

# Los beneficios de la automatización para aumentar la calidad de pruebas de software

David Morales Vindas  
Escuela de Ingeniería,  
ULACIT  
San José, Costa Rica  
dmoralesv153@ulacit.ed.cr

Juan Diego Chaves Rodríguez  
Escuela de Ingeniería,  
ULACIT  
San José, Costa Rica  
jchavesr883@ulacit.ed.cr

Ricardo Brizuela Hernández  
Escuela de Ingeniería,  
ULACIT  
San José, Costa Rica  
rbrizuelah015@ulacit.ed.cr

Julio Córdoba Retana,  
Escuela de Ingeniería,  
ULACIT,  
San José, Costa Rica  
jcordobar022@ulacit.ed.cr

**Resumen** - Se obtendrán resultados sobre el beneficio de la automatización en el sector del análisis de control de calidad, utilizando la comparación y análisis de un ambiente, con y sin este, que demostrarán el beneficio que esta presenta a un sector que ha permanecido sin grandes cambios a lo largo de la última década. Se investigarán factores que afecten su implementación y deban de ser tomadas en cuenta para una buena ejecución, se detallarán beneficios que se obtendrán luego de la implementación, además de presentar las consecuencias de ignorar esta mejora en el área de control de calidad.

**Palabras clave:** Control de calidad, automatización, mejora, prueba de software, pruebas unitarias, pruebas de integración.

**Abstract** – Results will be obtained about the benefits of automation in the quality control analysis sector using the comparison and analysis of an environment with and without it, that will demonstrate the benefit that this presents to a sector that has remained without major changes over the last decade. Factors that affect its implementation and must be considered for a good execution will be investigated, benefits that will be obtained after the implementation will be detailed and the consequences of ignoring this improvement in the quality control area will be presented.

**Keywords:** Quality control, automation, improvement, software testing, unit testing, integration testing.

## I. INTRODUCCIÓN

En la última década se ha utilizado a analistas de control de calidad para garantizar la calidad y seguridad del software desarrollado, pero estos incorporan defectos humanos que pueden causar o pasar desapercibidos; errores graves que afectarían significativamente el producto.

Las pruebas de software son tan antiguas como el desarrollo de software en sí, porque la forma más fácil de averiguar si el software funciona es probarlo [1]. Por lo que se puede apreciar la gran importancia que presenta el control de la calidad dentro del desarrollo de software y el potencial que este tiene para mejorar dentro de la industria.

En las últimas décadas, se han realizado grandes investigaciones de cómo automatizar estas pruebas. Por lo cual, ha habido un crecimiento de prácticas de cómo usar las herramientas de pruebas de software. Las pruebas de software

se han vuelto una parte esencial de la disciplina de la ingeniería de software.[13].

Se analizará el tema con los conocimientos obtenidos en análisis de control de calidad para detectar graves problemas que afectan a este sector del desarrollo de software. Dentro de los problemas encontrados están el gasto de recursos como tiempo, dinero y personal capacitado para la tarea, además de problemas inherentes al uso de recursos humanos que aumentan la probabilidad de errores en el producto.

Se estima que alrededor del 50% del costo de un proyecto de desarrollo de software va dirigido hacia el área de análisis de control de calidad, donde las empresas pueden llegar a tener pérdidas que disminuyen las ganancias hasta en un 10% [2]. El uso tan elevado de recursos dentro del área de calidad ha provocado que se busquen alternativas a la situación actual del mercado para poder optimizar este proceso.

En este trabajo se plantea el uso de la automatización para poder reducir los gastos y costos de los proyectos. La automatización es una herramienta que ha solventado problemas similares en distintos sectores del mercado obteniendo resultados eficaces y efectivos. En el área de control de calidad, esta automatización permite la reducción de costos en la realización de pruebas de software.

La calidad de las pruebas de software, así como el control de los gastos, es de gran importancia. Asegurar la calidad es una de las exigencias del mercado. Esta calidad no solo toma la funcionalidad completa del software, si no, también la reputación de la empresa o producto y el cumplimiento de los requerimientos de los clientes finales [12].

Para asegurar la calidad, como se mencionó anteriormente, se tiene que cumplir con los requerimientos del cliente y que el software sea funcional. Por esta razón, es importante reducir o eliminar la posibilidad de errores en este proceso.

La automatización disminuye la probabilidad de errores en el software en la etapa de pruebas o testing. Al ser las pruebas realizadas por una máquina, se elimina la interacción del probador o de la persona que hace pruebas de la ecuación.

Al considerar la automatización de las pruebas de software, se espera que se reduzcan los recursos utilizados. Además,

debería aumentar la calidad del software desarrollado. En definitiva, tiene como objetivo minimizar los errores en los que se produce el proceso. Las suposiciones realizadas deben ser eficaces para analizar el control de calidad de la unidad, el sistema y la integración.

## II. METODOLOGÍA

En el siguiente apartado se detalla el tipo de investigación que se realizará para poder llegar a los resultados deseados. Se definirá una investigación como “realizar actividades intelectuales y experimentales de modo sistemático con el propósito de aumentar los conocimientos sobre una determinada materia” [6]. Se utilizará el tipo de investigación que logre proveer resultados precisos y acertados tomando en cuenta el problema, el contexto y el tiempo utilizado para realizar la investigación.

Se utilizará la investigación cualitativa para poder llegar a los resultados deseados. Las investigaciones cualitativas buscan entender un problema de investigación con un enfoque humanista o idealista [10]. Este tipo de enfoque nos permitirá conseguir los resultados deseados dentro del tiempo límite, además, nos brindará una perspectiva más humana sobre los problemas que presenta actualmente el proceso de análisis de calidad.

Dentro del enfoque cualitativo se utilizará un alcance explicativo, este alcance nos permitirá observar las causas del fenómeno y como esté relaciona con otras variables o factores [11]. Se obtendrá conocimiento sobre cómo el poco o nulo cambio del sector de análisis de calidad ha afectado al área de desarrollo de software de forma negativa además de como este puede evolucionar por la automatización.

Esta investigación usará libros y artículos desde el 2018 hasta el 2021, esto ofrece información reciente con la que analizar el problema. Se utilizarán fuentes literarias tanto en español como en inglés, esto permitirá ampliar las fuentes de información para realizar la investigación.

## III. MARCO TEÓRICO

La automatización es la solución para solventar distintos problemas a través de la industria del software. Por medio de la enumeración de sus ventajas es posible saber cómo esta herramienta afecta al sector.

Con la automatización se consigue realizar una mayor cantidad de pruebas para el software desarrollado lo que nos brinda una mayor cobertura para asegurar su calidad. El costo inicial de las pruebas automatizadas es mayor pero el costo acumulado de estas pruebas será significativamente menor al de las pruebas manuales [3].

