



FACULTAD DE CIENCIAS EMPRESARIALES
CARRERA PROFESIONAL DE INGENIERÍA DE
SISTEMAS DE INFORMACIÓN Y GESTIÓN

“Herramientas de automatización de pruebas como propuesta para mejorar la revisión del control del área de Calidad de software en la empresa “Consultora KMC””

Trabajo de investigación para optar el grado académico de:
Bachiller en Ingeniería de Información y Gestión

Presentado por:

Katherine Jannett Mendoza Chiong (0000-0002-0850-329X)

César Augusto Mayorga Patroni (0000-0003-1246-2109)

Lima - Perú

2019



ANEXO 6

**RESULTADO DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN
PARA OPTAR EL GRADO DE BACHILLER**

Fecha: 14/11/2019

Facultad		Ciencias Empresariales	
Carreras de pre grado			
Ingeniería de Sistemas Empresariales	<input type="checkbox"/>	Ingeniería de Sistemas de Información y Gestión - CPE	<input checked="" type="checkbox"/>
Ingeniería Económica y de Negocios	<input type="checkbox"/>	Administración de Empresas - CPE	<input type="checkbox"/>
Administración de Negocios Internacionales	<input type="checkbox"/>		
Administración de Empresas	<input type="checkbox"/>		
Marketing y Administración	<input type="checkbox"/>		
Código	Alumno	Correo electrónico	Teléfono
100032969	Mendoza Chiong, Katherine	100032969@cientifica.edu.pe	942769810
100030220	Mayorga Patroni, César	100030220@cientifica.edu.pe	989021995

Título del trabajo de investigación:

"Herramientas de automatización de pruebas como propuesta para mejorar la revisión del control del área de Calidad de software en la empresa Consultora KMC".

El Jurado revisor concluye que el Trabajo de Investigación para optar el Grado de Bachiller es:

Aprobado (13 – 15)	<input type="checkbox"/>
Aprobado – Muy Buena (16 – 18)	<input checked="" type="checkbox"/>
Aprobado – Sobresaliente (19 – 20)	<input type="checkbox"/>
Desaprobado (0 – 12)	<input type="checkbox"/>

Calificándola con nota de: __16__ en letras (_DIESESIS_)

Comentario/sugerencias/recomendaciones/observaciones:

Atentamente,



 Profesor del curso
 Docente Arturo Martín Álvarez Espinoza

*Este documento puede ser enviado por correo electrónico

INDICE DE CONTENIDO

Contenido

INDICE DE CONTENIDO	3
INDICE DE TABLAS	5
INDICE DE FIGURAS	6
INTRODUCCION	7
CAPITULO I: DESCRIPCION DEL PROBLEMA	8
1.1. Descripción de la realidad problemática.....	8
1.1.1. Situación problemática.....	8
1.1.2. Problema general.....	8
1.1.3. Problemas específicos.....	8
1.2. Objetivos del proyecto.....	9
1.2.1. Objetivo general	9
1.2.2. Objetivos específicos	9
1.3. Justificación del proyecto.....	10
CAPITULO II: MARCO TEORICO	12
2.1. Antecedentes.....	12
2.2. Bases teóricas.....	15
2.2.1. Aplicación Web.....	15
2.2.2. Clasificación de las aplicaciones web	15
2.2.3. Calidad de Software.....	16
2.2.4. Estándares de Calidad.....	17
2.2.5. Verificación de Software.....	27
2.2.6. Validación de Software.....	28
2.2.7. Métodos de verificación y validación de Software.....	28
2.2.8. Proceso de pruebas.....	32
2.2.9. Herramientas de automatización de pruebas.....	33
2.3. Definiciones conceptuales	37
CAPITULO III: DESARROLLO DEL PROYECTO	39
3.1. Descripción de solución de negocio propuesta.....	39
3.1.1. Características para medir en el estudio comparativo	39
3.1.2. Análisis de Resultados	40

3.2.	Arquitectura empresarial.....	42
3.2.1.	Arquitectura de Información (Datos).....	42
3.2.2.	Arquitectura de Negocios (Procesos).....	45
3.2.3.	Arquitectura de Aplicación.....	47
3.2.4.	Arquitectura Tecnológica.....	48
•	Servidor Base de Datos: La capa de datos está hecha de las administraciones proporcionadas por el supervisor de base de datos. La información se guarda en una base de datos del marco de ejecutivos que ahora mismo MySQL	49
3.2.5.	Factibilidad Económica.....	50
CAPITULO IV: RECURSOS Y CRONOGRAMA		51
4.1.	Recursos	51
4.2.	Cronograma de ejecución.....	52
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES		53
5.1.	Conclusiones.....	53
5.2.	Recomendaciones.....	54
CAPITULO VI: FUENTES DE INFORMACION		56
Bibliografía		56
ANEXOS		57
6.1.	Anexo 1	57
6.2.	Anexo 2	58

INDICE DE TABLAS

Tabla 1 Comparativa de ISO 9126 por Objetivos	18
Tabla 2 Comparativa de Métodos de Verificación y Validación de Software	29
Tabla 3 Comparativo de Especificaciones Técnicas y Funcionales	41
Tabla 4 Comparativo de Especificaciones Conceptuales	41
Tabla 5 Factibilidad Económica.....	50
Tabla 6 Presupuesto de Hardware y Soporte.....	51
Tabla 7 Presupuesto de Recursos Humanos.....	51
Tabla 8 Presupuesto de Licenciamiento y Despliegue	51
Tabla 9 Modelo de la Evaluación de la Calidad de Software.....	57
Tabla 10 Herramienta Sahi Pro.	58

INDICE DE FIGURAS

Figura 1 Aplicación Web Estática.....	15
Figura 2 Aplicación Web Dinámica.....	16
Figura 3 Características de la Calidad Interna y Externa.	18
Figura 4 Características de Funcionalidad	19
Figura 5 Característica de Confiabilidad.....	20
Figura 6 Características de Usabilidad.....	21
Figura 7 Característica de Eficiencia	23
Figura 8 Característica de Mantenibilidad	24
Figura 9 Característica de Portabilidad	25
Figura 10 Característica de Calidad de Uso	26
Figura 11 Modelo de Gestión.....	43
Figura 12 Datos Físicos.....	43
Figura 13 Datos Lógicos	44
Figura 14 Flujo de Datos	44
Figura 15 Estructura Organizacional de la Consultora KMC	46
Figura 16 Diagrama de Flujo de Procedimiento de Atención de Proyectos Consultora KMC ...	46
Figura 17 Diagrama de Flujo de Procedimiento de Atención de Proyectos después de Implementación de Herramienta en la Consultora KMC	47
Figura 18 Arquitectura de Aplicación de Sagi Pro.....	47
Figura 19 Arquitectura Tecnológica de la Consultora KMC	48
Figura 20 Cronograma Gantt.....	52

INTRODUCCION

Las consultoras TI son proveedores de servicios, y son intermediarios entre el cliente y el usuario final, para ofrecer un elemento de programación de calidad, por tanto, esta consultora TI con años de experiencia en el mercado, contribuye al mercado y la economía peruana. Los productos de software desarrollados por esta consultora TI son el pilar del servicio, por tano deben mantener y superar un nivel de calidad competitivo, mejorando su desempeño en las pruebas que se realizan a su software, incluyendo herramientas que brinden funcionalidades que ayuden a mejorar cualidades como funcionalidad, confiabilidad, usabilidad, mantenibilidad y eficiencia. La consultora TI, ha elegido técnica para examinar los aparatos para la prueba de informatización, y así optimizar la ejecución de pruebas y evitar la alta demanda de observaciones generadas en el área de desarrollo.

CAPÍTULO I: DESCRIPCIÓN DEL PROBLEMA

1.1. Descripción de la realidad problemática

1.1.1. Situación problemática

La demanda del desarrollo de software se desenvuelve a un ritmo anual del 8,5% desde 2010 debido al interés de las administraciones innovadoras en el mercado, comprendiendo exigencias del sector financiero que buscan un crecimiento dentro y fuera del país, y para ello será necesario crear o mejorar la calidad de los aplicativos que son desarrollados. Una de las compañías es Consultora KMC situada en el Centro de Lima y cuenta con más de 11 años de experiencia construyendo software.

1.1.2. Problema general

Los gastos asociados al levantamiento de observaciones y entregas fuera de fecha que sufren los proyectos por esta razón, impacta considerablemente en la economía de la empresa, quienes deben responsabilizarse por el incumplimiento a los acuerdos (SLA's), este problema va incrementando y debido a esto se genera pérdida en competitividad, incremento de presupuesto para los proyectos de desarrollo, afectando la rentabilidad de la organización.

1.1.3. Problemas específicos

- El trabajo operativo actual afecta el nivel de funcionalidad de las pruebas realizadas en los aplicativos para la empresa y hace que se vea afectada la exactitud, seguridad y conformidad de la funcionalidad de pruebas en la Consultora KMC.

- El trabajo operativo actual afecta el nivel de confiabilidad de las pruebas realizadas en los aplicativos para la empresa y hace que se vea afectada la madurez, tolerancia de errores, recuperabilidad y conformidad de la fiabilidad en la Consultora KMC.

- El trabajo operativo actual afecta el nivel de usabilidad de las pruebas realizadas en los aplicativos para la empresa y influye en la impresión de clientes finales y clientes indirectos del producto en la Consultora KMC.

- El trabajo operativo actual afecta el nivel de mantenibilidad de las pruebas realizadas en los aplicativos para la empresa y hace que se vea afectada la naturaleza del producto que se va a cambiar, incluidas las correcciones o actualizaciones realizadas en el software en la Consultora KMC.

- El trabajo operativo actual afecta el nivel de eficiencia de las pruebas realizadas en los aplicativos para la empresa y hace que se vea afectado el desempeño de acuerdo con el número de recursos que utilizan el software en la Consulta KMC.

1.2. Objetivos del proyecto

1.2.1. Objetivo general

Identificar las herramientas disponibles en el mercado que ayuden a un correcto funcionamiento del software y mejorar el proceso de Pruebas de Calidad para empresa Consultora KMC.

1.2.2. Objetivos específicos

- Investigar softwares que cumplan con el nivel de funcionalidad suficiente para aplicarla a la empresa Consultora KMC.
- Investigar softwares que cumplan con el nivel de confiabilidad suficiente para aplicarla a la empresa Consultora KMC.
- Investigar softwares que cumplan con el nivel de usabilidad suficiente para aplicarla a la empresa Consultora KMC.
- Investigar softwares que cumplan con el nivel de mantenibilidad suficiente para aplicarla a la empresa Consultora KMC.
- Investigar softwares que cumplan con el nivel de eficiencia suficiente para aplicarla a la empresa Consultora KMC.

