplicabilidad de pruebas de caja negra.
- Analizar los resultados de las pruebas funcionales basados en los criterios de aceptación del cliente para determinar la fiabilidad de las mismas.

5. JUSTIFICACIÓN

Para ayudar a las organizaciones, que buscan implementar pruebas de software automatizadas. Es importante tener en cuenta estos seis conceptos clave que permiten lograr una implementación exitosa. Estas son: conocer los requerimientos de automatización de pruebas; desarrollar la estrategia de pruebas automatizadas; probar el entorno de trabajo sobre el cual se realizarán las pruebas de software automatizadas; realizar seguimiento continuo de los

avances y aplicar los ajustes necesarios; implementar el proceso de pruebas de software automatizadas; y seleccionar a las personas idóneas para la ejecución del proyecto. Si alguno de estos pasos no está bien definido o logrado puede resultar en el fracaso de la implementación de las pruebas automatizadas. Si bien, no establecen un modelo de costos parametrizable, para cada paso describen que aspectos se deben considerar a la hora de determinar los costos y los puntos más importantes que ayudan a determinar el retorno sobre la inversión como una primera estimación de la automatización de pruebas. (Martín, 2014)

Dadas las características anteriores, la automatización del testeado de software permite reducir los costes de cualquier organización que necesite probar sucesivas versiones de un mismo producto. Inviertes una vez en crear esas pruebas, y las automatizas múltiples veces. De igual manera la automatización de pruebas aporta otros beneficios tales como:

- **Rapidez:** Las herramientas de testing automatizado ejecutan pruebas significativamente más rápido que los testers humanos.
- **Fiabilidad:** Las pruebas ejecutan precisamente las mismas operaciones cada vez que se ejecutan, eliminando el error humano.
- **Repetición:** Se puede testear cómo reacciona el software bajo repetidas ejecuciones de las mismas operaciones.
- **Programable:** Se pueden programar pruebas sofisticadas y complejas que muestren información oculta de la aplicación.
- **Reusabilidad:** Se pueden reusar los scripts con pruebas automatizadas, las funciones, etc... (Andrea, Automatización de Pruebas, 2018)

Los productos de software son construidos, de forma manual, mediante la definición de una serie de rutinas ajustadas a las reglas sintácticas establecidas en el lenguaje de programación que les da soporte. Su construcción también puede ser automatizada mediante paradigmas

como el de programación, el que se centra en la implementación y diseño de software reusable mediante el cual se pueda generar nuevo software en lugar de la creación de cada uno desde cero. El análisis y diseño utilizado en esta automatización de pruebas de software corresponde a un grupo bien definido de sistemas y no a sistemas individuales.

CAPITULO III

Marco Metodológico

En el capítulo III se abarcará todo lo relacionado al análisis metodológico, para el desarrollo de cada uno de los objetivos implementados en el desarrollo de la aplicación de métodos para la automatización de pruebas de software dentro de la empresa Software Quality Assurance.

6. METODOLOGÍA

La Figura 3 resume en etapas, la metodología aplicada en el desarrollo del proyecto.

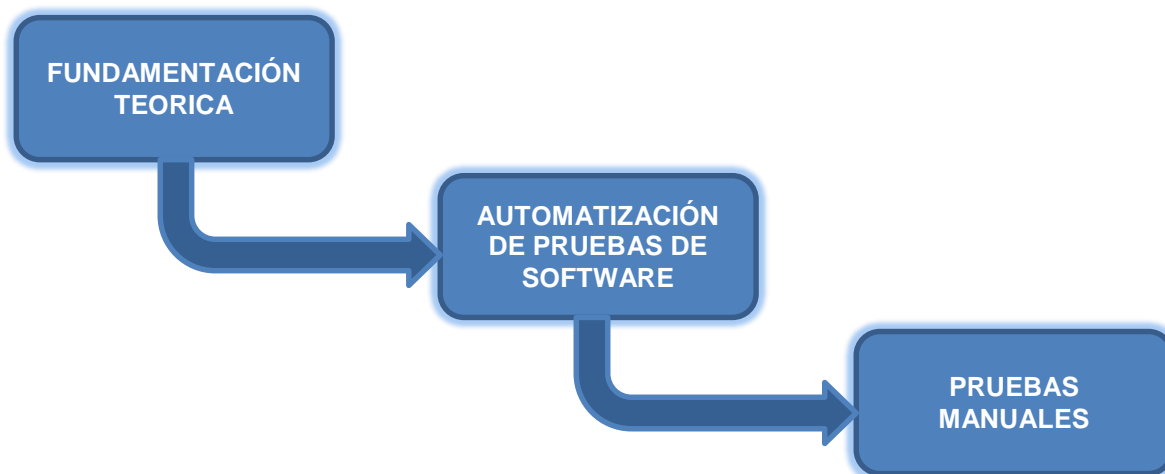


Figura 3 Esquema metodologico

6.1 FUNDAMENTACION TEORICA

Teniendo en cuenta que uno de los objetivos específicos del proyecto, es estudiar y profundizar las bases de conocimiento en conformidad con las características del mismo,

por lo tanto se procede a dar seguimiento al desarrollo de un “Plan de Inducción” dado por la empresa en el cual se presentara detalladamente lo relacionado a las responsabilidades dentro de SQA y por ende al proceso de apoyo al proyecto.

Para la elaboración de las bases de aprendizaje requeridas por la empresa se tendrá en cuenta el “Campus SQA”, el acompañamiento por parte de Gestión de Conocimiento e innovación y el equipo Elite de operaciones. En la Figura 3 Cursos y semilleros, se evidenciarán los estudios realizados durante el desarrollo de las prácticas empresariales desarrolladas en la empresa Software Quality Assurance (SQA).

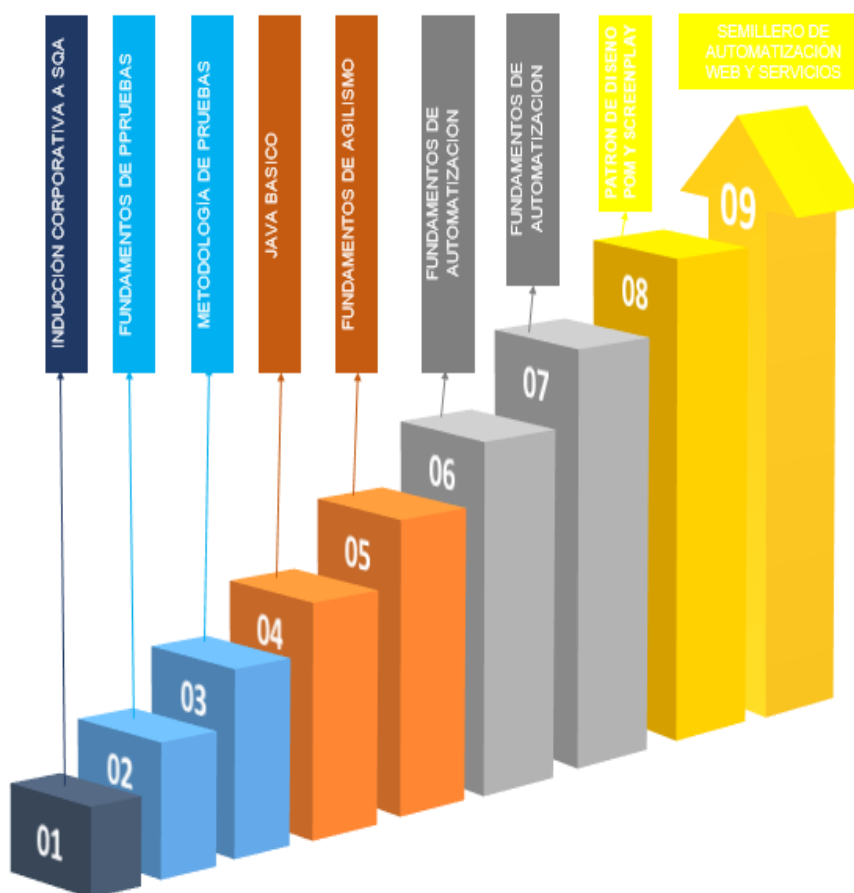


Figura 4 Cursos y semilleros

6.2 AUTOMATIZACIÓN DE PRUEBAS DE SOFTWARE

La industria del software, ha profesionalizado día a día su labor para otorgar calidad y confiabilidad al software que construye. Procesos de desarrollo bien administrados, análisis y diseño cuidadosamente elaborados, proporcionan un grado de certeza sobre la calidad de los desarrollos, pero la forma principal con la que se asegura la calidad del software es por medio de pruebas. Así, las pruebas de software se han convertido en una práctica habitual, no sólo realizada por los ingenieros especialistas en pruebas de software sino incorporada como una práctica habitual de los desarrolladores dentro del proceso de construcción.