Las pruebas manuales han demostrado ser ineficaces en un mercado que busca producir software con mayor velocidad [4].

La reutilización afecta el tiempo que el área se dedica a realizar los análisis de calidad. Esta disminución en el tiempo proporciona una mayor agilidad en el proceso de desarrollo permitiendo realizar los análisis en paralelo en lugar de al final del proceso. La comunicación con otras áreas también se ve mejorada debido al rápido análisis de las nuevas funciones desarrolladas.

El costo del proceso de análisis de calidad se ve reducido debido a que se puede realizar las pruebas automatizadas el doble de rápido que las manuales [5]. La implementación de la automatización va a tener un costo alto, pero este se compensa con la disminución parcial o total de las pruebas manuales en el área. Esto también permite una redistribución de los recursos del sector para ayudar al área de desarrollo de software en su totalidad.

La automatización presenta resultados rápidos lo que provee con realimentación para el equipo. La casi inmediata entrega de resultados permite al equipo de desarrollo actuar rápidamente para solucionar el problema, esto agiliza el proceso de desarrollo. Las pruebas manuales son lentas, esto es grave con errores que ya están en producción, pero la rápida retroalimentación permite actuar con prontitud para realizar las actualizaciones necesarias.

Uno de los principales beneficios de la automatización del análisis de calidad consiste en la mejora de la calidad a través de una mejor cobertura de pruebas. El poder reutilizar pruebas en todas las funcionalidades de un sistema permite eliminar los puntos ciegos donde la aplicación puede llegar a presentar fallos. Eliminar la dependencia hacia personas dentro del proceso de calidad también ayuda a mejorar la calidad ,ya que estas son susceptibles a fallas inherentemente de su capacidad como ingenieros o ingenieras.

La calidad del software va más allá de las pruebas. Cuando se habla de calidad del software, se debe considerar tres núcleos: la calidad del proceso de creación del código, la calidad del producto en sí y la calidad del equipo que lo está desarrollando. Sin métodos ágiles en el proceso y un equipo de trabajo motivado, la calidad del producto en sí disminuye, lo que se mide mediante pruebas. El aseguramiento de la calidad se puede definir como la “pauta planificada y sistemática de todas las acciones necesarias para proporcionar la confianza adecuada de que un elemento o producto se ajusta a las normas y requisitos establecidos” [28].

Las pruebas son importantes para la buena calidad del software, sí, pero las pruebas solo miden el software que se está ejecutando. Además, el hecho de que el software funcione no significa necesariamente que esté bien construido. La calidad del software abarca una serie de factores, como lo son: corrección, robustez, extensibilidad, reutilización, compatibilidad, eficiencia, portabilidad, facilidad de uso y funcionalidad. Cada uno de estos factores es esencial para el

conjunto, pero no determina el umbral de calidad por sí solo [18].

Actualmente, en el negocio del software, es muy común (o al menos cada vez más) implementar metodologías ágiles que permitan implementar un ciclo de vida iterativo en un proyecto en comparación con un ciclo de vida en cascada. En el mundo de las pruebas, cuando se escucha la palabra ágil, se conecta inmediatamente a todo lo relacionado con la automatización, porque es lógico que si, por ejemplo, hay ciclos de prueba repetitivos, la automatización de pruebas se acelera, mejoran los tiempos de ejecución y los tiempos de entrega.

Para automatizar, se puede iniciar cuando se tiene una aplicación donde los cambios no son muy frecuentes, es decir, se necesita cierta estabilidad en el entorno. Por ejemplo, los cambios de interfaz pueden hacer que falle la prueba automática. Por esta razón, primero se automatizan otras partes más estables de la aplicación, como el núcleo de la aplicación (pero eso no significa que en las aplicaciones no se puedan automatizar las pruebas de interfaz). Otro aspecto a considerar en esta fase es la duración del proyecto. Cuando un proyecto tiene muy poco tiempo para completarse, las pruebas dependen de la observación humana o solo requieren una única ejecución.

En estos casos, lo adecuado es no automatizar y optar por ciclos de prueba manuales, porque además, una de las cosas a tener presentes, es que la automatización de pruebas no es la eliminación del testing manual, éste debe existir e implantarse en proyectos donde, por características como las descritas anteriormente, sea necesario recurrir a él [19].

Esta es una de las preguntas más comunes que existen entre los evaluadores. Para saber cuándo automatizar las pruebas de software, es necesario evaluar la inversión potencial, el enfoque, los beneficios y, lo más importante, el conocimiento actual del proceso manual. Primero, el proceso de una prueba manual debe entenderse y conocerse completamente, y solo así, entonces, es posible la automatización. El conocimiento completo del proceso manual es la clave principal para saber cuándo se puede realizar una automatización, lo que significa que las pruebas manuales no se pueden reemplazar por completo [18].

La automatización del análisis de calidad es un proceso difícil que puede facilitarse utilizando las herramientas adecuadas. Se investiga cuáles son las herramientas más efectivas para llevar a cabo la automatización con el cuadrante mágico de Gartner [7]. Se toma en cuenta la facilidad de ejecución, la perspectiva completa de la herramienta y su posición en el mercado contra otras del estilo.

Eggplant ha logrado mantenerse en la posición de líder en el cuadrante mágico de Gartner por 2 años consecutivos. La herramienta ayuda a automatizar tecnologías en cualquier plataforma, ya sea web, de escritorio, móvil y más [8]. Este permite a analistas de calidad sin experiencia en automatización

utilizar sus conocimientos en una interfaz fácil e intuitiva con una herramienta con inteligencia artificial.

Tricentis ha logrado mantener su posición de líder en el cuadrante mágico de Gartner por cinco años consecutivos desde 2019. La herramienta permite a equipos de análisis de datos alcanzar tasas de automatización de pruebas sin precedentes de más del 90%, lo que les permite ofrecer la retroalimentación rápida requerida para metodologías ágiles [22]. La herramienta ha dado un paso más lejos optando por integrar visión artificial, un nuevo campo en la inteligencia artificial que hace uso de diferentes algoritmos, técnicas y métodos, logrando el procesamiento de la información contenida en imágenes digitales [21].

En el 2019 Smartbear se mantuvo como la opción más completa para poder automatizar el proceso de análisis de calidad. Esta posee cualidades que han sido populares en tiempos recientes como inteligencia artificial, integración con DevOps, desarrollo ágil y testeo continuo. Ofrece recomendaciones y estrategias al usuario para garantizar que se desarrollará software de calidad [9].

Es importante dejar claro ambos tipos de testing y además, no sería bueno emitir un juicio que dicte qué tipo de prueba es mejor para cada ocasión, por lo que, se requiere contar con conocimiento acerca de ambos tipos de pruebas.