1.3. Justificación del proyecto

Las consultoras de tecnologías de la información tienen como obligación, estar alineadas a cumplir y brindar estándares de calidad e innovación. El área de calidad debe tener un alto nivel de ocupación al momento de realizar sus pruebas para acortar la posibilidad de atrasos que pongan en riesgo la veracidad de sus clientes. Al presente, los procesos de prueba varían entre los casos manuales y automatizados, pero el enfoque de la metodología es conservar el 80% de las pruebas de forma automatizada y las pruebas manuales sean esos casos de prueba de gran complejidad o con parámetros especiales. Este cubrimiento en el proceso de aseguramiento de calidad conseguiría mantener la confianza que brinda a sus clientes.

Para conseguir este objetivo, es que se realiza este trabajo de investigación, y que por nuestra parte y en conjunto con un equipo de analistas de pruebas evaluaremos los softwares para automatización de pruebas para que sea usado por el área de tecnología de la empresa Consultora KMC.

Con esta investigación se obtendrán herramientas que fomentarán el avance de las pruebas automatizadas, la capacidad del grupo del área de calidad y la rapidez de los equipos que trabajan con la metodología SCRUM dentro de la compañía.

CAPITULO II: MARCO TEORICO

2.1. Antecedentes

Mendoza (2010), el hace un aporte intrigante con respecto a su exploración al caracterizar un modelo de referencia para la elección de aparatos de prueba como ayuda en el proceso de avance del producto, que se suma a mejorar el procedimiento de creación de estas organizaciones e incrementar la naturaleza de sus artículos, tomando como caso investigación de diseñadores de programación de pequeñas y medianas empresas (PYME). Muestra que la afirmación de la calidad ha sido un problema en la Ingeniería del Software, y que las pruebas son una fase que debe estar disponible en la mejora de la programación para garantizar la naturaleza del elemento, por lo que una investigación exhaustiva de los aparatos de prueba como lo indican los atributos y las medidas exhibidas en el modelo introducido que permitirán la evaluación de los instrumentos que existen en el mercado según sus atributos que realmente no se aplican a una amplia gama de pruebas ni a todo el proyecto.

Ciolti (2007), muestra que el desarrollo exponencial extraordinario en la web fue la motivación detrás de por qué varias organizaciones trasladaron sus aplicaciones de área de trabajo a un dominio web que les permite ser progresivamente transformadoras, efectivamente abiertas y cada vez más serias en contraste con enormes organizaciones que a partir de ahora tienen aplicaciones en estos etapas, hay algunos cambios de enfoque entre estos dos tipos de usos; que se extienden desde la forma en que crean hasta la forma en que son juzgados y aprobados. En este momento, la investigación demuestra una importancia extraordinaria para la etapa de prueba, reconociendo la caja blanca como la prueba

fundamental; donde las pruebas fueron informatizadas para la aprobación del código, lo que le permitió tener registros de prueba que luego se guardarían en un archivo de formulario que almacena los errores encontrados. Otro punto importante que se puede distinguir durante el período de mejora de los experimentos es que ahorra tiempo, ya que estos se estructuran durante el examen del marco y es un método para informar la aplicación. Además, en la estructura inicial de los experimentos existe la posibilidad de encontrar errores en el detalle de las necesidades por descuido o error en el significado de estos. Para la prueba de coordinación, utilizaron el dispositivo gratuito Rational Robot, que aprobó una exhibición superior en la aplicación difundida, ya que al ejecutar el robot en la aplicación web, hubo algunos enfrentamientos claros en el montón de la aplicación. Este instrumento también fue útil para las pruebas de recaída, permitiendo pruebas en varios programas

Esteve (2015), el proyecto surge en la necesidad de ocuparse del tema de la constante sin una acción de prueba explicada y suficiente. Últimamente, el gran acto de utilizar los recursos para tener equipo de testeo que hace el trabajo particular y satisfactorio para garantizar el mejor funcionamiento posible de los artículos y la transmisión de la programación de valor se está expandiendo. La acción de prueba computarizada asume un trabajo significativo en el ciclo de mejora; Uno de los enfoques para transmitir un artículo probado y creado para el cliente es tener un proceso de prueba mecanizado decente que ayude a una necesidad inexorablemente solicitada en la calidad renovada, sin embargo, en plazos progresivamente breves, junto con una base decente detrás de la cual puede apoyarse todo el tiempo de manera rápida y cumpliendo las limitaciones de tiempo establecidas con los clientes. Se ha caracterizado un procedimiento de prueba

dependiente de AP y confirmado por dispositivos con los que este procedimiento se ha archivado y delineado con modelos para una mejor comprensión. Como encuentro individual, puedo decir que después de haber hecho este TFG, he descubierto cómo estimar que es tan crítico aplicar las pruebas computarizadas a la mejora de la programación y ver que al final no es simplemente una hipótesis, pero poco a poco funciona y da resultados. Por ejemplo, voy a informarle sobre el progreso que la organización de la que soy individuo ha tenido que hacer, siendo una organización dedicada a hacer aplicaciones para el control de pacientes en arreglos de vivienda, un gran número de sus clientes han solicitado tienen una mayor versatilidad por parte del personal de los arreglos de vivienda, siendo una PC un marco estático, por lo que han utilizado una parte decente del personal para hacer un marco material para que el personal del hogar pueda acercarse a los datos silenciosos mientras los trata sin depender de una PC, un De manera más práctica y completamente renovada. En este momento, esperaba crear un módulo diferente para la informatización de dicho marco, ya que no se trata de controles como los utilizados para una aplicación de área de trabajo. Esta es la explicación detrás de la importancia que debe tener para el desarrollo de aplicaciones de programación, intentando continuamente crear marcos resumidos y con la visión de los posibles cambios que pueden ocurrir, en cualquier caso, a corto o mediano plazo, para reutilizar todo lo que debería ser posible en el uso futuro. En este momento, se ha reflejado la importancia del procedimiento de mecanización, ya que han descubierto cómo transmitir las entregas al cliente garantizando un nivel de valor genuinamente alto y llegando a tiempos de corte debido a la utilización de dicho proceso

2.2. Bases teóricas

2.2.1. Aplicación Web

Las aplicaciones web pueden ser clasificadas bajo ciertos razonamientos, que podría depender del modelo de arquitectura o configuración que use, según la tecnología o lenguaje utilizado o por la capacidad de comunicación que pueda tener con los usuarios en respuesta a sus interacciones con la interfaz que se les presenta. Son estos razonamientos según los autores mencionados que dan lugar a la clasificación de las aplicaciones web en: estáticas, dinámicas e interactiva (Lerma B, Murcia A, Mifsud T, 2013, P.16).

2.2.2. Clasificación de las aplicaciones web

- Aplicaciones web estáticas

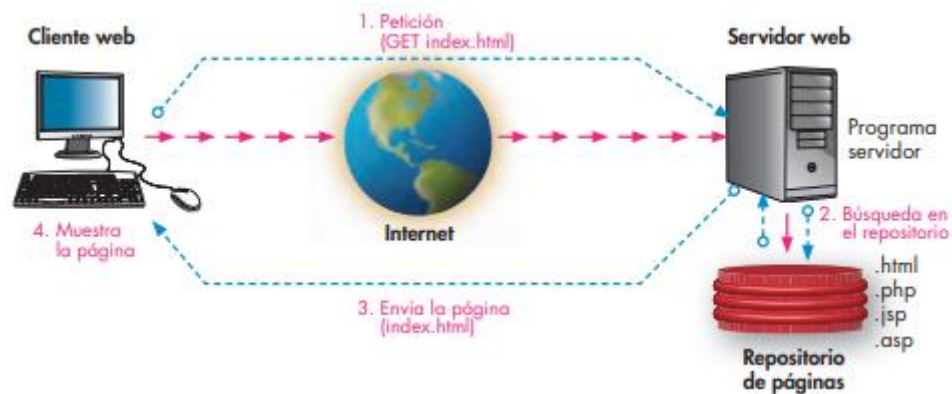


Figura 1 Aplicación Web Estática

Fuente: Elaborado por Lerma-Blasco, Murcia A, & Mifsud T, 2013

- Aplicaciones web estáticas



Figura 2 Aplicación Web Dinámica

Fuente: Elaborado por Lerma-Blasco, Murcia A, & Mifsud T, 2013

2.2.3. Calidad de Software

Según Rogers (2005): “es según las necesidades de los requisitos funcionales de ejecución creadas con los indicadores de mejora expresamente documentados y las características entendidas que se anticipan de toda la programación desarrollada por profesionales”.

Existen estándares internacionales como la ISO 25000:2005 SQuaRE, que permite el detalle de los requisitos previos y la evaluación de la naturaleza del elemento (software), dando el manual para los controles de una progresión de los estándares internacionales, por ejemplo:

- ISO/IEC 9126 donde forma un modelo hacia la siguiente evaluación de la calidad del producto(software), mediante características de calidad, existe un modelo llamado MECABIC que hace uso de una matriz para evaluar software (Anexo 1) que permite enfocarse en atributos que dependen únicamente de ingeniería, y esta ser una estándar fomenta la correspondencia con las cualidades consideradas por las técnicas contempladas.

- IEEE 829-2008 sigue un plan adecuado para evitar los posibles problemas que puedan presentarse en los sistemas informáticos y otros tipos relacionados, desarrollando una metodología de prueba, que consta de planes de pruebas para así diseñarlas después, ejecutarlas esperando los resultados para poder en la siguiente fase evaluar las pruebas.
- ISO/IEC 20119 se encarga de definir el proceso de que se sigue para la presente puesta en marcha de las pruebas de software, distinguiendo metodologías y enfoques, incorporando redacción, documentación y sistemas para todo el ciclo de vida de las pruebas de programación.

2.2.4. Estándares de Calidad

Son parte importante de Ing. Software, criterios que dirigen la forma que deben ser aplicadas tanto la metodología y las técnicas, con el fin de permitir reunir la teoría del trabajo, se necesita poder lograr una perfecta confiabilidad, viabilidad y simplicidad más notables. Aumentando la eficiencia para el avance como también el seguimiento de la calidad de la programación.

2.2.4.1 ISO 9126

un estándar mundial para el seguimiento de la programación, que establece los atributos de calidad para los elementos de programación.

Tiene 4 partes:

1. (FDIS 9126-1) que es el modelo de calidad, representa un modelo de las secciones para programar la calidad del elemento:
 - i. Calidad interna y externa

- ii. Calidad en uso
- 2. (9126-2 PDTR) que son mediciones externas.
- 3. (9126-3 PDTR) que son mediciones internas.
- 4. (9126-4 PDTR) que son mediciones de calidad en uso.

Tabla 1 Comparativa de ISO 9126 por Objetivos

Estándar	Objetivo Principal
ISO/IEC 9126-1	Modelo de calidad
ISO/IEC 9126-2	Métricas externas
ISO/IEC 9126-3	Métricas internas
ISO/IEC 9126-4	Métricas de calidad de uso

Fuente: Elaboración propia

Calidad Interna y Externa

El estándar ISO-9126 desarrolla los atributos para la calidad interna y externa, atributos que permiten una evaluación más profunda de la naturaleza de los elementos de programación. Ver Figura 4.