La definición y propósito de las pruebas pueden tener orígenes muy diversos, como especificaciones y requerimientos, el código fuente, diseño de los artefactos o definiciones no funcionales; las que, además, se realizan en distintas etapas del proceso de desarrollo. Para facilitar el alcance y contexto de las pruebas, se definen los siguientes niveles:

- **Pruebas de aceptación:** Evalúan el software de acuerdo a sus requerimientos.
- **Pruebas de integración:** Evalúan el software de acuerdo al diseño de los subsistemas.
- **Pruebas unitarias:** Evalúan el software respecto a su implementación. (Martín, 2014)

Dado que en los objetivos específicos dos y tres se pretende abarcar el desarrollo de las pruebas funcionales y por ende el análisis de resultados. Es por esto que es necesario nombrar el “Ciclo de Vida de Pruebas” como un método fundamental para la aplicación en la automatización de pruebas de software.

6.2.1 DESARROLLO DE METODOLOGÍA POR FASES, MEDIANTE EL CICLO DE VIDA DE PRUEBAS

Un ciclo de vida de pruebas es la secuencia de cambios que atraviesa una entidad de una forma a otra. Estas actividades incluyen verificar el software

desarrollado para ver si cumple con los requisitos específicos. Si hay algún defecto en el producto, los probadores (tester) trabajan con el equipo de desarrollo.

Validar cada módulo de software o aplicación es imprescindible para garantizar la precisión y exactitud del producto. Dado que el software testing en sí mismo es un proceso complicado, los probadores lo llevan a cabo en fases. Pueden surgir complejidades si las pruebas carecen de organización. Esos problemas pueden incluir errores no resueltos, errores de regresión no detectados o, en el peor de los casos, un módulo que se saltó las pruebas porque se acercaba la fecha límite. (Anónimo, 2021)

Al momento de realizar el CICLO DE VIDA DE PRUEBAS se deben tener en cuenta 3 objetivos fundamentales, como lo son: inicio, ejecución y terminación del ciclo. A continuación se describen las actividades que se desarrollan dentro del CICLO DE VIDA para la ejecución de pruebas.

6.2.1.1 CONOCIMIENTO Y ANALISIS DE REQUISITOS

En esta primera fase el automatizador de pruebas observa, estudia y analiza cada una de las especificaciones dadas o solicitadas por el cliente. Es decir, los tester estudiarán los requisitos de pruebas funcionales y no funcionales, para después elegir qué tipo de automatización de prueba se va a realizar.

El objetivo de esta actividad es definir el conjunto de ciclos funcionales o funcionalidades que se probarán con las pruebas automatizadas. Esta actividad consiste en definir con el Cliente qué pruebas se van a automatizar (en alto nivel), considerando los ciclos funcionales o funcionalidades que involucran. Para ello se priorizan y evalúan las pruebas y las funcionalidades a probar. (Farías, 2016)

Las actividades que se desarrollan dentro de esta fase son básicamente realizar una lista de ideas junto con el cliente de los requisitos y prioridades que se llevaran a cabo en la realizaran del proyecto.

6.2.1.2 PLANEACIÓN DE PRUEBAS

En esta segunda fase el equipo de pruebas después de analizar detalladamente cada una de las especificaciones empieza a desarrollar un plan de pruebas para la ejecución del proyecto.

Esta fase se divide en dos sub-fases las cuales son:

- PLANIFICACIÓN DE PRUEBAS: En esta parte se visualiza y describe el alcance que tendrá el proyecto, los riesgos involucrados, como y cuando se hará (mediante la definición de horarios y entornos de prueba).
- ESTIMACIÓN: Se planean las etapas y se le asignan roles y responsabilidades a cada persona.

Dentro de esta etapa se encuentra también la RE-ESTIMACIÓN, la cual se lleva a cabo cuando el cliente no acepta los parámetros que se realizaron

El objetivo de esta actividad es definir las suites y scripts que conformarán las pruebas automatizadas. A partir de los ciclos funcionales y funcionalidades seleccionados para las pruebas, se especifican las suites y scripts que las ejecutarán. (Ignacio Esmite, 2016)

6.2.1.3 DISEÑO

En esta tercera etapa se empieza el diseño y desarrollo de los casos de prueba, se deben tener en cuenta unos datos mínimos como lo son: ID (único),

nombre de las variables, resumen o descripción, precondition, paso a paso, resultados, post condición.

Dentro del diseño de pruebas, podemos encontrar 3 casos de flujos:

- FLUJOS BASICOS: Dentro de este flujo se realiza todo lo que se espera lograr dentro del proyecto.
- FLUJOS ALTERNOS: Dentro de este flujo se tienen en cuenta acciones que no fueron creadas dentro del código, pero que son necesarias.
Ejemplo: Al ingresar mal la contraseña de inicio de sesión, arroje un mensaje que diga “contraseña incorrecta”.
- FLUJO EXCEPCIONAL: Dentro de este flujo se agregan las condiciones de salida del código.

Basándose en el plan de prueba, los probadores diseñan y desarrollan casos de prueba. Los casos de prueba deben ser extensos y cubrir casi todos los casos posibles. Se deben recopilar todas las permutaciones y combinaciones aplicables. (Anónimo, 2021)

6.2.1.4 EJECUCIÓN DE LA PRUEBA

En esta cuarta etapa se empiezan a probar cada uno de los ciclos que se fueron desarrollando en la ejecución de las pruebas, verificando paso a paso que todo esté funcionando bien o de lo contrario re ejecutar las pruebas y corregir los errores encontrados. Algo importante que se debe tener en cuenta dentro de esta etapa es comparar los resultados esperados vs los resultados reales, con el fin de verificar si se cumplió a cabalidad lo pactado junto al cliente.

Dentro de la ejecución de pruebas se deben tener en cuenta 2 casos:

- **RETEST**: Es volver a re ejecutar la prueba, este paso se realiza si al momento de realizar la ejecución nos arroja algún error, por ende se debe verificar paso a paso hasta que toda la automatización este correcta.
- **REGRESIÓN**: Estas pruebas se utilizan para garantizar que la automatización esté funcionando de forma correcta.

En esta etapa es de suma importancia tomar evidencias del paso a paso (pantallazos con hora y fecha) para poder tener pruebas de cada uno de los pasos realizados.

El objetivo de esta actividad es ejecutar una prueba completa de las suites correspondientes al ciclo funcional y verificar su correcto funcionamiento. Esta actividad consiste en realizar la prueba completa del ciclo funcional en el entorno preparado para dicho fin. En caso de un funcionamiento incorrecto deben realizarse los ajustes necesarios. (Farías, 2016)

6.2.1.5 CERTIFICACIÓN O CIERRE

En esta quinta y última etapa del ciclo de vida se realiza la entrega final del proyecto y se inicia la fase de cierre del mismo.