El testing manual, consiste en pruebas que son ejecutadas y que tienen interacción directa del tester con el software, en muchos casos, utilizando herramientas adecuadas para cada propósito. Los testers se encargan de configurar un entorno en el que se llevará a cabo la ejecución de pruebas, por supuesto, haciendo un recorrido exhaustivo y completamente interactivo con el software objetivo de las pruebas [18].

Es importante destacar que este tipo de pruebas se encuentran expuestas a errores humanos, como la omisión de alguno de los pasos en el ciclo de prueba o un error al cargar datos. Por lo que esta es una de las razones principales por la cual se opta por un testing automatizado, para poder lograr una reducción de errores los cuales puedan generar algún tipo de inconveniente en el software [27].

Realizar las pruebas automatizadas consiste en utilizar una o varias herramientas de software para hacer pruebas a otro software. Estas herramientas permiten regular la ejecución de pruebas según las necesidades o planificación del tester, permitiendo comparar los resultados obtenidos con los resultados esperados.

No hay un mejor testing que otro, ambos pueden ser buenos si se implementan de forma adecuada, o bien ser improductivos si se aplican en el momento o caso equivocado. Sin embargo, existe una métrica llamada ROI (Return of Investment o Retorno de la Inversión) que aplicado a la automatización de pruebas, será un punto clave tanto para decidir si se debe llevar

a cabo, como para presentar resultados y sustentar la inversión que conlleva aplicar o no, este tipo de testing [19].

Para calcular el ROI de la automatización de pruebas específicamente, se tomarán en cuenta factores como el número de pruebas de regresión (en los que se aplicará la automatización de pruebas), tiempos aproximados de ejecución de estas pruebas si se hicieran de forma manual, el ahorro que representa la ejecución automatizada y el tiempo invertido en la automatización [20].

Los programadores que hacen pruebas de forma manual, en el momento en que haya una actualización en el software, necesitarán hacer una prueba de regresión, para depurar errores. Esto requiere un mayor esfuerzo y recursos, por lo cual se necesita un mejor mecanismo que evite el aumento de costos y probabilidad de errores [15].

El surgimiento de errores puede causar que disminuya la calidad, la funcionalidad e incluso aumenten los costos al tener que destinar más recursos a reparar el daño. Esta es la razón por la cual se necesita un enfoque de automatización en el proceso de pruebas de software.

Una prueba automatizada es mucho más rápida que una manual, reduce la participación de los probadores, el tiempo de pruebas y ofrece en general una mejor calidad en el software. Permite detectar de manera más rápida errores en el software, ya que proveen herramientas que permiten simular el comportamiento de múltiples usuarios. En contraste con las pruebas manuales, permite probar más en menos tiempo, lo cual permite que la detección y eliminación de errores sea mucho más eficiente [16]. Esto ayuda al equipo de desarrollo a cumplir con los requerimientos del proyecto y asegurar un producto de calidad.

El control de calidad se realiza con el fin de saber si el producto está cumpliendo con los requerimientos del cliente. Todos los procesos tienen que cumplir con las especificaciones y tener la funcionalidad esperada. Esto aplica tanto al producto final, como al proceso de desarrollo. Por ello, el proceso de testing también debe ser de la mayor calidad posible. [17].

Para asegurar la calidad, es importante saber qué tipo de prueba se va a hacer. Esta puede ser a nivel unitario, de integración o sistema. Asimismo, se hacen pruebas de regresión para identificar y solucionar los errores en el software. Esto permite preservar la funcionalidad del software.

Una prueba de regresión es una prueba que tiene como objetivo descubrir errores en el software, carencia de funcionalidades, o funcionalidad no esperada. Con esta evalúa el correcto funcionamiento del software cada vez que este se actualiza o modifica [24].

Una prueba unitaria se realiza a un componente específico de un código, los cuales pueden ser subprogramas o métodos.

Este tipo de pruebas sirven para verificar el funcionamiento de esa porción pequeña de código [23].

Para poder realizar una prueba unitaria, primero se debe seleccionar la unidad de código, o método, que se va a probar y tener definida la funcionalidad esperada de esta y los criterios de aceptación. Luego, se preparan los datos de entrada que se utilizarán para la prueba. Se ejecuta la prueba, la cual se puede hacer utilizando una herramienta o manualmente [25].

Una prueba de integración es una prueba que tiene como base las pruebas unitarias y es elaborada para validar el funcionamiento de un conjunto de módulos. Existen dos tipos: la incremental y la no incremental. En el primer tipo, se va incrementando la cantidad de módulos que se prueban hasta que se prueban todos a la vez. Se comienzan las pruebas con los módulos de bajo nivel. En la no incremental, se prueban todos los módulos a la vez [14]. Las pruebas de sistema son aquellas que evalúan la funcionalidad de todo el software como conjunto.

Para las pruebas de integración, al igual que las pruebas unitarias, se necesitan entradas. Con una herramienta, se prueba si el conjunto de varios componentes trabaja entre sí, y su funcionalidad es la esperada. Se suele probar la funcionalidad de interfaces e infraestructura. Las de sistema funcionan igual, solo que a nivel global. En general, para realizar estas pruebas se necesita preparar los datos de entrada, se definen los casos de prueba y se implementan. Luego se comparan los resultados esperados con los reales y se corrige y documenta cualquier discrepancia. [26].

Al utilizar las pruebas automatizadas se obtiene una mejor calidad y fiabilidad del ciclo de vida del software. También, se genera un control y seguimiento de las fallas y su manera de resolverlas.

#### IV. ANÁLISIS DE RESULTADOS

Parte esencial de esta investigación es contar con datos de fuente primaria que sirvan para obtener un análisis de resultados óptimo. Para ello, luego de hacer la investigación teórica, se realizaron pruebas automatizadas de tipo unitario, de integración y de sistema. Su propósito fue probar su funcionamiento y analizar si los resultados coincidían con lo obtenido en la revisión de literatura. También, se realizaron entrevistas a varios expertos que tienen experiencia en el área de testing o control de calidad para tener una comprensión más amplia del impacto de la automatización de las pruebas en entornos laborales.

Para el desarrollo de estas pruebas, se utilizaron varias herramientas como JMeter [29], BlazeMeter [30] y TestComplete [31]. Se realizaron pruebas unitarias que simulaban el comportamiento de un usuario al entrar en una aplicación web. Además, se crearon grupos de pruebas para probar el rendimiento y funcionalidad de algunas partes del

código. Finalmente, se hizo una prueba de sistema que probaba la funcionalidad total del código. Se hicieron estas pruebas tanto de forma manual como automatizada con el fin de comparar el tiempo de ejecución y cantidad de recursos que consumían, entre ambas.

Para realizar las pruebas unitarias se utiliza TestComplete donde el primer paso es identificar un método del sistema que se quiere probar, en este caso, el inicio de sesión. Una vez identificado este método, se realiza una vez la prueba para que el sistema registre todos los pasos que se realizaron.

