



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

“Sistema para la Automatización de pruebas de regresión basado en web en la empresa Pandora Technologies durante el periodo 2019”

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero de Sistemas

AUTORES:

Hoyos Hipolito, Marco Giovanni (ORCID: 0000-0002-3926-6426)

Henriquez Palomino, Carlos Alberto (ORCID: 0000-0003-3840-4273)

ASESORA:

Mg. Acuña Meléndez, María (ORCID: 0000-0002-5188-3806)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y Comunicaciones

LIMA – PERÚ

2019

### **Dedicatoria**

Nuestro proyecto está dedicado a nuestros padres, a la familia, amigos más cercanos y a nuestra asesora quienes depositaron su confianza en nosotros. Por ello agradecemos la oportunidad de llevar a cabo esta investigación.

### **Agradecimiento**

A la Facultad de Ingeniería de la Universidad por la excelente formación.

A todos mis compañeros y amigos que fuimos haciendo durante toda la carrera y que gracias a ellos las jornadas de estudio siempre resultaron más agradables.

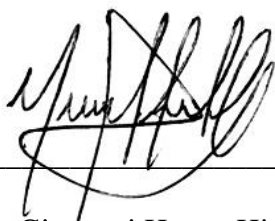
## **Página del Jurado**

## DECLARATORIA DE AUTENTICIDAD

Nosotros, Marco Giovanni Hoyos Hipolito, con DNI N.º 72699018 y Carlos Alberto Henriquez Palomino, con DNI N.º 73338825, a efecto de cumplir con las disposiciones vigentes consideradas en el Reglamento de Grados y Títulos de la Universidad Cesar Vallejo, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, declaramos bajo juramento que toda la documentación que acompañamos es veraz y auténtica.

Asimismo, declaramos también bajo juramento que todos los datos e información que se presenta en la presente tesis son auténticos y veraces. En tal sentido, asumimos la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada por lo cual nos sometemos a lo dispuesto en las normas académicas de la Universidad Cesar Vallejo.

Lima, 14 de julio de 2019



Marco Giovanni Hoyos Hipolito

DNI. 72699018



Carlos Alberto Henriquez Palomino

DNI. 73338825

## Índice General

I.	INTRODUCCIÓN .....	10
1.1	Realidad Problemática .....	11
1.2	Situación actual.....	13
1.3	Propuesta .....	13
1.4	Trabajos Previos .....	13
1.4.1	Internacionales.....	13
1.4.2	Nacionales .....	18
1.5	Teorías relacionadas al tema .....	23
1.5.1	Sistema web.....	23
1.5.2	Tecnología para el desarrollo .....	27
1.5.3	Java .....	28
1.5.4	MongoDB.....	29
1.5.5	Automatización de pruebas .....	30
1.5.6	Cross Browser Testing .....	37
1.5.7	Rational Unifed Process (RUP).....	38
1.5.8	Scrum .....	40
1.5.9	Metodología XP.....	41
1.6	Formulación del problema .....	44
1.7	Justificación del estudio .....	44
1.8	Hipótesis.....	46
1.9	Objetivos .....	46
II.	MÉTODO .....	47
2.1	Método de investigación .....	48
2.1.1	Tipo de estudio .....	48
2.1.2	Enfoque .....	48
2.1.3	Alcances de estudio .....	48
2.1.4	Diseño de Investigación .....	49
2.2	Variables y Operacionalización.....	49
2.3	Población y Muestra .....	51
2.4	Técnicas e instrumentos de recolección de datos, validez y confiabilidad .....	52
2.5	Método de análisis de datos.....	53
2.6	Aspectos Éticos.....	53
III.	RESULTADOS.....	55

IV. DISCUSIÓN .....	63
V. CONCLUSIONES .....	66
VI. RECOMENDACIONES .....	68
REFERENCIAS .....	70
ANEXOS.....	77

## Resumen

Este estudio analiza, desarrolla e implementa un sistema que automatiza los procesos de pruebas de regresión en el proceso de desarrollo de un software en la empresa Pandora Technologies E.I.R.L, donde actualmente los procesos son realizados por desarrolladores, los cuales omitían ciertos errores, no terminaban de realizar todos los casos de prueba, de probar todas las funcionalidades y no conformidades, generando costos adicionales y retrasos en los tiempos establecidos. El objetivo de esta investigación es determinar el impacto de un sistema basado en la automatización de pruebas (Test Automation) utilizando frameworks y demás librerías para que reemplace a los desarrolladores que se encargaban de todos los procesos de testing en la regresión y que se adapten a las necesidades del equipo de la programación de software. Los resultados son positivos, obteniendo la reducción de costos a través de disminuir el tiempo y mejorar la eficacia mediante el proceso de pruebas de regresión al haber reducido el tiempo en la generación de data en 55% y aumentado el porcentaje de data en 7%. Se sugiere evaluar el impacto de la automatización en otros sectores.

**Palabras clave:** automatización de pruebas, test automation, software de automatización de pruebas, pruebas de regresión, robot de software, testing.



## Abstract

This study included the analysis, development and implementation of a system that automates the regression testing processes in the software development process in the company Pandora Technologies EIRL, where currently the processes are carried out by developers, which omitted certain errors, They did not finish carrying out all the test cases, testing all the functionalities and non-conformities, generating additional costs and delays in the established times. The objective of this research is to determine the impact of a system based on test automation (Test Automation) using frameworks and other libraries to replace the developers who were in charge of all the testing processes in the regression and that adapt to the needs of the software programming team. The results were positive, achieving cost reduction through time reduction and improved efficiency during the regression testing process by reducing the time in data generation by 55% and increasing the percentage of data in 7%. It is suggested to evaluate the impact of automation in other sectors.

Keywords: test automation, test automation, test automation software, regression testing, software robot, testing.

## **I. INTRODUCCIÓN**

## 1.1 Realidad Problemática

Las pruebas en la calidad de software actualmente están en constante evolución. Las Tecnologías de Información y Comunicaciones (TIC), debido a este cambio se da el desarrollo de pruebas más sofisticadas a la hora de testear un software, como son las pruebas automatizadas de software, la cual beneficia enormemente a los desarrolladores de software en una empresa; ayudando las 24 horas los 7 días de la semana, estos procesos automatizados pueden ser programados innumerables veces para cualquier horario del día.

“Desde la década de los 70 del siglo pasado, se ha ido insertando sistemas informáticos en las empresas, principalmente para realizar operaciones administrativas concretas como elaboración de nóminas y automatización de la contabilidad” (Solana Álvarez, 2014). Las empresas manejan sistemas de información, estos SI son clave para la gestión de los procedimientos que realizan a diario estas áreas. Estos sistemas son un gran apoyo para las empresas, ya que permiten agilizar los procedimientos y reducir costos de personal, ya que todo se realiza de manera sistematizada.

“Es importante tener implantada la automatización en las entidades que así lo requieran, pero es mejor que se necesite de un sistema de información que rijan, interconecte, controle y administre cada uno de los entes automatizados” (Pardo & Gonzales, 2014). Es por ello que los sistemas de información son vitales en toda empresa, deben ser manejados de manera adecuada y realizar un óptimo uso de ellas. Ya que dependiendo de esos factores se determinará el éxito del negocio.

Es necesario testear de forma correcta cuando se está probando los casos de prueba en un software, para evitar la filtración de posibles bugs e inconsistencia en el código, como redundancia de datos y/o posibles incompatibilidades en algunos navegadores.

La empresa “Pandora Technologies” ubicada en el Cercado de Lima, tiene una población de 20 trabajadores y cuenta con 10 técnicos.

La institución fundó la empresa a mediados del año 2015, comenzando con esto a ampliar su cartera de cliente poco a poco en el mercado peruano, exponencialmente fue

creciendo la demanda para el desarrollo de software, la cual requeriría de herramientas informáticas que ayuden al proceso de testing.

Debido a la fuerte tarea de automatizar los procesos de pruebas de regresión se puso a disposición trabajar con la dimensión de eficacia donde analizamos el tiempo durante las pruebas la cual se mide con los segundos transcurridos entre el inicio y fin de la prueba; y la dimensión productividad donde analizamos la cantidad de data generada dividiendo la cantidad de data generada entre la data solicitada en las pruebas.

## **1.2 Situación actual**

Actualmente se hacen las pruebas de regresión mediante un desarrollador, que manualmente realiza cada uno de los casos de prueba, donde se registra los fallos que se encuentren, los bugs, inconsistencias del aplicativo, entre otros. Careciendo evidentemente de un adecuado testing, así mismo no cuentan con un inventario óptimo de los errores de los casos de prueba, esto puede ocasionar que se vayan entorpeciendo las labores de testing. Lo que posibilita el peligro los procesos de pruebas de regresión.

## **1.3 Propuesta**

El objetivo fundamental es desarrollar un sistema que permite automatizar los procesos que han encontrado en la empresa Pandora Technologies. La automatización de pruebas es el empleo de las TICS en las fases de calidad de software. El objetivo se logrará en el período de 10 meses, período que se ha determinado y donde se han tomado preocupaciones para evitar riesgos o retrasos en el mismo para que no pueden establecer las fechas ya pactadas en el calendario.

La automatización de los procesos de pruebas de regresión busca:

- Reducir los problemas que podrían generar las actividades hechas por el hombre, debido a la enorme cantidad de datos que es generada durante las horas trabajadas.
- Mejorar los procesos y recursos, para evitar el trabajo doble que puede darse en los procesos actuales.
- Ofrecer información actual y en tiempo real del funcionamiento de los casos de prueba del software mediante la utilización de indicadores e estadísticas de rendimiento.

Cuando se automatizan los procesos, prevenir los datos duplicados y mejorar el performance en las actividades y servicios en el testing.

## **1.4 Trabajos Previos**

### **1.4.1 Internacionales**

(Rivera Martínez, 2018) realizó la investigación “Automatización de pruebas de regresión.”, para lograr el título grado profesional de magíster en tecnologías de la información, en la Universidad de Chile. Esta investigación se realizó con la

finalidad de formalizar, instrumentar, definir y conseguir un mejor nivel de eficacia y eficiencia en la ejecución en las pruebas de regresión, debido a la implementación de esta nueva herramienta se pudo formalizar un nuevo proceso en las pruebas que permitían cubrir un porcentaje más grande en las pruebas, también la reducción en los tiempos de validar los sistemas previamente testeados y la correcta mantenibilidad de los códigos de las pruebas automatizadas. Gracias al nuevo sistema las horas de trabajo de las mantenciones correctivas lograron una significativa disminución. El sistema tuvo que ser validado por el personal de QA que participó en el desarrollo del prototipo.

(Jerez Mayorga, 2017) realizó el trabajo de investigación “Análisis de métodos, técnicas y herramientas de verificación y validación de software aplicados en la dirección de tecnología de información y comunicación de la Universidad técnica de Ambato”, su objetivo general fue distinguir las herramientas de técnicas de validación y verificación de software en la cual se identificó que las características en los distintos tipos de prueba que se aplica fue desarrollado en la facultad de ciencia y tecnología. Se propuso que la aplicación de distintas herramientas técnicas; métodos de validación y verificación para poder proteger las necesidades del negocio. Se pudo crear un método para evaluar que permitía elegir las técnicas con las cuales se escogían las características del modelo, la que brindaba calidad al software.

(CÁRDENAS, 2016) en su trabajo de investigación “Automatización de casos de prueba para mejorar el proceso de calidad de software”, identificó y mantuvo los casos de prueba que actualmente tenían errores durante la ejecución. Había un total de 25 casos ya mezclados en la rama principal, pero con inconvenientes en la ejecución y que actualmente se realizan manualmente. Los procesos de prueba se alternan entre los casos manuales y automatizados, pero el enfoque de la metodología fue mantener el 80% de la suite de regresión automatizada y las pruebas manuales sean de aquellas funcionalidades en proceso de desarrollo y casos de prueba de gran complejidad. Este cubrimiento en el proceso de aseguramiento de calidad lograría mantener la confianza que brinda a sus clientes. Para lograr este objetivo, los equipos del área de tecnología deben incluir puntos dentro de su planeación para automatizar las pruebas de regresión y evitar la deuda técnica de casos continúe en crecimiento, por mi parte en conjunto con un equipo de analistas de pruebas planteamos una estrategia para mantener todos aquellos casos que ya se encuentran automatizados,

pero presentan fallos e iniciar el proceso de disminución de la suite de regresión manual. Con este plan emergente se reactivará y fomentará el progreso de las pruebas automatizadas, la capacidad del grupo de calidad y la velocidad de los equipos Scrum dentro de la compañía. Con el gran aporte de confianza que brindan las pruebas de regresión a los procesos de software, y al encontrarse migradas a un sistema automatizado en su mayoría y complementadas con pruebas complejas manuales se lograra una cobertura bastante amplia.

(Laureano Aguilar & Urquilla Osorio, 2018) en El Salvador, se publicó la tesis titulada “Aplicación Web para la Gestión Administrativa de las Unidades de Créditos y Tesorería del fondo Universitario De Protección de la Universidad de El Salvador”, para conseguir el título profesional de la carrera de ingeniería de sistemas en la cual concluye que el problema está sobrellevando la actual gestión administrativa, por la falta de digitalización en los procesos involucrados de la institución, y tiene como objetivo desarrollar un aplicativo web para optimizar estos procesos, ya que esto ayudara a tener un mejor servicio al cliente disminuyendo los tiempos de espera para conocer su estado financiero y reducir el costo porque el llenado de estos informes conlleva el uso de horas de un recurso humano. En el cual se concluyó, de análisis de viabilidad se demostró que el aplicativo es factible, dando paso al desarrollo del aplicativo con los requisitos encontrados, los cuales nos facilitan la diseño y construcción de la aplicación informática que solventa el problema principal.

(Rosero et al., 2016) en su artículo “15 años de técnicas de prueba de regresión de software – Una encuesta”, buscó, presentar un relevamiento de las técnicas de prueba de regresión de software aplicadas en los últimos 15 años; teniendo en cuenta su dominio de aplicación, tipo de métricas que utilizan, sus estrategias de aplicación y la fase del proceso de desarrollo de software donde se aplican. A partir de un resultado de 460 artículos, se identificó un conjunto de 25 artículos que describen el uso de 31 técnicas de regresión de pruebas de software. Los resultados de esta encuesta sugieren que, al momento de aplicar una técnica de prueba de regresión, métricas como el costo y la eficiencia de detección de fallas son las más relevantes. La mayoría de las técnicas fueron evaluadas con programas instrumentados (casos experimentales) en entornos académicos. Por el contrario, observamos un conjunto mínimo de técnicas de regresión de software aplicadas en entornos industriales, principalmente, bajo enfoques correctivos y de mantenimiento. Finalmente,

observamos una tendencia utilizando algunas técnicas de regresión bajo enfoques ágiles.

(Lacity et al., 2017) Service Automation: Cognitive Virtual Agents at SEB Bank. En el artículo dado por la academia de investigación Outsourcing (OU), donde tuvieron como objetivo evaluar los efectos actuales y a largo plazo el uso del software RPA y la Inteligencia Cognitiva(CA) en los servicios empresariales en la organización de los clientes. En el 2017 se analizaron e investigaron aproximadamente cerca de 20 casos de automatización. En una encuesta a principios de 2016, encontramos que el 5% de las organizaciones desarrollan capacidades de automatización cognitiva (CA) maduras, el 11% inician investigaciones de prueba de concepto y el 17% considera CA. Si bien a principios del 2017 CA es de considerar o implementar podría darse una investigación más profunda, pero contrasta con la adopción de RPA que claramente alcanzó un punto de inflexión en 2016, al ver un aumento del 68% en la adopción en todos los sectores. Cabe destacar que RPA y CA son aun tecnologías que continúan en constante crecimiento cuyo empírico en el 2017 encuentra que solo el 5% de los trabajos son totalmente automatizables utilizando las tecnologías disponibles en la actualidad, mientras que el 60% son 30% automatizables. En este sentido, solo porque un trabajo, o parte del mismo, se pueda automatizar, no significa necesariamente que lo será. Este es un corrector útil para las proyecciones más dramáticas basadas en gran medida en la capacidad tecnológica, desarrollada rápida y perfectamente, solo. Su enfoque principal en tareas, actividades y procesos, en lugar de en el número de puestos de trabajo y cómo se definen los empleos existentes, también es bienvenido y se adecua a la práctica real en las organizaciones que estudiamos, incluyendo la SEB (Willcocks, Lacity, & Craig, 2017, pág. 22).

(AMARICAI & CONSTANTINESCU, 2014) “Designing a Software Test Automation Framework” pudieron desarrollar un framework que permitía automatizar las pruebas que le hacían al software empleado por Page Object para evitar posibles fallos como redundancias y errores de tipos humanos, esto empleando distintas herramientas y técnicas. Para poder entender el proyecto se hizo los flujos de las pruebas como fuentes externas para archivos que son de localizar los datos de prueba. La cual estos Mantuvieron todos los recursos muy altos para archivos externos. Se concluye que la introducción de un framework tuvo gran impacto



verdadero en las pruebas de software, ya que evitaba al personal de desarrollo hacer tareas repetitivas, además de que las pruebas se hacen más rápidas y confiables.

(Moreano Camacho, 2014) en su tesis “Diseño de un Sistema de Gestión de Procesos de Negocio Para la Solicitud de Procura de Bienes y Servicios”, buscó el desarrollo de un sistema para la gestión de procesos de negocio que permita automatizar el rendimiento, la inspección y la eficiencia de sus procesos para progresar la capacidad de respuestas frente a diversos cambios del mercado para mejorar sus amplitudes para la solicitud de bienes y servicios de la empresa. Éste sistema se desarrolló usando la metodología BPM. En conclusión, se mejoró la gestión automatizando el proceso de proyectar bienes y asistencia para la empresa la cual genero considerablemente las mejoras en el tiempo de atención, optimizar las funciones, incrementando la productividad, salvaguardando los datos y apoyando en la toma de decisiones, satisfaciendo las expectativas del cliente y los colaboradores de la empresa con la ventaja de contribuir con el buen resultado de la empresa.

Kishenkumar (2009), en su reporte de tesis “Test Automation in Practice” presenta la automatización de pruebas en metodologías y técnicas utilizadas para introducir la automatización de pruebas son el desarrollo impulsado por pruebas y las pruebas dirigidas por datos. Los costos y beneficios de esta prueba de automatización han sido descritos y estimados. Se pudo decidir si automatizar o no las pruebas basadas en estas estimaciones. Debido a que la lógica de negocios en la aplicación web se implementó en la capa UI, no fue posible detectar errores en la implementación de la lógica de negocios mediante pruebas unitarias. Por lo tanto, se introdujo la prueba de interfaz de usuario automatizada. La prueba tuvo algunos inconvenientes, como el pedido de que la funcionalidad se construya primero. Debido a que las pruebas automatizadas de UI fueron difíciles de mantener se propuso como objetivo solo tener un 15% de las pruebas automatizadas. Para que sea de manera beneficiosa, el código fue refactorizado (la lógica de negocios se movió de la capa de UI a otra capa) para realizar pruebas de unidad elaboradas. Los costos y beneficios fueron los estimados en las entrevistas.