El equipo de control de calidad verifica los resultados de la prueba y los analiza con otros miembros del equipo. Algunos otros factores que consideran son la calidad del producto, la cobertura de la prueba y el costo del proyecto. Si hay una desviación de los valores estimados, se pueden realizar más análisis para identificar lo que no salió como se esperaba. Es una práctica esencial para los evaluadores reunirse y discutir la conclusión después de la prueba. (Anónimo, 2021)

En resumen, en esta etapa se pueden considerar los siguientes factores:

- Verificar que todas las etapas se cumplieran de forma correcta.
- Evaluar cada uno de los factores.
- Documentar cada paso que se realizó en el desarrollo de pruebas.
- Hacer entrega formal del proyecto.

6.3 PRUEBAS MANUALES

El término pruebas manuales, al igual que automatizadas, significa algo más que la ejecución de casos de prueba, y la idea de que este proceso podría hacerse sin la intervención humana es una cuestión de amplio debate en la comunidad. Si bien es cierto que los sistemas complejos requieren más de la automatización, mucho software trivial y con interfaces sencillas se puede probar sin estas herramientas, porque el costo es un ítem que se debe tener en cuenta. Por eso es que en todo proyecto de desarrollo se debe tener la posibilidad de elegir entre unas u otras para garantizar la fiabilidad del producto. La automatización acelera el proceso de prueba y una de las razones por las que es importante es que asegura de mejor forma la fiabilidad del software, especialmente cuando se realizan actualizaciones. Otra manera de lograrlo es utilizando pruebas redundantes, lo que disminuye los costos de mantenimiento y asegura la integridad del conjunto de pruebas. (Martínez, 2021)

Pues bien a razón de esto, se realizará la explicación del paso a paso de lo que es una prueba manual y más adelante como resultado esperado se hará de manera automatizada.

1. En el navegador principal realizo el ingreso a la página que se va automatizar, como se evidencia en la figura 5.



Figura 5 Ingreso página automatizar

Link: <https://core.dev.bolsamercantil.com.co/#/home>

2. Seguidamente realizo el "Login", es decir, el ingreso del Usuario y la Contraseña; como se evidencia en la figura 6.

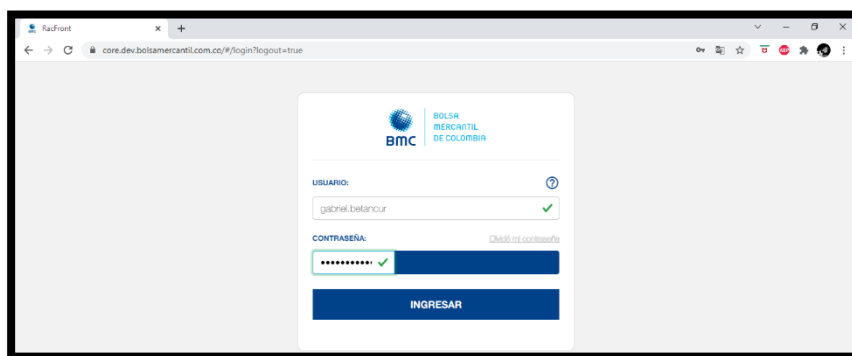


Figura 6 Login

3. Después de realizar el ingreso a la página, sale un letrero que dice "INFORMACIÓN" en este elijo la opción "BMC" y le doy click en aceptar, como se muestra en la Figura 7.

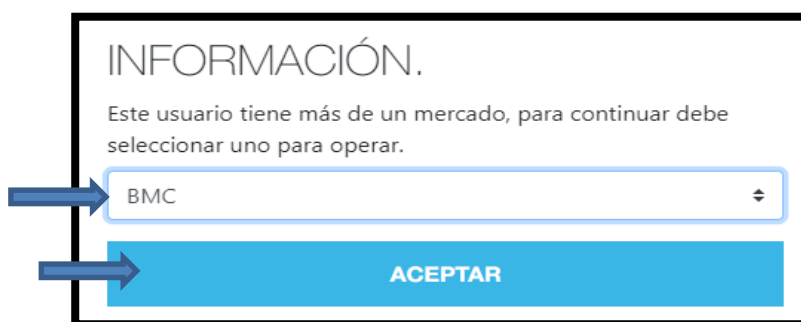


Figura 7 Selección "BMC"

- Al realizar la selección del usuario “BMC”, me dirijo a menú y doy click (se encuentra en la parte superior izquierda), como se muestra en la figura 8.



Figura 8 Click en menú

- Al desplegarse el menú, doy click en “Motor de Registro”, como se muestra en la figura 9.



Figura 9 Click en “Motor de Registro”

- Luego doy click en “Registro”, como se muestra en la Figura 10.

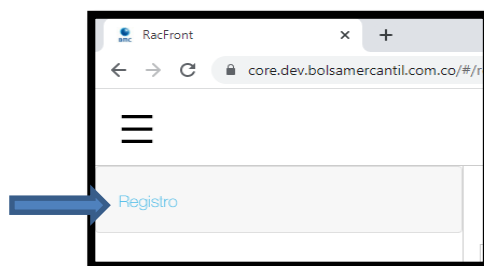


Figura 10 Click en “Registro”

- Después de desplegarse las funciones, doy click en “Prueba Auto” seguidamente en “Ítem 1” y por último en “subítem”, como se muestra en la figura 11.

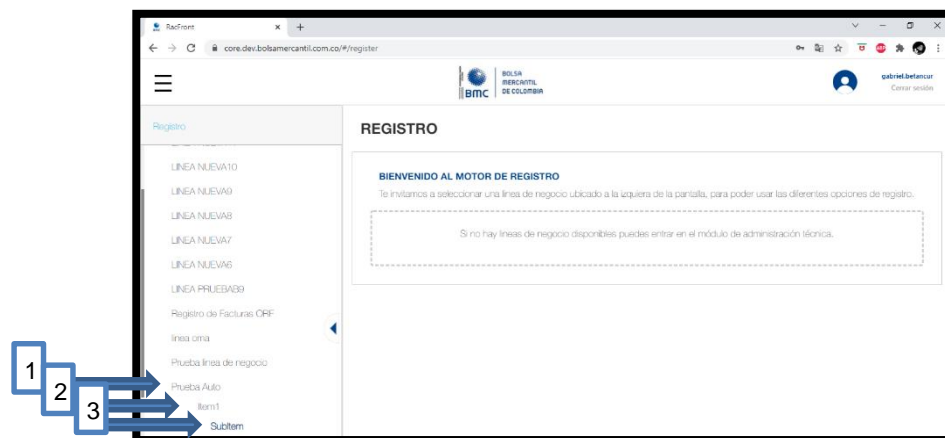


Figura 11 Click en “Prueba Auto-Ítem1-Subítem”

- Al ingresar en “subítem” nos aparecerán varias opciones de registro, doy click en “Registro por Demanda”, como se muestra en la figura 12.