Figura 3 Características de Calidad Interna y Externa.

Fuente: Largo & Marin ,2005

Definiciones según el modelo de calidad del estándar ISO-9126:

1. Funcionalidad

Conjunto de atributos que nos permite evaluar si un artículo del producto trata satisfactoriamente la disposición de las capacidades que cumplen los requisitos para los cuales fue estructurado.



Figura 4 Características de Funcionalidad

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- **Adecuación**
Se centra en evaluar si el producto tiene muchas capacidades de adaptación para ejecutar las tareas las cuales fueron indicadas.
- **Exactitud**
Evalúa que el artículo exhiba resultados según las necesidades para las cuales fue creado. Este crédito evalúa si el producto presenta resultados o impactos según los requisitos para los que se creó.
- **Interoperabilidad**
Evalúa la capacidad del software para colaborar con varios sistemas recientemente indicados.

- Cumplimiento

Evalúe si el producto cumple con las normas, espectáculos o pautas de leyes y soluciones comparables.

- Seguridad

Alude a la capacidad de prevenir el acceso no aprobado, independientemente de si es casual o planificado, a proyectos e información.

2. Confiabilidad

La calidad inquebrantable une muchos créditos que aluden la capacidad de un producto para sostener su nivel durante un tiempo establecido.



Figura 5 Característica de Confiabilidad

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- Nivel de desarrollo

Permite estimar la recurrencia de la decepción debido a errores que nos brinda el producto.

- **Aceptación a fallos**
Alude la capacidad de sostener un grado particular de actividad de fallos de programación o talvez de presentar una transgresión de su interfaz particular.
- **Límite de recuperación**
La capacidad de restituir el nivel de acción y recuperar todo aquello que haya sido directamente dañado por la frustración, así como el tiempo y el esfuerzo para hacerlo posible.
- **Cumplimiento de calidad inquebrantable**
Alude a la capacidad del elemento del producto para aferrarse a indicadores, espectáculos o pautas identificadas con fiabilidad.

3. Usabilidad

El uso acumula muchos créditos que dejan evaluar todo esfuerzo que es importante para que el cliente pueda contribuir para utilización del programa.



Figura 6 Características de Usabilidad

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- Capacidad de comprensión
Alude la capacidad del software(producto) que aprueba que el cliente comprenda si el producto es razonable y cómo tiende a utilizarse para tareas específicas y estados de utilización.
- Capacidad de ser aprendido
Capacidad del software que permita que el cliente se familiarice con su aplicación.
- Capacidad de ser operado
Capacidad del producto que permita que el cliente trabaje y lo controle.
- Límite de atracción
Alude a la capacidad del artículo del producto para atraer al cliente.
- Cumplimiento de usabilidad
Alude la capacidad del producto para adjuntarse a los estándares, espectáculos o pautas de los suegros y pautas que se identifican con conveniencia.

4. Eficiencia

Une muchas propiedades que permiten evaluar la conexión entre el grado de actividad de programación y la medida de los activos utilizados.



Figura 7 Característica de Eficiencia

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- Comportamiento del tiempo
Alude la capacidad del artículo del producto los tiempos de manipulación y porcentaje de ejecución en exhibición de su capacidad en condiciones obtenidas.
- Uso de recursos
Alude a la capacidad del artículo del producto para utilizar sumas de ajuste y tipos de activos cuando el producto desarrolla su capacidad en condiciones que se establecen.
- Cumplir con la eficiencia
Alude la capacidad del artículo para aferrarse a los principios, espectáculos o pautas de los seguros y pautas identificadas con la productividad.

5. Mantenibilidad

Une muchas cualidades que permiten estimar el esfuerzo requerido para realizar alteraciones en el producto, independientemente de si se debe al ajuste

de errores, actualizaciones, cambios en las necesidades o la expansión de la utilidad.



Figura 8 Característica de Mantenibilidad

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- Capacidad para ser desglosado
 La capacidad del artículo para poder ser analizada por razones para decepciones de programación, o para reconocer sus partes para ser ajustadas.
- Capacidad de cambio
 Capacidad del artículo(software) para poder permitir que se actualice un cambio particular.
- Estabilidad
 Capacidad del artículo(software) para evadir los impactos imprevistos de las alteraciones de programación.
- Capaz a ser probado
 Capacidad del artículo para que permita la aprobación de las modificaciones que se realicen.

- Cumplir en la mantenibilidad

Capacidad del artículo para apearse a los principios, espectáculos o pautas de los suegros y pautas identificadas con Mantenimiento.

6. Portabilidad

Une muchas propiedades que permiten estimar la capacidad del elemento del producto para moverse comenzando con un medio y luego al siguiente, un dominio puede incorporar asociación, equipo o programación.



Figura 9 Característica de Portabilidad

Fuente: Elaborado por Borbón, N., 2014

Subcategorías:

- Ser capaz de adaptación

Capacidad del artículo para ajustarse a varias situaciones explícitas, sin poder aplicar actividades o medios distintos de los esperados por este motivo.

- Ser capaz de Instalación

Capacidad del artículo para poder ser introducido dentro de un dominio particular.

- Convivencia

Capacidad del elemento del producto para poder coincidir con otra programación autónoma en una situación típica al compartir activos regulares.

- Capaz de ser reemplazado

Capacidad del artículo que se utilizará en lugar de otro artículo del producto por un motivo similar en un dominio similar.

- Cumplimiento de portabilidad

La capacidad del producto de software que se ciñe a las políticas, acuerdos o normas en leyes y ordenanzas asociados a la portabilidad.

Calidad de uso

Calidad utilizada alude al reconocimiento del cliente que es el usuario final.

Modelo de calidad del estándar nos dice que las características que lo acompañan para la calidad que se utiliza:



Figura 10 Característica de Calidad de Uso

Fuente: Elaborado por Borbón, N., 2014

- Efectividad:

Capacidad del artículo del producto para consentir a los clientes lograr sus metas con precisión y culminación, en un entorno predeterminado de utilización.

- **Productividad:**

Capacidad del artículo del producto para consentir a los clientes aprovechar una medida satisfactoria de los activos de acuerdo con la adecuación lograda, en un entorno predefinido de utilización.

- **Seguridad física:**

Capacidad del artículo del producto para llegar a los grados adecuados de peligro de dañar a las personas, los negocios, la programación, las propiedades o la tierra en un entorno de utilización predefinido.

- **Satisfacción:**

Capacidad del artículo del producto para complacer a los clientes en un entorno predeterminado de utilización.

2.2.5. Verificación de Software

La verificación del producto consiste en decidir si las técnicas que se originan a partir de un ciclo de vida del producto son según sus determinaciones y satisfacen las necesidades prácticas y no útiles que se han indicado, en otras palabras, que está libre de errores.

2.2.5.1 Verificación Estática

Gestiona la investigación de representaciones estáticas del marco para encontrar problemas. Pueden complementarse con informes que dependen de los dispositivos y el análisis (S. Rakitin,2001).

2.2.5.2 Verificación Dinámica

Gestiona la actividad y la percepción de la conducta del artículo. Permite que el descubrimiento de imperfecciones se una con otros controles de calidad (I. Sommerville,2005).

Este tipo de confirmación se fija en la percepción y la conducta del último producto.

2.2.6. Validación de Software

La aprobación de la programación es para garantizar que el producto creado cumpla con los deseos del cliente, esto implica que la aprobación va más allá de verificar si el marco es según sus detalles. La aprobación de programación se concentra en demostrar que el producto realizar lo que el cliente necesita y no simplemente en verificar lo indicado.

2.2.6.1 Validación Simple

La aprobación directa concentra su trabajo en investigaciones a través de procedimientos establecidos para aprobar la coherencia con las necesidades acumuladas en el avance.

2.2.6.2 Validación Cruzada

La aprobación cruzada tiene un sistema utilizado para ajustar las secuelas de un examen fáctico. Utilizado en empresas de Inteligencia Artificial para aprobar modelos producidos.

2.2.7. Métodos de verificación y validación de Software

Tabla 2 Comparativa de Métodos de Verificación y Validación de Software

Test	Objetivo Principal	Participación	Ambiente	Método
Pruebas Unitarias	Correcto funcionamiento del código	Desarrollador	Desarrollo	Caja Blanca
Pruebas de Integración	Correcto funcionamiento de lo probado unitariamente también funcione en conjunto	Desarrollador	Desarrollo	Caja Blanca/Top-Down
Pruebas Funcionales	Correcto funcionamiento según lo definido	Testers y Analistas	Desarrollo	Herramientas Funcionales
Pruebas de Sistema		Testers y Analistas	Desarrollo	Herramientas Funcionales
Pruebas de Aceptación		Testers, analista y cliente	Producción	Herramientas Funcionales

Fuente: Elaboración propia

2.2.7.1 Método de Caja Negra

Los ensayos de descubrimiento se fijan en lo que se anticipa de un módulo, es decir, intentan descubrir asuntos en los que el módulo no concuerda a su determinación. Esa es la razón por la que se llaman pruebas utilitarias, el analizador está obligado a ingresar información como fuentes de datos y considerar los rendimientos, sin preocuparse por lo que el módulo podría estar haciendo dentro (S. Rakitin,2001).

Las pruebas de descubrimiento intentan descubrir errores, por ejemplo:

- Capacidades incorrectas o faltantes.
- Errores de interfaz.
- Errores en estructuras de información o acceso a base de datos externas.
- Errores de rendimiento.

2.2.7.2 Método de Caja Blanca

Llamados también pruebas de cobertura en las que se conoce la estructura interior del software, ya que se crea de manera no normal para la estrategia de

descubrimiento en este momento, se detalla el uso. El objetivo principal es probar la lógica del programa desde la perspectiva algorítmica (S. Rakitin,2001).

La posibilidad de esta técnica es estructurar un plan de prueba en el que el código se ejecute deliberadamente hasta que se haya ejecutado todo o con mucho, esto se convierte en un sistema confuso cuando el programa contiene código difícil de entender.

2.2.7.3 Método Top-Down

Cada pieza del sistema se ensaya con detalles más destacados considerando características y ejemplos de calidad, hasta que el detalle completo sea punto por punto suficiente para aprobar el modelo. El modelo Top Down se plantea con la ayuda de elementos secretos (S. Rakitin,2001).

La desventaja de esta técnica es: no reconocer la mayoría de errores.

2.2.7.4 Método Act-like-a-customer

Una estrategia donde las pruebas se crean dependiendo de la información de como utilizan el software los clientes, es decir, se centran en encontrar las imperfecciones que los clientes pueden descubrir (S. Rakitin,2001)..

Uno de los impedimentos es que los elementos de programación desconcertantes tienen numerosos defectos y los clientes habitualmente localizan un pequeño nivel de ellos.