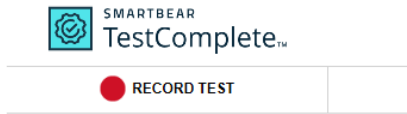


Figura 1. Pantalla del botón para grabar los pasos de una prueba automatizada. Fuente: Elaboración propia, 2021.

Dentro de la grabación de los pasos también se definen los “flags” o condicionales para que la herramienta verifique cuando el proceso se realizó con éxito, en nuestro caso la herramienta busca el título de la aplicación que aparece una vez que se inicia sesión. Después de definir los pasos, se agregan nuevas funcionalidades, como crear un ciclo donde se repiten los pasos a través de múltiples navegadores de internet, así se asegura de que el método funciona bien a través de múltiples ambientes.

| Item                 | Operation             |
|----------------------|-----------------------|
| Browser Loop         | Chrome, Firefox, Edge |
| browser              |                       |
| pageLogin            |                       |
| emailinputMail       | Click                 |
| emailinputMail       | Keys                  |
| emailinputMail       | SetText               |
| panelInicioDeSesiN   | Click                 |
| passwordboxContraseA | Click                 |
| passwordboxContraseA | Keys                  |
| passwordboxContraseA | SetText               |
| panelInicioDeSesiN4  | DbfClick              |
| submitButtonLogin    | ClickButton           |
| pagelocalhost44335   | Wait                  |
| Property Checkpoint  |                       |
| browser              |                       |
| pagelocalhost44335   |                       |
| textnodeAdmin12      | Click                 |
| linkLogout           | Click                 |
| linkLogout2          | Click                 |
| BrowserWindow        | Close                 |

Figura 2. Pantalla de los pasos grabados. Fuente: Elaboración propia, 2021.

Para las pruebas de sistemas se busca que funcione de manera correcta la interacción entre uno o más componentes del sistema, donde se utiliza el proceso el proceso de inicio de sesión y creación de nuevos usuarios dentro de un software de prueba. Como en el proceso anterior, se utilizará la herramienta TestComplete para grabar todos los pasos realizados y se definirán las condicionales para asegurarse de que el proceso se ejecutó completamente, una vez grabados los pasos se utilizará

la herramienta para generar una cantidad de datos aleatorios, estos se usarán para probar el proceso y asegurarse de que actúa correctamente ante los datos generados  $n$  cantidad de veces, donde  $n$  será el número de veces que el analista de datos considera necesario para asegurarse de que el proceso funciona bien, en este caso el valor de  $N$  será 50.

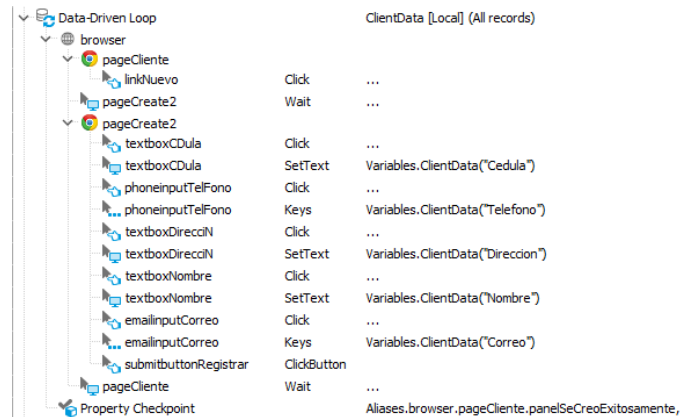


Figura 3. Pantalla del código que se ejecuta en la prueba automática. Fuente: Elaboración propia, 2021.

También, al utilizar JMeter se busca analizar y medir el rendimiento de un software a través de pruebas de carga. Esta aplicación funciona en un inicio creando un Testplan el cual consiste en configurar en la aplicación, la prueba que se requiere realizar. Posteriormente se ingresa un parámetro, el cual sería la cantidad de veces que se necesita ejecutar el Testplan.



Figura 4. JMeter Fuente: Elaboración propia, 2021.

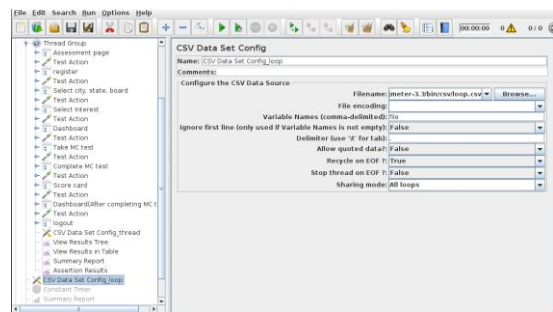


Figura 5. Jmeter Creación de Testplan Fuente: Elaboración propia, 2021.

Con esto la herramienta empieza a realizar lo configurado y, al final de ello, se muestran en gráficos los resultados. Se

pueden obtener resultados de tiempo, de estrés e incluso de carga.

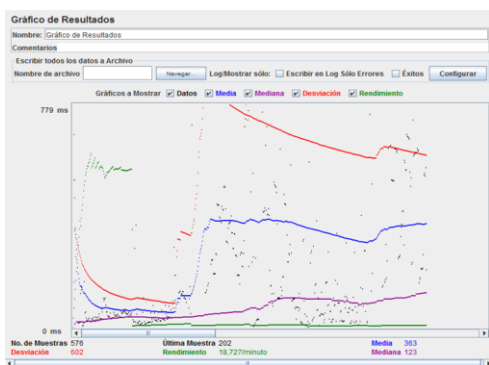


Figura 6. JMeter Gráficos de resultados  
Fuente: Elaboración propia, 2021.

Al realizar las pruebas, se encontró que las que las automatizadas en general eran más rápidas que las manuales, ya que no contaban con la intervención de un probador. En la “TABLA I” se encuentra una comparativa del tiempo de ejecución entre las diferentes pruebas manuales y automatizadas realizadas para la investigación. Asimismo, las pruebas automatizadas que simulaban el comportamiento de una persona pueden ser utilizadas para hacer pruebas de estrés o carga. Estas, permiten ejecutar la misma prueba un número elevado de veces simultáneamente para probar los límites del sistema, lo cual no se puede hacer en las pruebas manuales.