(Ciolli, 2007) en su tesis titulada “Testing de migración de aplicaciones distribuidas a entornos Web”, Propuso como objetivos específicos desarrollar meta modelos para poder abstraer las características en común de las aplicaciones web, diseñar una

metodología que le permita reutilizar los casos de prueba para generar herramientas automáticas para la ejecución. El sistema que estuvo puesto a prueba estuvo desarrollado en UML, por lo que había documentación en la que se pudo apoyar Ciolli para poder realizar los respectivos casos de prueba. Se vio tipos de prueba como regresión, de caja negra y de interfaces gráficas. La investigación realizada se basa del conocimiento previo sobre testing de regresión, caja negra e interfaces gráficas con el cliente. Se logró documentar el manejo de errores en la gran totalidad, se pudo adentrar en los procesos de validación y verificación de los sistemas cumpliendo los estándares requeridos por los jefes del área de desarrollo de software.

(Dalebout, 2018), en su investigación de su tesis “The effects of Robotic Process Automation on FTE effort and the design of the jobs involved”, buscó explorar los efectos de la implementación de la automatización robótica de procesos (RPA) en el esfuerzo de FTE y la influencia en los trabajos involucrados. De este modo, se esclarece el campo de la RPA y se construye una base para futuras investigaciones sobre el tema. La investigación se basa en una filosofía interpretivista y utiliza un enfoque inductivo, donde los datos cualitativos se recolectan a través de entrevistas con gerentes y empleados que están directamente involucrados en la implementación de RPA. Los hallazgos indican que la implementación de RPA contribuye al enriquecimiento y ampliación de los trabajos involucrados. No hubo observaciones que indicaran la existencia de una brecha en el esfuerzo FTE. Sin embargo, los proyectos de RPA discutidos tienen poca madurez y tamaño. Las implicaciones gerenciales de los hallazgos indican la necesidad de anticipar el aumento de las habilidades requeridas de los empleados y la redistribución de tareas después de la implementación de una RPA. Además, los hallazgos contribuyen al enriquecimiento de la literatura sobre automatización al esclarecer los beneficios y amenazas de la RPA. Las limitaciones de los hallazgos se deben principalmente al tamaño de la muestra. Finalmente, se formulan las limitaciones del trabajo y recomendaciones para futuras investigaciones.

#### **1.4.2 Nacionales**

(Capcha Coronado, 2018) en su trabajo de investigación “Implementación del Framework de automatización de proceso de QA en un proyecto de diseño de

software en una consultora” identificó que tenían problemas en el proceso de QA con inconvenientes en la ejecución ya que actualmente se realizaban manualmente. Para el presente proyecto se decidió utilizar una metodología ágil Scrum por que se adapta con el perfil de la empresa Belatrix, ya que cuenta con un equipo de desarrolladores de software. Este cubrimiento en el proceso de aseguramiento de calidad logró mantener la confianza que brindo a sus clientes. Entre las actividades de QA se halló las siguientes: implementación de casos de prueba, realización de las pruebas exploratorias, aplicación de casos de pruebas creados, registra las incidencias y las pruebas de regresión. Gracias a la implementación se pudo reducir los tiempos en la realización de casos de prueba, de tres días a 3 horas, la intervención del trabajador durante la aplicación no fue necesaria, también se agregó una actividad en el transcurso de calidad de software se capacitó a todo el equipo del área de QA para poder ejercer correctamente las funciones de automatización.

Según (Quispe Tapara, 2018) en su tesis “Diseño de un sistema de riego automatizado por aspersión para viveros de café utilizando la tecnología Arduino en la empresa Viveros Ortiz –Pasco”, con el motivo de optar por el título de ingeniero de sistemas de la “Universidad Católica Los Ángeles de Chimbote”, su objetivo fue realizar el Diseño de riego automatizado por aspersión utilizando la tecnología Arduino, que ayudara a mejorar el tiempo y el ahorro del agua a la hora de trabajar, la cual menciona que su investigación es cuantitativa de diseño no experimental, descriptivos, transaccional. La población de esta investigación son los empleados de la empresa y la muestra son 5 de ellos, recolectaremos datos mediante el cuestionario o también con la técnica de encuestas ya que esto permitió conocer el resultado que la dimensión de la satisfacción se observó que el 100% acepta que está conforme con el sistema de riego manual; con la siguiente dimensión nos muestra que el 100% está de acuerdo en conocer y mejorar el sistema de riego. Este resultado, corresponde con la hipótesis específica y en consecuencia confirma la hipótesis general, quedando así demostrada y justificada la investigación de diseño.

(Ventura Labrin, 2015) en su tesis” Automatización del Proceso de Ventas y Distribución utilizando tecnología móvil y Geolocalización para la empresa líder SRL”, buscó diseñar un Sistema Web Móvil para generar los Pedidos aplicada a la Empresa Casinillo en base al uso del Framework JQuery Mobile, donde se realizó varios pruebas y se llegó a detectar el problema que tenía la empresa que era que se

tomaba bastante tiempo en realizar el pedido, mediante este sistema se pudo solucionar el problema gracias a la automatización, agilizando los pedidos del cliente, de esta manera se permitió que toda la información que se tomaba en la línea principal de la empresa cuya facilidad de uso era de manera ágil y que estuviera presente en todos los dispositivos móviles y sea de manera dependiente. En la investigación se utilizó la metodología RUP. En conclusión, mediante la aplicación del entorno sumamente cerrado por los usuarios y los colaboradores de esta empresa, calculó el tiempo que demora la atención de los clientes y también se calculó el costo de su atención.

(Peña Rondoy, 2018) en Piura en su tesis titulada “Implementación de un Sistema Informático de Gestión de Pagos en el Instituto de Educación Superior Tecnológico Privado Santa Úrsula-Sullana”, para obtener por el título de ingeniero de sistemas, En la problemática que se encontró en dicha institución está por ejemplo los procesamientos de datos al encontrar dificultad a la hora de registrar las evaluaciones y pagos de los estudiantes, es tedioso y en muchas ocasiones no se logra el objetivo planificado. Para este problema se desarrolló un sistema que automatice y sea capaz de registrar, almacenar, conservar y procesar los datos, a través de un programa computarizado que optimice los tiempos y los procesos que existen en él. Y se concluye que la mejora del sistema de gestión de pagos se logra reducir la pérdida de información y se pueden agilizar el proceso de pagos mucho más eficaz y segura.

(Cahill, 2017) A Multi Stakeholder Perspective on Audit and Automated Compliance: Bank of Ireland. El siguiente informe demuestra la utilidad RPA para automatizar una gama de procesos en el Banco de Barclays en Irlanda. Verificando desde la detección de fraudes y el monitoreo de riesgos, hasta la automatización de la apertura de cuentas. La tecnología de RPA permite a Barclays bank escalar rápidamente su capacidad para procesar las solicitudes de los clientes y las crecientes necesidades comerciales, al tiempo que mantiene procesos de calidad para sus clientes. La adopción exitosa de la automatización robótica aún enfrenta desafíos, particularmente durante la fase de implementación. Por lo general, surgen problemas cuando las organizaciones pasan por alto el grado en que las partes interesadas han "comprado" el nuevo sistema o se han preparado para ello. Los ejecutivos que no se apropian del concepto, por ejemplo, probablemente dificultarán la implementación multifuncional. Dando como conclusión que los datos si apoyaron a los objetivos de

la empresa, sentido de que la administración y los usuarios se alinearon estratégica y operativamente con la automatización estratégica y operativamente con la automatización, la auditoría y el cumplimiento automatizado. Sin embargo, las perspectivas sobre Auditoría, Técnica y Riesgo se opusieron. La regulación sobre el cumplimiento automatizado será un desafío para el Banco de Irlanda en el futuro.

(Calva Carhuamaca, 2017) en su tesis titulada “Sistema que reemplaza funciones de un operador humano durante la validación de documentos digitales en Core Andina Group”, comprendió el análisis, desarrollo e implementación de un sistema que reemplazó funciones de un operador, esta tesis tuvo un impacto positivo en favor de la empresa, ya que gracias al sistema implementado se pudo ahorrar cerca de 40000 nuevos soles, además que los reportes a los cliente mejoró en un 70%, en los documentos registrados se llegó a la importante cifra de 96% de éxito, lo que comprende un incremento de 20.10% en el proceso, 12.79% nos dice que ha disminuido porque los documentos fueron rechazados, dejando solo la cifra de 3.30% en documentos sin éxito de entrega. El autor confirma que los robots son más confiables que los desarrolladores debido a que estos no sienten cansancio y tienen menor índice de error durante sus actividades laborales. El sistema fue implementado en JAVA y se utilizó librerías que eran libres.

Según (Bughin, Jacques; Hazan, Eric; Manyika, James; Woetzel, 2017) en su investigación titulada “Artificial Intelligence – The next digital frontier?”. Tiene como propósito principal el siguiente objetivo principal, dar a describir cómo las compañías ya han comenzado a implementar estas nuevas tecnologías basadas en IA y RPA en todos los sectores y apunte a explorar su potencial midiendo el alcance de la adopción y el uso de la IA en 13 sectores en los 10 países que fueron encuestados (n= 3,073). Para ello, observamos la IA a través de varios lentes. Analizando la panorámica de la especulación total reuniendo tanto la especulación de considerables organizaciones como la de subvención de haberes de contingencia y bienes de haberes reservados. También revisamos los juegos de cartera de las principales compañías de Internet, la dinámica de la holocenosis de IA de Shenzhen a Nueva York y una extensa escala de formación de casos. Como parte de nuestra investigación principal, encuestamos a más de 3,000 ejecutivos sénior sobre el uso de las tecnologías de inteligencia artificial, las perspectivas de sus empresas para un mayor despliegue y la impresión de la inteligencia artificial en los mercados, los

gobiernos y las personas. Llegando a la conclusión que RPA mejora aproximadamente en un 20% la productividad. Usar la IA para mejorar la eficiencia de los humanos es un habilitador crítico de la productividad y la calificación dada a través de 16,000 ensayos que los instructores realizaron se dio por hecho que mejora en un 85% el rendimiento en el tiempo.

(Rosa Raymundo, 2013) realizó la tesis “Gestión de la empresa QA e implementación del proceso de auditoría, servicio de automatización de pruebas y administración del conocimiento”, se propuso examinar el efecto de la automatización de pruebas en los probadores de software. Por lo que, a lo largo del tiempo, las pruebas habían creado una disciplina distinta al desarrollo de software. El software lo ha vuelto cada vez más complejo y distribuido. La presión para lanzar en producción varias veces al día ha aumentado la importancia estratégica de la automatización de pruebas. Para utilizar métodos tanto cualitativos como cuantitativos, surge una imagen que contrasta con la estrategia de desarrollo promovida por la Compañía de Servicios en la Nube donde se realizaron las entrevistas semiestructuradas. Después del análisis de los datos, el hecho de que una gran cantidad de pruebas llevadas a cabo sea todavía manual fue inesperado y significa que aún se requieren comprobadores manuales. Sin embargo, los problemas que impiden que la automatización sea más exitosa e implementada de manera general pueden resolverse y pueden solucionarse. Abordado en los próximos cinco años o menos. La conclusión de esta investigación sugiere que las personas que actualmente realizan pruebas manuales de software están en peligro de ser reemplazadas. Para sobrevivir en la industria del software, deberán volver a formarse y aprender a codificar. De lo contrario, sus posiciones quedarán obsoletas.

(Tasato Cánepa, 2014) realizó la investigación “Desarrollo de una infraestructura de software para realizar pruebas automatizadas de sistemas de información desarrollados en lenguaje cobol en el contexto bancario.”, se propuso mejorar la infraestructura que permitirá hacer la automatización de pruebas en sistemas web que estaban presentes en Cobol; para esto se desarrolló un elemento que permitió registrar los casos de pruebas con el principal propósito de poseer permisividad de sustituirlos en diversas ejecuciones de pruebas, la gestión de grupo de pruebas como crear, agregar o eliminar prueba, la gestión de la ejecución de los casos de prueba y los resultados logrados; y la generación de reportes para colaborar la gestión del

personal de pruebas de software. Para validar la infraestructura se desarrolló un sistema reducido en Cobol de modo que se permitirá verificar la facilidad que brinda la infraestructura de pruebas.

(Cárdenas Ospina & Rodríguez Beltrán, 2011) en su trabajo de investigación “Sistema de Gestión de pruebas para productos de software”, identificó y mantuvo los casos de prueba que tenían errores cuando estaba ejecutada. Se obtuvo un total de 25 casos de prueba que tenían inconvenientes en la depuración del sistema. Se encontró varios problemas durante el desarrollo del presente trabajo, los analistas de sistemas tenían que realizar distintas funciones, lo cual perjudicaba sus labores en la empresa, además no se contaba con tecnología adecuada para resolver las demandas del negocio que necesitaban ser puesta en ejecución en computadoras de última generación. En el presente trabajo de grado se explicó de manera detallada la necesidad de las empresas de software y los beneficios que trajo a todo el equipo de desarrollo. Se presentó un prototipo del sistema que reunía las demandas actuales de tecnología.

## **1.5 Teorías relacionadas al tema**

### **1.5.1 Sistema web**

Según (Berzal et al., 2010) señalan que:

Define como sistema web a aquellas aplicaciones cuya interfaz está desarrollada por parte de los sistemas. Las páginas web son archivos de texto en un único tamaño designado HTML [HyperText Markup Lenguaje]. Estos datos se acumulan en uno o varios servidores web los cuales se acceden mediante el protocolo HTTP [HyperText Transfer Protocol], como el reglamento más usado de Internet (p. 1)

Según (Luján Mora, 2002) señala también que:

Una aplicación web se menciona como un sistema donde el cliente o usuario final mediante un explorador web ejecuta peticiones a un sistema cualquiera de otro servidor mediante internet (o a través de un servidor

local) y este recibe respuestas que se muestran en el propio terminal de los usuarios (p. 2)

Según (Talledo Pascual, 2018) indica que:

Otra manera de definir podría ser como un grupo de aplicaciones web que están en constante comunicación unas con otras y con diversos medios de un servidor online con otros o consigo mismo, incluidas las data bases. Las interacciones permiten desarrollar pequeñas aplicaciones como catálogos de servicios, productos virtuales o una tienda virtual, la administración de contenidos y noticias. Pueden beneficiar optimizar tareas administrativas, etc. (p. 71)

#### **1.5.1.1 Ventajas**

Según (Berzal et al., 2010) algunas ventajas de utilizar sistemas web son:

Entre tantas ventajas que existen al usar los sistemas web las más destacadas son su accesibilidad (se puede acceder desde cualquier terminal), su fácil mantenimiento (las aplicaciones cada cierto tiempo se actualizan para mejorar procesos), su relativa seguridad (el código fuente de las aplicaciones no puede ser cambiado ni modificado por el usuario final) y su escalabilidad (se utiliza una arquitectura con varias capas y clusters de PCs que permiten ampliar el número de clientes para que puedan acceder al servicio web)(p. 19)

#### **1.5.1.2 Estructura de las aplicaciones web**

Según (ARANDA et al., 2009) en su proyecto de investigación señalan que:

Debido a que se pueden encontrar un sinnúmero de posibilidades, un Sistema Web básicamente está compuesto en una estructura de 3 capas. Mayormente se puede encontrar, el explorador o navegador de internet donde está la primera capa y una tecnología Web con un motor web capa de leer varios lenguajes como (ejemplo: Java Servlets o ASP, PHP, embPerl, Python (lenguaje de programación), Ruby on Rails, ASP.NET, CGI o ColdFusion,) que están en la capa media. Finalmente, las bases de datos constituyen la tercera capa (p. 15)



### **1.5.1.3 Protocolo SOAP**

SOAP es un acrónimo de Simple Object Access Protocol. Es un reglamento del mensaje basado en XML para cambiar información entre el computador. SOAP es una aplicación de la especificación XML.

A continuación, se mencionan algunos puntos importantes que el usuario debe tomar en cuenta.

SOAP es un protocolo de comunicación diseñado para comunicarse vía internet.

SOAP puede extender HTTP para mensajes XML.

SOAP proporciona transporte de datos para servicios web.

SOAP puede intercambiar documentos completos o llamar a un procedimiento remoto.

SOAP se puede utilizar para transmitir un mensaje.

SOAP no depende y es libre de la plataforma y del lenguaje.

SOAP es la forma XML de definir qué información se envía y cómo.

SOAP permite a las aplicaciones cliente conectarse fácilmente a servicios remotos e invocar métodos remotos.

Si bien SOAP se puede usar en una variedad de sistemas de mensajería y se puede entregar a través de una variedad de protocolos de transporte, el enfoque inicial de SOAP es el de las llamadas a procedimientos remotos transportadas a través de HTTP.

Otros marcos, incluidos CORBA, DCOM y Java RMI, proporcionan una funcionalidad similar a SOAP, pero los mensajes SOAP están escritos completamente en XML y, por lo tanto, son únicamente independientes de la plataforma y el idioma.

### **1.5.1.4 Arquitectura MVC**

“Es un patrón arquitectónico comúnmente utilizado para desarrollar interfaces de usuario que divide una aplicación en tres partes interconectadas. Esto se hace para separar las representaciones internas

de información de las formas en que la información se presenta y acepta del usuario. El patrón de diseño MVC desacopla estos componentes principales permitiendo la reutilización del código y el desarrollo paralelo”. (Eslava, 2013, p.109).

Podemos decir que el modelo vista controlador es popular porque aísla la lógica de la aplicación de la capa de interfaz de usuario y admite la separación de información. Aquí, el Controlador recibe todas las solicitudes de la aplicación y luego trabaja con el Modelo para preparar los datos que necesita la Vista. La Vista luego usa los datos preparados por el Controlador para generar una respuesta presentable final.

Los autores James Coplien y Trygve Reenskaug nos dicen que se componen de 3 capas.

●**Modelo:** “Las clases de modelo se utilizan para implementar la lógica de los dominios de datos. Estas clases se utilizan para recuperar, insertar o actualizar los datos en la base de datos asociada con nuestra aplicación.”. (Flórez, 2012, p.212)

El modelo hace posible el acceso a la información y contienen las siguientes funciones (insertar, presentar, actualizar y eliminar) que interactúan con las bases de datos.

●**Vista:** “Las vistas se utilizan para preparar la interfaz de nuestra aplicación. Al usar esa interfaz los usuarios interactúan con nuestra aplicación” (Eslava, 2013, p.109).

Las vistas son todas aquellas capas de visualización que muestran la información al usuario final que interactúa con ellas.

●**Controlador:** “las clases de controlador se utilizan para responder a las solicitudes del usuario. Las clases de controladores realizan las acciones solicitadas por los usuarios. Estas clases trabajan con clases modelo y seleccionan la vista apropiada que debe mostrarse al usuario de acuerdo con las solicitudes del usuario.”. (Flórez, 2012, p.212).

Por lo que el controlador es el nexo que une las capas del modelo y la vista en las aplicaciones. En él yace toda nuestra lógica de negocio.