2.2.7.5 Método Act-like-a-customer

El método de análisis de compensación de ingeniería es una estrategia que nos hace elegir un diseño razonable para un marco de producto que aborde problemas de calidad. La técnica ATAM es útil cuando se realiza desde el principio en la mejora de la programación, cuando el gasto de cambiar los modelos es insignificante.

2.2.7.6 Método ADR

La estrategia Active Design Review, al igual que la técnica ATAM garantiza la naturaleza del artículo del producto, ADR se utiliza para evaluar las unidades de programación, por ejemplo, módulos o segmentos.

En este momento, se les pide que en general se coordinen hacia:

- La calidad y la honestidad de la documentación.
- La adecuación de las administraciones otorgada por el plan.

2.2.7.7 Método ARID

La estrategia de revisión activa para el diseño intermedio es la mixtura entre la técnica ATAM y la técnica ADR que explota los beneficios de cada una de estas estrategias.

Las ventajas de emplear ARID son las siguientes:

- Esta técnica es razonable para evaluar planes intermedios en los tiempos iniciales de avance.

- Permite la divulgación de errores a tiempo, lo que ayuda a saber si el plan es razonable.

2.2.8. Proceso de pruebas

2.2.8.1 Niveles de Pruebas

Pruebas Unitarias

Contrastan la utilidad y la estructura de cada parte por separado luego de haber sido codificado.

Pruebas de Integración

Confirman la unión correcta entre los diversos segmentos de un marco, una vez que se han intentado de manera unitaria, con una consideración poco común de las interfaces, internas como externas.

Pruebas de Sistema

Prueba completamente el software, verificando la utilidad y rectitud universalmente, en un dominio lo más cercano posible a la última condición de creación.

Pruebas de Implantación

Verifican el funcionamiento correcto del marco dentro de la condición de creación genuina.

Pruebas de Aceptación

Contrastan que el sistema ejecute todas las necesidades mostradas y permite a los clientes del marco dar el último aval.

Pruebas de Regresión

Observe los cambios en una parte del sistema no presentan conductas indeseables o caídas en otros segmentos no se modifican.

Pruebas de Valores de Frontera

Prueba de borde conocido como prueba de límite, es una estrategia para los experimentos de configuración que hacen hincapié en las pruebas que se acercan lo más posible a las cualidades permitidas para ver la conducta del sistema.

2.2.9. Herramientas de automatización de pruebas

Hay dispositivos que ayudan a diferentes partes de la prueba. La siguiente es una disposición potencial para los instrumentos:

- Administración de las pruebas y el procedimiento de las pruebas: dispositivos para la organización de las pruebas, para el seguimiento de episodios, para la administración de la configuración y para la organización de las necesidades.
- Pruebas estáticas: dispositivos para ayudar al procedimiento de auditoría, instrumentos para examen estático y dispositivos de demostración.
- Especificación de las pruebas: aparatos para la estructura de las pruebas y para la disponibilidad de la información de los ensayos.

- Ejecución de prueba: aparatos de ejecución de experimentos, dispositivos de prueba unitaria, comparadores, dispositivos de estimación de inclusión, instrumentos de seguridad.
- Rendimiento y verificación: aparatos de investigación dinámica, dispositivos de ejecución, carga y estrés, dispositivos de observación.

Para la mecanización de pruebas prácticas, se muestran particularmente los dispositivos para la ejecución de pruebas de captura y proliferación. Estos instrumentos permiten al analizador capturar y registrar pruebas, y luego alterarlas, cambiarlas y replicarlas en varias condiciones. Durante la cuenta, se capturan las actividades realizadas por el analizador, lo que naturalmente hace un contenido en un lenguaje de nivel significativo. En ese punto, el analizador altera el contenido para hacer una prueba reutilizable y viable. Este contenido se convierte en el indicador y luego se imita en otra adaptación, contra la cual se mira. Cuando todo está dicho, estos aparatos se unen por un comparador, que naturalmente observa el rendimiento a la hora de ejecutar el contenido con el rendimiento registrado.

Los aspectos más destacados que debe cubrir un útil instrumento de informatización de prueba son los siguientes:

- Integración con otros aparatos de avance. El procedimiento de prueba es uno de los períodos del ciclo de avance del producto. Regularmente, esta tarea de prueba la realiza el experto en pruebas (o analizador) que forma parte de un grupo de mejora del producto, que probablemente utilizará un dispositivo en particular. De esta manera, es fascinante que el producto utilizado por el analizador

se incorpore al marco que se está utilizando para el avance de la tarea.

- Automatización de pruebas prácticas sobre diversos tipos de innovación de uso. Es ventajoso que el aparato no esté restringido claramente a un tipo particular de uso. Es fascinante que pueda trabajar con aplicaciones creadas en varias innovaciones, por ejemplo, Java, .NET, etc.
- Ejecución de pruebas en premisa circulada. El dispositivo debe permitir un proceso de prueba difundido. Al final del día, debe aceptar la ejecución sincronizada de pruebas en varias PC. Es decir, dar la alternativa de tener la opción de ejecutar algunas pruebas se extiende en algunos destinos de trabajo, al mismo tiempo o sincronizados, lo que permite realizar una prueba razonable de múltiples clientes de aplicaciones diseminadas.
- Varias opciones de idioma para los contenidos del examen. Es intrigante que el dispositivo no utilice un idioma particular para el contenido de la prueba, sin embargo, consolida diferentes dialectos de prueba, utilizados adicionalmente para el avance de la aplicación.
- Opción para configurar el lugar de trabajo. Cada aparato debe ofrecer una alternativa para diseñar la tierra en la que se trabajarán las pruebas. Por ejemplo, qué programa se utilizará para la crónica y la propagación de las pruebas, donde se ubicará la bóveda en la que se guardarán las pruebas y los resultados, y así sucesivamente.

- Creación de pruebas guiadas y scripts programados. Debería permitir la producción de pruebas de forma guiada, sin la necesidad de escribir líneas de código, de esta manera alentando el camino hacia la realización y el control de las pruebas para clientes nuevos. Es fascinante que tenga una grabadora, cuya capacidad es registrar las actividades que el cliente actúa en la aplicación. Además, debe almacenar datos sobre elementos existentes.

Por otra parte, debe tener una alternativa para crear el contenido de prueba del documento que contiene el registro realizado por el programa.

- Script Explorer. Es un tablero que contiene todos los elementos enlistados durante la cuenta de la prueba y donde puede verificar las propiedades de cada uno de ellos.
- Prueba del director editorial. El aparato debe proporcionar un corrector de prueba para que se encargue de ellos una vez que la cuenta esté completa. En el momento en que el contenido es accesible, que contiene las actividades del cliente, al igual que los objetos de la aplicación, la sustancia de esta sustancia debe ser controlada. Por ejemplo, ingrese aprobaciones, puntos de control, etc. Por otra parte, el administrador de la prueba debe ofrecer la alternativa para controlar igualmente el contenido de la prueba. Modifique el contenido trabajando directamente sobre el código fuente.
- Edad automática de los resultados de la prueba y la capacidad. El aparato debe tener un sistema que cree informes de los resultados de la prueba después de la ejecución. Además, debe dar la opción de ahorrar estos informes y guardarlos en la bóveda correspondiente.

2.3. Definiciones conceptuales

- **Aplicación web.** Es un software utilizado por la mayoría de usuarios, quienes mediante un programa de navegación son capaces de ejecutarlo a través de internet. Esto sucede porque el software está codificado en un lenguaje compatible con los programas de navegación. (Niño, 2010).
- **Eficiencia.** Grupo de créditos identificados con la conexión entre el grado de ejecución de la programación y la medida de los activos requeridos bajo las condiciones señaladas (ISO/IEC 9126-1:2001).
- **Fiabilidad.** Muchos adscritos identificados con la facultad del producto para conservar su grado de ejecución en condiciones señaladas durante un tiempo determinado (ISO/IEC 9126-1:2001).
- **Funcionalidad.** Muchos adscritos que se identifican con la presencia de muchas capacidades y sus atributos particulares. Las capacidades son aquellas que satisfacen las necesidades entendidas o inequívocas (ISO/IEC 9126-1:2001).
- **Usabilidad.** Muchos adscritos identificados con las condiciones para su utilización, y en la evaluación individual de dicho uso, por un arreglo establecido o incluido de los clientes (ISO/IEC 9126-1:2001, 2001).
- **Selenium.** Es una suite de herramientas que permite la automatización de pruebas de software en diferentes lenguajes de programación. Además, es un software de código libre (Open Source).
- **Sahi.** El método de actividad depende de la forma en que Sahi introduce un servidor intermediario en la máquina en la que se impulsa la robotización. Al diseñar el programa para utilizar esta asociación intermediaria, Sahi tendrá la

opción de infundir código javascript en él, realizar el procedimiento de comparación en la aplicación web y manejar las ocasiones.

- **QuickTest Professional (QTP).** Presta automatización de ensayos funcionales y de regresión para aplicaciones y ambientes de software. HPE Unified Functional Testing se puede usar para garantizar la calidad de la empresa.

CAPITULO III: DESARROLLO DEL PROYECTO

3.1. Descripción de solución de negocio propuesta

Lo que se propone es realizar la investigación de un Software que sea óptimo, por esa razón en esta investigación se hará la comparación de 3 herramientas de software que sirvan para realizar las pruebas automatizadas. Posteriormente se van a listar y definir sus características que tendremos en consideración al realizar el estudio comparativo; luego se puntualizará cada herramienta con sus características definidas en un cuadro comparativo, esto con el fin de disminuir el tiempo de procesos y la cantidad de defectos encontrados en la etapa de pruebas en la Consultora KMC.

3.1.1. Características para medir en el estudio comparativo

En esta comparación se tienen en cuenta 15 funcionalidades de las herramientas para realizar automatización de pruebas.

1. **Dirección URL:** Es el sitio web donde se encuentran las guías, definiciones e instaladores de herramientas
2. **Tipo de licencia y precio:** periodo de prueba y finalmente el costo.
3. **Acceso al código fuente:** Se accede al código fuente de las herramientas y se realizan las modificaciones en ella.
4. **Plataforma:** Sistemas operativos que son soportados para la instalación de la herramienta. Generalmente son Windows, Unix, linux, etc.
5. **Multiplataforma:** Son las características que nos dice que tan portable es la herramienta.