TABLA I. Comparativa entre pruebas manuales y automatizadas.

|                                    | Tiempo de aprendizaje | Tiempo de ejecución |
|------------------------------------|-----------------------|---------------------|
| Prueba unitaria manual             | 20 minutos            | 30 minutos          |
| Prueba unitaria automatizada       | 1 hora                | 5 a 10 minutos      |
| Prueba de integración manual       | 30 minutos            | 40 minutos          |
| Prueba de integración automatizada | 1 a 2 horas           | 10 a 20 minutos     |
| Prueba de sistema manual           | 1 a 2 horas           | 40 minutos          |
| Prueba de sistema automatizada     | 3 horas               | 25 a 30 minutos     |

En los últimos años se han comenzado a implementar alternativas para complementar las pruebas manuales en el sector de análisis de calidad. Según las entrevistas realizadas, como se muestra en el gráfico 1, dentro de las alternativas más populares se encuentra la automatización del sector, donde más del 60% de las empresas presenta cierto nivel de

automatización. Por otro lado, más del 70% de las personas con experiencia en el sector de análisis de calidad, según la información recaudada, tal y como se muestra en el gráfico 2, afirman tener experiencia implantado la automatización de pruebas. Se puede apreciar cómo la automatización ha estado reemplazando o cooperando con las pruebas manuales para poder realizar un mejor trabajo en el sector de análisis de calidad.

### Casos de automatización en el entorno laboral actual de los participantes

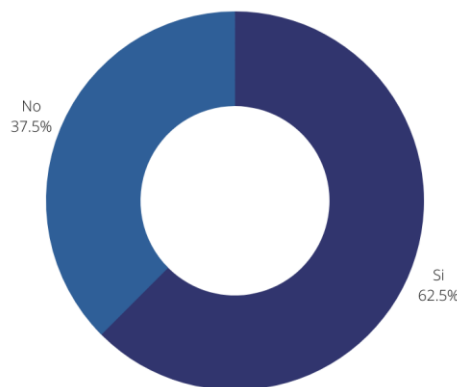


Gráfico 1. Casos de automatización en el entorno laboral actual de los participantes  
Fuente: Elaboración propia, 2021.

### Experiencia en automatización del sector de análisis de calidad de software

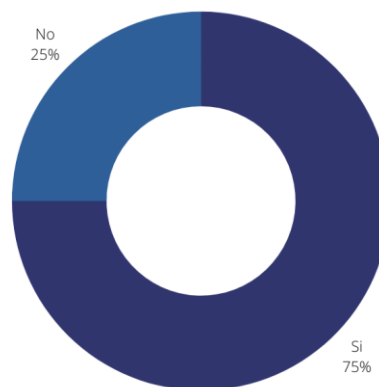


Gráfico 2. Experiencia en automatización del sector de análisis de calidad de software.  
Fuente: Elaboración propia, 2021.

Se han utilizado las pruebas manuales para analizar la calidad del software desde el inicio, pero estas siempre han presentado problemas que, con el pasar del tiempo, se han vuelto más y más claros. El 60% de los entrevistados calificó la duración de las pruebas manuales como un problema grave que está retrasando al sector, como se muestra en el gráfico 3. Mientras tanto, se han encontrado problemas constantes por factores humanos, donde la mala realización de una prueba o la falta de realización de estas debido a diversas razones ha afectado la calidad del software. También, se encontraron

desventajas debido a la seguridad, recursos, dificultad para repetir la prueba y confianza con el cliente.

### Problemas más comunes en pruebas manuales de software

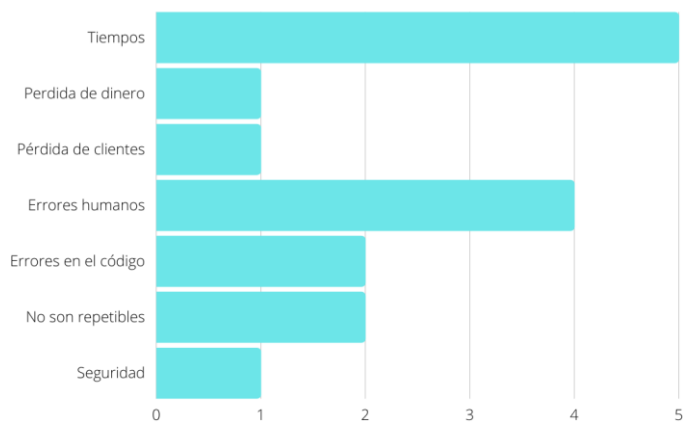


Gráfico 3. Problemas más comunes al realizar pruebas manuales de software  
Fuente: Elaboración propia, 2021

Con base en las entrevistas realizadas, se obtuvo como resultado que la gran mayoría de los consultados consideran a la automatización como una herramienta efectiva para reducir los recursos en las pruebas de software. Esto debido a que permiten ejecutar una cantidad mucho mayor de casos de prueba en menos tiempo, lo que significa que ofrece resultados positivos en un menor tiempo.

### ¿LA AUTOMATIZACIÓN ES UNA HERRAMIENTA EFECTIVA PARA REDUCIR RECURSOS?

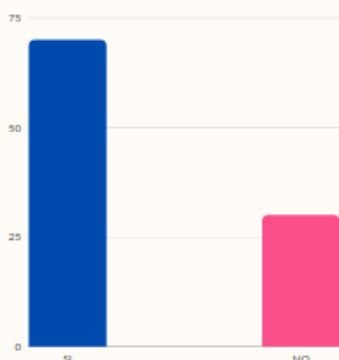


Gráfico 4 . La automatización como una fuente efectiva para ahorrar recursos  
Fuente: Elaboración propia, 2021.

### LA AUTOMATIZACIÓN AYUDA A AHORRAR RECURSOS ¿CÓMO?

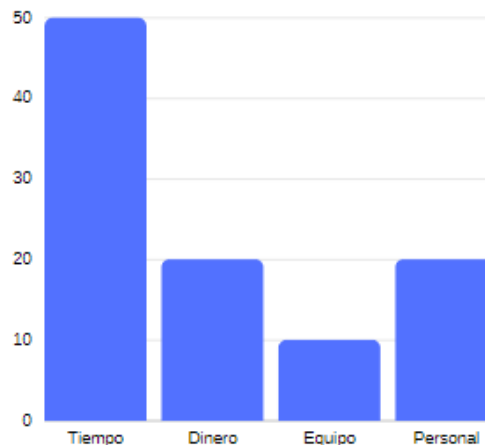


Gráfico 5 . Como es la automatización una fuente efectiva para ahorrar recursos  
Fuente: Elaboración propia, 2021.

A largo plazo, es posible ver grandes resultados a favor con respecto a lo económico. Esto debido a que se pueden crear bloques de pasos para realizar las pruebas y volverlos a utilizar como objetos cuando sea necesario. Lo que conlleva a un ahorro económico al no tener que crear nuevas automatizaciones desde cero.

Igualmente, se obtuvieron resultados en el ámbito general de la calidad del software, por ejemplo, según los entrevistados, como se muestra en el gráfico 6, el 80% concuerda en que una reducción de la interacción humana en las pruebas ayudaría a evitar los errores que ocurrían con anterioridad al realizar pruebas manuales. Esto sucede debido a que se evitan en gran porcentaje la posible presencia de interacciones con humanos, lo que genera que el software cada vez sea más adecuado a lo que se requiere y que cuente con la menor cantidad de fallas posibles.