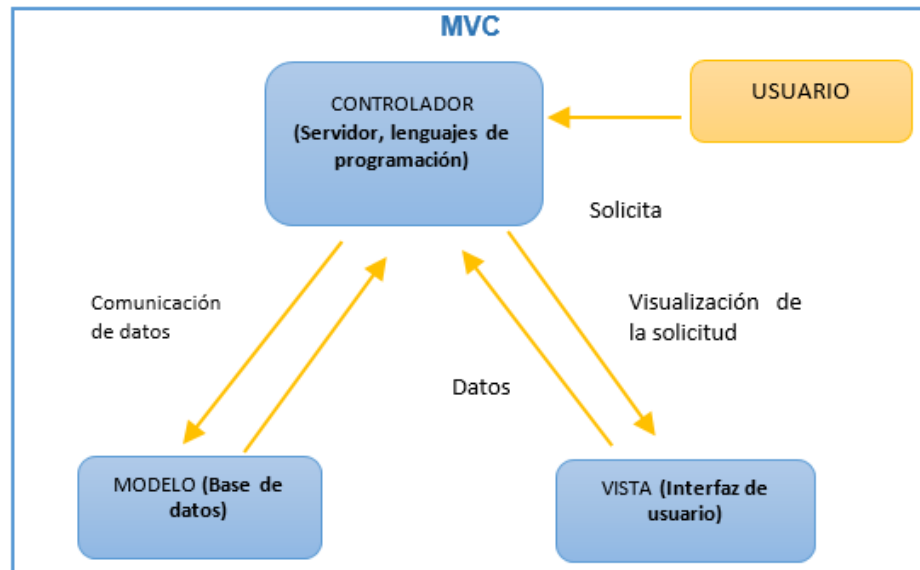


Figura 1. Arquitectura MVC

### 1.5.2 Tecnología para el desarrollo

Para el desarrollo del presente sistema se utilizaron las siguientes herramientas:

a) TypeScript: Es un lenguaje de programación orientado a objetos desarrollado y mantenido por Microsoft Corporation. Es un superconjunto de JavaScript y contiene todos sus elementos y podemos decir que TypeScript es un JavaScript moderno con clases, tipos opcionales, interfaces y más.

b) JQuery: Es una biblioteca de código abierto de JavaScript que simplifica las interacciones entre un documento HTML / CSS o, más precisamente, el Modelo de objetos de documento (DOM) y JavaScript.

c) NPM: Es el Administrador de paquetes de nodos desde el cual puede obtener millones de paquetes de JavaScript al instalarlo en su aplicación.

d) Microsoft Visual Studio: Es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft para desarrollar GUI (interfaz gráfica de usuario), consola, aplicaciones web, aplicaciones móviles, nube y servicios web, etc. Con la ayuda de este IDE, puede crear un código administrado como así como el código nativo. Utiliza las distintas plataformas de software de desarrollo de software de Microsoft como Windows store, Microsoft Silverlight y Windows API, etc.

e) Node.js: Es un entorno de ejecución de código abierto y multiplataforma para el desarrollo de aplicaciones de servidor y de red. Las aplicaciones Node.js están escritas en JavaScript y se pueden ejecutar dentro del tiempo de ejecución de Node.js en OS X, Microsoft Windows y Linux.

f) Git bash: Git Bash es una aplicación para entornos de Microsoft Windows que proporciona una capa de emulación para una experiencia de línea de comandos de Git.

g) Docker: Es una plataforma de tipo contenedora que agrupa su aplicación y todas sus dependencias en forma de contenedor acoplable para garantizar que su aplicación funcione a la perfección en cualquier entorno..

h) MongoDB: Es el sistema de base de datos NoSQL más popular. Es una base de datos multiplataforma orientada a documentos. Almacena documentos similares a JSON con esquemas dinámicos, lo que hace que la integración de datos en ciertos tipos de aplicaciones sea más fácil y más rápida.

i) ExpressJS: Se utiliza para el enrutamiento y el servidor de una aplicación, por ejemplo, Express se usaría para enviar solicitudes y respuestas cada vez que escriba la URL de un sitio web.

j) Chai.Js: Es una biblioteca de aserción utilizada con marcos de prueba como Mocha. La afirmación en las pruebas es una expresión que se aplicará en un estado específico de la aplicación si no hay un error en el programa. Mientras probamos las API de Node.JS, podemos usar esta biblioteca de aserciones para verificar si la respuesta recibida de un punto final de API para una solicitud específica es la esperada o no.

### **1.5.3 Java**

(Bocángel Lamas, 2016) expone que el lenguaje de programación java fue creado por Sun Microsystems publicado en la segunda mitad del año 1995 donde en ese momento se volvió uno de los lenguajes de programación con mayor reconocimiento. Siendo un lenguaje muy valorado ya que los programas se pueden ejecutar en diferentes plataformas con sistemas operativos (Windows, Mac OS, Linux o Solaris. James Gosling), el cabecilla del conjunto de la labor apoderado de hacer crecer a Java, prometió un

lenguaje libre de la plataforma. Buscando proyectar un lenguaje que acogiera programar un empleo una sola vez y que luego se mandara a realizar en distintas computadoras y software operativos. Para conseguir trasladar los programas Java se utiliza un ambiente de realización para los programas agrupados. Esta distribución se llama Java Runtime Environment (JRE), siendo gratis y libre para los sistemas operativos primordiales. Afirmando que el programa exacto Java conseguirá realizarse en Mac OS, Windows, Linux o Solaris (Martínez, 2015, p. 2). 3 “Write Once, Run Anywhere”, traducándose tanto “programar una sola vez y luego ejecutar los programas en diferentes sistemas operativos”, era el punto del conjunto de elaboradores de Java resumiendo la idea de portabilidad. Explicando que el programa de Java es portable, quiere declarar que son libres de la plataforma pudiéndose realizar en cualquier computadora o aparato móvil, personalmente del sistema operativo que esté establecido: Java es capaz de ser efectuado en una PC de escritorio, portátil, tablet, receptor, aparato de música o en diferente dispositivo móvil con cualquier S.O.

#### **1.5.4 MongoDB**

MongoDB es un sistema de base de datos de almacén de documentos codificado en BSON. Una base de datos MongoDB está configurada por varias colecciones y cada una contiene documentos que transportan datos. MongoDB cuenta con fragmentación automática, replicación, soporte para consultas enriquecidas e índices completos, y más.

Según el sitio web oficial de MongoDB, un conjunto de réplicas es un grupo de instancias que llevan el mismo conjunto de datos. Un conjunto de réplicas se compone de varios nodos portadores de datos, uno y solo un nodo se asigna como nodo primario, los otros nodos se asignan como nodos secundarios. El nodo primario tiene todas las operaciones de escritura. Los nodos secundarios replicarán las operaciones del nodo primario y las aplicarán a sus propios conjuntos de datos. Cuando el nodo primario no está disponible, un nodo secundario elegible se convertirá automáticamente en un nuevo nodo primario, lo que garantiza que siempre haya un nodo primario disponible para todas las operaciones de escritura.

### 1.5.5 Automatización de pruebas

El primer concepto relacionado al *testing* automatizado es *script* de prueba, el cual representa el paso a paso que se debe ejecutar de forma manual para validar una condición del sistema. Cuando se transforman estos *scripts* de prueba, en *scripts* automatizados utilizando herramientas especializadas aparece el concepto de *testing* automatizado y es usado generalmente para automatizar las pruebas de regresión (McKay & Bath, 2014).

Cuando estos *scripts* automatizados son utilizados en una iteración de un proceso de prueba, considerando, además los datos que se utilizarán para la realización de éstas, aparece el término de test de ejecución.



Figura 2. Transformaciones de un CP en test de ejecución.

La transformación de los *scripts*, que se visualiza en la Figura 2, es lo que se considera tradicionalmente como automatización en las pruebas de sistemas.

Se puede destacar como principal beneficio al realizar la automatización de las pruebas de regresión: la reducción del tiempo de duración y esfuerzo en la ejecución, posibilidad de repetición automática y amplitud de cobertura, permitiendo entregar software de mayor calidad en comparación con las pruebas manuales (Madi, 2013).

#### 1.5.5.1 Métodos de testing automatizado

Las herramientas de automatización no son suficientes para realizar la automatización de pruebas, también es necesario considerar el método a usar para realizar correctamente el *testing* automatizado y así evitar realizar doble trabajo.

En esta sección se presentarán tres métodos que pudieran usarse en beneficio de una correcta automatización de pruebas: descomposición funcional, keyworddriven y action-based testing, cuyos enfoques permiten identificar

elementos a considerar en la construcción de los *scripts* automatizados. Estos son: funcionalidades comunes en los casos de prueba de un sistema, pruebas estándares, formatos y reglas generales de cómo las pruebas se deben documentar y administrar, estas últimas tareas son muy importantes pero pocas veces se consideran en las automatizaciones de prueba (Yépez La Rosa & Márquez Caicedo, 2004).

#### **1.5.5.1.1 Método de descomposición funcional**

Para construir un *script* de prueba con el método de descomposición funcional, se debe reducir todos los casos de prueba a sus tareas más esenciales y escribir *scripts* definidos por el usuario, *scripts* de funciones de negocio y *scripts* de rutinas que efectúan esas tareas independientes unas de las otras. Para lograr la descomposición es necesario separar los datos de las funciones. Esto permite escribir un *script* de prueba automatizado para una función, usando archivos de datos para proporcionar tanto el input y la verificación de los resultados esperados. Se emplea un diseño estructurado o modular (Hayes, 1996).

#### **1.5.5.1.2 Método keyword-driven**

Para construir un *script* de prueba con el método keyword-driven, los casos de prueba deben estar escritos en una hoja de cálculo la que contiene keyword y/o valores especiales que son leídas por los *scripts* automatizados, estas palabras claves tienen un significado especial y generan una acción. El proceso completo es totalmente dirigido por los datos (Zambelich, 1998).

#### **1.5.5.1.3 Método Action-based testing (ABT)**

La principal diferencia de este método en comparación de los anteriores es su evolución en cuanto a la abstracción en la escritura de los *scripts*. No se muestran detalles sobre cómo sería ejecutada la prueba, cuáles serán los botones a presionar o qué campos se deben rellenar en los cuadros de diálogo. De existir acciones de verificación de estado

durante la ejecución de la prueba, no necesita especificar dónde capturar el valor real de comparación. Este valor podría ser capturado desde una ventana, pero también podría ser recuperada de una base de datos con una consulta SQL.

En ABT, las pruebas están organizadas en módulos de prueba. En cada módulo de prueba, se escribe como una serie de acciones, compuesto de una palabra clave de la acción y argumentos que representan los datos de entrada o los resultados esperados. Esta forma de trabajo mejora la lectura de las pruebas y tiende a ahorrar enormemente la capacidad de mantenimiento debido a que los esfuerzos de automatización se centran exclusivamente en las palabras de acción (Buwalda, 2011).

#### **1.5.5.2 Herramientas relacionadas**

Al realizar una investigación para el desarrollo de este proyecto, se han encontrado muchas herramientas de soporte para los distintos aspectos relacionados a procesos de automatización de pruebas de software, según su función o enfoque las podemos clasificar en:

Herramientas enfocadas al desarrollador (útiles para las pruebas unitarias).

Herramientas para la administración de pruebas (útiles para gestionar las pruebas).

Herramientas que permiten desarrollar las pruebas funcionales (útiles en las pruebas de regresión).

Herramientas para aplicar pruebas de carga.

Y, por último, herramientas para el manejo de defectos.

Todas estas herramientas pueden funcionar de forma individual o integrada ejecutando las distintas funciones para las que fueron construidas. En los siguientes párrafos de esta sección se discute únicamente las herramientas de *testing* funcional, que es el tipo de *testing* que se enfoca el proyecto de tesis.

Estas herramientas permiten revisar que las aplicaciones funcionan de acuerdo a lo esperado, proveen facilidades para captura y ejecución



automática (reproducir la captura) permitiendo grabar una aplicación a nivel funcional (tradicionalmente se utilizan para grabar pruebas de regresión). Una condición importante cuando se elige la herramienta de automatización, es la plataforma de ejecución de las aplicaciones que se requiere automatizar y donde estas tienen posibilidad de alcance, siendo hoy en día la plataforma web, el más común donde se han enfocado estas herramientas.

En este contexto existen varias herramientas gratuitas que permiten realizar las pruebas funcionales, emular el usuario final y poseen interfaz de grabación, por nombrar algunas importantes: Sahi (Sahi, 2016), WatiN (Watin, 2013) y Selenium (Selenium, 2016).

Por mencionar solo algunas características de estas herramientas, todas ellas permiten testear una aplicación web a través de cualquiera de los navegadores más importantes en la actualidad: IE, Firefox, Opera, Chrome y Safari. Se diferencian principalmente en el lenguaje de programación que ha sido usado en la construcción del *script* automatizado. Sahi implementa los *scripts* en JavaScript, WatiN implementa los *scripts* en .Net y Selenium implementa los *scripts* en Java como lenguaje nativo y también en C#, Perl, PHP, Python y Ruby.

Al respecto, (Mendoza, 2011) realizó una evaluación de todas estas herramientas, asignándoles a los diferentes criterios de evaluación una puntuación entre 1 y 10 y a cada criterio un peso entre 1 y 100. La herramienta que obtuvo una ponderación mayor fue Selenium en los conceptos que se visualizan en la Figura 3. Esta ponderación fue muy útil para la selección de la herramienta a considerar para el proyecto de tesis, sobre todo en lo que representa la estabilidad y mantenimiento de los *scripts* y la facilidad de uso que se indica ya que no se cuenta con experiencias previas dentro de TI en materia de automatización.

Concepto	Peso	Sahi	Selenium	Watir
Documentación	5%	8	10	7.4
Soporte online	5%	8	7	5
Actualizaciones de software	5%	9	9	9
Compatibilidad con lenguajes de programación	10%	8	10	7.4
Estabilidad y mantenimiento de los scripts	30%	4.7	8	8
Organización de los tests	10%	7	8	4
Compatibilidad de la interfaz de grabación	5%	10	4	4
Calidad de la interfaz de grabación	5%	6.2	7.1	5.5
Ejecución	10%	5.5	10	3.5
Facilidad de uso	15%	7.2	8.6	7.2
	100%	6.3	8.345	6.515

Figura 3. Resumen del análisis de comparación (Mendoza, 2011).

A continuación, se proporciona más detalle solo de la herramienta Selenium.

Selenium está compuesta por diferentes herramientas o componentes de software, cada una con un enfoque diferente para el apoyo a la automatización de pruebas, proporcionando distintas soluciones para abordar los problemas de automatización.

Estas herramientas dan lugar a un gran conjunto de procesos de prueba dirigidos exclusivamente a las necesidades de las pruebas de aplicaciones web de todo tipo, proporcionando operaciones muy flexibles para la localización de elementos de interfaz de usuario y la comparación de resultados de la prueba contra comportamiento de la aplicación real. Una de las más importantes características de Selenium es el apoyo para la ejecución de las pruebas en múltiples plataformas de navegadores.

La principal diferencia de Selenium y otras herramientas de prueba para aplicaciones en web, esta utiliza un navegador real impulsado a través de JavaScript para reproducir los *scripts* de pruebas. Esto significa que tiene algunas características únicas que no están disponibles en ninguna otra herramienta de prueba web (Gheorghiu, 2005).

Las herramientas o componentes de la suite Selenium son:

Selenium IDE: Es un entorno de desarrollo integrado para *scripts* de Selenium. Se implementa como una extensión de Firefox por intermedio de un plugin que permite grabar, editar y reproducir las pruebas. Provee el entorno ideal para crear *scripts* automatizados de manera rápida.

Selenium Remote Control (RC): Corresponde a la primera versión de Selenium que permitió la ejecución de *scripts* automatizados considerando una arquitectura cliente – servidor. Hoy esta versión de Selenium está oficialmente obsoleta, la nueva versión es Selenium WebDriver, sin embargo, como contexto, el servidor aceptaba comandos vía http desde el cliente. La principal ventaja de RC, permitía la ejecución de los *scripts* en varios lenguajes de programación (Java, C#, Perl, PHP, Python y Ruby) para una mejor integración de Selenium a entornos de prueba existentes.

Selenium WebDriver: Es una API orientada a objeto que permite ser usada desde otras herramientas IDE de programación. La principal ventaja de esta herramienta es que permite construir *scripts* de automatización de manera escalable, extensible, modificables y reutilizables. Estas características impulsaron a esta herramienta a ser considerada como el sucesor de Selenium RC. Selenium WebDriver tiene componentes que permite la captura de páginas web dinámicas, donde los elementos de una página pueden cambiar, sin que la propia página se vuelva a cargar. El objetivo de WebDriver es suministrar una API orientada a objetos bien diseñado que proporciona un soporte mejorado para los modernos problemas avanzados de pruebas de aplicaciones web.

Selenium Grid: permite ejecutar las pruebas en diferentes máquinas con diferentes navegadores en paralelo. Es decir, la ejecución de varias pruebas al mismo tiempo considerando distribuir la ejecución de las prueba.

A continuación solo se realiza un bosquejo de la arquitectura física y lógica de la herramienta Selenium IDE, que se utilizará para una prueba conceptual.

#### Arquitectura física de Selenium IDE

Selenium IDE en la documentación formal de su sitio web, no muestra de forma explícita un diagrama de arquitectura que permita respaldar

teóricamente el diseño generado. Sin embargo, de acuerdo a las instrucciones de instalación podemos definir que se trata de una arquitectura física monolítica mostrada en la Figura 4.

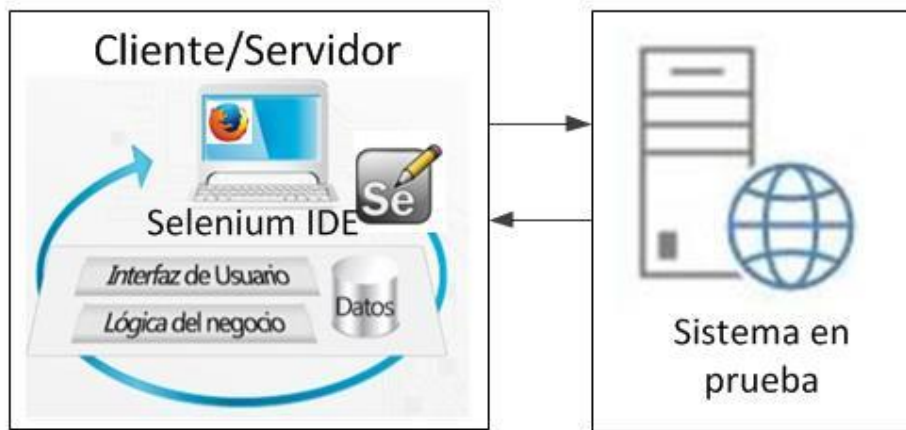


Figura 4. Diagrama de arquitectura monolítica de Selenium IDE.