6. **Requisitos de hardware:** En general, el grado de equipo requerido para que el dispositivo funcione. Alude al tipo de procesador, RAM y disco duro.
7. **Idiomas que soporta:** El lenguaje en el que interactúan con el cliente es una parte de las funcionalidades atractivas de estos dispositivos.
8. **Visualización a tiempo real:** Es el cual permita visualiza los resultados de las pruebas en paralelo con la ejecución.
9. **Pruebas que son programadas:** Es el que permite poder programar tanto el inicio como las pausas y también el final de un procedimiento de pruebas.
10. **Scripting:** Al poder efectuar el diseño de pruebas, la manera de programar o estructurar tiene mucho que ver con la fácil manipulación que se le puede dar a la misma. Algunas herramientas cuentan con lenguajes más conocidos que permiten realizar el diseño de las pruebas de forma más amplia y fácil, con la ventaja que no se requiere aprender un nuevo lenguaje programático para generarlas.
11. **Diseño de informes personalizados:** Es todo el informe específico para el estudio de resultados de las pruebas.
12. **Pruebas funcionales:** Que permita primero realizar las pruebas que se basan en requisitos, casi siempre programables mediante el script.
13. **Bibliotecas por navegador:** Necesita bibliotecas separadas para cada navegador
14. **Conocimientos en programación:** Hace referencia a la necesidad de programar en un nivel avanzado.
15. **Soporte disponible:** El servicio del proveedor para dar un buen servicio de soporte.

3.1.2. Análisis de Resultados

Tabla 3 Comparativo de Especificaciones Técnicas y Funcionales

#	Nombre	Selenium IDE	Sahi Pro	Test Complete
1	Dirección URL del fabricante	https://www.seleniumhq.org/	https://sahipro.com/	https://smartbear.com/product/testcomplete
2	Tipo de licencia y precio	Open Source	Prueba de 30 días. Luego el precio es desde \$ 695.00	Prueba de 30 días. Luego el precio es desde \$ 2,398
3	Acceso al código fuente	SI	NO	NO
4	Plataforma	Windows/Linux/IOS	Windows/Linux/IOS	Windows/Linux/IOS
5	Multiplataforma	SI	SI	SI
6	Requisitos de hardware	No especificado	No especificado	No especificado
7	Idiomas soportados	Inglés	Inglés	Inglés
8	Visualización en tiempo real	NO	SI	SI
9	Pruebas programadas	SI	SI	SI
10	Scripting	SI	NO	NO
11	Diseño de informes personalizados	NO	SI	NO
12	Pruebas funcionales	SI	SI	SI
13	Bibliotecas por navegador	SI	NO	SI
14	Conocimientos de programación avanzados	SI	NO	SI
15	Soporte disponible	NO	SI	SI

Fuente: Elaboración Propia.

Tabla 4 Comparativo de Especificaciones Conceptuales

CUADRO COMPARATIVO DE LAS HERRAMIENTAS INVESTIGADAS

		HERRAMIENTAS		
CRITERIOS	Selenium IDE	Sahi Pro	Test Complete	
¿Qué es?	Selenium es una suite de herramientas que permite la automatización de pruebas de software en diferentes lenguajes de programación. Además, es un software de código libre (Open Source).	Sahi Pro es una gran cantidad de aparatos de desarrollo, preparados para negocios, para pruebas computarizadas de aplicaciones web, administraciones web, versátiles, área de trabajo de Windows y Java. Para hardware de prueba que necesita una informatización rápida y confiable. Sahi disminuye el tiempo con un avance más rápido, propagación diseminada rápida.	TestComplete es una herramienta de prueba de IU automatizada que hace que sea rápido y fácil de crear, mantener y ejecutar pruebas funcionales en aplicaciones de escritorio, web y móviles.	
Ventaja	<ul style="list-style-type: none"> - Desarrollo regido por pruebas, o desarrollo impulsado por pruebas. - Definir conjunto de pruebas que cubran la mayoría de las funciones del sistema, que puedan ser repetidas la cantidad de veces que sea, siguiendo los mismos pasos, evitando trabajo manual o susceptible a errores. - Al realizar un cambio al sistema, se pueden ejecutar las pruebas para comprobar los impactos contemplados y no contemplados. 	<ul style="list-style-type: none"> - Permite la grabación y ejecución de múltiples programas. - Ejecución por lotes (formas de grupo). - Ejecución paralela (varias ocasiones de programas sin demora de un momento). - Los resultados concebibles ofrecidos para el área de componentes dentro del DOM del programa. - Realizar las mismas pruebas en todos los navegadores compatibles. - Realizar la misma prueba con distintos clientes al mismo tiempo. - Recopilar evidencias físicas (logs, resultados, capturas de pantalla). - Evitar cosas como “eso ya lo había probado”. 	Ofrece la capacidad de robotizar pruebas prácticas y pruebas de recaída para situaciones de programación y prueba. Incorpora aspectos destacados, por ejemplo, Vista experta, pruebas de procesos de negocios, grabado de pantalla (para capturar pruebas de prueba), entre diferentes perspectivas.	
Desventaja	<ul style="list-style-type: none"> - Para mayor cantidad de funcionalidades las pruebas llegarán a ser demasiado complejas para realizarse fácilmente en Selenium. 	<ul style="list-style-type: none"> - Las pruebas llegarán a ser demasiado complejas para realizarse fácilmente. 	<ul style="list-style-type: none"> - Las pruebas llegarán a ser demasiado complejas para realizarse fácilmente. 	