Gran parte de los consultados considera que es de suma importancia evitar lo más que se pueda la interacción humana en las pruebas, sin embargo, una parte de estos concuerdan en que es importante implementar ciertas pruebas manuales, ya que, según el tipo de prueba a realizar, o bien, el resultado que se requiera, las manuales pueden brindar resultados más certeros y exactos que cuando se realiza una prueba automatizada.

### ¿ES NECESARIA LA DISMINUCIÓN DE LA INTERACCIÓN HUMANA PARA AUMENTAR LA CALIDAD DEL SOFTWARE?

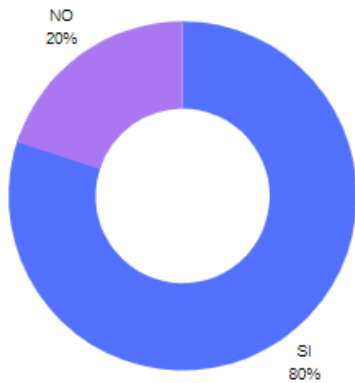


Gráfico 6. Disminución de interacción humana para aumentar la calidad  
Fuente: Elaboración propia, 2021.

Como se muestra en el gráfico 7, la mayoría de los entrevistados, a excepción de uno solo, están de acuerdo en que las pruebas automatizadas permiten tener una mejor cobertura a la hora de probar el funcionamiento de un software. Permiten abarcar más componentes del sistema y revisarlos en un tiempo más corto. Sin embargo, la persona que estaba en desacuerdo comentó que algunos elementos sólo pueden ser probados de manera manual. En estas situaciones, explican que los probadores pueden trabajar en conjunto al hacer pruebas manuales en esos casos específicos, mientras que se automatizan las demás pruebas.

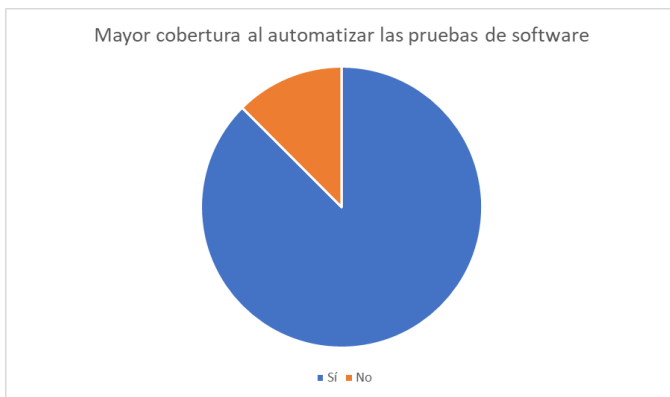


Gráfico 7. ¿Se presenta una mayor cobertura al automatizar las pruebas de software?  
Fuente: Elaboración propia, 2021.

También, se comenta que llegar a una cobertura total del sistema es una tarea difícil y la cobertura depende de cómo estén diseñadas las pruebas. El conjunto de pruebas debe tener un diseño óptimo y estas deben ser significativas. Si una prueba no es significativa, no agrega valor al proceso de testing y no va a aportar nada cuando se hagan actualizaciones al sistema.

Si el grupo de pruebas es de calidad, este podría probar la mayoría de los escenarios y cubrir la mayoría de las líneas de código.

En cuanto a las herramientas de automatización que los entrevistados conocen o han utilizado y que han presentado resultados positivos en su entorno laboral, nombraron Jest [32], JMeter, Selenium [33] y TestComplete como las más utilizadas. Manejan Jest para pruebas unitarias; Selenium para automatización de flujos; JMeter para pruebas de estrés y rendimiento; TestComplete para probar aplicaciones web. Otras herramientas que citan son: GitLab CI/CD, Karma, Jasmine, Postman, Nightwatch.js, Moq, JUnit, Protractor y Ranorex. Esto se refleja en el gráfico 8.

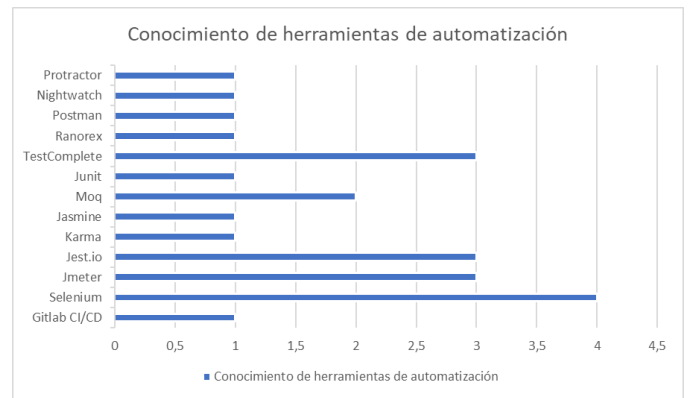


Gráfico 8. Conocimiento sobre herramientas de automatización de pruebas de software,  
Fuente: Elaboración propia, 2021.

Cabe destacar que nombraron a JMeter y TestComplete, las cuales, como se dijo anteriormente, fueron las herramientas que se utilizaron para hacer las pruebas en la investigación primaria. Los entrevistados coinciden en que estas herramientas dan un gran rendimiento a la hora de hacer pruebas automatizadas. TestComplete, la cual es una herramienta de SmartBear, y JMeter, de Apache, parecen ser las herramientas más utilizadas para hacer pruebas automáticas.

#### IV. CONCLUSIONES Y RECOMENDACIONES

El sector de análisis de calidad no ha recibido cambios significativos en la última década, esto contrasta con el resto del área de desarrollo de software que está en constante evolución para ser más eficaz y eficiente. La automatización es una herramienta que se ha popularizado en varios sectores de la industria por sus resultados efectivos y el sector de análisis de calidad ha demostrado ser uno más de ellos.

La investigación demuestra las mejoras que otorga la automatización al sector de análisis de calidad. Los resultados obtenidos mostraron que se puede reducir el tiempo y los errores en el sector de análisis de calidad. Por lo tanto, se concluye que la automatización puede mejorar el proceso de análisis de calidad, reducir el gasto de recursos, identificar



errores y, sobre todo, otorgar velocidad, aspectos que se han vuelto fundamentales en los últimos años donde ha aumentado la demanda de creación de software. Se recomienda su implementación teniendo en cuenta la situación actual del sector en la empresa donde se requiera implementar y las ventajas que esta ofrece al sector para poder llevar a cabo una implementación donde se aproveche al máximo su potencial.