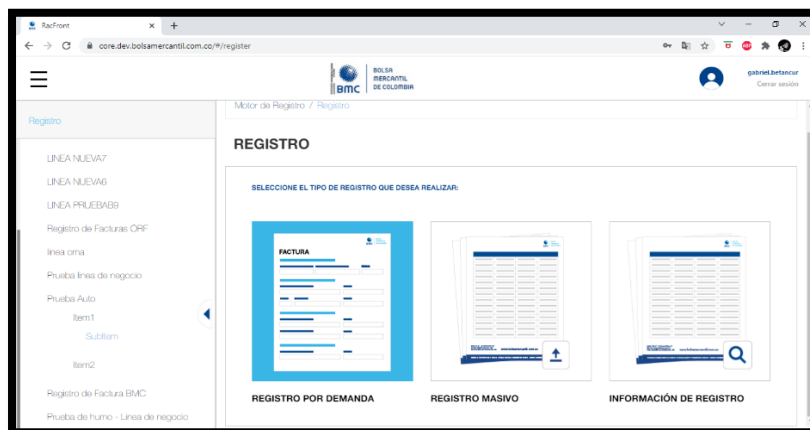
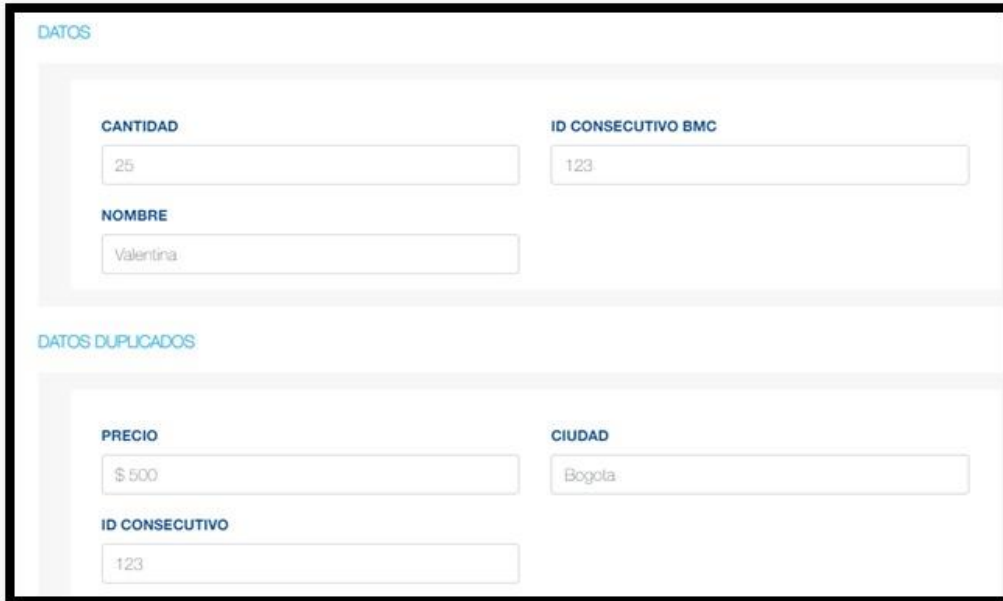


Figura 12 Click en “Registro por Demanda”

- Seguidamente se empieza con el llenado de campos, como se muestra en la figura 13.



DATOS

CANTIDAD
25

ID CONSECUTIVO BMC
123

NOMBRE
Valentina

DATOS DUPLICADOS

PRECIO
\$ 500

CIUDAD
Bogota

ID CONSECUTIVO
123

Figura 13 Llenado de Campos

10. Al terminar de llenar cada uno de los campos requeridos, en la parte inferior derecha doy click en “Guardar”, como se muestra en la figura 14.



Figura 14 Click en “Guardar”

11. Finalmente se muestra el “Mensaje de validación”, como se ve en la figura 15.



Figura 15 Validación

6.4 DESCRIPCIÓN DE ACTIVIDADES

Tabla 1 Descripción de actividades

OBJETIVOS	DESCRIPCIÓN
Fundamentar la base de conocimiento en conformidad con las características del proyecto.	<ol style="list-style-type: none"> 1. Elaboración de bases de aprendizaje requeridas por la empresa como cursos a través del Campus SQA, semilleros de automatización, capacitaciones por parte de talento humano e investigación propia. 2. Realización de actividades de formación dentro el semillero de SQA inherentes al proyecto del cliente. 3. Análisis del levantamiento de requerimientos por parte del cliente.
Desarrollar las pruebas funcionales como parte fundamental en el aseguramiento de la calidad del producto de software del cliente a través de la aplicabilidad de pruebas de caja negra.	<ol style="list-style-type: none"> 1. Revisión de las sesiones de contextualización acerca del entorno el cliente. 2. Conocimiento de criterios de aceptación del cliente. 3. Verificación de tipos de prueba que se van a emplear dentro del proyecto. 4. Realización de las pruebas funcionales empleando técnicas de automatización.
Analizar los resultados de las pruebas funcionales basados en los criterios de aceptación del cliente para determinar la fiabilidad de las mismas.	<ol style="list-style-type: none"> 1. Ejecución de pruebas, según corresponda. 2. Recopilación de los resultados e información de las pruebas. 3. Comparación los resultados obtenidos vs los resultados esperados.

CAPITULO IV

Análisis e interpretación de resultados

Este último capítulo es de suma importancia para el proyecto ya que en él se logra evidenciar los resultados obtenidos en base a los objetivos planteados explicando detalladamente como se dio cumplimiento a cada uno de ellos.

7. RESULTADOS OBTENIDOS

Con el propósito de fundamentar el aprendizaje y de adaptarse rápidamente a las necesidades de la empresa y a su vez del cliente, es de suma importancia aumentar el

aprendizaje y de esta manera ir en sintonía para resolver las necesidades del cliente. Por medio del siguiente listado se pretende describir el proceso formativo, donde se desarrollaron cursos y seminarios.

Tabla 2 Listado de cursos

CURSOS	MODULOS	OBJETIVOS	DESCRIPCIÓN	FECHA
INDUCCIÓN CORPORATIVA A SQA	<ul style="list-style-type: none"> Lo que orienta nuestra gestión. Acuerdos con SQA. Procesos en SQA. Herramientas de SQA. Seguridad de la información en SQA. Seguridad y Salud en el Trabajo en SQA. Procedimientos comunices en SQA. Zoom al proceso al que perteneces. 	<ul style="list-style-type: none"> Visualizar a los nuevos empleados a que brinden al cliente un servicio que les genere confianza y que le ayuden a cumplir los objetivos de cada negocio. Conocer más de SQA, orientando la confianza y satisfacción de los clientes. 	<ul style="list-style-type: none"> Proporciona información en relación a la empresa, sus servicios y sus procesos internos. Conocerás los principales acuerdos con SQA, sus herramientas, sus procedimientos más comunes, entre muchos otros. 	Agosto
FUNDAMENTOS DE PRUEBAS	<ul style="list-style-type: none"> Conceptos básicos de pruebas Principios de las pruebas Taxonomía de pruebas Niveles de Pruebas Procesos de Pruebas 	<ul style="list-style-type: none"> Comprender las bases fundamentales que rigen las pruebas de software. Definir términos comunes en el ámbito de pruebas para lograr un mejor entendimiento. 	<ul style="list-style-type: none"> Proporciona las bases fundamentales que rigen las pruebas de software y los términos comunes en este ámbito. 	Agosto
METODOLOGÍA DE PRUEBAS	<ul style="list-style-type: none"> Estimación de pruebas funcionales. Planeación de pruebas. Diseño de casos de pruebas. Ejecución de Pruebas. Evaluación y cierre de pruebas. 	<ul style="list-style-type: none"> Presentar la metodología tradicional para realizar pruebas funcionales, con el fin de que el analista de pruebas de SQA, tenga los criterios básicos para el desarrollo de su labor, y este en capacidad de comprender, justificar y mejorar el proceso de pruebas que lleva 	<ul style="list-style-type: none"> Presenta la metodología tradicional para realizar pruebas funcionales que tenemos en SQA, la cual comprende las etapas de estimación, planeación, diseño, ejecución y cierre de pruebas. 	Agosto