La arquitectura soporta múltiples usuarios ya que solo usa los recursos de la máquina cliente, permitiendo ejecutarse en varias instancias del navegador Firefox.

#### Arquitectura lógica de Selenium IDE

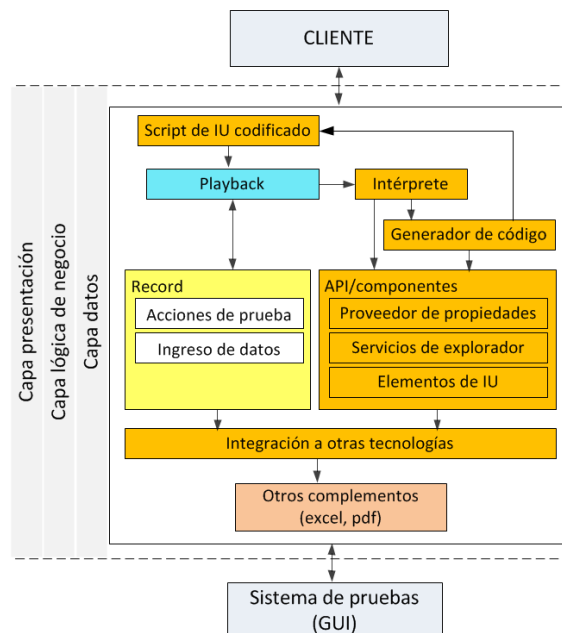


Figura 5. Diagrama de arquitectura lógica de la aplicación Selenium.

Selenium IDE por ser una herramienta de entorno de desarrollo integrado y ejecutado en un 100% desde el cliente, contiene en un solo programa las 3 capas en el contexto de arquitectura lógica de sistemas multicapa: capa de presentación, capa de negocio y capa de datos. Como se visualiza en la Figura 5.

Dentro de los beneficios de esta arquitectura se encuentra el bajo costo que significa su implementación, opera en una misma máquina tanto el cliente como servidor, sin embargo, una desventaja importante, no permite aislar la lógica de un *script* de prueba en componentes separados y hacer del proceso de automatización de testing escasamente escalable, extensible y modificable.

Es decir, no se podría separar la capa de presentación de una capa de lógica de negocio o de los datos. En varios casos de estudio citados por (Graham & Fewster, 2012), se indica a las herramientas utilizadas como responsables del no éxito de los proyectos, por lo cual, es importante considerar para quienes se inician en la automatización de pruebas de software; el uso exitoso de una herramienta en una organización, no implica que este será de beneficio para otra organización. Depende de muchos otros factores que se deben considerar, como son: los procesos, la calidad de las pruebas, el conocimiento de las personas, la arquitectura y la abstracción con que se escriben los *test* de prueba. Este trabajo expone los factores a considerar para no ser parte de las estadísticas fallidas.

Es importante entonces considerar también estos otros factores, además de la herramienta Selenium, de esta forma probablemente se lograría alcanzar los objetivos propuestos en la presente tesis.

### **1.5.6 Cross Browser Testing**

Según Paul (2018) es simplemente lo que significa su nombre, es decir, probar su sitio web o aplicación en múltiples navegadores, y asegurarse de que funcione de manera consistente y como se pretende sin ninguna dependencia, o compromiso en la calidad.

El Cross browser testing es un tipo de prueba no funcional que nos ayuda a garantizar que nuestro sitio web o aplicación web funcione como se espera en varios navegadores web. Podríamos realizar pruebas de navegador cruzado en

diferentes navegadores tanto de forma manual como automatizada. Para realizar las pruebas manualmente, nosotros (Software Testers) creamos pruebas para cada navegador y las ejecutamos manualmente en cada navegador. Para hacerlo de forma automatizada, podríamos crear pruebas de Selenium con varias declaraciones condicionales que ejecuten casos de prueba según el tipo de navegador especificado. Las pruebas en varios navegadores son simplemente lo que significa su nombre, es decir, probar su sitio web o aplicación en múltiples navegadores, y asegurarse de que funcione de manera coherente y como se pretende sin ninguna dependencia, o compromiso con la calidad. Esto es aplicable tanto a aplicaciones web como móviles.

### **¿Por qué se realizan pruebas de navegador cruzado?**

Para saber qué está mal y poder solucionarlo.

Para mejorar la eficiencia y la experiencia del usuario y, por tanto, las empresas.

Ser informado de cualquier posible trampa.

### **Importancia de las pruebas de navegador cruzado**

Facilita la gestión del sitio web y su actualización.

Proporciona funcionalidad en todas las páginas web.

Los clientes obtienen un buen acceso con facilidad de visualización

Mejora el ratio de conversión del cliente.

### **1.5.7 Rational Unifed Process (RUP)**

(Toro López, 2013) nos menciona sobre la metodología RUP que:

Esta metodología en español como como Proceso Unificado de Rational (con algunas variaciones) ya se conocía desde los años 90's. Es un proyecto de software de varios años desarrollado y utilizado por la IBM. Esta metodología de software de Rational (RUP) tiene ya varios años desde que ha sido implementado en proyecto informáticos, pero debido a su complejidad varios expertos los consideran complejo de interpretar, la

mayoría de veces por tener un concepto abstracto y lenguaje en pseudocódigo nemotécnico.

La estructura de la metodología RUP (Rational Unified Process) cuenta con 7 fases para el desarrollo de proyectos de software:

- Business Modeling (Modelo de negocio): En esta fase se trata de dejar las políticas del negocio bien en claro con lo que refiere a la información.
- Requirements (Requisitos funcionales): Esta fase tiene como propósito esclarecer los requisitos de los procesos del proyecto.
- Analysis/Design (Análisis y Diseño): En esta fase se trata de proyectar todo lo que concierne a los módulos, procesos, actividades, trabajadores y demás piezas del sistema, hablando un tipo de arquitectura óptimo según convenga.
- Implementation (Implementación): En esta etapa se montan las bases de datos, así como las interfaces previamente diseñadas.
- Test (Pruebas): El objetivo de esta etapa es realizar tantas pruebas posibles a nivel de módulos para pobrar el rendimiento de estos.
- Configuration & Change Manage (Configuración y Administración de Cambios): El objetivo de esta etapa es configurar los equipos en general y programas en las distintas interfaces y módulos.
- Deployment (Instalación): La finalidad de esta etapa es poner en marcha el producto en óptimas condiciones.

Es preferible que cada una de las interacciones estén clasificadas correctamente y sean ordenadas según sea la prioridad, además que cada una pueda ser convertida más adelante en un documento para el cliente. Esto favorecerá como una conveniente y necesaria retroalimentación que se tendría en cada paso que signifique un documento en cada fase. (p. 27)

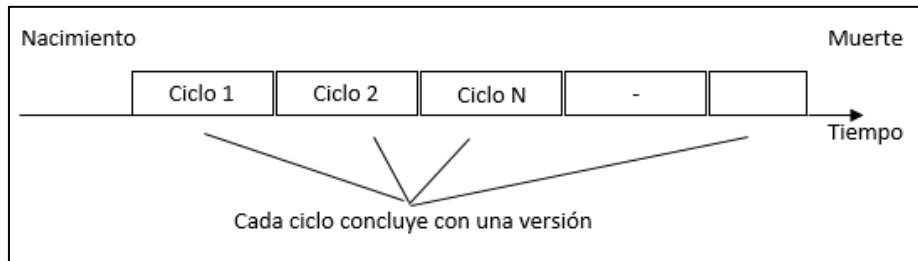


Figura 6. Ciclo de vida RUP

Fuente: Modificado de Pérez (2011)

### 1.5.8 Scrum

Con respecto a la metodología Scrum los autores (Álvarez García, Alonso; Lasa Gómez, Carmen; de las Heras del Dedo, 2011) sostienen que:

Esta metodología sostiene un marco de trabajo que da apoyo a lo ingenioso, se apoya en equipos con gestión automática. Con la metodología Scrum se obtienen resultados con alto índice de calidad en períodos cortos de tiempo (entre una y cuatro semanas) llamadas **Sprints**

De forma muy simplificada se podría resumir el flujo del trabajo con Scrum de la siguiente manera:

- El **Product Owner** escribe en el **Product Backlog** todos los servicios y requerimientos que quisiera que su entregable considere. Eso sí, debería de priorizarla indicando el orden en que quiere que se vaya construyendo su proyecto. Los aditamentos más primarios deben estar resaltados como también descritos pasa por paso de los que no que no tan importantes.
- Los programadores estimaran cada uno de estos requerimientos en congruencia a su complejidad. Priorizando los ítems marcados por el **Product Owner** y la consideración realizada por el equipo, se planificará las partes que se vaya a abordar en el siguiente Sprint. Ojo, los requisitos seleccionados no podrán cambiarse durante el Sprint.
- Empieza el Sprint y el equipo se sincronizará diariamente con la **Daily Meeting**.
- Cuando concluya el Sprint, los del equipo mostrarán el **Product Owner** el trabajo realizado que deber ser un producto potencialmente entregable. Con la opinión y sugerencias del **Product Owner** (p. 39).



### 1.5.9 Metodología XP

La revista Agile 101 (2016) nos menciona sobre la metodología XP que:

La programación extrema (XP) es un marco de desarrollo de software ágil que tiene como objetivo producir software de mayor calidad y mayor calidad de vida para el equipo de desarrollo. XP es el marco ágil más específico en relación con las prácticas de ingeniería adecuadas para el desarrollo de software. (p. 1)

Según la revista Agile 101 (2016) menciona sobre los roles XP que:

Aunque la Programación Extrema especifica prácticas particulares que debe seguir su equipo, en realidad no establece roles específicos para las personas de su equipo. Dependiendo de la fuente que lea, no hay orientación o hay una descripción de cómo se comportan los roles que normalmente se encuentran en los proyectos más tradicionales en los proyectos de Programación Extrema. Aquí hay cuatro funciones más comunes asociadas con la programación extrema:

#### El cliente

El rol del Cliente es responsable de tomar todas las decisiones comerciales relacionadas con el proyecto. Se asume que el Cliente XP es una sola persona, sin embargo, la experiencia ha demostrado que una persona no puede proporcionar adecuadamente toda la información relacionada con el negocio de un proyecto. Su equipo necesita asegurarse de que obtiene una imagen completa de la perspectiva del negocio, pero tiene algún medio de resolver los conflictos en esa información para que pueda obtener una dirección clara.

#### El desarrollador

Debido a que XP no necesita mucha definición de rol, todos los integrantes del equipo (con la excepción del cliente y un par de roles

secundarios que se enumeran a continuación) están etiquetados como desarrolladores. Los desarrolladores son responsables de realizar las historias identificadas por el Cliente. Debido a que los diferentes proyectos requieren una combinación diferente de habilidades, y debido a que el método XP se basa en un equipo multifuncional que proporciona la combinación adecuada de habilidades, los creadores de XP no sintieron la necesidad de una mayor definición de roles.

#### El rastreador

Algunos equipos pueden tener un rastreador como parte de su equipo. Este es a menudo uno de los desarrolladores que dedica parte de su tiempo cada semana a desempeñar este rol adicional. El propósito principal de esta función es realizar un seguimiento de las métricas relevantes que el equipo considera necesarias para realizar un seguimiento de su progreso e identificar áreas de mejora. Las métricas clave que su equipo puede rastrear incluyen la velocidad, las razones de los cambios en la velocidad, la cantidad de horas extra trabajadas y las pruebas de aprobación y fracaso.

#### El entrenador

Si su equipo acaba de comenzar a desarrollar en XP, puede resultarle beneficioso contar con un Coach en el equipo. Por lo general, el entrenador viene a ser un consultor externo o uno con experiencia previa en la metodología XP, este coach ayuda a respetar los principios básicos de XP, así como ayuda a mantener los roles dentro del equipo, con los clientes y usuarios finales del software.

La principal fortaleza del entrenador es la experiencia que tiene con proyectos de investigación previos, conoce de los errores más comunes y ayuda a evitar a los más nuevos del equipo a cometerlos.

Tabla 1: Cuadro comparativo RUP, Scrum y XP

Característica	RUP	SCRUM	XP
Se hereda modelos	x	-	-
Independencia de tecnologías	-	X	-
Rigurosa documentación	X	-	-
Es sistemático	X	-	X
Son enfocados en los procesos	X	-	-
Son enfocados en las personas	-	X	X
Resultados instantáneos	-	X	X
Apoyo del cliente	-	X	X
Conduce el tiempo	X	X	X
Reestructurar código fuente	-	-	X
Iteración	X	X	X
Réplica a cambios	-	X	X

Fuente: Adaptado de Pérez (2011)

## **1.6 Formulación del problema**

Problema General:

¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en las pruebas de regresión?

Problemas específicos:

¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en el tiempo de generación de data?

¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en la productividad de data generada?

## **1.7 Justificación del estudio**

Justificación Tecnológica

El nuevo sistema de automatización de pruebas que se implementará, será desarrollado en una plataforma web que actualmente es muy demandada debido a las ventajas que posee su utilización.

Como mencionaron (Guevara Bolaños et al., 2006):

La situación de las empresas actuales son los nuevos requerimientos en cuanto a calidad de testing para los programas, y los cambios constantes de la tecnología para el tratamiento de la información, son razones por el cual proponen realización de nuevos sistemas en donde se debe utilizar tecnologías y herramientas modernas, en las que se dé respuesta al difícil equilibrio planteado. El software es el producto más importante de este tiempo: información, transforma los datos personales (por ejemplos, las transacciones financieras de una empresa) de forma que los datos sean más útiles en un contexto local. (p. 121)

Justificación Económica

El sistema de información será alojado en un servidor Web y se podrá acceder a ella a través de un navegador web, no se tendrá que instalar un aplicativo o adicionales para el uso de la herramienta de pruebas de regresión.

(Martínez Orozco et al., 2014) comentan que:

La justificación económica de un proyecto es basado en la comparación de los beneficios y costos a medida que ocurren a lo largo del tiempo y con el descuento correspondiente. El principal beneficio económico de un software es el ahorro de costos laborales. El cálculo del ahorro de mano de obra debe incluir tanto la tarifa por hora como los beneficios adicionales para el ahorro de mano de obra para todos los turnos. Una inversión en un sistema se justifica económicamente en la gran mayoría de aplicaciones, los clientes pueden realizar tareas de monitoreo o control de montajes complejos situados a enormes distancias, a partir de simples estaciones de cómputo con recursos de hardware mínimos y una conexión estable a Internet. (p.77)

#### Justificación Operacional

La herramienta web será fácil de manipular, ya que un sistema web se puede programar con interfaces amigables y se mostrará un previo tutorial de su uso al ingresar al sistema para los usuarios.

(Aguaded & Fuentes, 2006) señalan que:

Para un sistema de información es de vital importancia la optimización de los procesos manuales llevado a cabo en la empresa, en este caso Pandora Technologies EIRL, esto llevará a un mejor manejo en los procesos de pruebas de la empresa, lo que permitirá reducción de costos y tiempos ayudando a los objetivos planteados por las entidades. El objetivo de controlar un proyecto es que este bien gestionado, sin deteriorar la calidad de gestión de las actividades de tipo continuo, para conseguir esto la gestión de proyectos debe apoyarse con técnicas de control. (p. 3)

#### Justificación Metodológica

El método experimental, es muy utilizado en investigaciones en donde el propósito del estudio es dar a conocer los efectos que causan estímulos o acciones sobre una realidad. Además de la simplicidad nos ayuda a brindar soluciones reales a problemáticas de todo ámbito. El diseño pre experimental nos da la facilidad de poder manipular de manera

mínima una variable conociendo los efectos y las causas de los diferentes resultados que se obtienen después de realizar una investigación científica.

## **1.8 Hipótesis**

### **Hipótesis General**

El desarrollo de un Sistema de automatización de pruebas tendrá influencia en las pruebas de regresión en la empresa Pandora Technologies.

HE1:

El desarrollo de un Sistema de automatización de pruebas tendrá influencia en el tiempo de generación de data en la empresa Pandora Technologies.

HE2:

El desarrollo de un Sistema de automatización de pruebas tendrá influencia en la productividad de data generada en la empresa Pandora Technologies.

## **1.9 Objetivos**

### **Objetivo General**

Determinar el impacto de un sistema de automatización de pruebas de regresión en la empresa Pandora Technologies.

OE1:

Determinar el impacto de un sistema de automatización de pruebas en el tiempo de generación de data en la empresa Pandora Technologies.

OE2:

Determinar el impacto de un sistema de automatización de pruebas en la productividad de data generada en la empresa Pandora Technologies.

## **II. MÉTODO**

## **2.1 Método de investigación**

El diseño experimental proviene de la palabra experimento. Un experimento refiere al estímulo o acción que se aplica en un objeto, para luego poder visualizar los efectos. (Gómez, 2006). El diseño de la investigación será experimental, ya que se manipulará de manera mínima con un estímulo o acción a la variable dependiente “pruebas de regresión” para posterior analizar los resultados y ver el comportamiento que está presente. El tipo de investigación en que nos basaremos será el pre experimental, ya que tomaremos un solo grupo para realizar un pre y post test para luego analizar y ver los resultados obtenidos.

### **2.1.1 Tipo de estudio**

(Valderrama Mendoza, 2007) sobre el tipo de estudio menciona que: “Cuando se habla de tipos de investigación, nos referimos a la clasificación (según el diccionario de investigación) [...]” (p. 28). El tipo de estudio que se utilizara es la investigación aplicada, por lo que nos basaremos en la solución a un problema real que padece la empresa Pandora Technologies. Dando a conocer nuestros conocimientos en informática con la finalidad de brindar dicha solución.

### **2.1.2 Enfoque**

El enfoque cuantitativo toma una idea y las deriva a interrogantes de investigación sobresalientes, a partir de estas se generan hipótesis y se definen variables; impone un plan para probar las hipótesis (diseño la investigación); se miden las variables en un definido entorno; se analizan las mediciones (utilizadas en estadística), además se formulan conclusiones con respecto a las hipótesis planteadas mediante informes estadísticos en donde se utilizan gráficos para su mejor interpretación. (Gómez, 2006). En el presente trabajo de investigación se mostrará un enfoque cuantitativo, debido a que se seguirá un proceso de investigación para probar las hipótesis planteadas mediante fichas de observación, análisis de datos. Se utilizarán gráficos y estadísticas para poder sustentar las conclusiones.

### **2.1.3 Alcances de estudio**

(Arias.f, 2006) con respecto al nivel de la investigación refiere que: “El nivel de investigación se refiere al grado de profundidad con que se aborda un fenómeno u objeto de estudio” (p. 23). El alcance del presente proyecto de investigación es de tipo explicativo, debido a que el propósito es poder dar a conocer las causas que originan un efecto, en las cuales participaran la variable dependiente y la variable independiente.



### 2.1.4 Diseño de Investigación

Los diseños pre experimentales se caracterizan por tener muy bajo control sobre la manipulación de las variables (Soto, 2015). El tipo de investigación será pre experimental debido a que se tomara una muestra seleccionada a una cantidad de registros de prueba y se evaluaran las vías del procedimiento por los que fueron registrados o atendidos antes y después de implementar el sistema de automatización de pruebas.