Para llevar a cabo la automatización, se identifican múltiples herramientas usando fuentes como el cuadrante mágico de Gartner. Se puede ver resultados efectivos con JMeter y TestComplete que ayudaron a realizar las pruebas de manera eficiente. Se concluye que las herramientas vistas dentro del cuadrante como TestComplete, entre otras, otorgan grandes facilidades para la automatización del sector, además, ofrecen un proceso más eficiente. Se recomienda buscar dentro de fuentes confiables, como lo es Gartner, herramientas para la automatización, una vez localizada una fuente confiable se debe ver dentro de las características y funcionamientos de la herramienta para así obtener las que se adapten mejor a la empresa y ofrezcan los mejores resultados.

Uno de los factores que más dañan al sector es la probabilidad de errores en el mismo. Luego del análisis de resultados, se puede ver cómo la automatización logró disminuir los errores dentro del software. Con esto, se concluye que la automatización logra evitar errores dentro del software gracias a dos factores importantes, esta permite reducir la interacción entre las pruebas y las personas lo que logra reducir la probabilidad de error humano, también se logra un aumento de las pruebas dentro del sistema asegurando un funcionamiento correcto. Se recomienda el uso de la automatización si el sector de análisis de calidad está presentando errores humanos ocasionales poniendo el software en riesgo o si las pruebas manuales realizadas no logran detectar los errores dentro del software.

Si bien las pruebas manuales son más lentas y menos eficientes, estas, en situaciones específicas, no pueden ser reemplazadas por completo, por lo que no se pueden automatizar. Se recomienda que se sigan haciendo las pruebas manuales en estas situaciones específicas y se automatice el resto de las pruebas. Es decir, puede haber una coexistencia entre los dos tipos de pruebas.

Los pasos para realizar las pruebas unitarias, de integración y de sistema siguen un mismo patrón. Se identifica el tipo de prueba que se requiere hacer, se selecciona el código al que se le va a aplicar la prueba, se debe tener en claro los criterios de aceptación o requerimientos, se realiza la prueba y se decide si cumple con estos para tomar decisiones futuras. Con base en la investigación, se concluye que existen diferentes herramientas para poder realizar estas pruebas que ayudan a la automatización de estas. Tomando como base las pruebas realizadas y los testimonios de los entrevistados, se recomienda el uso de estas herramientas, como TestComplete, JMeter o Selenium, para realizar estas pruebas de software.

Ciertos factores son de suma importancia a la hora de intentar realizar una prueba automatizada. Algunos de estos son: la carga de trabajo manual, transacciones de alto volumen, gran posibilidad de errores como consecuencia del factor humano.

En el análisis de resultados se aprecia como efectivamente estos factores son fundamentales a la hora de realizar una automatización, debido a que la mayoría de los entrevistados concuerdan en que estos factores son como las “señales” que indican que algún proceso de prueba se debe automatizar. Esto para buscar reducir los problemas anteriormente mencionados.

Se concluye que estos factores pueden indicar la falta de una automatización en el área de pruebas, esto debido a la presencia de fallas en el proceso. Además, se recomienda analizar el proceso que se esté realizando actualmente en el sector de pruebas y verificar si se están presentando algunos de los factores mencionados para así poder iniciar el proceso de implementar la automatización de pruebas cuando sea necesario, buscando efectivamente la reducción de estos.

Asimismo, se puede concluir que al poner en evidencia los factores anteriormente mencionados, se puede contribuir a la mejora de la calidad del software. Esto debido a que se ve una disminución en la presencia de errores al momento de ejecutar una prueba como tal, lo que conlleva a tener un software con la menor cantidad de incidencias posibles, aumentando en gran medida la calidad y rendimiento de este.

## VI. TRABAJOS FUTUROS

Para tener una visión más amplia sobre los resultados obtenidos en esta investigación, se requiere involucrar a científicos de datos, estadísticos y personas expertas en el área de control de calidad para que hagan un estudio más profundo y obtengan un análisis con datos cuantitativos con el fin de corroborar los resultados obtenidos.

Para poder obtener mayor variedad de datos, se busca ampliar la investigación de cualitativa a mixta para así poder obtener información más variada que ofrezca resultados más profundos en el tema.

## VI. REFERENCIAS

- [1] D. Beyer. First international competition on software testing. *International Journal on Software Tools for Technology Transfer*, 1-14. 2021
- [2] E. Serna, R. Martínez & P. Tamayo. Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas*, 23(1), 169-183. 2019
- [3] V. Llobell & A. T. Pruebas funcionales automatizadas para aplicaciones Web: usando Selenium para aplicar pruebas de