		a cabo en su día a día.		
JAVA BASICO	<ul style="list-style-type: none"> • Java Básico. • Variables. • Operadores. • Recolección de Datos. • Estructuras (condicionales e iterativas). • Declaraciones de salto. 	<ul style="list-style-type: none"> • Proporciona los conceptos básicos asociados al lenguaje de programación Java, para que los interesados, puedan entender esta tecnología de desarrollo y empezar a implementar pequeños proyectos de software, que ayuden a mejorar los tiempos de y la calidad de sus actividades de pruebas. 	<ul style="list-style-type: none"> • Presenta los conceptos básicos asociados al lenguaje de programación Java: variables, operadores, condicionales, ciclos, etc. 	Septiembre
FUNDAMENTOS DE AGILISMO	<ul style="list-style-type: none"> • Manifiesto Ágil. • Procesos Predictivos VS Procesos Adaptativos. • Marcos de Desarrollo ágil. • Historias de usuario. • Pruebas ágiles. 	<ul style="list-style-type: none"> • Interiorizar los conceptos fundamentales en entornos de desarrollo ágil y como es la interacción de un tester dentro de un equipo scrum. • Aprender las bases y principios de la filosofía y mentalidad ágil. 	<ul style="list-style-type: none"> • Presenta los conceptos fundamentales usados en entornos de desarrollo ágil y cómo es la interacción de un tester dentro de un equipo scrum. 	Septiembre
FUNDAMENTOS DE AUTOMATIZACIÓN	<ul style="list-style-type: none"> • Introducción a la automatización de pruebas. • HTML Básico. • Localizadores básicos. 	<ul style="list-style-type: none"> • Conocer los aspectos básicos de automatización de pruebas. • Comprender los fundamentos de HTML. • Desarrollar habilidades para entender localizadores de elementos. 	<ul style="list-style-type: none"> • Presenta los aspectos básicos de la automatización de pruebas, los fundamentos del lenguaje de marcación html y los conceptos básicos de localizadores de elementos en web. 	Septiembre
LOCALIZADORES	<ul style="list-style-type: none"> • Árbol HTML. • Localizadores para web. • Localizador CSS. • Localizador XPATH. • Localizadores para móviles. 	<ul style="list-style-type: none"> • Conocer los principales localizadores de elementos en web y móviles. • Desarrollar habilidades para usar los localizadores en 	<ul style="list-style-type: none"> • Presenta los principales localizadores que pueden usarse para encontrar y relacionar los elementos de una página web (name, id, 	Octubre

		pruebas automatizadas.	xpath, etc.) y en dispositivos móviles.	
PATRÓN DE DISEÑO POM Y SCREENPLAY	<ul style="list-style-type: none"> Patrón de diseño POM (Conceptos básicos, configuraciones, creación del “feature -page-step definitios” y ejecución) Patrón de diseño Screenplay (Principios SOLID, capas de screenplay, creación de proyecto “feature-runner-steps”, configuración del driver). 	<ul style="list-style-type: none"> Conocer las principales aplicaciones de POM y Screenplay. Comprender las bases teóricas fundamentales que rigen este patrón. 	<ul style="list-style-type: none"> Presenta los patrones de Diseño POM y ScreenPlay y su aplicación a una automatización básica. 	Noviembre

Por otra parte se desarrolló también un semillero de automatización.

Tabla 3 Lista de semilleros

SEMILLEROS	MODULOS	OBJETIVOS	FECHA
SEMILLERO DE AUTOMATIZACIÓN WEB Y DE SERVICIOS (POM Y SCREENPLAY)	<ul style="list-style-type: none"> Java Básico Localizadores BDD + Cucumber Patrones de Diseño Automatización web con Serenity. Automatización de servicios con Serenity. 	<ul style="list-style-type: none"> Conocer los principales aspectos de automatización de pruebas. Desarrollar habilidades para la fundamentación de pruebas. 	<ul style="list-style-type: none"> Septiembre-Octubre

Cada uno de estos complementa minuciosamente una base fundamental para el conocimiento y de igual manera ayudan a complementar el desarrollo de los procesos de automatización de pruebas de software.

7.1 PLAN DE DESARROLLO, PROYECTO AUTOMATIZACIÓN

Para el desarrollo del proyecto de automatización realizado, se utilizó el enfoque conocido como como “Desarrollo guiado por comportamiento (BDD)”, este resuelve

considerablemente los inconvenientes que se presentan en los cambios de interfaz gráfica, que se van dando a medida que el software evoluciona. Sin embargo, esto no implica que no se deban realizar mantenimientos posteriores a los scripts de automatización, los cuales cumplen con estándares de calidad, entre estos el de mantenibilidad.

El “Desarrollo guiado por comportamiento (BDD)”, permite la creación de escenarios a través del lenguaje Gherkin, basado en ‘Given-When-Then’. En otras palabras, Gherkin presenta el comportamiento de la aplicación, el cual puede ser manipulado por un analista funcional o un usuario de negocio.

Para el desarrollo del proyecto de automatización, se recolecto información del cliente para conocer los requerimientos del mismo y así nosotros como testers poder hacer el PLAN DE DESARROLLO como se muestra a continuación.

7.1.1 INFORMACIÓN DEL PROYECTO

Tabla 4 Información general del proyecto

Nombre del requerimiento	Automatización BMC Plataforma Web Bolsa Mercantil
Tipo de Servicio	Desarrollo Automatización
Aplicación Principal	Portal Web Bolsa Mercantil
Fábrica de Desarrollo	Software Quality Assurance S.A

En la anterior tabla se encuentra la información general de cada uno de los principales requerimientos dados por el cliente.

7.1.2 ALCANCE DEL DESARROLLO DE LA AUTOMATIZACIÓN DE PRUEBAS

Se realizó el Proyecto de Desarrollo de Automatización bajo el enfoque “Automatización guiada por comportamiento (BDD)”, para el alcance relacionado en los requerimientos solicitados por el cliente sobre la aplicación de prueba.

Plataforma: Portal Web Bolsa Mercantil

- Los flujos definidos como alcance se encuentran relacionados en la siguiente tabla.

Tabla 5 Alcances

ID CASO DE PRUEBA	MÓDULO	PLATAFORMA	CASOS DE PRUEBA	RESULTADO ESPERADO
CP001	Registro por demanda con duplicación de datos.	Portal Web Bolsa Mercantil	Validar y verificar que, al realizar el registro e ingresar los datos, y clic en el botón guardar, quede guardado el registro	Se evidencia que, al realizar un registro por demanda, se visualiza un mensaje de registro exitoso
CP002	Registro por demanda con duplicación de datos fallida.	Portal Web Bolsa Mercantil	Validar y verificar que, al hacer el registro con datos inválidos, y clic en el botón guardar, no quedara guardado	Se evidencia que, al realizar un registro por demanda con datos inválidos, se visualiza un mensaje de error en registro

➤ **Versión de navegador:**

- Google Chrome V.94 en adelante

➤ **Características técnicas de los dispositivos:**

- El desarrollo de la automatización se realizó en la capa de planeación.

➤ **Generación de reportes de evidencias, front de ejecución de la Automatización.**

- Los cambios de alcance generaron reestimación. Las modificaciones realizadas en el proyecto tuvieron impacto en los tiempos estimados de entrega.

7.1.3 FUERA DEL ALCANCE DEL DESARROLLO DE AUTOMATIZACIÓN DE PRUEBAS

No se contempla dentro del alcance:

- Desarrollo de automatizaciones de las transacciones, soluciones o sistemas no especificados dentro de alcance.
- La automatización de transacciones, soluciones o sistemas en una capa diferente a la definida en el alcance.
- Mantenimiento de scripts entregados a conformidad, que se deriven de posteriores modificaciones que se realicen a la versión del software sobre el que se desarrolló la automatización.

7.1.4 REQUISITOS Y RESTRICCIONES

Tabla 6 Requisitos y restricciones

REQUISITOS	RESTRICCIONES
<ul style="list-style-type: none"> • BMC realizara una contextualización de la aplicación objeto del desarrollo de Automatización. • Toda la información utilizada en la ejecución se encuentra previamente validad. • Los datos suministrados para realizar el proyecto de desarrollo de automatización solo podrán ser utilizado por el equipo de automatización, los cambios sobre estos tendrán que ser informados de forma oportuna. • Los perfiles de usuarios suministrados por el cliente para las diferentes aplicaciones no deben cambiar durante el desarrollo de la automatización. • El cliente contará con el personal necesario durante el desarrollo de automatización de scripts para atender cualquier eventualidad que se presente, restablecer los procesos/servicios y/o servidor en caso de presentarse inconvenientes. • El ambiente es accesible. • Las funcionalidades definidas para el desarrollo de automatización de 	<ul style="list-style-type: none"> • Los scripts de automatización deben ser entregados en las fechas pactadas. • No se realizarán despliegues en ambiente de QA Normal durante el proceso de levantamiento de scripts de Automatización. En caso de hacerse, se informará a SQA previamente. • No se presentarán ni solicitarán requerimientos prioritarios que impacten la ejecución del proyecto.

pruebas fueron sometidas a pruebas funcionales.	
---	--

En la tabla anterior se evidencian con claridad cada uno de los requisitos que se pactaron junto con el cliente y a su vez las restricciones que se tuvieron dentro del proyecto, con el fin de aclarar futuros malos entendidos y evitar verse afectadas ambas partes. Ya que de lo contrario se podrán ver exigencias de uno de los dos lados (cliente o desarrollador de software) que no se podrán cumplir por motivos de tiempo o costos.