## 2.2 Variables y Operacionalización

### Dimensiones de la prueba de software

Según (Fernandez Avalos, 2018) menciona ciertos indicadores pero uno de los más importantes y lo cuales se tomaran en cuenta son los siguientes:

**Eficacia:** Es todo acerca de la comparación entre lo que realmente se está produciendo o realizarse con lo que se produce teniendo en cuenta la misma cantidad de recursos, tales como: dinero, tiempo y mano de obra. En términos más simples, la eficiencia mide si hay algún desperdicio en su empresa. Dependiendo de la industria en la que trabaja, la eficiencia puede ser más deseable que la productividad, pero por lo general su importancia es proporcional.

**Productividad:** En su esencia, la productividad le muestra la velocidad a la que se están desarrollando productos o una tarea se está completando. Cuando se mide la productividad, las cosas se vuelven un poco más complejas que eso. Hay que tomar en cuenta si se trata de un trabajo físico o en la oficina, si el trabajo requiere un cierto factor de calidad o el impacto de los requisitos de una industria específica pueden tener sobre sus trabajadores. No obstante, la productividad es una parte integral de cualquier empresa de éxito. (pág. 22)

VARIABLE INDEPENDIENTE						
	Definición conceptual	Definición Operacional	Dimensiones	Indicadores	Formula	I. E.
AUTOMATIZACIÓN DE PRUEBAS	La automatización de pruebas es el uso de un software especial (separado del software que se está probando) para controlar la ejecución de las pruebas y la comparación de los resultados reales con los resultados previstos. Las herramientas de	Las automatizaciones de pruebas de software poseen características los cuales miden su aceptabilidad, su confiabilidad, disponibilidad, consistencia. Mediante estos parámetros				

automatización ayudan a acelerar las pruebas unitarias, las pruebas API y las pruebas de GUI. ( Saravanan y Poorna, 2016, p. 104)

podremos deducir la calidad de la automatización.

Tabla 2: Operacionalización de la Variable Independiente

Elaboración propia

		VARIABLE DEPENDIENTE				I.	E.
	Definición conceptual	Definición Operacional	Dimensión	Indicadores	Formula		
PRUEBAS DE REGRESION	“La prueba de regresión es una práctica de prueba de software que garantiza que una aplicación siga funcionando como se espera después de cualquier cambio, actualización o mejora de código.” (ISQTB, 2017,p. 1)	La prueba de regresión se evaluará teniendo en cuenta los diferentes indicadores que se especifican.	Eficacia	Tiempo de generación de data.	<b>T(t)= T(f)-T(i)</b> Donde: T(t) = Tiempo de Generación de Data en segundos. T(f) = Tiempo final. T(i) = Tiempo inicial.	FICHA DE OBSERVACIÓN	RAZÓN

Tabla 3: Operacionalización de la Variable Dependiente

Elaboración Propia

		VARIABLE DEPENDIENTE				I.	E.
	Definición conceptual	Definición Operacional	Dimensión	Indicadores	Formula		
PRUEBAS DE REGRESION	“La prueba de regresión es una práctica de prueba de software que garantiza que una aplicación siga funcionando como se espera después de cualquier cambio, actualización o mejora de código.” (ISQTB, 2017,p. 1)	La prueba de regresión se evaluará teniendo en cuenta los diferentes indicadores que se especifican.	Productividad	Productividad de data generada.	<b>P=(C*100)/D</b> Donde: P=Productividad de Data Generada. C= Cantidad de data generada. D = Data Solicitada.	FICHA DE OBSERVACIÓN	RAZÓN

Tabla 4: Operacionalización de la Variable Dependiente

Elaboración propia

## **2.3 Población y Muestra**

### **Población:**

La población para este proyecto de investigación hace referencia a los casos de pruebas en el área de testing de la empresa Pandora Technologies. El área de testing se revisa alrededor de 20 registros de prueba al día y un aproximado de 600 registros de prueba al mes.

### **Muestra**

(Ortíz-Uribe & García-Nieto, 2008) señalan que:” [...] En lugar de examinar a todo el grupo llamado población o universo, se evalúa solo una cierta parte del grupo, la que se llamará como muestra” (p. 134). Tomaremos los 40 primeros registros de prueba en un rango de 7 días hábiles.

### **Muestreo**

El muestreo es el proceso que se realiza para poder escoger al conjunto de individuos que formaran parte de la muestra con el fin de evaluarlos, para ello utilizaremos el muestreo probabilístico.

El muestreo probabilístico es caracterizado por tomar la muestra circunstancialmente, por ello todos los individuos poseen la misma probabilidad de formar parte de la muestra. (BISQUERRA, 2009. p. 145). La muestra seleccionada en esta investigación posee la característica mencionada, pues serán seleccionados al azar y todas las papeletas generadas tendrán la misma probabilidad de conformar parte de la muestra.

Según (Gómez, 2006) sobre el muestreo señala que:

Se llamará aleatoria simple a una muestra que contenga las próximas características:

- Cada uno de los miembros del conjunto de la población deben de haber tenido las mismas posibilidades de ser incluido en la muestra.
- La incorporación de un agente externo en la muestra no debe de haber sido influenciado por otro agente.

- El total de las muestras de tamaño n que pudieron formarse con los elementos de la población, deberían de haber tenido las mismas posibilidades de ser seleccionadas.

De modo que todos los registros generados se tendrán en cuenta sin discriminación alguna, se realizara un muestreo aleatorio simple.

## **2.4 Técnicas e instrumentos de recolección de datos, validez y confiabilidad**

### **Técnicas**

#### **Observación**

“La observación es el método principal en la obtención de datos de la realidad, toda vez que consiste en obtener información mediante la percepción intencionada y selectiva, ilustrada e interpretativa de un objeto o de un fenómeno determinado” (Ortíz-Uribe & García-Nieto, 2008).

#### **Instrumentos:**

#### **Ficha de Observación**

La ficha de observación es un instrumento fundamental y de mayor utilidad para realizar una investigación documental. Nos permite poder recolectar y registrar información previamente analizada. (Ortíz-Uribe & García-Nieto, 2008). Los datos que se obtendrán serán plasmados en una ficha de observación, en ella se realizara el análisis de los datos conseguidos y la síntesis de los datos conseguidos a través del instrumento.

#### **Validez y Confiabilidad**

##### **Validez**

“La validez se refiere a si el instrumento vale o sirve para medir lo que realmente quiere medir” (Soto, 2015).

##### **Juicio de Expertos**

(Valderrama Mendoza, 2007) con respecto al juicio de expertos señala que:

Vendría a ser las rectificaciones y contribuciones que realizan los desarrolladores, técnicos, equipo de trabajo o el especialista en la materia de investigación, con el objetivo de que los instrumentos previamente declarados puedan tener coherencia a la hora de colecta de datos y estos tengan sentido común. Estas contribuciones ayudaran a los instrumentos en la investigación y podrán ser expeditos para que se puedan emplear en una muestra predeterminada. (p. 223)

### **Confiabilidad**

“La confiabilidad se refiere a la credibilidad que brinda el instrumento, y esto se verifica si al aplicar repetidas veces dicho instrumento, brinda los mismos resultados o valores muy cercanos”. (Soto, 2015).

“Como colofón, cabe destacar que, existen instrumentos para recabar datos que por su naturaleza no ameritan el cálculo de la confiabilidad, como son: entrevistas, escalas de estimación, listas de cotejo, guías de observación, hojas de registros, inventarios, rúbricas, otros “(Revista Ciencias de la Educación, 2009, p. 45). En el presente trabajo de investigación se utilizará la guía o ficha de observación, de modo que confirmamos la validez del instrumento.

### **2.5 Método de análisis de datos**

La información que se obtendrá de los registros será plasmada en fichas de registro y se aplicaran fórmulas para medir los tiempos y la eficacia de las pruebas, comparando dichos resultados con estudios anteriores al sistema desarrollado.

### **2.6 Aspectos Éticos**

(Soto, 2015), sobre los aspectos éticos menciona:

Se refiere, por ejemplo, si el proyecto de investigación que estamos trabajando cuenta con la aprobación de especialistas y docentes calificados de la institución donde se desarrolla el proyecto, si el desarrollo que estamos haciendo es verídico, si el estudio es el verdadero, si no hay copia en otras tesis, también si se mantiene en anonimato a los sujetos que son puestos a prueba, entre otro, etc. (p. 77).

Nos basamos a la teoría citada previamente. El sistema de automatización de pruebas tiene como finalidad mejorar los procesos de la gestión de la calidad de la empresa, es por ello que contaremos con la participación de los trabajadores de la firma Pandora Technologies E.I.R.L., técnicos, y colaboradores de la institución. Se respetará la ética profesional manteniendo los datos de los usuarios seguros y protegidos. Los resultados que se obtengan respetaran la confidencialidad de la información personal de identidad de los usuarios con quienes se trabajará.

### **III. RESULTADOS**

En este capítulo se presentan los resultados conseguidos de la investigación realizada pudiendo usar los indicadores “Tiempo de generación de data” y “Productividad de data generada” realizándose por medio del procesamiento de los datos obtenido de las muestras de cada indicador (pre-test y post-test) con el software IBM SPSS Statistics v.25.0

## **Indicador Tiempo de generación de data**

### **Prueba de Normalidad**

Para la siguiente prueba de normalidad se hizo la elección del método Shapiro-Wilk, por el motivo del tamaño de la muestra, que en nuestro caso fue de 40 registros, en el caso de que una muestra sea inferior a 50 se utiliza el método mencionado, caso contrario se hubiera empleado el método de Kolmogórov-Smirnov. A continuación se muestran el análisis para los datos del PRE y POST-TEST:

Tabla 5

Prueba de Shapiro-Wilk - Tiempo de generación de data

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Segundos	,912	40	,004
Post_Segundos	,744	40	,000

Donde:

#### **Pre-test**

En la tabla podemos apreciar el resultado de aplicar la prueba de Shapiro-Wilk en los segundos que demora una prueba antes del desarrollo del sistema y nos dice que el nivel de significancia en Pre-test es menor a 0.05, lo cual nos indica que nuestra muestra no se adecua como una de distribución normal.



## **Post-test**

En la tabla podemos apreciar el resultado de aplicar la prueba de Shapiro-Wilk en los segundos que demora una prueba después del desarrollo del sistema y nos dice que el nivel de significancia en post-test es menor a 0.05, lo cual nos indica de que nuestra muestra no se adecua como una de distribución normal.

## **Prueba de hipótesis**

**H(0):** El desarrollo de un Sistema de automatización de pruebas no tendrá influencia en el tiempo de generación de data en la empresa Pandora Technologies.

**H(1):** El desarrollo de un Sistema de automatización de pruebas tendrá influencia en el tiempo de generación de data en la empresa Pandora Technologies.

Tabla 6

Estadísticos descriptivos - Tiempo de generación de data

Estadísticos descriptivos					
	N	Media	Desviación estándar	Mínimo	Máximo
Pre_Segundos	40	219,7750	22,75115	187,00	257,00
Post_Segundos	40	94,1000	,90014	93,00	95,00

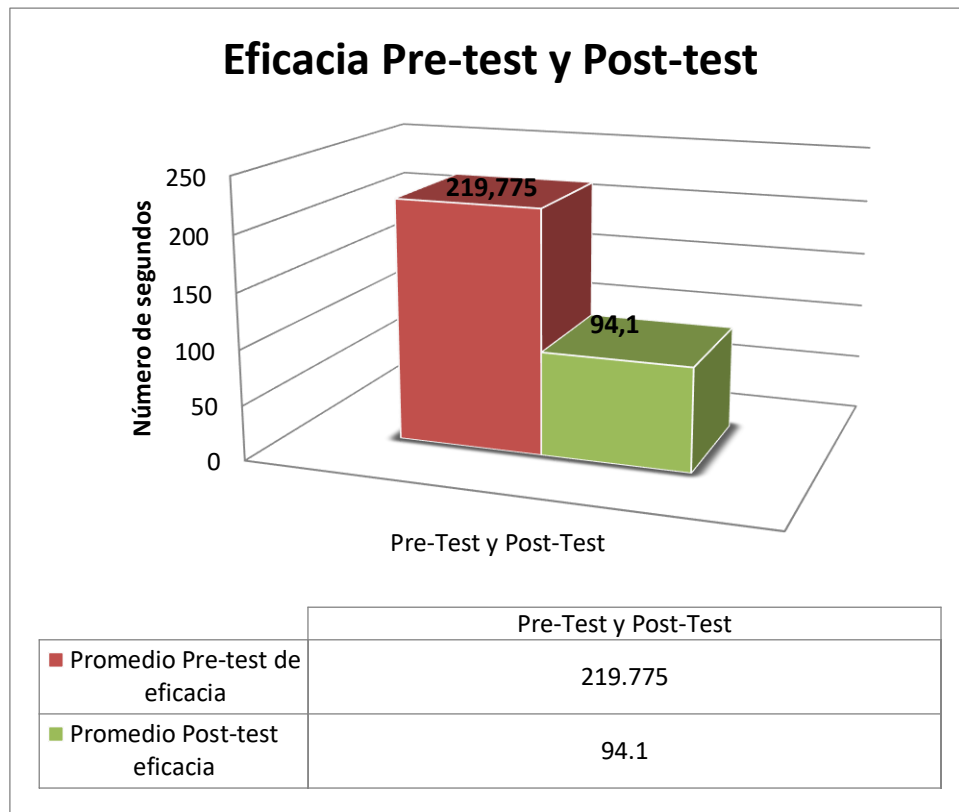


Figura 7. Análisis comparativo para el indicador Tiempo de generación de data

Como se observa en la figura 7, se muestra la estadística descriptiva de los tiempos promedios en segundos del indicador Tiempo de generación de data antes y después de desarrollar el sistema de automatización de procesos para las pruebas de regresión en la empresa Pandora Technologies E.I.R.L. Sin desarrollar el sistema los programadores se demoraban en promedio 220 segundos, sin embargo cuando se implementó el sistema este tiempo disminuyó a 94 segundos, que es equivalente a una mejora del 55%.

### Prueba de Wilcoxon

Tabla 7

Rangos de prueba de signos - Tiempo de generación de data

Rangos de prueba de signos				
		N	Rango promedio	Suma de rangos
Post_Segundos - Pre_Segundos	Rangos negativos	40 <sup>a</sup>	20,50	820,00
	Rangos positivos	0 <sup>b</sup>	,00	,00

Empates	0 <sup>c</sup>
Total	40
a. Post_Segundos < Pre_Segundos	
b. Post_Segundos > Pre_Segundos	
c. Post_Segundos = Pre_Segundos	

Tabla 8

Estadísticos de prueba Z - Tiempo de generación de data

Estadísticos de prueba	
	Post_Segundos - Pre_Segundos
Z	-5,512
Sig. asintótica (bilateral)	,000

Como el valor de Sig. (bilateral) es 0 (menor a 0.5), se rechaza la hipótesis nula y se acepta la hipótesis alterna: El desarrollo de un Sistema de automatización de pruebas tendrá influencia en el tiempo de generación de data en la empresa Pandora Technologies.

### **Indicador Productividad de data generada**

#### **Prueba de Normalidad**

Para la siguiente prueba de normalidad se hizo la elección del método Shapiro-Wilk, por el motivo del tamaño de la muestra, que en nuestro caso fue de 40 registros, en el caso de que una muestra sea inferior a 50 se utiliza el método mencionado, caso contrario se hubiera empleado el método de Kolmogórov-Smirnov. A continuación se muestran el análisis para los datos del PRE y POST-TEST:

Tabla 9

Prueba de Shapiro-Wilk - Productividad de data

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Productividad	,971	40	,039
Post_Productividad	,889	40	,001

Donde:

### **Pre-test**

En la tabla podemos apreciar el resultado de aplicar la prueba de Shapiro-Wilk en el porcentaje de la productividad de generación de data antes del desarrollo del sistema y nos dice que el nivel de significancia en Pre-test es menor a 0.05, lo cual nos indica que nuestra muestra no se adecua como una de distribución normal.

### **Post-test**

En la tabla podemos apreciar el resultado de aplicar la prueba de Shapiro-Wilk en el porcentaje de la productividad de generación de data después del desarrollo del sistema y nos dice que el nivel de significancia en post-test es menor a 0.05, lo cual nos indica que nuestra muestra no se adecua como una de distribución normal.

### **Prueba de hipótesis**

**H(0):** El desarrollo de un Sistema de automatización de pruebas no tendrá influencia en la productividad de data generada en la empresa Pandora Technologies.

**H(1):** El desarrollo de un Sistema de automatización de pruebas tendrá influencia en la productividad de data generada en la empresa Pandora Technologies.

Tabla 10

Estadísticos descriptivos - Productividad de data generada

Estadísticos descriptivos					
	N	Media	Desviación estándar	Mínimo	Máximo
Pre_Productividad	40	91,4773	2,51642	84,80	96,43
Post_Productividad	40	98,5282	1,36757	94,29	100,00

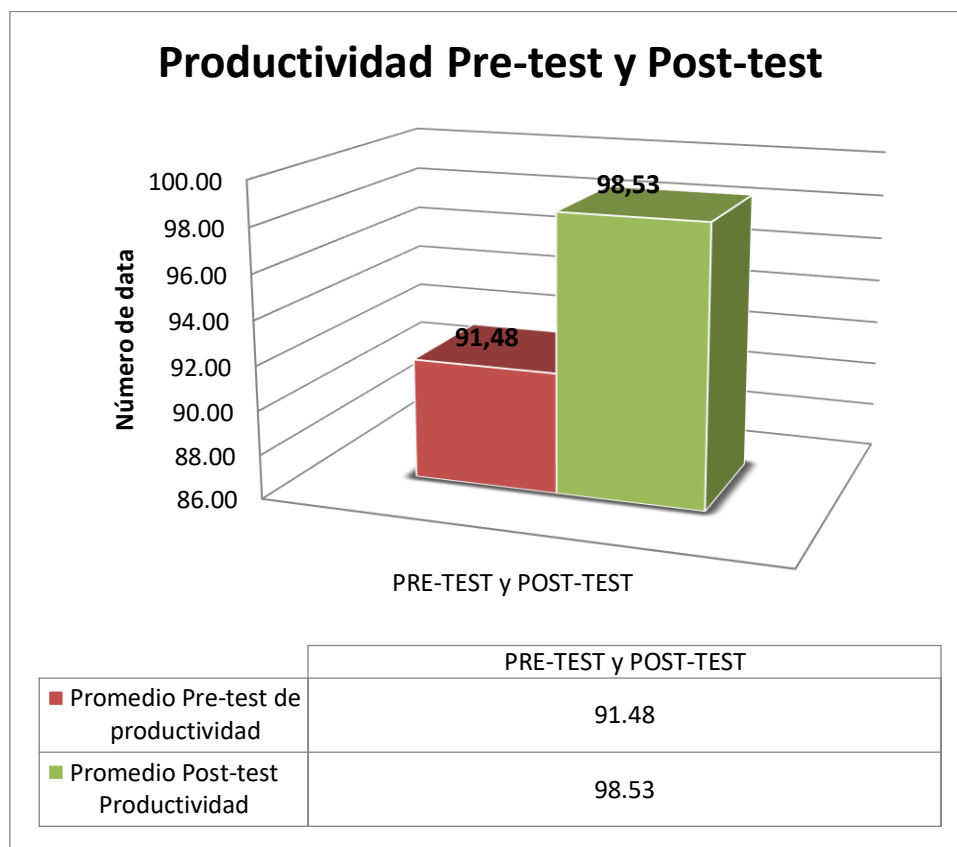


Figura 8. Análisis comparativo para el indicador Productividad de generación de data

Como se observa en la figura 8, se muestra la estadística descriptiva del número de datos promedio del indicador Productividad de generación de data antes y después de desarrollar el sistema de automatización de procesos para las pruebas de regresión en la empresa Pandora Technologies E.I.R.L. Sin desarrollar el sistema los programadores obtenían en promedio 91.48 datos, sin embargo cuando se implementó el sistema la productividad aumento a 98.53 datos, que es equivalente a una mejora del 7%.