- regresión automatizadas (Doctoral dissertation, Universitat Politècnica de València). 2018
- [4] King, T. M., Arbon, J., Santiago, D., Adamo, D., Chin, W., & Shanmugam, R.. AI for testing today and tomorrow: industry perspectives. In 2019 IEEE International Conference on Artificial Intelligence Testing (AITest). Abril, 2019, pp. 81-88. IEEE.
- [5] S. Şentürk, A. Akin, A. Karagöz, A. B. & V. Garousi. Pratikte Model Tabanlı Test: Bankacılık Alanından Bir Deneyim Raporu.. 2019
- [6] Real Academia Española. Investigar. [En Línea] Rae. Disponible en <https://dle.rae.es/investigat>
- [7] Herschmann, J. (2019, 25 noviembre). Magic Quadrant for Software Test Automation. [En Línea] Gartner. Disponible en: <https://www.gartner.com/en/documents/3975493/magic-quadrant-for-software-test-automation>.
- [8] Bloomberg. (2019, 2 diciembre). Eggplant Recognized as a Leader in Gartner's 2019 Magic Quadrant for Software Test Automation for the Second Consecutive Year. [En Línea] Bloomberg. Disponible en: <https://www.bloomberg.com/press-releases/2019-12-02/eggplant-recognized-as-a-leader-in-gartner-s-2019-magic-quadrant-for-software-test-automation-for-the-second-consecutive-year>
- [9] Smartbear. (2019, 25 noviembre). Magic Quadrant for Software Test Automation 2019 | Gartner Report | SmartBear Software. [En Línea] SmartBear. Disponible en: <https://smartbear.com/resources/white-papers/gartner-magic-quadrant-2019/>.
- [10] B. Mohammed. How to Write Qualitative Research. International Journal of Quantitative and Qualitative Research Methods, 9(3), 1-6. 2021
- [11] R. Hernández-Sampieri & C. P. M. Torres. Metodología de la investigación (Vol. 4). México^ eD. F DF: McGraw-Hill Interamericana. 2018.
- [12] G. del Roció Veloz-Remache, J. A. Menéndez-Verdecia, y L. N. Aguilar-Moncayo. 2021. Mejores prácticas de calidad en el desarrollo de software integradas al conocimiento de la ingeniería. Polo del Conocimiento, vol 6, no. 1, pp. 656-668.[En Línea] <https://polodelconocimiento.com/ojs/index.php/es/article/view/2170/4340#> [Accedido: octubre 2021]
- [13] J. J. Li, A. Ulrich, X. Bai y A. Bertolino. Advances in test automation for software with special focus on artificial intelligence and machine learning. Software Quality Journal, vol. 28, no. 1, pp. 245-248, 2020. [En Línea] Disponible en: <https://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=143095779&lang=es&site=ehost-live> .[Accedido: octubre 2021].
- [14] C. P. Suárez, y C. Fontela. 2020. Documentación y pruebas. [En línea]. Universidad de Buenos Aires. [http://materias.fi.uba.ar/7507/content/20101/lecturas/documentacion\\_pruebas.pdf](http://materias.fi.uba.ar/7507/content/20101/lecturas/documentacion_pruebas.pdf). [Accedido: Noviembre 2021].
- [15] E. Gómez Cortez Kenia. Implementación del Nightwatch como una estrategia para Testing en las empresas de desarrollo de software. 2020. [En Línea].Disponible en: <http://repositorio.upsin.edu.mx/Fragmentos/tesinas/A003GOMEZCORTEZKENIAESTEFANIA9896.pdf> [Accedido: octubre 2021]
- [16] E. Serna, R. Martínez y P. Tamayo. 2019. Una revisión a la realidad de la automatización de las pruebas del software. Computación y Sistemas, vol 23, no. 1, pp. 169-183.[En Línea] <https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/2782/2602> .[Accedido: octubre 2021]
- [17] W. A. Lucero, C. H. Salgado, A. Sánchez, y M. Peralta. 2020. Fiabilidad en la calidad del software: modelos, métodos y estrategias.[En línea]. XXII Workshop de Investigadores en Ciencias de la Computación. [http://sedici.unlp.edu.ar/bitstream/handle/10915/104045/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/104045/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y) [Accedido: noviembre 2021].
- [18] Jiménez, J., 2021. *Testing Manual vs. Automatizado: ¿cuál es mejor?* / *Abstracta Chile*. [online] Blog de Testing y Calidad de Software | Abstracta Chile. Available at: <<https://cl.abstracta.us/blog/testing-manual-vs-testing-automatizado/>> [Accessed 29 October 2021].
- [19] Interware.com.mx. 2018. *Software Testing: Pruebas manuales vs pruebas automatizadas*. [online] Available at: <<https://www.interware.com.mx/blog/pruebas-manuales-vs-pruebas-automatizadas>> [Accessed 29 October 2021].
- [20] Kumar, V., 2019. *Costos de la automatización de pruebas*. [online] Encora. Available at: <<https://www.encora.com/es/blog/costos-de-la-automatizacion-de-pruebas>> [Accessed 30 October 2021].
- [21] J. Arias, L. R. A. Fonseca & J. Moreno. Object Recognition Through Artificial Intelligence Techniques. Facultad de Ingeniería, 29(54), 3. 2020
- [22] E. Coleman. (2020, 26 marzo). Tricentis Recognized as a 2020 Gartner Peer Insights Customers' Choice for Software Test Automation. [En línea] Tricentis. Disponible en: <https://www.tricentis.com/news/tricentis-recognized-as-2020-gartner-peer-insights-customers-choice-for-software-test-automation/>
- [23] J. Muñoz Marín. 2020. Desarrollo de una aplicación web y móvil para que sirva como referencia técnica en proyectos de desarrollo de software emergentes dentro de la compañía Cidenet S.A.S. [En línea]. Universidad de Antioquia. [http://repositorio.udea.edu.co/bitstream/10495/18509/1/MunozJuliana\\_2020\\_AplicacionWebMovil.pdf](http://repositorio.udea.edu.co/bitstream/10495/18509/1/MunozJuliana_2020_AplicacionWebMovil.pdf) [Accedido: noviembre 2021].
- [24] E. Díaz Asencio. 2021. Automatización de pruebas de regresión. [En línea]. Universidad de Sevilla. <https://idus.us.es/handle/11441/125114> [Accedido: Noviembre 2021].
- [25] *Pruebas unitarias*. (s. f.). Servicio de Informática. [En línea]. Universidad de Alicante. <https://si.ua.es/es/documentacion/c-sharp/documentos/pruebas/07pruebasunitarias.pdf> [Accedido: noviembre 2021].
- [26] A. P. Duquino Sánchez. 2020. Automatización de un Sistema de Pruebas de Software para la Optimización del Proceso de Calidad de DetectID. [En Línea]. Universidad Nacional de Colombia, Facultad de Ingeniería. <https://repositorio.unal.edu.co/bitstream/handle/unal/79288/Tests%20Maestri%CC%81a%20Angela%20Paola%20Duquino%20S.pdf?sequence=1&isAllowed=y> [Accedido: noviembre 2021].
- [27] M. Medina Yacupoma. 2020. Programa Especial de Titulación: "AUTOMATIZACIÓN DE PRUEBAS PARA PROYECTOS ÁGILES APLICANDO EL DESARROLLO DIRIGIDO POR COMPORTAMIENTO PARA UNA COMPANIA DE LINEAS DE BELLEZA". [En línea]. Universidad Tecnológica del Perú, Facultad de Ingeniería de Software. [https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3166/Michael%20Medina\\_Trabajo%20de%20Suficiencia%20Profesional\\_Titulo%20Profesional\\_2020.pdf?sequence=1&isAllowed=y](https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3166/Michael%20Medina_Trabajo%20de%20Suficiencia%20Profesional_Titulo%20Profesional_2020.pdf?sequence=1&isAllowed=y) .[Accedido: noviembre 2021].
- [28] I. López Baldominos. 2020. Aplicación de herramientas y análisis de datos para el aseguramiento de la calidad de software. [En línea]. Escuela Politécnica Superior. [https://ebuah.uah.es/xmlui/bitstream/handle/10017/43889/TFG-Lopez\\_Baldominos\\_2020.pdf?sequence=1&isAllowed=y](https://ebuah.uah.es/xmlui/bitstream/handle/10017/43889/TFG-Lopez_Baldominos_2020.pdf?sequence=1&isAllowed=y) [Accedido: noviembre 2021].

- [29] “Apache JMeter,” Apache.org. [En línea]. Available: <https://jmeter.apache.org/index.html>. [Accedido: diciembre 2021].
- [30] “The complete continuous testing automation platform,” Blazemeter.com. [En línea]. Disponible en: <https://www.blazemeter.com/>. [Accedido: Diciembre 2021].
- [31] “TestComplete Automated UI Testing Tool,” Smartbear.com. [En línea]. Disponible en: <https://smartbear.com/product/testcomplete/overview/>. [Accedido: Diciembre 2021].
- [32] “Jest,” Jestjs.io. [En línea]. Disponible en: <https://jestjs.io/>. [Accedido: Diciembre 2021].
- [33] “Selenium,” Selenium.dev. [En línea]. Disponible en: <https://www.selenium.dev/>. [Accedido: Diciembre 2021].