7.1.5 VISTA GENERAL DEL PROYECTO

Entregables del proyecto:

- Plan de desarrollo del proyecto de automatización: Se entrega el presente documento, para la verificación y aprobación del área usuario.
- Informes de avance.
- Ejecutable con los scripts desarrollados en el proyecto de automatización, en extensión <<.rar>>.
- Acta de entrega a conformidad de los scripts automatizados.

Y en caso de presentarse algún inconveniente que obligue a la cancelación del proyecto, se hará entrega de los siguientes documentos:

- ✓ Carta de cancelación.

7.1.5.1 HERRAMIENTAS

Para el Proyecto de desarrollo de Automatización, se encontró que para la aplicación Portal Web Bolsa Mercantil las herramientas que mejor se acoplaron fueron: Cucumber con Serenity integrado con el patrón POM. El desarrollo de la

automatización se realizó con el lenguaje de java y la generación de reportes se ejecutó a través de la herramienta de Serenity.

7.1.5.2 ANALISIS DE RIESGOS

Tabla 7 Análisis de riesgos

RIESGOS DE LA PRUEBA		
Riesgo de la prueba	Solución Propuesta	Responsable
Estimaciones de alto nivel basada en el alcance de la implementación del sistema por el proveedor.	Detallar cada uno de los elementos que componen el sistema y reestimar las pruebas.	BMC
Ambiente de pruebas inestable	Entregar un ambiente de pruebas estable con datos similares al de producción con pruebas mínimas.	BMC
No se tiene conocimiento en la aplicación.	Realizar contextualización o reunión con el cliente.	BMC SQA
No contar con datos reales para la elaboración de las pruebas	Revisar con anterioridad los datos entregados por el cliente, verificando que cumple con las condiciones y características mínimas para las pruebas.	BMC
Consumo de los recursos planeados sin finalizar la ejecución de todos los casos de prueba	Priorización de casos de prueba.	BMC SQA
Dificultad en la asignación de permisos y/o perfiles para la ejecución de las pruebas	Solicitar con antelación al proveedor la configuración de los permisos y perfiles requeridos	BMC
Genere error en la integración con navegadores	Solicitar con antelación al proveedor la configuración de los permisos y perfiles requeridos	BMC SQA
No tener cubrimiento del 100% de los escenarios	Garantizar cubrimiento entre ruta crítica.	BMC SQA

La anterior tabla es un estudio de las causas o efectos no deseados, y las soluciones dadas para cada una de ellas. La tabla 7 hace referencia al proceso que planifica, identifica, analiza, responde y controla los riesgos de un proyecto.

7.1.5.3 ARQUITECTURA DEL DESARROLLO DE AUTOMATIZACIÓN

Para la automatización de “BMC Portal Web Bolsa Mercantil” se implementó una arquitectura basada en el patrón POM (Page Object Model), utilizando herramientas como Cucumber integrado con la herramienta Serenity. Los escenarios estuvieron diseñados bajo el lenguaje Gherkin para mayor entendimiento del flujo de los analistas funcionales o de negocio.



Figura 16 BDD+Cucumber

7.1.5.4 CRONOGRAMA DE HORAS

Tabla 8 Cronograma en horas

TIEMPOS POR ACTIVIDAD	
ACTIVIDADES A REALIZAR	TIEMPO ESTIMADO (HORAS)
ETAPA DE DEFINICIÓN Y ENTENDIMIENTO	8
ETAPA DE DISEÑO Y CONSTRUCCION	135
ETAPA DE EJECUCIÓN	10
ACTIVIDADES ADMINISTRATIVAS Y TRANSVERSALES	8
TOTAL TIEMPO ESTIMADO (HORAS)	161

En la tabla 8 se hace una descripción detallada de cada una de las actividades realizadas en pro a las funciones requeridas para la ejecución del

proyecto. Allí se desglosa por número de horas como se fue trabajando para dar cumplimiento a dichas actividades.

7.1.6 ORGANIZACIÓN DEL PROYECTO DE AUTOMATIZACIÓN

7.1.6.1 ROLES Y RESPONSABILIDADES

Tabla 9 Roles y responsabilidades

Nombre	Rol	Responsabilidades
✓ Gissell Valentina Montañez Vera ✓ Sol Serna Vargas	Analista de automatización	Apoyo en el desarrollo de la Automatización.
John Anderson Duque	Gerente de Servicio	Responsable de los escalamientos al cliente.

7.1.6.2 AMBIENTE

El ambiente en donde se ejecutó la automatización de prueba de software es el QA (Quality Assurance) en este se realizaron pruebas donde se garantizó que el software cumpliera con cada uno de los requerimientos establecidos. En la siguiente tabla se evidencian las especificaciones establecidas para este ambiente.

Tabla 10 Ambiente

Ambiente	Recurso	Cantidad	Comentarios o Descripción
Ambiente de QA	Servidor disponible en ambiente de QA	1	Ambiente funcionando correctamente y homologado al de producción

7.1.6.3 BACKUP DE LA INFORMACIÓN

Todo el respaldo de la documentación fue responsabilidad de la Empresa Contratante, para esto, se suministró una ruta centralizada donde los Analistas de Automatización almacenaron toda la información referente al proyecto, en una estructura de carpetas tal como lo indica la Metodología de SQA S.A.

7.2 PRUEBAS AUTOMATIZADAS

En las pruebas de software, la automatización de pruebas consiste en el uso de software especial para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos y los resultados esperados. (Kolawa & Huizinga, 2007)

Teniendo presente que en el inciso 6.3 de la metodología se habló acerca de las Pruebas Manuales y un procedimiento que se debe realizar en dichas pruebas, ahora se realizará la explicación de lo que es la creación de una prueba automatizada para la Creación de un nuevo Proyecto en base a los requerimientos del cliente.

7.2.1 PROCEDIMIENTO

- a. Se crea un nuevo proyecto tipo Gradle, en la opción File/New/Project, como se muestra en la figura 17.

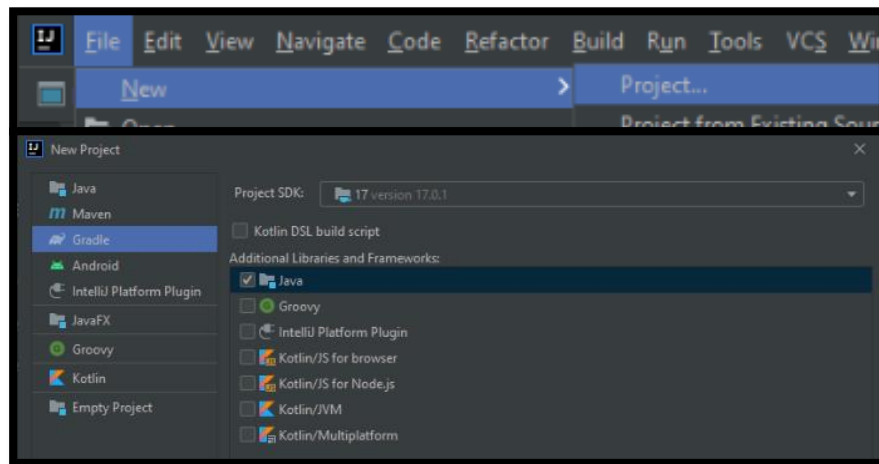


Figura 17 Creación Nuevo Proyecto

- b. Se realiza la configuración del *build.gradle* (este archivo viene por defecto en el proyecto, solo se realiza la configuración dentro de este), como se muestra en la figura 18.