## Prueba de Wilcoxon

Tabla 11

Rango de prueba de signos - Productividad de data generada

		Rangos- Prueba de signos		
		N	Rango promedio	Suma de rangos
Post_Productividad - Pre_Productividad	Rangos negativos	0 <sup>a</sup>	,00	,00
	Rangos positivos	40 <sup>b</sup>	20,50	820,00
	Empates	0 <sup>c</sup>		
	Total	40		
a. Post_Productividad < Pre_Productividad				
b. Post_Productividad > Pre_Productividad				
c. Post_Productividad = Pre_Productividad				

Tabla 12

Estadísticos de prueba Z - Productividad de data generada

Estadísticos de prueba	
Post_Productividad - Pre_Productividad	
Z	-5,511
Sig. asintótica (bilateral)	,000

Como el valor de Sig. (bilateral) es 0 (menor a 0.5), se rechaza la hipótesis nula y se acepta la hipótesis alterna: El desarrollo de un Sistema de automatización de pruebas tendrá influencia en la cantidad de data generada en la empresa Pandora Technologies.

## **IV. DISCUSIÓN**

Con los datos obtenidos en el análisis de la actual investigación se realiza un comparativo sobre el tiempo y productividad de data generada en las pruebas de regresión basado en web en la empresa Pandora Technologies.

**Primera:** La cantidad de data generada en las pruebas de software, en la medición pre-test se alcanzó 219.78 segundos y en la medición pos-test, con la automatización de procesos con la herramienta se alcanzó los 94.1 segundos.

La prueba de normalidad menciona que el Sig. de la muestra del tiempo de generación de data en el pre-test fue de 0.004, teniendo el valor menor que el error asumido de 0.05, por consiguiente se rechaza la hipótesis nula (H0) y se acepta la hipótesis alternativa (H1), concluyendo que dicha prueba con el indicador del tiempo de generación de data de prueba no tienen un comportamiento normal.

También, la prueba indica que el Sig. de la muestra del tiempo de generación de data en el post-test fue de 0.000, teniendo el valor menor que el error asumido de 0.05, por consiguiente se rechaza la hipótesis nula (H0) y se acepta la hipótesis alternativa (H1), concluyendo que dicha prueba con el indicador del tiempo de generación de data de prueba no tienen un comportamiento normal.

La prueba Wilcoxon indica que después de la automatización de las pruebas la media del tiempo de data generada ha disminuido de 217.50 a 94.00 Siendo la significancia Sig 0.000 en donde es menor que el valor alfa de 0.05, conllevando a que se rechace la hipótesis nula y se acepte la hipótesis alterna con un 95% de confianza. Vale decir que al utilizar la herramienta reduce 123.5 segundos en tiempo de data generada, por consiguiente, aplicar automatización de pruebas mejora significativamente el tiempo de generación de data dentro de las pruebas de regresión en la empresa Pandora Technologies al disminuir en 55% respecto a las pruebas sin el sistema.

Según la investigación realizada por (Fernandez Avalos, 2018) en su tesis “Automatización de Procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito”, los resultados determinaron que implementar una automatización de procesos mejora en un 56.78% en los tiempo de sus pruebas con el fin de ser eficiente en el área de calidad de software; al implementar la automatización de pruebas para mejorar la calidad de software del Banco de Crédito de Perú.



**Segunda:** La cantidad de data generada para las pruebas de software, en la medición Pre-Test logró un 91.48 y con la automatización de procesos incremento a 98.53; los resultados conseguidos revelan que existe un incremento de 7.05 en la productividad de data generada.

La prueba de normalidad menciona que el Sig. de la muestra de la cantidad de data generada de prueba pre-test fue de 0.039, teniendo el valor menor que el error asumido de 0.05, por consiguiente se rechaza la hipótesis nula y se acepta la hipótesis alternativa (H1), concluyendo que dicha prueba con el indicador de productividad de data generada no tienen un comportamiento normal. De la misma manera, la prueba menciona que el Sig. de la muestra de la cantidad de data generada de prueba post-test fue de 0.001, teniendo el valor menor que el error asumido de 0.05, por consiguiente se rechaza la hipótesis nula (H0) y se acepta la hipótesis alternativa (H1), concluyendo que dicha prueba con el indicador de cantidad de data generada no tienen un comportamiento normal.

Para comprobar la hipótesis se aplicó la prueba Wilcoxon que indica que después de la automatización de procesos la media de la productividad de data generada ha aumentado de 91,48 a 98.53 Siendo la significancia Sig 0.000 en donde es menor que el valor alfa de 0.05, conllevando a que se rechace la hipótesis nula y se acepte la hipótesis alterna con un 95% de confianza. Vale decir que al utilizar el sistema aumenta en un 7% la productividad de data generada. En consecuencia, al implementar la automatización de pruebas se tuvo un significativo incremento en la productividad de data generada en un 7%, lo que significa que existe una mejora de manera significativa la cantidad de data generada en la dimensión de productividad en las pruebas de regresión basado en web en la empresa Pandora Technologies .

Según la investigación realizada por (Calva Carhuamaca, 2017) en su tesis “Sistema que reemplaza funciones de un operador humano durante la validación de documentos digitales en Core Andina Group”, permitió obtener resultados satisfactorios, como la reducción de costos y tiempos gracias a aumentar la eficacia en el proceso de validación de registros de documentos en un 16.09%.

## **V. CONCLUSIONES**

Las conclusiones de la investigación fueron las siguientes:

1. Con respecto a los datos obtenidos, el desarrollo de un sistema de automatización de pruebas regresión basado en web ha tenido un impacto favorable en el proceso de la calidad del software teniendo como resultado el ahorro de casi 10,000 dólares americanos, además se mejoró el tiempo en las pruebas de los registros de la empresa en aproximadamente 55% y se mejoró la productividad en los procesos de desarrollo de software.
2. El porcentaje en la reducción de los tiempos en las pruebas fue disminuido en 55%, logrando llegar a los 94.1 segundos en los tiempos de generación de data, esto nos dice que el software de automatización trabaja mucho más exacto y rápido que un desarrollador promedio, evitándole el trabajo repetitivo y teniendo tiempo para actividades de mayor complejidad.
3. El porcentaje de productividad fue aumentado en 7%, gracias a esto podemos afirmar que un sistema con la automatización en los procesos de prueba de regresión en cuanto a la productividad en la data generada es más precisa que un humano, debido a que puede trabajar con menos margen de error, con más rapidez y sin algún tipo de cansancio que pueda afectar en los procesos de prueba.

## **VI. RECOMENDACIONES**

Las siguientes recomendaciones para el trabajo a futuro:

1. Extender la investigación para poder implementar una interfaz gráfica que ayude al desarrollador a mejorar sus procesos de testing y se le facilite la labor de programar los scripts de prueba.
2. Que el sistema capture toda evidencia relevante sobre fallos durante el testing como fotos o videos, además de dar detalles de los tiempos de las pruebas y los recursos consumidos por él.
3. Tener la capacidad de importar datos de diferentes tipos ya sean Access, archivos de texto, bases de datos no relacionales, etc.
4. Profundizar en la investigación y agrandar el tamaño de la población y muestra para saber el comportamiento durante un entorno de trabajo de tipo más demandante, que podría ser usado por bancos, hospitales, universidades, empresas transaccionales, el gobierno, etc.
5. Ampliar el número de las variables para poder medir con exactitud el desempeño del sistema en diferentes procesos como costos, cantidad de transacciones al día, semana, mes y año, recursos consumidos, entre otros para comprobar la viabilidad de tener un sistema en diferentes sectores de la empresa.

## **REFERENCIAS**

- Aguaded, C., & Fuentes, C. (2006). *Páginas Web : Heurística Para Analizar Un Sitio Web , Aplicación En*.
- Álvarez García, Alonso; Lasa Gómez, Carmen; de las Heras del Dedo, R. (2011). *Métodos Ágiles. Scrum, Kanban, Lean (Manuales Imprescindibles)* (edición (2)). Grupo Anaya Publicaciones Generales. <https://www.amazon.es/Métodos-Ágiles-Kanban-Manuales-Imprescindibles/dp/8441538883?tag=igravilan-21>
- AMARICAI, S., & CONSTANTINESCU, R. (2014). Designing a Software Test Automation Framework. *Informatica Economica*, 18(1/2014), 152–161. <https://doi.org/10.12948/issn14531305/18.1.2014.14>
- ARANDA, L. H., BUGGIA, D. E., & GHELLA, M. A. (2009). *APLICACIONES WEB 2.0 WIKIS RSS*. [https://www.academia.edu/22586799/APLICACIONES\\_WEB\\_2\\_0\\_WIKIS\\_RSS](https://www.academia.edu/22586799/APLICACIONES_WEB_2_0_WIKIS_RSS)
- Arias.f. (2006). *El Proyecto de Investigación. Introducción a la Metodología Científica* (6a Edición). [https://books.google.com.pe/books?id=W5n0BgAAQBAJ&printsec=frontcover&dq=METODOLOGIA+CIENTIFICA&hl=es\\_419&sa=X&ved=0ahUKEwicqNWbj8DbAhVJx5AKHZSDCTkQ6AEIJjAA#v=onepage&q=METODOLOGIA+CIENTIFICA&f=false](https://books.google.com.pe/books?id=W5n0BgAAQBAJ&printsec=frontcover&dq=METODOLOGIA+CIENTIFICA&hl=es_419&sa=X&ved=0ahUKEwicqNWbj8DbAhVJx5AKHZSDCTkQ6AEIJjAA#v=onepage&q=METODOLOGIA+CIENTIFICA&f=false)
- Berzal, F., Cortijo, F. J., & Cubero, J. C. (2010). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. <http://elvex.ugr.es/decsai/csharp/pdf/web/web-book-a4.pdf>
- Bocángel Lamas, E. (2016). *JAVA Y BASE DE DATOS [UNIVERSIDAD JOSÉ CARLOS MARIÁTEGUI]*. [http://repositorio.ujcm.edu.pe/bitstream/handle/ujcm/435/Ericka\\_Trabajo\\_de\\_suficiencia\\_titulo\\_2016.pdf?sequence=1&isAllowed=y](http://repositorio.ujcm.edu.pe/bitstream/handle/ujcm/435/Ericka_Trabajo_de_suficiencia_titulo_2016.pdf?sequence=1&isAllowed=y)
- Bughin, Jacques ; Hazan, Eric; Manyika, James; Woetzel, J. (2017). Artificial Intelligence – The next digital frontier? *McKinsey Global Institute Discussion Paper*, 2017(4), 14–17. [www.mckinsey.com/mgi](http://www.mckinsey.com/mgi).
- Buwalda, H. (2011). Action Based Testing: The Solution for Agile Test Automation? | LogiGear Magazine. *LogiGear Magazine*. <https://www.logigear.com/magazine/action-based-testing/action-based-testing-the->

solution-for-agile-test-automation/

- Cahill, S. M. (2017). *A Multi Stakeholder Perspective on Audit and Automated Compliance: Bank of Ireland*.  
<https://www.scss.tcd.ie/publications/theses/diss/2017/TCD-SCSS-DISSERTATION-2017-048.pdf>
- Calva Carhuamaca, J. G. (2017). Sistema que reemplaza funciones de un operador humano durante la validación de documentos digitales en Core Andina Group [Universidad César Vallejo]. In *Universidad César Vallejo*.  
<http://repositorio.ucv.edu.pe/handle/20.500.12692/26911>
- Capcha Coronado, E. (2018). Implementación del Framework de automatización de proceso de QA en un proyecto de diseño de software en una consultora [Universidad Nacional Mayor de San Marcos]. In *Universidad Nacional Mayor de San Marcos*. <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/10210>
- CÁRDENAS, O. L. (2016). A “missing” family of classical orthogonal polynomials. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8).  
<https://doi.org/10.1088/1751-8113/44/8/085201>
- Cárdenas Ospina, L. A., & Rodríguez Beltrán, J. A. (2011). Sistema de gestión de pruebas para productos de software. In *reponame:Repositorio Institucional Universidad Libre*. <http://repository.unilibre.edu.co/handle/10901/8839>
- Ciolli, M. E. (2007). *Testing de migración de aplicaciones distribuidas a entornos Web* [Universidad Nacional de la Plata].  
<https://pdfs.semanticscholar.org/8b2f/258ef270a7b9cc7f6b9c473a011b5b0ff86f.pdf>
- Dalebout, W. (2018). The effects of Robotic Process Automation on FTE effort and the design of the jobs involved. *Electronic Engineering*, 74.
- Fernandez Avalos, J. L. M. (2018). Automatización de Procesos para mejorar las Pruebas de Software en el área de calidad del Banco de Crédito [Universidad César Vallejo]. In *Universidad César Vallejo*.  
<http://repositorio.ucv.edu.pe/handle/20.500.12692/23871>
- Gheorghiu, G. (2005). A Look at Selenium . *Better Software*, 08.



<https://www.stickyminds.com/better-software-magazine/look-selenium>

Gómez, M. M. (2006). *Introducción a la metodología de la investigación científica*

(Editorial Brujas (ed.)).

[https://books.google.com.pe/books?id=9UDXPe4U7aMC&pg=PA11&dq=METODOLOGIA+CIENTIFICA&hl=es\\_419&sa=X&ved=0ahUKEwicqNWbj8DbAhVJx5AKHZSDCTkQ6AEIQjAF#v=onepage&q=METODOLOGIA+CIENTIFICA&f=false](https://books.google.com.pe/books?id=9UDXPe4U7aMC&pg=PA11&dq=METODOLOGIA+CIENTIFICA&hl=es_419&sa=X&ved=0ahUKEwicqNWbj8DbAhVJx5AKHZSDCTkQ6AEIQjAF#v=onepage&q=METODOLOGIA+CIENTIFICA&f=false)

Graham, D., & Fewster, M. (2012). *Experiences of Test Automation: Case Studies of Software Test Automation*. [https://play.google.com/store/books/details?id=ryVM-XTNiI0C&source=gbs\\_api](https://play.google.com/store/books/details?id=ryVM-XTNiI0C&source=gbs_api)

Guevara Bolaños, J. C., Flórez Fernández, H., Pinzón Nuñez, S. A., Pérez Castillo, J. N., & Espinel, Á. (2006). La web semántica y sus posibles aplicaciones en el ámbito universitario. *La Web Semántica y Sus Posibles Aplicaciones En El Ámbito Universitario*, 10(19), 120–131. <https://doi.org/10.14483/22487638.6251>

Jerez Mayorga, D. S. (2017). *Análisis de métodos, técnicas y herramientas de Verificación y validación de software, aplicados en la Dirección de Tecnología de Información y Comunicación de La Universidad Técnica de Ambato*. <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/26424>

Lacity, M., Willcocks, L., & Craig, A. (2017). Service Automation : Cognitive Virtual Agents at SEB Bank. *The Outsourcing Unit Working Research Paper Series*, 17/01(February), 1–29. <http://www.lse.ac.uk/management/research/outsourcingunit/>

Laureano Aguilar, E. A., & Urquilla Osorio, N. A. (2018). *APLICACION WEB PARA LA GESTION ADMINISTRATIVA DE LAS UNIDADES DE CREDITOS Y TESORERIA DEL FONDO UNIVERSITARIO DE PROTECCION DE LA UNIVERSIDAD DE EL SALVADOR* [UNIVERSIDAD DE EL SALVADOR]. [http://ri.ues.edu.sv/id/eprint/16667/1/Tomo de tesis.pdf](http://ri.ues.edu.sv/id/eprint/16667/1/Tomo%20de%20tesis.pdf)

Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. <http://rua.ua.es/dspace/handle/10045/16995>

Madi, R. (2013). *Learning Software Testing with Test Studio*. Packt Publishing. <https://www.worldcat.org/title/learning-software-testing-with-test->

studio/oclc/859390755

- Martínez Orozco, J., Henao Toro, D. F., & Quintero Salazar, E. A. (2014). Aplicativo para el Acceso vía Web a Observatorios Astronómicos . *Scientia et Technica Año XIX, 19*(1), 77–83. <http://www.utp.edu.co/observatorioastronomico/observatorio->
- McKay, J., & Bath, G. (2014). Software Test Engineer’s Handbook, 2nd Edition. In *Rocky Nook* (2nd editio).  
[https://books.google.com.pe/books?id=x0C4BAAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.pe/books?id=x0C4BAAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Moreano Camacho, E. M. (2014). *Análisis, diseño, construcción e implementación de un portal de información y venta de aplicaciones de entrenamiento para el club de video juegos de la ESPE* [Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática.].  
<http://repositorio.espe.edu.ec/jspui/handle/21000/9041>
- Ortíz-Uribe, F. G., & García-Nieto, M. P. (2008). Metodología de la investigación: el proceso y sus técnicas. In *Limusa* (13a. ed.).  
<http://biblio.upmx.mx/library/index.php/312631>
- Pardo, P., & Gonzales, R. (2014). La importancia de los sistemas de información en las organizaciones automatizadas. *Revista Ingenio Libre, 12*.  
<http://www.unilibre.edu.co/revistaingeniolibre/revista-12/ar6.pdf>
- Peña Rondoy, E. Y. (2018). Implementación de un sistema informático de gestión de pagos en el instituto de educación superior tecnológico privado Santa Úrsula-Sullana; 2015 [Universidad Católica los Ángeles de Chimbote]. In *Universidad Católica Los Ángeles de Chimbote*.  
<http://repositorio.uladech.edu.pe/handle/123456789/4763>
- Quispe Tapara, H. (2018). Diseño de un sistema de riego automatizado por aspersión para viveros de café utilizando la tecnología arduino en la empresa Viveros Ortíz – Pasco; 2018 [Universidad Católica los Ángeles de Chimbote]. In *Universidad Católica Los Ángeles de Chimbote*.  
<http://repositorio.uladech.edu.pe/handle/123456789/3024>
- Rivera Martínez, C. A. (2018). *Automatización de pruebas de regresión* [Universidad de Chile]. <http://repositorio.uchile.cl/handle/2250/165608>