Fuente: *Elaboración propia*

3.2. Arquitectura empresarial

3.2.1. Arquitectura de Información (Datos)



Figura 11 Modelo de Gestión

Fuente: Elaboración propia

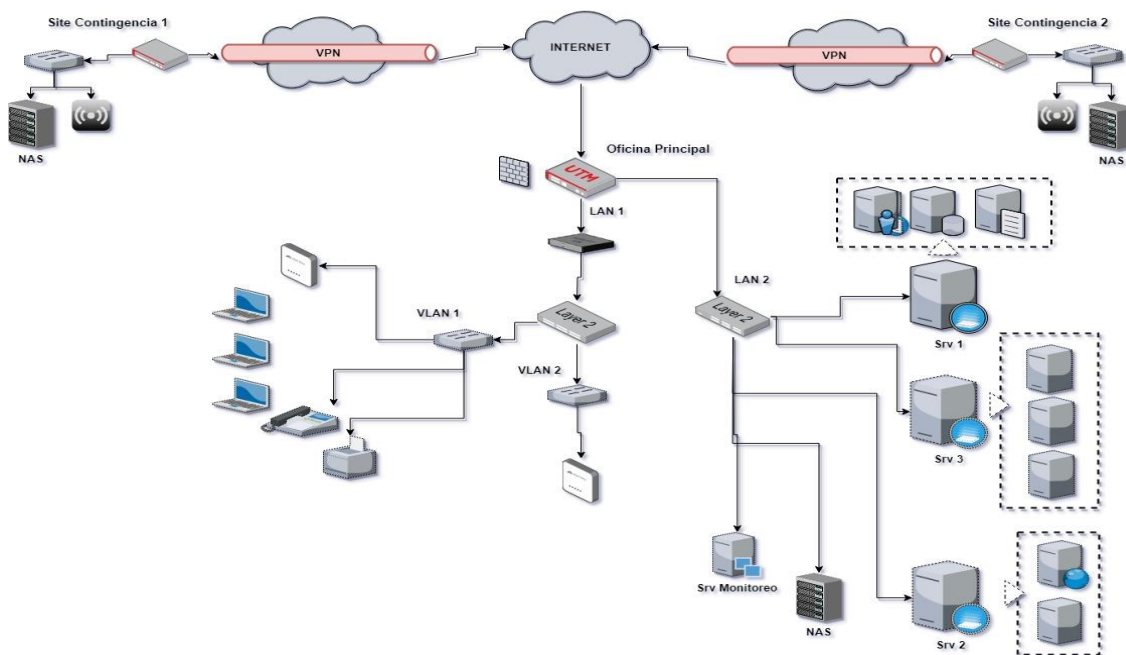


Figura 12 Datos Físicos

Fuente: Elaboración propia

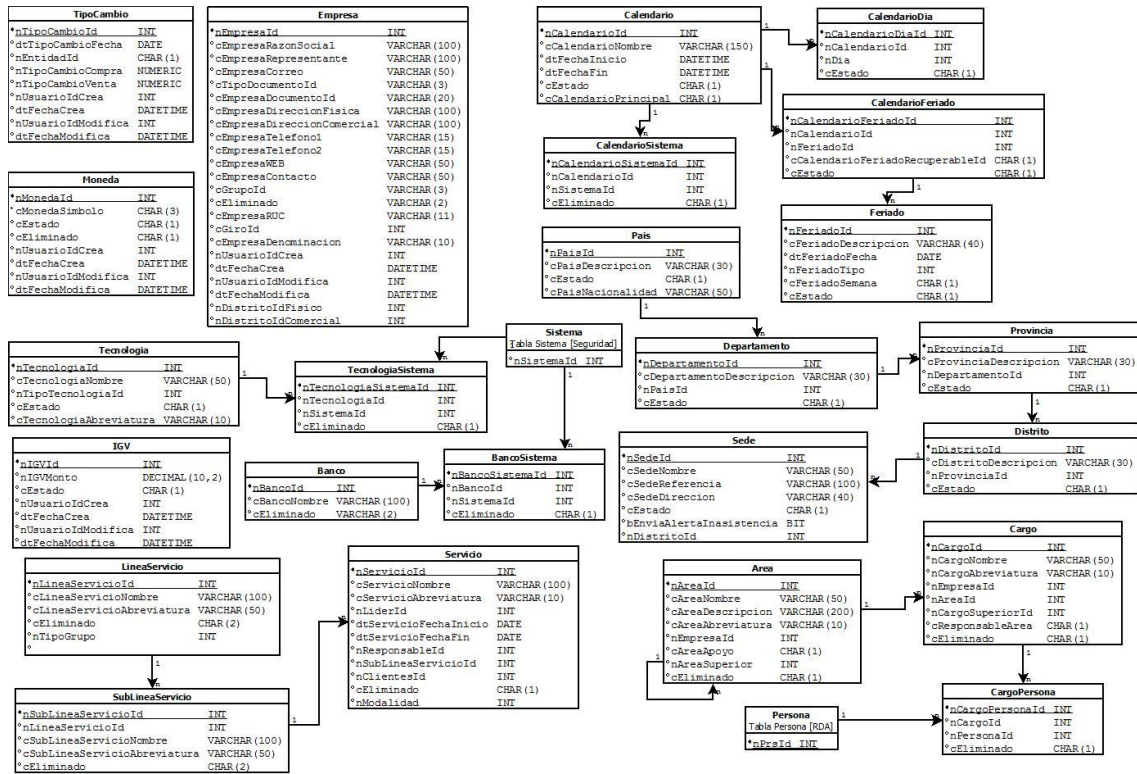


Figura 13 Datos Lógicos

Fuente: Elaboración propia

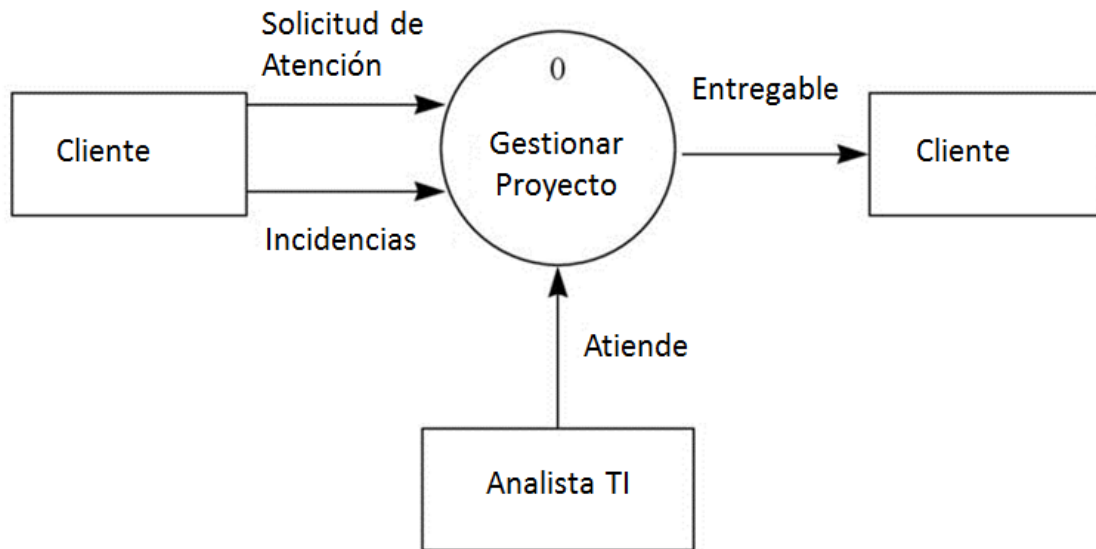


Figura 14 Flujo de Datos

Fuente: Elaboración propia

3.2.2. Arquitectura de Negocios (Procesos)

3.2.2.1 Estrategia del negocio

Consultora KMC tiene claro cuáles son los beneficios de sus servicios y a cuáles necesidades o deseos estará satisfaciendo con cada uno de ellos a los clientes.

Convertimos a sus colaboradores y usuarios en protagonistas activos de del negocio, hacer y compartir sustancia, investigar, interesarse y relacionarse en todos los procedimientos interiores de la organización.

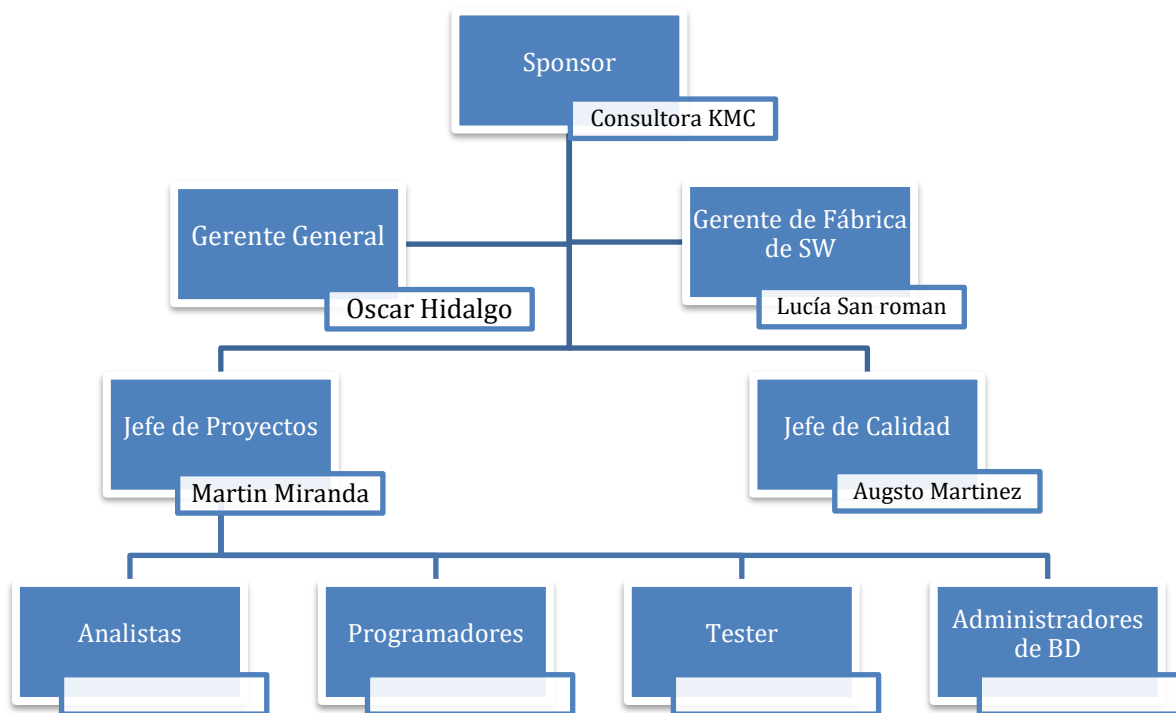
Visualizamos la trazabilidad de sus procesos de punta a punta bajo un estricto nivel de seguridad y protección de sus datos empresariales.

Utilizamos un procedimiento de modelo de ayuda que depende de nuestros largos períodos de experiencia, afirmaciones globales y familiaridad para lidiar con la zona de marcos de nuestros clientes con el desarrollo.

Contamos con expertos excepcionalmente calificados y continuamente preparados en procedimientos de mejora de procesos y etapas innovadoras que les permite dar un nivel significativo de valor para el desarrollo de la programación.

El modelo de servicio de consultores de KMC respalda las aplicaciones de administración de PC que nos permiten mecanizar una parte de los ejercicios que, sumados a la experiencia de nuestros socios y su mejora continua, nos permiten trasladar incentivos de calidad a nuestros clientes.

3.2.2.2 Estructura organizacional



3.2.2.3 Los procesos clave de la organización

Figura 15 Estructura Organizacional de la Consultora KMC

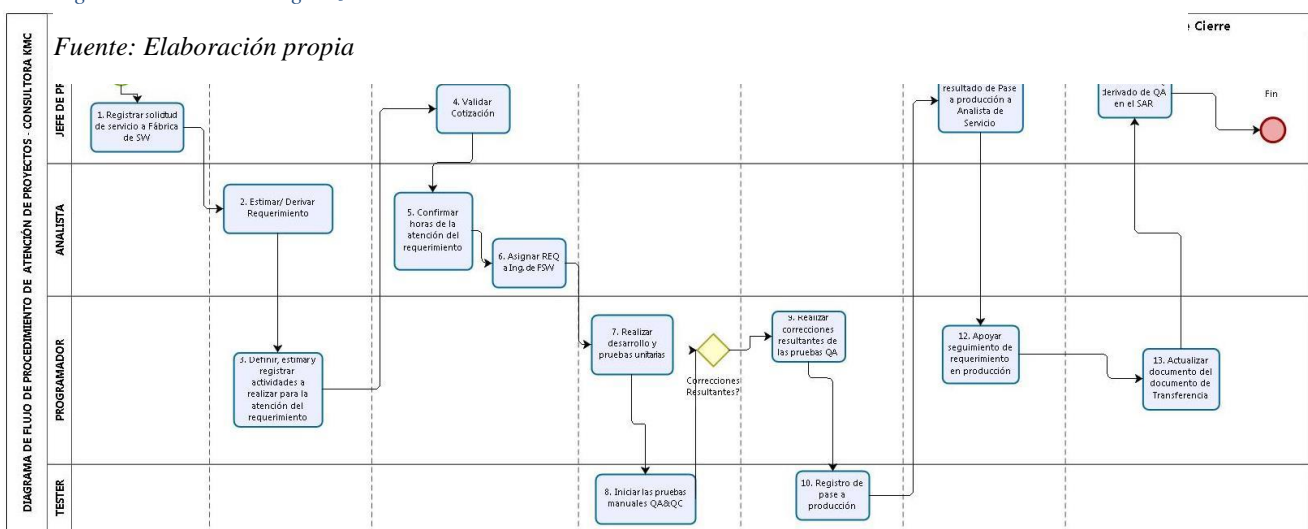


Figura 16 Diagrama de Flujo de Procedimiento de Atención de Proyectos Consultora KMC

Fuente: Elaboración propia

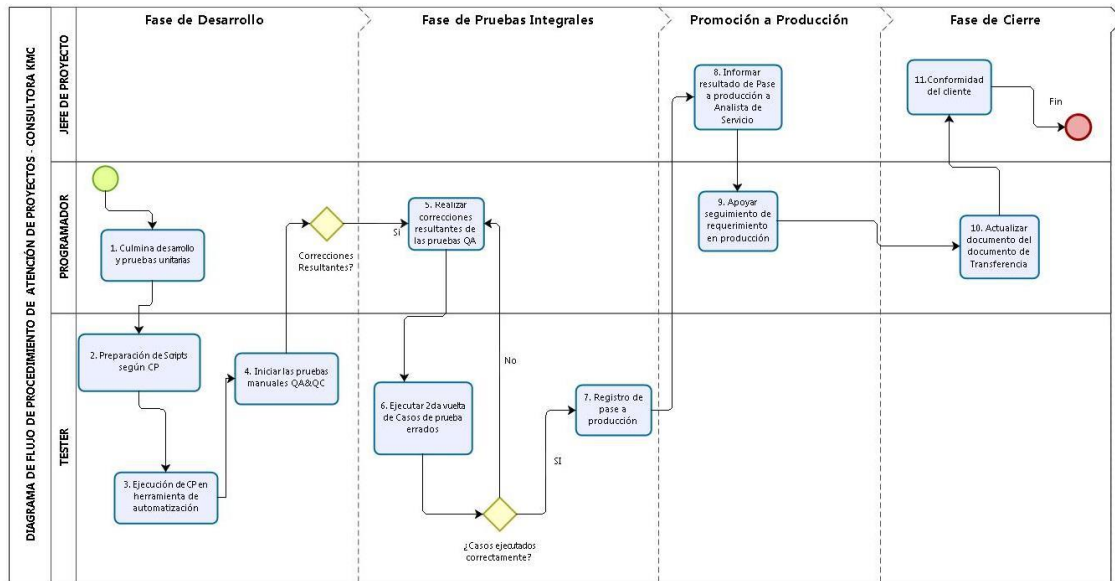


Figura 17 Diagrama de Flujo de Procedimiento de Atención de Proyectos después de Implementación de Herramienta en la Consultora KMC

Fuente: Elaboración propia

3.2.3. Arquitectura de Aplicación

El javascript inyectado agrega manejadores de eventos para la grabación agrega las API de Sahi para la simulación de eventos se burla de algunos cuadros de diálogo de ventana como alerta, aviso, confirmación, impresión para que la reproducción no se vea afectada

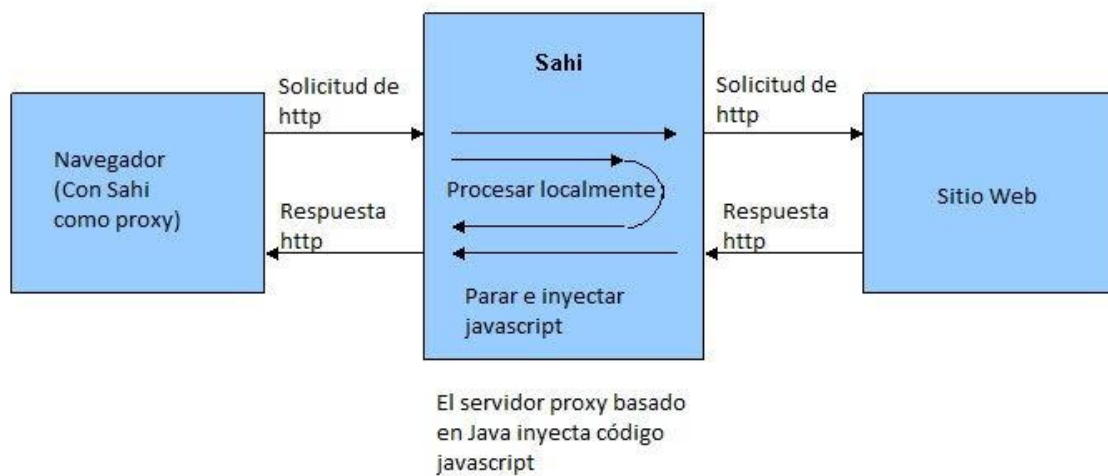


Figura 18 Arquitectura de Aplicación de Sagi Pro.