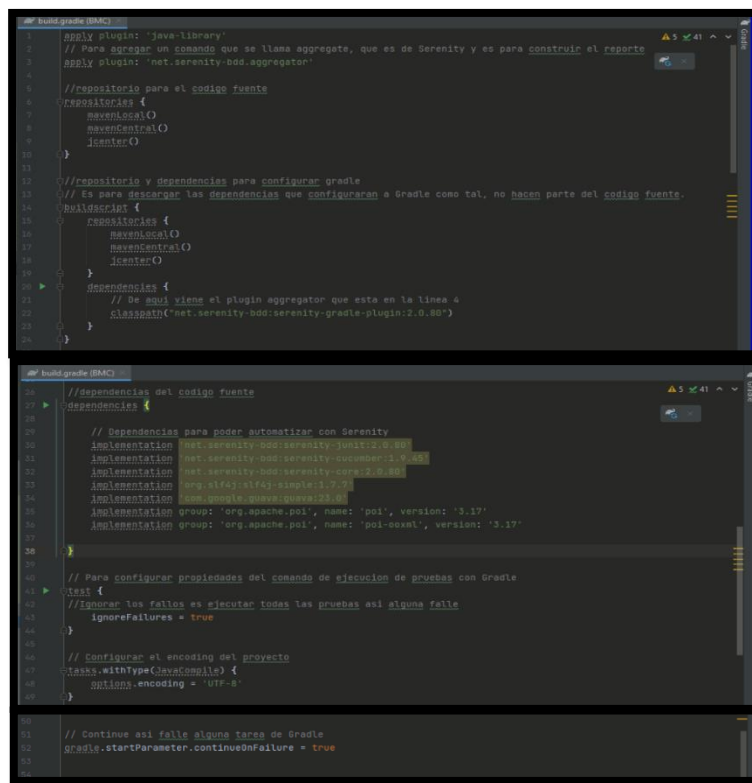
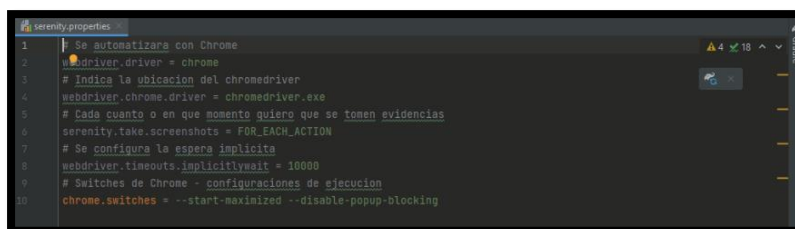


Figura 18 Creación build.gradle

- c. Se realiza la creación y configuración del `serenity.properties`, como se muestra en la figura 19.



```
1 | Se automatizara con Chrome
2 | driver.driver = chrome
3 | # Indica la ubicacion del chromedriver
4 | webdriver.chrome.driver = chromedriver.exe
5 | # Cada cuanto o en que momento quiero que se tomen evidencias
6 | serenity.take.screenshots = FOR_EACH_ACTION
7 | # Se configura la espera implicita
8 | webdriver.timeouts.implicitlywait = 10000
9 | # Switches de Chrome - configuraciones de ejecucion
10 | chrome.switches = --start-maximized --disable-popup-blocking
```

Figura 19 Creación `serenity.properties`

- d. Seguidamente se verifica la versión del navegador y se descarga el chrome driver según la versión correspondiente, como se muestra en la figura 20.

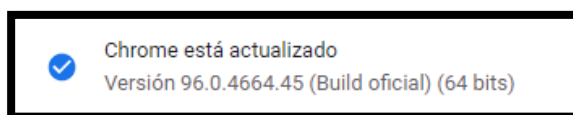


Figura 20 Versión Chrome Driver

Link descarga ChromeDriver: <https://chromedriver.chromium.org/downloads>

- e. Después de descargar la versión correspondiente, se agrega a la raíz del proyecto, como se muestra en la figura 21.

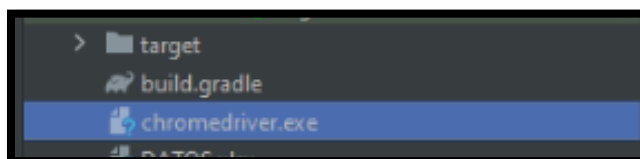


Figura 21 Chrome Driver en la estructura del proyecto

- f. Luego se realiza la creación de carpetas y clases, como se muestra en la figura 22.

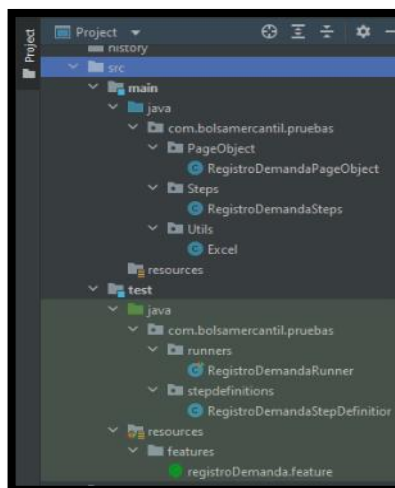


Figura 22 Creación de carpetas y clases

- g. En este proyecto en especial se utilizó la herramienta Excel (el documento se guardó como “Datos”) para la ejecución de datos, esta se agrega a la raíz del proyecto, como se muestra en la figura 23.

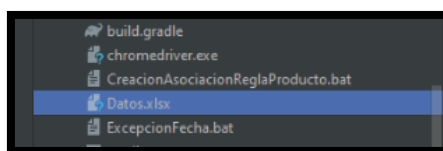


Figura 23 “Datos” en la estructura del proyecto

Estos incisos son esenciales en la creación de cada proyecto de automatización, sin embargo, dentro de cada clase (paso f) se realiza la configuración para la automatización del proyecto, se debe tener en cuenta el siguiente orden y contexto:

- Configuración Feature: Se realiza la configuración del lenguaje Gherkin (Given-When-Then), en las cuales se idealiza el comportamiento que llevara la automatización.
- Configuración Runner: Se colocan las principales rutas de ejecución del código, es decir, la ruta del Feature y el StepDefinition.
- Configuración stepDefinitions: Se llaman los métodos de la clase steps los cuales tienen la validación en ellos.
- Configuración Page Object: Se declaran cada una de las variables mapeadas dentro de la página automatizada.
- Configuración Steps: Se configuran cada una de las acciones a realizar dentro de la página automatizar.
- Configuración Utils: Su finalidad es colocar las esperas o métodos que se usan eventualmente, también esta configuración se utiliza para llamar los datos del Excel.

7.2.2 HERRAMIENTAS

- IntelliJ IDEA Community Edition 2021 2.3
- Microsoft Excel

A través del “**VIDEO CODIGO**” que se encuentra en el apartado de anexos se realizó una explicación más detallada de todo lo que se ha dicho anteriormente a través del software de programación IntelliJ IDEA Community Edition 2021 2.3.

CONCLUSIONES

- Con el propósito de fundamentar toda la etapa de aprendizaje para adaptarse rápidamente al entorno y a la filosofía que tiene la empresa para desempeñar sus labores y buscar la satisfacción de sus clientes, se logró realizar todos los cursos y semilleros durante el proceso de práctica, ya que estos facilitaban la elaboración del proyecto e influían proporcionalmente al crecimiento formativo.
- Para el desarrollo de las pruebas funcionales y el aseguramiento de la calidad del software, se utilizaron las denominadas Pruebas de Caja Negra las cuales permiten aplicarse en la interfaz del software para poder examinar el aspecto funcional del mismo, demostrando que las funcionalidades del software son operativas, es decir, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Un ejemplo de ello es que al ingresar en el Portal Web de BMC con nuestros datos correctos produce un ingreso exitoso y nuestra salida por ende sería el correcto funcionamiento de la página y al final el mensaje de validación. Cabe recalcar que todo lo relacionado a los métodos de aplicación de pruebas se hizo bajo el conocimiento de los criterios de aceptación del cliente
- Al validar la funcionalidad del software se determinó que cumple con todas las características y especificaciones que han sido establecidas por el cliente ya que al realizar ambas pruebas (manuales y automatizadas) se obtuvo un resultado satisfactorio ya que los resultados esperados como los resultados obtenidos fueron muy similares por ende se concluye que la aplicación de métodos para la automatización de pruebas fue exitosa.