- Rosa Raymundo, E. O. de la. (2013). Gestión de la empresa QA e implementación del proceso de auditoría, servicio de automatización de pruebas y administración del conocimiento [Universidad Peruana de Ciencias Aplicadas (UPC)]. In *Universidad Peruana de Ciencias Aplicadas (UPC)*.  
<http://hdl.handle.net/10757/315031>
- Rosero, R. H., Gómez, O. S., & Rodríguez, G. (2016). 15 Years of Software Regression Testing Techniques - A Survey. In *International Journal of Software Engineering and Knowledge Engineering* (Vol. 26, Issue 5, pp. 675–689). World Scientific Publishing Co. Pte Ltd. <https://doi.org/10.1142/S0218194016300013>
- Saravanan, K., & Poorna Chandra, P. E. (2016). Open Source Software Test Automation Tools: A Competitive Necessity. *Scholedge International Journal of Management & Development ISSN 2394-3378*, 3(6), 110.  
<https://doi.org/10.19085/journal.sijmd030601>
- Solana Álvarez, J. M. (2014). El sistema de información de una organización. Necesidad de implicación de la dirección. *Anuario Jurídico y Económico Escurialense*, 471–480. <https://dialnet.unirioja.es/descarga/articulo/4639730.pdf>
- Soto, R. (2015). *La tesis de maestría y doctorado en 4 pasos*. (2da Edición). Colección Nuevo Milenio. <https://es.slideshare.net/YOLYSP/la-tesis-de-maestra-y-doctorado-en-4-pasos-segunda-edicion-dr-roger-soto>
- Talledo Pascual, R. M. (2018). Sistema web auto-response de gestión de incidencias en la empresa Limtek servicios integrales S.A. [Universidad César Vallejo]. In *Universidad César Vallejo*.  
<http://repositorio.ucv.edu.pe/handle/20.500.12692/19641>
- Tasato Cánepa, K. (2014). Desarrollo de una infraestructura de software para realizar pruebas automatizadas de sistemas de información desarrollados en lenguaje Cobol en el contexto bancario [Pontificia Universidad Católica del Perú]. In *Pontificia Universidad Católica del Perú*.  
<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/5424>
- Toro López, F. J. (2013). *Administración de proyectos de informática*. ECOE Ediciones. <https://www.ecoediciones.com/libros/libros-de-informatica/ebook-administracion-de-proyectos-en-informatica-1ra-edicion/>

- Valderrama Mendoza, S. (2007). *Pasos para Elaboración Proyectos y Tesis de Investigación Científica* (1ra edición). San Marcos E.I.R.L. .  
<http://sisbiblio.utea.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=3273>
- Ventura Labrin, L. A. (2015). Automatización del proceso de ventas y distribución utilizando tecnología móvil y geolocalización para la empresa líder SRL [Universidad Privada Antenor Orrego - UPAO]. In *Universidad Privada Antenor Orrego - UPAO*. <http://repositorio.upao.edu.pe/handle/upaorep/643>
- Watin. (2013, September 25). *An Automation Testing in .NET*. Code Project.  
<https://www.codeproject.com/Tips/658947/Watin-An-Automation-Testing-in-NET>
- Yépez La Rosa, M. P., & Márquez Caicedo, A. G. (2004). *Mejoras en los procesos de aseguramiento de la calidad en el desarrollo de aplicaciones*.  
<https://docplayer.es/6006823-Mejoras-en-los-procesos-de-aseguramiento-de-la-calidad-en-el-desarrollo-de-aplicaciones.html>
- Zambelich, K. (1998). *Totally Data-Driven Automated Testing*. [www.kaner.com](http://www.kaner.com)

## **ANEXOS**

Anexo 1: Matriz de Operacionalización

TÍTULO: “Sistema para la Automatización de pruebas de regresión basado en web en la empresa Pandora Technologies durante el periodo 2019”											
AUTORES: Marco Hoyos Hipolito, Carlos Henriquez Palomino											
Problema General	Objetivo General	Hipótesis General	VARIABLE INDEPENDIENTE							Es cal a	Método
			Definición Conceptual	Definición Operacional	Dimens iones	Indicad ores	Formul a	I.			
¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en las pruebas de regresión?	Determinar el impacto de un sistema de automatización de pruebas de regresión en la empresa Pandora Technologies.	El desarrollo de un Sistema de automatización de pruebas tendrá influencia en las pruebas de regresión en la empresa Pandora Technologies.	Automatización de pruebas	La automatización de pruebas es el uso de un software especial (separado del software que se está probando) para controlar la ejecución de las pruebas y la comparación de los resultados reales con los resultados previstos (Saravanan & Poorna Chandra,	Las pruebas automatizadas poseen características las cuales miden su aceptabilidad, su confiabilidad, disponibilidad, consistencia. Mediante estos parámetros podremos deducir la						Enfoque de investigación: Cuantitativo  Tipo de Investigación: Pre experimental

				2016)	calidad del sistema.						
Problemas Específicos	Objetivos Específicos	Hipótesis Especificas	VARIABLE DEPENDIENTE							I.	Escala
			Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Formula				
¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en el tiempo de generación de data?	Determinar el impacto de un sistema de automatización de pruebas en el tiempo de generación de data en la empresa Pandora Technologies.	El desarrollo de un Sistema de automatización de pruebas tendrá influencia en el tiempo de generación de data en la empresa Pandora Technologies.	Pruebas de regresión	“La prueba de regresión es una práctica de prueba de software que garantiza que una aplicación siga funcionando como se espera después de cualquier cambio, actualización o mejora de código.” (ISQTB, 2017,p. 1)	La prueba de regresión se evaluará teniendo en cuenta los diferentes indicadores que se especifican.	Eficacia	Tiempo de generación de data.	$T(t) = T(f) - T(i)$ Donde: $T(t) =$ Tiempo de Generación de Data en segundos.  $T(f) =$ Tiempo final.  $T(i) =$ Tiempo inicial.	Ficha de Observación	Razón	

<p>¿De qué manera el desarrollo de un Sistema de automatización de pruebas influye en la cantidad de data generada?</p>	<p>Determinar el impacto de un sistema de automatización de pruebas en la cantidad de data generada en la empresa Pandora Technologies.</p>	<p>El desarrollo de un Sistema de automatización de pruebas tendrá influencia en la cantidad de data generada en la empresa Pandora Technologies.</p>				<p>Productividad</p>	<p>Productividad de data generada</p>	<p><b><math>P=(C*100)/D</math></b>  Donde:  P= Productividad de Data Generada.  C= Cantidad de data generada.  D = Data Solicitada.</p>	<p>Ficha de observación</p>	<p>Razón</p>	
---	---	---	--	--	--	----------------------	---------------------------------------	---	-----------------------------	--------------	--

Fuente: Elaboración propia



Anexo 2: Arquitectura física del sistema



Fuente: Elaboración Propia

Anexo 3: Ficha de Observación 1

FICHA DE OBSERVACION				
Investigador:		Carlos Alberto Henríquez Palomino		
PRE TEST				
DIMENSION		INDICADOR		FORMULA
Eficacia		Tiempo de Generación de Data		$T(t) = T(f) - T(i)$
N°	FECHA	Tiempo Inicial	Tiempo Final	Tiempo de Generación de Data en segundos
	(dd/mm/aa)	T(i)	T(f)	T(t)
1	10/05/2019	10:00:00	10:03:40	220
2	10/05/2019	10:05:00	10:08:54	234
3	10/05/2019	10:09:05	10:13:08	243
4	10/05/2019	10:13:58	10:17:52	234
5	10/05/2019	10:19:02	10:23:09	247
6	10/05/2019	10:24:00	10:27:53	233
7	10/05/2019	10:28:35	10:32:31	236
8	10/05/2019	10:33:12	10:37:28	256
9	10/05/2019	10:38:19	10:42:27	248
10	10/05/2019	10:43:02	10:46:57	235
11	11/05/2019	13:17:25	13:21:42	257
12	11/05/2019	13:25:14	13:29:01	227
13	11/05/2019	13:31:04	13:34:45	221
14	11/05/2019	13:36:00	13:39:07	187
15	11/05/2019	13:41:04	13:44:17	193
16	11/05/2019	13:45:22	13:48:38	196
17	11/05/2019	13:50:10	13:54:01	231
18	11/05/2019	13:55:14	13:58:42	208
19	11/05/2019	13:59:35	14:03:42	247
20	11/05/2019	14:04:18	14:07:31	193
21	12/05/2019	11:08:13	11:11:32	199
22	12/05/2019	11:12:21	11:15:35	194
23	12/05/2019	11:16:01	11:19:32	211
24	12/05/2019	11:20:19	11:23:34	195
25	12/05/2019	11:24:35	11:27:45	190
26	12/05/2019	11:28:09	11:31:49	220
27	12/05/2019	11:32:05	11:35:23	198
28	12/05/2019	11:37:11	11:40:27	196
29	12/05/2019	10:19:02	10:23:09	247
30	12/05/2019	10:43:02	10:46:57	235
31	13/05/2019	12:17:25	12:21:42	257
32	13/05/2019	12:25:14	12:29:01	227

33	13/05/2019	12:31:04	12:34:45	221
34	13/05/2019	12:36:00	12:39:07	187
35	13/05/2019	12:41:04	12:44:17	193
36	13/05/2019	12:45:22	12:48:38	196
37	13/05/2019	12:50:10	12:54:01	231
38	13/05/2019	12:55:14	12:58:42	208
39	13/05/2019	12:59:35	13:03:42	247
40	13/05/2019	13:04:18	13:07:31	193

Fuente: Elaboración Propia

Anexo 4: Ficha de Observación 2

FICHA DE OBSERVACION				
Investigador:		Carlos Alberto Henriquez Palomino		
POST TEST				
DIMENSION		INDICADOR		FORMULA
Eficacia		Tiempo de Generación de Data		$T(t) = T(f) - T(i)$
N°	FECHA	Tiempo Inicial	Tiempo Final	Tiempo de Generación de Data en segundos
	(dd/mm/aa)	T(i)	T(f)	T(t)
1	02/06/2019	9:00:00	9:01:34	94
2	02/06/2017	9:01:35	9:03:09	94
3	02/06/2019	9:03:10	9:04:44	94
4	02/06/2019	9:04:45	9:06:19	94
5	02/06/2019	9:06:20	9:07:54	94
6	02/06/2019	9:07:45	9:09:19	94
7	02/06/2019	9:09:20	9:10:54	94
8	02/06/2019	9:10:55	9:12:29	95
9	02/06/2019	9:12:30	9:14:04	95
10	02/06/2019	9:14:05	9:15:39	95
11	05/06/2019	9:00:00	9:01:35	95
12	05/06/2019	9:01:36	9:03:11	95
13	05/06/2019	9:03:12	9:04:47	95
14	05/06/2019	9:04:48	9:06:23	95
15	05/06/2019	9:06:24	9:07:59	95
16	05/06/2019	9:08:00	9:09:35	95
17	05/06/2019	9:09:36	9:11:11	95
18	05/06/2019	9:11:12	9:12:47	95
19	05/06/2019	9:12:48	9:14:23	93
20	05/06/2019	9:14:24	9:15:59	93
21	06/06/2019	9:00:00	9:01:33	93
22	06/06/2019	9:01:34	9:03:07	93
23	06/06/2019	9:03:08	9:04:41	93
24	06/06/2019	9:04:42	9:06:15	93
25	06/06/2019	9:06:16	9:07:49	93
26	06/06/2019	9:07:50	9:09:23	93
27	06/06/2019	9:09:24	9:10:57	93
28	06/06/2019	9:10:58	9:12:31	93
29	06/06/2019	10:19:02	10:23:09	93
30	06/06/2019	10:43:02	10:46:57	93
31	07/06/2019	9:00:00	9:01:35	95
32	07/06/2019	9:01:36	9:03:11	95
33	07/06/2019	9:03:12	9:04:47	95

34	07/06/2019	9:04:48	9:06:23	95
35	07/06/2019	9:06:24	9:07:59	95
36	07/06/2019	9:08:00	9:09:35	95
37	07/06/2019	9:09:36	9:11:11	95
38	07/06/2019	9:11:12	9:12:46	94
39	07/06/2019	9:12:47	9:14:22	93
40	07/06/2019	9:14:23	9:15:58	93

Fuente: Elaboración Propia

Anexo 5: Ficha de Observación 3

FICHA DE OBSERVACION				
Investigador:		Marco Giovanni Hoyos Hipolito		
PRE TEST				
VARIABLE		INDICADOR		FORMULA
Productividad		Productividad de data generada		$P=(C*100)/D$
N°	FECHA	Cantidad de data generada	Data Solicitada	Productividad de Data Generada
	(dd/mm/aa)	C	D	P
1	08/05/2019	120	130	92,31
2	09/05/2019	121	130	93,08
3	10/05/2019	120	135	88,89
4	11/05/2019	119	130	91,54
5	12/05/2019	125	135	92,59
6	15/05/2019	122	130	93,85
7	16/05/2019	124	140	88,57
8	17/05/2019	121	135	89,63
9	18/05/2019	118	130	90,77
10	19/05/2019	123	130	94,62
11	22/05/2019	104	115	90,43
12	23/05/2019	140	155	90,32
13	24/05/2019	114	125	91,20
14	25/05/2019	146	155	94,19
15	26/05/2019	105	115	91,30
16	29/05/2019	131	140	93,57
17	30/05/2019	118	130	90,77
18	31/05/2019	128	135	94,81
19	01/06/2019	110	125	88,00
20	02/06/2019	112	120	93,33
21	05/06/2019	117	130	90,00
22	06/06/2019	135	140	96,43
23	07/06/2019	116	125	92,80
24	08/06/2019	130	145	89,66
25	09/06/2019	106	125	84,80
26	12/06/2019	138	145	95,17
27	13/06/2019	135	150	90,00
28	14/06/2019	131	145	90,34
29	15/06/2019	146	155	94,19
30	16/06/2019	135	145	93,10
31	17/06/2019	120	133	90,23
32	18/06/2019	129	140	92,14
33	19/06/2019	110	125	88,00
34	20/06/2019	125	133	93,98
35	21/06/2019	115	122	94,26
36	22/06/2019	120	136	88,24
37	23/06/2019	114	121	94,21
38	24/06/2019	122	136	89,71

39	25/06/2019	126	140	90,00
40	26/06/2019	118	134	88,06

Fuente: Elaboración Propia

Anexo 6: Ficha de Observación 4

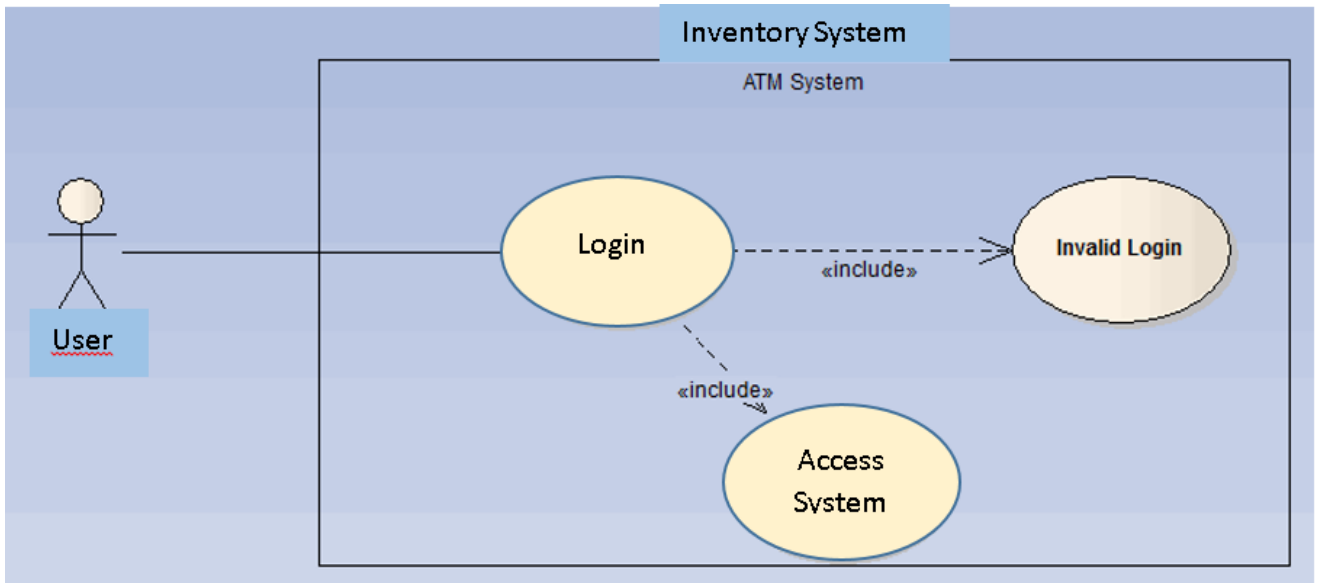
FICHA DE OBSERVACION				
Investigador:		Marco Giovanni Hoyos Hipolito		
POS TEST				
VARIABLE		INDICADOR	FORMULA	
Productividad		Productividad de data generada	$P=(C*100)/D$	
N°	FECHA	Cantidad de data generada	Data Solicitada	Productividad de Data Generada
	(dd/mm/aa)	C	D	P
1	05/06/2019	130	130	100,00
2	05/06/2019	120	120	100,00
3	05/06/2019	125	125	100,00
4	05/06/2019	140	140	100,00
5	05/06/2019	90	90	100,00
6	06/06/2019	110	110	100,00
7	06/06/2019	80	80	100,00
8	06/06/2019	115	115	100,00
9	06/06/2019	65	65	100,00
10	06/06/2019	40	40	100,00
11	07/06/2019	124	125	99,20
12	07/06/2019	129	130	99,23
13	07/06/2019	94	95	98,95
14	07/06/2019	44	45	97,78
15	07/06/2019	149	150	99,33
16	08/06/2019	144	145	99,31
17	08/06/2019	119	120	99,17
18	08/06/2019	74	75	98,67
19	08/06/2019	89	90	98,89
20	08/06/2019	49	50	98,00
21	09/06/2019	123	125	98,40
22	09/06/2019	143	145	98,62
23	09/06/2019	128	130	98,46
24	09/06/2019	148	150	98,67
25	09/06/2019	63	65	96,92
26	12/06/2019	73	75	97,33
27	12/06/2019	148	150	98,67
28	12/06/2019	123	125	98,40
29	12/06/2019	143	145	98,62
30	12/06/2019	133	135	98,52
31	13/06/2019	112	115	97,39
32	13/06/2019	78	80	97,50
33	13/06/2019	89	90	98,89
34	13/06/2019	86	88	97,73
35	13/06/2019	98	101	97,03
36	14/06/2019	102	103	99,03



37	14/06/2019	77	80	96,25
38	14/06/2019	59	61	96,72
39	14/06/2019	59	62	95,16
40	14/06/2019	66	70	94,29

Fuente: Elaboración Propia

Anexo 7: CASO DE USO 01 – ACCEDER AL SISTEMA

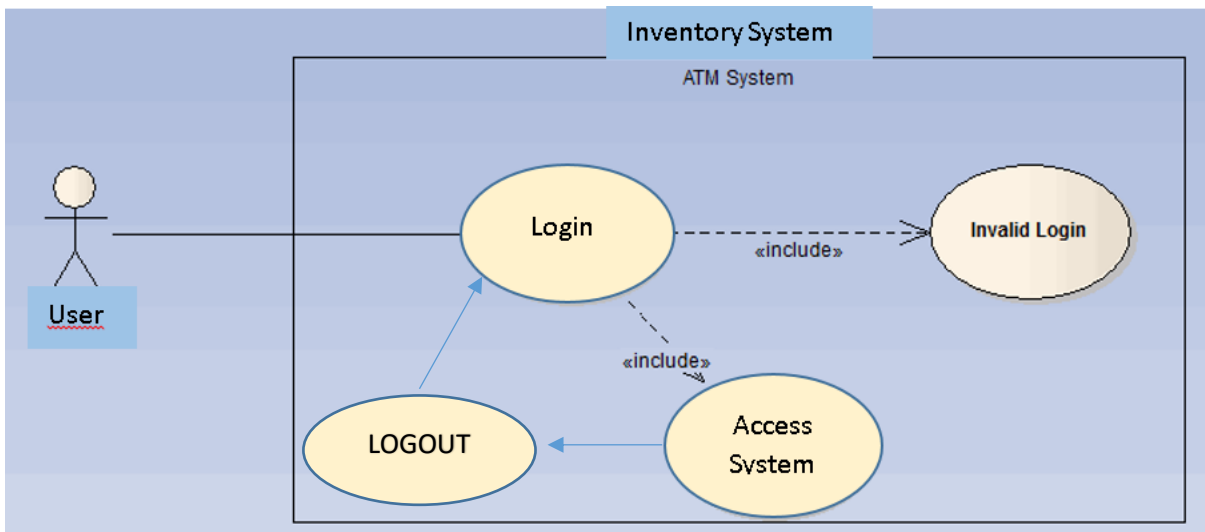


Anexo 8: CASO DE PRUEBA 01 – LOGIN DEL SISTEMA

CP001 – LOGIN AL SISTEMA

Pasos	Resultados Esperados
1.Ingresar con usuario incorrecto	El usuario accede al sistema correctamente.
2.Ingresar con usuario correcto	El usuario no puede acceder al sistema y le mostrarà un mensaje de credenciales incorrectas.