Fuente: Elaboración Propia.

3.2.4. Arquitectura Tecnológica

Se muestra la arquitectura tecnológica de la organización en donde es posible visualizar en la (figura 14) las capas de la arquitectura y la cantidad de capas que posee las aplicaciones web de la organización.

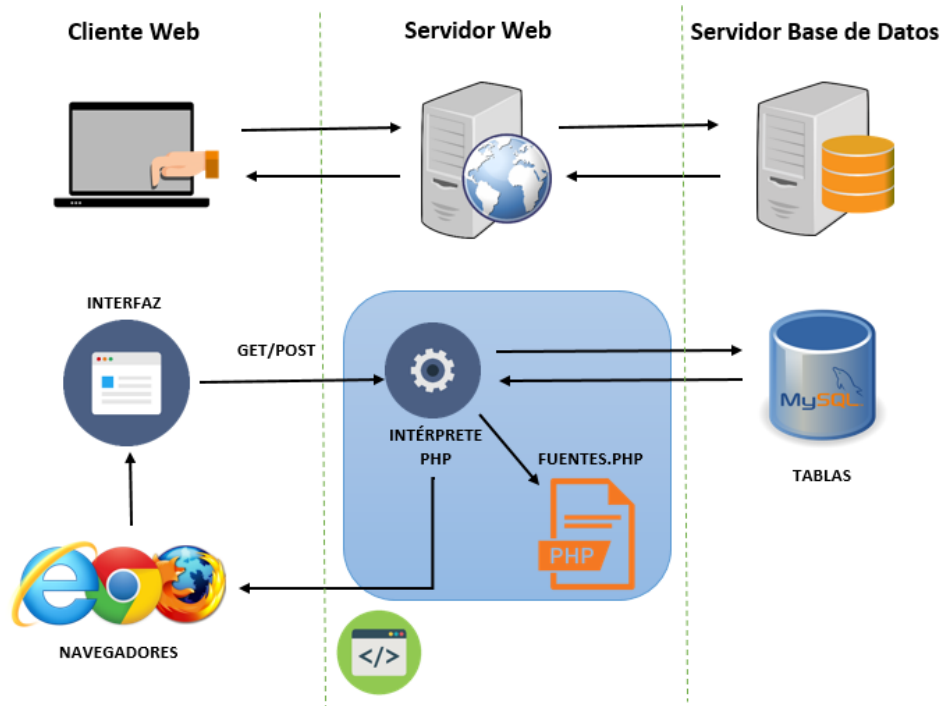


Figura 19 Arquitectura Tecnológica de la Consultora KMC

Fuente: Elaboración propia

A continuación, describiremos las tres capas sensibles que aparecen en la figura:

- **Cliente web:** esta capa se enmarca en la lógica de negocio inmersa dentro de la aplicación que efectúa el interfaz de usuario. La lógica de la capa del cliente incorpora programas, por ejemplo, Internet Explorer, Google Chrome, Mozilla Firefox, etc.

- **Servidor web:** Esta capa está hecha de la justificación que representa los elementos primordiales de la aplicación, ya sea: preparación de información, ejecución de capacidades comerciales y activos del consejo, por ejemplo, la base de datos. En esta capa se procesan las peticiones enviadas desde la capa de cliente. La lógica en esta capa está formada por el componente Hypertext Preprocessor (PHP).
- **Servidor Base de Datos:** La capa de datos está hecha de las administraciones proporcionadas por el supervisor de base de datos. La información se guarda en una base de datos del marco de ejecutivos que ahora mismo MySQL.

3.2.5. Factibilidad Económica

De acuerdo al objetivo de esta investigación, encontramos resultados satisfactorios en los distintos proyectos de implementación utilizando la herramienta de automatización, se reflejan la disminución en el tiempo de prueba y los scripts generados para los distintos proyectos sirvieron para disminuir la cantidad de observaciones que generaba el equipo de desarrollo.

Tabla 5 Factibilidad Económica

Proyecto	Fecha inicio	Fecha fin	Fecha fin real	Dif. en días	CP propuestos	Tiempo de ejec. Manual (min)	Tiempo de ejec. Autom. (min)	Hrs. de ejec. Manual	Hrs. de ejec. automatizada	Precio por Hora	Hrs. de ejec. manual ²	Hrs. de ejec. automatizada ³	Dif. de Horas	%		
Préstamos Hipotecarios	15/04/2019	18/05/2019	12/05/2019	5	65	12	4	33	11	S/. 45	S/.	1.462,50	S/.	487,50	22	200%
Emboce de tarjetas	02/06/2019	10/02/2019	08/02/2019	2	28	15	6	18	7	S/. 45	S/.	787,50	S/.	315,00	11	150%
Pagos de Servicios	03/06/2019	15/06/2019	12/06/2019	3	25	13	5	14	5	S/. 45	S/.	609,38	S/.	234,38	8	160%
Nueva opción registro – Recaudaciones	02/05/2019	13/05/2019	07/05/2019	4	15	15	5	9	3	S/. 45	S/.	421,88	S/.	140,63	6	200%
Ajuste de Céntimos	18/05/2019	28/05/2019	27/05/2019	1	18	10	4	8	3	S/. 45	S/.	337,50	S/.	135,00	5	150%
Sistema de Compras	14/06/2019	24/06/2019	22/06/2019	2	24	11	6	11	6	S/. 45	S/.	495,00	S/.	270,00	5	83%
Sistema de Atención de Requerimientos	08/06/2019	17/06/2019	14/06/2019	3	17	20	7	14	5	S/. 45	S/.	637,50	S/.	223,13	9	186%

Fuente: Elaboración Propia.

CAPITULO IV: RECURSOS Y CRONOGRAMA

4.1. Recursos

El presupuesto del desarrollo del proyecto se basó principalmente en el personal requerido para la investigación, implementación, capacitación.

Tabla 6 Presupuesto de Hardware y Soporte

Servicios TI	Cantidad	Costo x unidad	Costo Total
Servidor de aplicación	1	S/. 6.000	S/. 6.000
Soporte TI	1	S/. 2.500	S/. 2.500
TOTAL			S/. 8.500

Fuente: Elaboración Propia.

Tabla 7 Presupuesto de Recursos Humanos

Costo de Investigación	Cant.	Horas	Precio Hora	Costo Variable
Líder QA&QC	1	189	S/ 65.00	S/ 12,285.00
Analista QA	1	210	S/ 40.00	S/ 8,400.00
Analista Funcional	1	210	S/ 40.00	S/ 8,400.00
TOTAL				S/. 29,085.00

Fuente: Elaboración Propia.

Tabla 8 Presupuesto de Licenciamiento y Despliegue

Concepto	Tipo	Horas	Precio	Precio Final Soles
Precio de Licencia	Anual	-	\$ 695.00	S/ 2,314.35
Implementación y capacitación	-	18	S/ 40.00	S/ 720.00
TOTAL				S/ 3034.35

Fuente: Elaboración Propia.

Considerando el gasto global del proyecto con una duración de 1 mes y 15 días Apróx.