RECOMENDACIONES

- Para desarrollar un proceso de automatización de pruebas de software la columna vertebral esta en empaparse del conocimiento que esta derivada, ya que todos los casos pueden variar, puede que muchos errores sean indescifrables por esto es importante mantener nuestra mente llena de recursos para saber en qué momento los podemos utilizar.
- Es de vital importancia tener una minuciosa observación al momento de realizar cada una de las cinco etapas del ciclo de vida de pruebas ya que si no se realiza de forma correcta prolongaría el tiempo de entrega, porque se deben realizar más pruebas y por ende aumentarían los gastos.
- Al momento de realizar el plan de desarrollo del proyecto de automatización se debe tener siempre en cuenta los parámetros de aceptación del cliente para poder lograr cumplir sus expectativas y por ende tender buenos resultados en la aplicación de métodos para la automatización de pruebas de software.

REFERENCIAS BIBLIOGRÁFICAS

Alicia Salamon, P. M. (Agosto de 2016). *PDF*. Obtenido de PDF:

http://sedici.unlp.edu.ar/bitstream/handle/10915/42358/Documento_completo.pdf?sequence=1&isAllowed=y

Andes, U. d. (16 de Mayo de 2019). *UNIANDES*. Obtenido de UNIANDES:

<https://sistemas.uniandes.edu.co/foro/miso/2019/>

Andrea, C. (2018). *Automatización de Pruebas*. Obtenido de Automatización de Pruebas:

<https://www.excentia.es/automatizacion-de-pruebas>

Andrea, C. (11 de Enero de 2018). *Reflexionamos acerca de la automatización de pruebas funcionales de software. Qué es exactamente esta práctica, cuándo aplicarla y por qué.*

Obtenido de Reflexionamos acerca de la automatización de pruebas funcionales de software. Qué es exactamente esta práctica, cuándo aplicarla y por qué.:

<https://www.excentia.es/automatizacion-de-pruebas>

Anónimo. (11 de Enero de 2021). *Trans-ti*. Obtenido de Tran-ti: [https://trans-](https://trans-ti.com/2021/01/11/software-testing-conoce-el-ciclo-de-vida-de-las-pruebas-de-software/)

[ti.com/2021/01/11/software-testing-conoce-el-ciclo-de-vida-de-las-pruebas-de-software/](https://trans-ti.com/2021/01/11/software-testing-conoce-el-ciclo-de-vida-de-las-pruebas-de-software/)

Castillo, J. F. (Noviembre de 2020). *DESARROLLO DE DEMO PORTABLE Y*

AUTOMATIZADO PARA LAS APLICACIONES KACTUS-HCM FUNDAMENTADO EN CONTENEDORES SOBRE DOCKER PARA LA EMPRESA DIGITALWARE.

Díaz, D. J. (16 de Mayo de 2021). *INTEGRACIÓN DE AUTOMATIZACIÓN Y DISTRIBUCIÓN DE PRUEBAS UTILIZANDO MICROSOFT AZURE Y CONTENEDORES DOCKER EN LA EMPRESA DIGITAL WARE.*

Edgar Serna, R. M. (26 de Febrero de 2021). *Scielo*. Obtenido de Scielo:

http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462019000100169

Esteve Ambrosio, D. A. (Julio de 2015). *PDF*. Obtenido de PDF:

<https://riunet.upv.es/bitstream/handle/10251/55321/Esteve%20-%20Implantaci%3%b3n%20de%20un%20proceso%20de%20automatizaci%3%b3n%20de%20pruebas%20para%20una%20aplicaci%3%b3n%20software.pdf?sequence=2&isAllowed=y>

Farías, M. (28 de Marzo de 2016). *Automatización y Gestión de las Pruebas Funcionales.*

Obtenido de Automatización y Gestión de las Pruebas Funcionales:

http://sedici.unlp.edu.ar/bitstream/handle/10915/21787/Documento_completo.pdf?sequence=1&isAllowed=y

Haro, J. D. (10 de Marzo de 2021). *Automatización de pruebas para optimizar los procesos de desarrollo de*. Obtenido de Automatización de pruebas para optimizar los procesos de desarrollo de:

https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/4517/Jesus_Carrion_Trabajo_de_Suficiencia_Profesional_Titulo_Profesional_2021.pdf?sequence=3&isAllowed=y

Ignacio Esmite, M. F. (2016). *PDF*. Obtenido de PDF:

http://sedici.unlp.edu.ar/bitstream/handle/10915/21787/Documento_completo.pdf?sequence=1&isAllowed=y

Kolawa, A., & Huizinga, D. (2007). *Automated Defect Prevention*.

Luciano Baresi, M. P. (01 de Febrero de 2006). *sciencedirect*. Obtenido de An Introduction to Software Testing: <https://www.sciencedirect.com/science/article/pii/S1571066106000442>

Martín, R. F. (Julio de 2014). *Modelo de Desarrollo de Software Guiado por Pruebas y Apoyado con*. Obtenido de Modelo de Desarrollo de Software Guiado por Pruebas y Apoyado con: <https://www.mti.cl/wp-content/uploads/2017/12/5.-Documento-Tesina-MTI-Rau%CC%81I-Catala%CC%81n.pdf>

Martínez, R. (26 de Febrero de 2021). *scielo.org.mx*. Obtenido de scielo.org.mx:

http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462019000100169

Maximiliano A. Mascheron, E. I. (Octubre de 2018). *SEDICI UNLP*. Obtenido de SEDICI UNLP:

<http://sedici.unlp.edu.ar/handle/10915/73270>

Moreno, E. G. (Marzo de 2012). *PDF*. Obtenido de PDF:

https://d1wqtxts1xzle7.cloudfront.net/54693882/automatizacion_de_procesos_industriales-with-cover-page-v2.pdf?Expires=1632005794&Signature=BPXXno01lr~0lcaeh2qc9aFvRBBAQO3yJAx1ytfXhM9II9u5UhE8vufh2cXwm1SdFx4C1vmaMrXSRhlpAaiUMm30mVNCEQTHEERzOYXWHZa~9ae7E7cz

Rodriguez, C. (2018). *Abstracta*. Obtenido de Abstracta:

<https://cl.abstracta.us/blog/automatizar-pruebas-de-software/>)

Serna. (Enero de 2012). *Social control for science and technology*. Obtenido de Social control

for science and technology: [http://www.scielo.org.mx/scielo.php?pid=S1405-](http://www.scielo.org.mx/scielo.php?pid=S1405-55462017000200337&script=sci_arttext#B1)

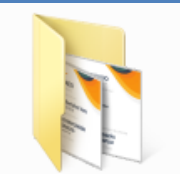


[55462017000200337&script=sci_arttext#B1](http://www.scielo.org.mx/scielo.php?pid=S1405-55462017000200337&script=sci_arttext#B1)

Serna, S. E. (2022). *PLAN DE DESARROLLO DE AUTOMATIZACIÓN*. Medellin Antioquia.

SQA. (Octubre de 2021). *SQA S.A.* Obtenido de SQA S.A: <https://sqasa.co/>

ANEXOS

Tabla 11 Anexos

ANEXOS	
 <p>CERTIFICADOS SQA</p>	<p>Certificados de cursos y semilleros.</p>
 <p>VIDEO CODIGO</p>	<p>Video explicativo del código mostrando su funcionamiento.</p>
 <p>PROYECTO FINAL - GISSELL VALENTINA MONTAÑEZ VE...</p>	<p>Documento final de trabajo de grado.</p>