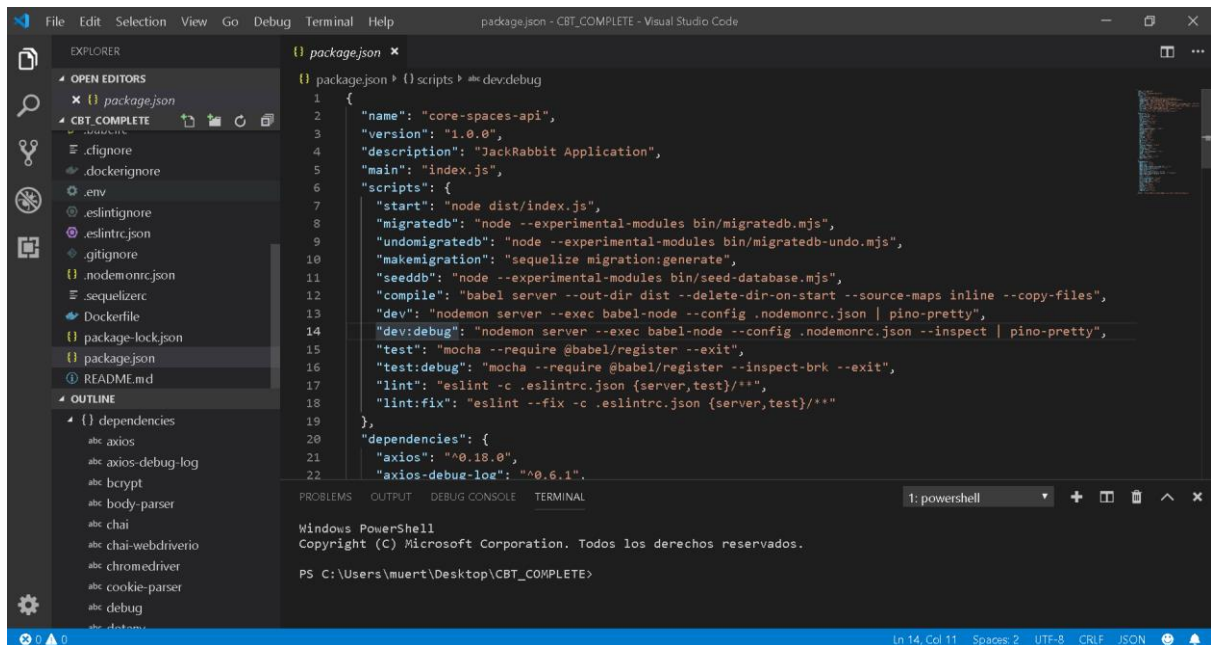
Anexo 9: CASO DE USO 02 – CERRAR SESION DEL SISTEMA



## Anexo 10: CASO DE PRUEBA 02 – CERRAR SESION DEL SISTEMA

### CP002 – LOGIN AL SISTEMA

Pasos	Resultados Esperados
1. Dirigirse a la opción salir del sistema	El sistema debería mostrar un mensaje de confirmación para completar el proceso.
2. Confirmar si desea salir del sistema	El sistema arrojarà al usuario al Login.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the file structure of a project, including files like .gitignore, .env, .eslintrc.json, package-lock.json, and package.json. The main editor area shows the content of package.json, which includes fields for name, version, description, main, and scripts. The scripts section lists various commands such as start, migratedb, undomigratedb, makemigration, seeddb, compile, dev, dev:debug, test, test:debug, lint, and lint:fix. The terminal window at the bottom shows the Windows PowerShell prompt, indicating the current directory is C:\Users\muert\Desktop\CBT\_COMPLETE.

```
{} package.json x
1  {}
2  {
3    "name": "core-spaces-api",
4    "version": "1.0.0",
5    "description": "JackRabbit Application",
6    "main": "index.js",
7    "scripts": {
8      "start": "node dist/index.js",
9      "migratedb": "node --experimental-modules bin/migratedb.mjs",
10     "undomigratedb": "node --experimental-modules bin/migratedb-undo.mjs",
11     "makemigration": "sequelize migration:generate",
12     "seeddb": "node --experimental-modules bin/seed-database.mjs",
13     "compile": "babel server --out-dir dist --delete-dir-on-start --source-maps inline --copy-files",
14     "dev": "nodemon server --exec babel-node --config .nodemonrc.json | pino-pretty",
15     "dev:debug": "nodemon server --exec babel-node --config .nodemonrc.json --inspect | pino-pretty",
16     "test": "mocha --require @babel/register --exit",
17     "test:debug": "mocha --require @babel/register --inspect-brk --exit",
18     "lint": "eslint -c .eslintrc.json {server,test}/**",
19     "lint:fix": "eslint --fix -c .eslintrc.json {server,test}/**"
20   },
21   "dependencies": {
22     "axios": "^0.18.0",
23     "axios-debug-logs": "^0.6.1"
24   }
25 }
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
PS C:\Users\muert\Desktop\CBT\_COMPLETE>

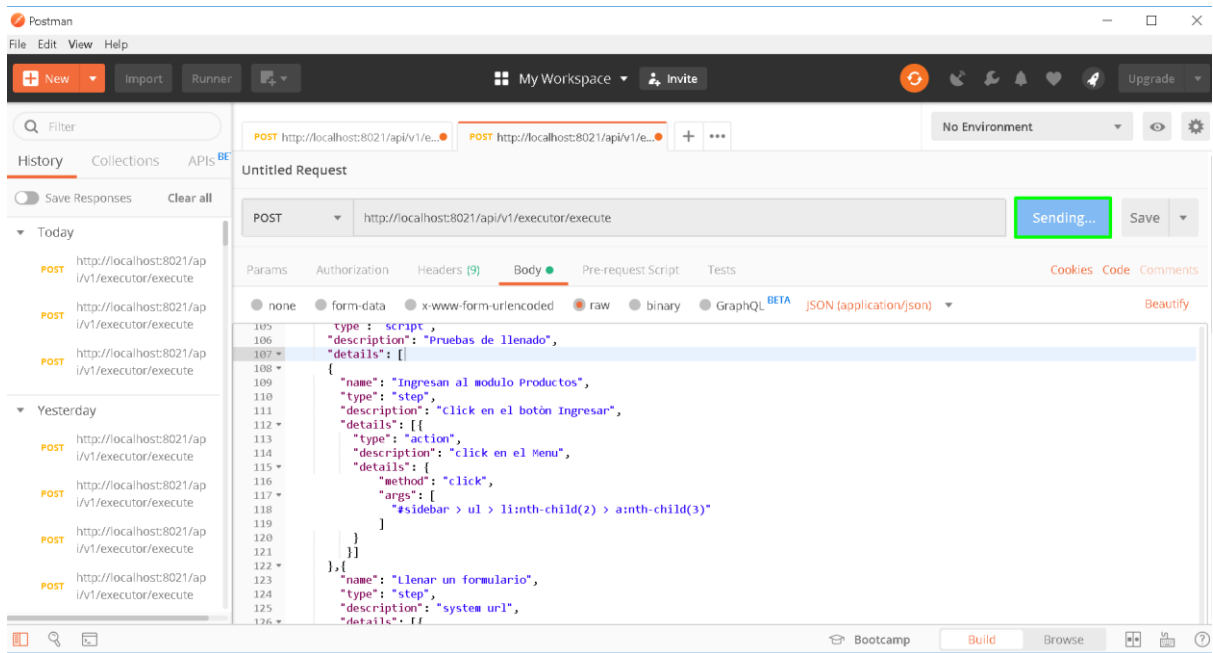
```
server > api > JS router.js
1 import * as express from 'express';
2
3 import executorRouter from './controllers/executor';
4
5
6 export default express
7   .Router()
8   .use('/v1/executor', executorRouter)
9
```

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
PS C:\Users\muent\Desktop\CBT\_COMPLETE>

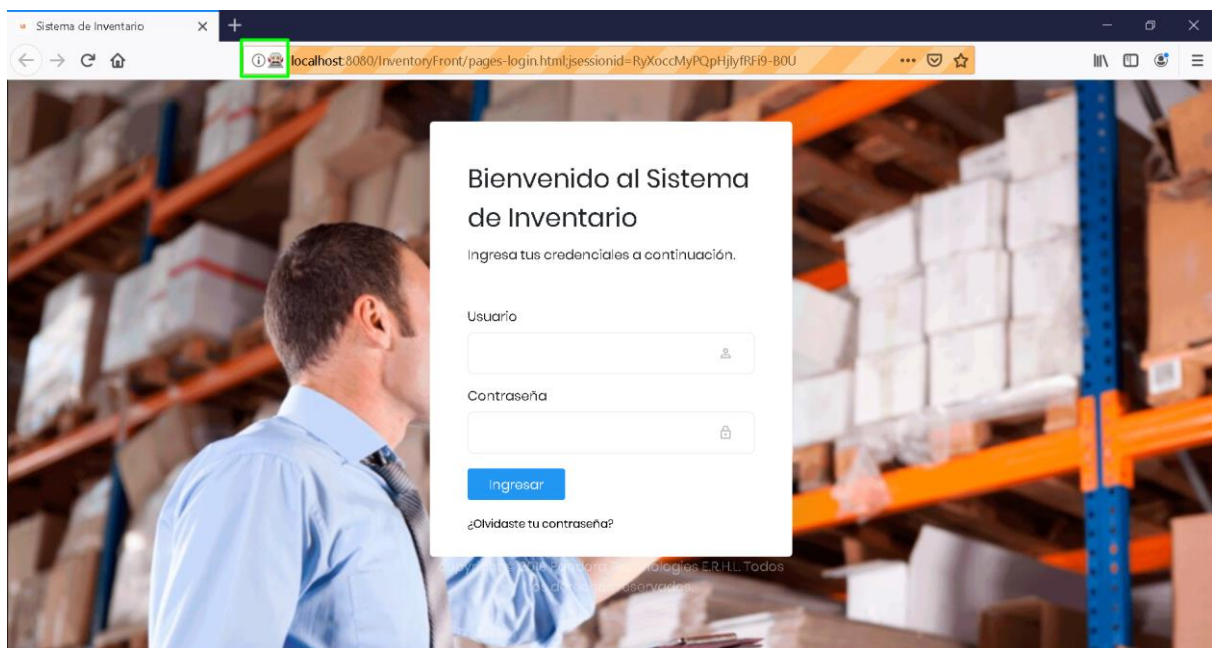
POST http://localhost:8021/api/v1/executor/execute

Body

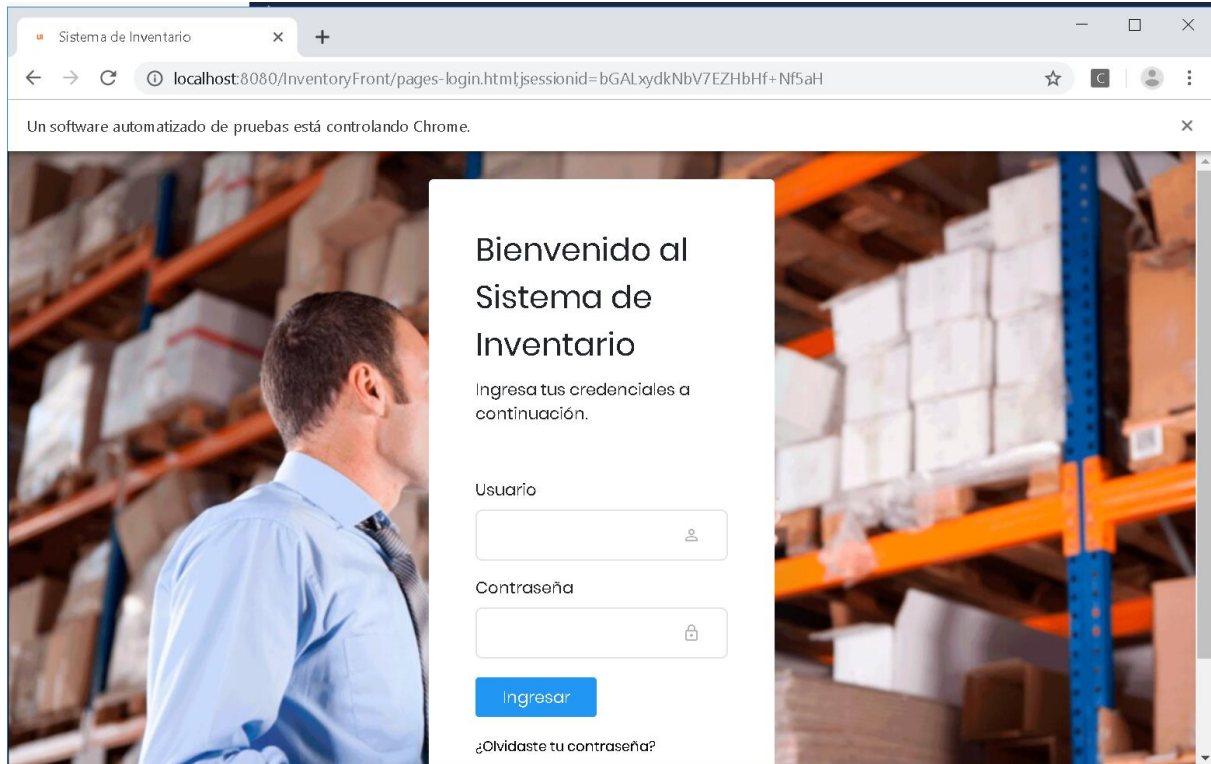
```
1 {
2   "config": {
3     "maxParallelBrowserNumber": 2
4   },
5   "browsers": [{
6     "desiredCapabilities": {
7       "browserName": "firefox"
8     },
9     "desiredCapabilities": {
10      "browserName": "chrome"
11    }
12  }],
13  "dataContext": {
14    "name": "BVL - DATABASE",
15    "description": "A CONNECTION TO THE BVL DATABASE",
16    "type": "sql",
17    "setup": {
18      "database": "Inventory",
19      "host": "127.0.0.1",
20      "port": 5432,
21    }
22  }
23 }
```



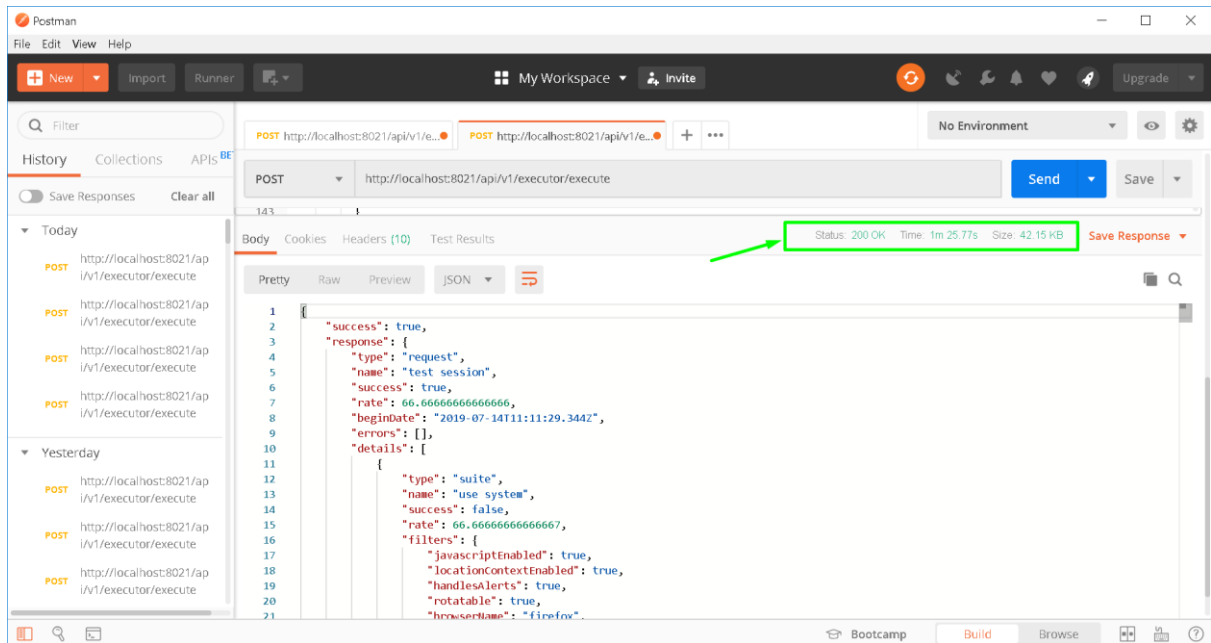
## Anexo 11: FIREFOX



## Anexo 12: CHROME



## Anexo 13: RESULTADOS



## Anexo 14: BASE DE DATOS DE PRUEBAS (POSTGRES)