El gasto total ascendería a **S/ 40,619.35**

4.2. Cronograma de ejecución

	EDT	Nombre de tarea	% completac	Comienzo	Fin
1	1	➤ HERRAMIENTA DE AUTOMATIZACIÓN DE PRUEBAS COMO PROPUESTA PARA MEJORAR LA REVISIÓN DEL CONTROL DE CALIDAD DE SOFTWARE LA EMPRESA "CONSULTORA KMC"	0%	mié 5/06/19	vie 12/07/19
2	1.1	➤ INICIACIÓN	0%	mié 5/06/19	mar 2/07/19
3	1.1.1	➤ Estudio de Funcionalidad	0%	mié 5/06/19	lun 10/06/19
4	1.1.1.1	Análisis del problema	0%	mié 5/06/19	mié 5/06/19
5	1.1.1.2	Estudio de Funcionalidad	0%	mié 5/06/19	jue 6/06/19
6	1.1.1.3	Resultados de Funcionalidad	0%	jue 6/06/19	vie 7/06/19
7	1.1.1.4	Estudio de Funcionalidad Aprobado	0%	vie 7/06/19	lun 10/06/19
8	1.1.2	➤ Estudio de Usabilidad	0%	lun 10/06/19	vie 14/06/19
9	1.1.2.1	Análisis del problema	0%	lun 10/06/19	mar 11/06/19
10	1.1.2.2	Estudio de Usabilidad	0%	mar 11/06/19	mié 12/06/19
11	1.1.2.3	Resultados de Usabilidad	0%	mié 12/06/19	jue 13/06/19
12	1.1.2.4	Estudio de Usabilidad Aprobado	0%	jue 13/06/19	vie 14/06/19
13	1.1.3	➤ Estudio de Mantenibilidad	0%	vie 14/06/19	jue 20/06/19
14	1.1.3.1	Análisis del problema	0%	vie 14/06/19	lun 17/06/19
15	1.1.3.2	Estudio de Mantenibilidad	0%	lun 17/06/19	mar 18/06/19
16	1.1.3.3	Resultados de Mantenibilidad	0%	mar 18/06/19	mié 19/06/19
17	1.1.3.4	Estudio de Mantenibilidad Aprobado	0%	mié 19/06/19	jue 20/06/19
18	1.1.4	➤ Estudio de Eficiencia	0%	jue 20/06/19	mié 26/06/19
19	1.1.4.1	Análisis del problema	0%	jue 20/06/19	vie 21/06/19
20	1.1.4.2	Estudio de Eficiencia	0%	vie 21/06/19	lun 24/06/19
21	1.1.4.3	Resultados de Eficiencia	0%	lun 24/06/19	mar 25/06/19
22	1.1.4.4	Estudio de Eficiencia Aprobado	0%	mar 25/06/19	mié 26/06/19
23	1.1.5	➤ Estudio de Fiabilidad	0%	mié 26/06/19	mar 2/07/19
24	1.1.5.1	Análisis del problema	0%	mié 26/06/19	jue 27/06/19
25	1.1.5.2	Estudio de Fiabilidad	0%	jue 27/06/19	vie 28/06/19
26	1.1.5.3	Resultados de Fiabilidad	0%	vie 28/06/19	lun 1/07/19
27	1.1.5.4	Estudio de Fiabilidad Aprobado	0%	lun 1/07/19	mar 2/07/19
28	1.2	➤ PLANIFICACIÓN	0%	mar 2/07/19	vie 12/07/19
29	1.2.1	➤ Identificación de herramientas de automatización	0%	mar 2/07/19	vie 12/07/19
30	1.2.1.1	Identificar características	0%	mar 2/07/19	mié 3/07/19
31	1.2.1.2	Identificar Riesgos	0%	mié 3/07/19	jue 4/07/19
32	1.2.1.3	Identificar probabilidad de los Riesgos	0%	jue 4/07/19	vie 5/07/19
33	1.2.1.4	Comparar con Criterios de aceptación de la consultora	0%	vie 5/07/19	lun 8/07/19
34	1.2.1.5	Identificar compatibilidad	0%	lun 8/07/19	mar 9/07/19
35	1.2.1.6	Plasmar estudio (Funcionalidad, usabilidad, Mantenibilidad, Eficiencia, Fiabilidad)	0%	mar 9/07/19	mié 10/07/19
36	1.2.1.7	Realizar cuadro comparativo con propuestas	0%	mié 10/07/19	jue 11/07/19
37	1.2.1.8	Realizar documentación de Investigación	0%	jue 11/07/19	vie 12/07/19
38	1.3	➤ SEGUIMIENTO Y CIERRE	0%	vie 12/07/19	vie 12/07/19
39	1.3.1	➤ Reunión con Cliente - Precisando Resultado de Investigación	0%	vie 12/07/19	vie 12/07/19
40	1.3.1.1	Reunión Con Usuario - Precisando Definición Funcional 1	0%	vie 12/07/19	vie 12/07/19

Figura 20 Cronograma Gantt

Fuente: Elaboración propia.

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

En esta ejecución los casos de pruebas de software en las fábricas de software, a pesar de que se cuenta con un nivel de documentación avanzado, con certificaciones como CMMI, en el día a día son poco formales y poco acopladas a los estándares de buenas prácticas de pruebas de software.

En la actualidad las herramientas de automatización de pruebas están orientadas a detectar y prevenir ciertos tipos de errores en el software, que permita la realización de pruebas, y la posibilidad del monitoreo de sistemas, la automatización de pruebas. También la portabilidad de estas herramientas marca diferencia e impone entre las características esperadas por las organizaciones.

La prueba ejecutada posterior a la etapa de desarrollo implica altas posibilidades de caer en el retraso del proyecto, costos elevados, la baja calidad del software.

La etapa de pruebas automatizadas cumple una labor importante en el ciclo de desarrollo de SW; La mejor forma de cumplir con un entregable probado y desarrollado para el cliente es teniendo un buen proceso de pruebas automatizadas que aporte a esta necesidad con gran demanda en el mercado, acompañada de una infraestructura donde apoyarse en el proceso de pruebas de manera ágil y cumpliendo plazos acordados con el cliente.

De los objetivos establecidos hacia el inicio de este trabajo de exploración, y los resultados relacionados obtuvieron de la evaluación de Software (**Anexo 2**) para automatización de pruebas en la empresa Consultora KMC, exponemos nuestras conclusiones:

- La funcionalidad de las pruebas a realizar con este Software influyó correctamente en el todo proceso de Ejecución de cada Casos de Prueba para el área de Control de Calidad con el Software Sahi Pro; mejorando la exactitud, seguridad y conformidad de la funcionalidad de pruebas en la Consultora KMC.
- La confiabilidad de las pruebas a realizar con este Software influyó satisfactoriamente en todo proceso de ejecución de los grupos de prueba en el área de Control de Calidad con el Software Sahi Pro; mejorando madurez, resistencia de errores, recuperabilidad y consistencia de la calidad inquebrantable en el Consultor KMC.
- La usabilidad de las pruebas a realizar con este Software ejerció correctamente en el proceso de ejecución de Casos de Prueba en el área de Control de Calidad, ya que se detectó que la usabilidad del Software Sahi Pro, ya que se orienta en la navegación comprensiva y sencilla, además mejoró la impresión de los interesados finales y los interesados indirectos del software en la Consultora KMC.
- La mantenibilidad de las pruebas a realizar con este Software ejerció correctamente en el proceso de ejecución de Casos de prueba para el área de Control de Calidad con el Software Sahi Pro; mejorando la manera que tiene el software para ser corregido, incluyéndose las mejoras que se realicen en el software en la Consultora KMC.
- La eficiencia de las pruebas a realizar con este Software ejerció correctamente en el proceso de Ejecución de grupos de Prueba para el área de Control de Calidad, ya que esta permite la generación de reportes y mejoró a la disminución de tiempos de contestación en la ejecución de pruebas con el Software Sahi Pro a un 60%; mejorando el desempeño de acuerdo al número de recursos que utilizan el software en la Consultora KMC.

5.2. Recomendaciones

- Se prescribe para actualizar un instrumento de robotización que garantiza la crítica de los datos adquiridos durante el proceso de prueba del producto, se puede garantizar el uso correcto de las pruebas del producto y el plan de prueba en el que se analizan. Existen numerosos aparatos para el uso de pruebas, con sus puntos focales e inconvenientes individuales, sin embargo, lo importante cuando se utiliza cualquier dispositivo, es utilizar el que mejor se adapte al grupo de trabajo y el que permita lograr los destinos ideales. A pesar del hecho de que esto implica hacer cambios a uno previamente establecido, sin embargo, seguirlo exactamente de manera continua, de manera similar, es importante aclarar el grupo de trabajo desde el principio, ya que será utilizado. Del mismo modo, el objetivo fundamental de lograr una calidad ideal que satisfaga al cliente final nunca debe pasarse por alto, ya que este será el que colaborará con el producto y en el que dependerá del producto que se utiliza o se deja atrás.
- Fomentar el interés de las zonas involucradas en el proceso de prueba del producto, al igual que los colegas de cada territorio para que perciban su importancia simultáneamente y puedan ser participativos cuando se ejecuta la aplicación.
- Fomentar el apoyo de las regiones asociadas con el proceso de prueba del producto, al igual que los colegas de cada territorio para que perciban su importancia todo el tiempo y puedan ser participativos cuando se actualiza el dispositivo.

CAPITULO VI: FUENTES DE INFORMACION

Bibliografía

- Ciolti, M. (2007). Testing de migración de aplicaciones distribuidas a entornos Web. La Plata, Argentina: Universidad Nacional de la Plata.
- Esteve Ambrosio, D. (2015). Implantacion de un proceso de automatizacion de pruebas para una aplicacion software. Valencia, España: Universidad politecnica de Valencia.
- Lerma-Blasco, R., Murcia Andres, J., & Mifsud Talon, E. (2013). Aplicaciones Web. Basauri, España: McGraw-Hill/Interamericana de España, S.L.
- Mendoza, M. (2010). Modelo de referencia para la selección de herramientas de pruebas como soporte al proceso de desarrollo de software en pymes venezolanas. Caracas, Venezuela, Venezuela: Universidad Católica Andrés Bello.
- Rogers, P. (2005). Ingeniería de Software un Enfoque Práctico. Madrid: Editorial McGraw.
- ISO/IEC 9126-1:2001. (2001). ISO/IEC 9126-1:2001. Software engineering - Product quality - Part 1: Quality model. Suiza: International Organization for Standardization.
- Largo, C. & Marin, E. (2005). Guía Técnica para Evaluación de Software. Uruguay: PuntoExe.
- Niño, J. (2010). Aplicaciones web: Ciclos Formativos. España: Editorial Editex.
- “Iso 9126-1 software quality characteristics and metrics part 1 – quality characteristics and sub-characteristics. (1997).”
- S. Rakitin, Software Verification and Validation for Practitioners and Managers. Artech House, 2001.
- I. Sommerville, Ingeniería de Software. Pearson Educación, 2005.
- Borbón Ardilla, N. (2013). *Evaluacion de Software*. Obtenido de <http://actividadreconocimiento-301569-8.blogspot.com/>

ANEXOS

6.1. Anexo 1

Tabla 9 Modelo de la Evaluación de la Calidad de Software

Tipo de Calidad	Características	Sub-Características	Puntaje	Software 1 Puntaje	Software 2 Puntaje
Calidad Interna y Externa	Funcionalidad	Adecuación			
		Exactitud			
		Interoperabilidad			
		Seguridad			
	Usabilidad	Documentación			
		Operabilidad			
		Interfaz Grafica			
	Fiabilidad	Recuperabilidad			
		Tolerancia a Fallas			
	Eficiencia	Comportamiento en el tiempo			
		Utilización de Recursos			
	Manteniabilidad	Acoplamiento			
		Modularidad			
		Facilidad de Pruebas			
	Portabilidad	Adaptabilidad			
		Facilidad de Instalación			
Coexistencia					
Facilidad de Reemplazamiento					
Calidad en uso	Productividad				
	Seguridad				
	Satisfacción				
TOTAL					

Fuente: Elaboración propia

6.2. Anexo 2

Tabla 10 Herramienta Sahi Pro.

FUNCIONALIDAD	USABILIDAD	MANTENIBILIDAD	EFICIENCIA	FIABILIDAD
El software cumple y provee todas las aplicaciones para desempeñar las necesidades explícitas e implícitas cuando es usado.	El software se probó en instalaciones del cliente con algunos aplicativos y se detectó que puede usarse para aplicativos que se ejecuten en cualquier explorador.	El software tiene la cualidad de ser configurado para adaptarse al flujo de pruebas que se realicen, por lo cual cumple con la mantenibilidad.	El Software es eficiente ya que es posible que sea utilizado por el número de analistas de testing que haya en la empresa y por cualquier sistema operativo y navegador de internet.	Un 90% ha determinado estar muy de acuerdo en que la aplicación cumplió en su gestión de controlar de manera efectiva la información de las pruebas realizadas; mientras que un 8.1% determinan estar de acuerdo, por último, un 1.9% concluye que no están ni de acuerdo ni en desacuerdo..

Fuente: Elaboración Propia tras análisis de métodos, técnicas y herramientas.