

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE COMPUTACIÓN

PROGRAMA DE MAESTRÍA EN GERENCIA DE TI



Propuesta de Metodología de Control de Calidad para el  
Departamento de Automatización de QA

Proyecto para optar por el grado de Maestría Profesional  
con Énfasis en Gerencia de Tecnologías de Información

Carmen María Mok Zheng

Prof. Adriana Álvarez Figueroa

San José, Costa Rica

Octubre, 2024

## ACTA DE APROBACION DE PROYECTO

Propuesta de Metodología de QA para el departamento de Automatización de QA

CARMEN MARIA MOK ZHENG

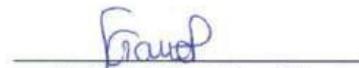
### TRIBUNAL EXAMINADOR



MAP. Adriana Álvarez Figueroa  
Profesor Asesor



Dr. Mauricio Arroyo Herrera  
Profesor Lector



Mgr. Silvia Elena Garro Pifa  
Lector Externo



Dra.-Ing. Lilliana Sancho Chavarría  
Presidente  
Tribunal Evaluador Proyecto Final de Graduación



22 de octubre, 2024



## **Dedicatoria**

A mis queridos padres, cuyo apoyo incondicional y amor han sido la base de cada uno de mis logros. Estoy eternamente agradecida por creer en mí, por sus sacrificios, y por enseñarme el valor del esfuerzo y la perseverancia.

## **Agradecimientos**

En primer lugar, quiero agradecer a mi familia, cuyo apoyo incondicional ha sido el pilar sobre el cual he construido cada uno de mis logros. A todos aquellos que han estado a mi lado, brindándome su amor y comprensión, incluso en los momentos más difíciles, mi gratitud eterna.

Quiero expresar mi más sincero agradecimiento a mi profesora Adriana Álvarez Figueroa, cuya guía y apoyo constante han sido invaluableles en la realización de esta tesis. Su dedicación y pasión por la enseñanza me han inspirado a superar los desafíos y alcanzar nuevos niveles de conocimiento.

Agradezco profundamente a la empresa donde se llevó a cabo este proyecto por brindarme la oportunidad de desarrollarlo en sus instalaciones, lo que me permitió aplicar y expandir mis conocimientos en un entorno profesional.

También quiero reconocer a todos los profesores de la maestría, así como a los expertos que me brindaron su apoyo, por compartir su experiencia y sabiduría. Su conocimiento y orientación han sido fundamentales en mi formación académica y profesional.

Finalmente, agradezco a mis compañeros, quienes con su amistad, colaboración y apoyo hicieron que este recorrido fuera más enriquecedor y gratificante.

## **Epígrafe**

“Around here, however, we don't look backwards for very long. We keep moving forward, opening up new doors and doing new things, because we're curious...and curiosity keeps leading us down new paths.”

– Walt Disney.

## **Resumen**

El objetivo de este proyecto es proponer una metodología de control de calidad (QC) para el departamento de automatización del QA (*QA Automation*) de una empresa, a la cual denominaremos Empresa A. Esta propuesta se basa en un análisis detallado de la situación actual del departamento, en la revisión de diversos marcos de referencia y en las opiniones de expertos en el campo del aseguramiento de la calidad.

Para alcanzar este objetivo, se realizó un análisis exhaustivo del proceso de QC actualmente implementado en dicho departamento, documentando la metodología utilizada.

Como parte del análisis preliminar, se investigaron distintos marcos de referencia que sirvieron de guía para el desarrollo de la nueva metodología. Además, se aplicaron encuestas a los miembros del departamento de *QA Automation* y se llevaron a cabo entrevistas con expertos de la industria para obtener una visión más amplia sobre las prácticas actuales en el campo y para identificar las necesidades y expectativas del departamento.

Con base en la información obtenida, se propone una metodología de QC adaptada al contexto específico del departamento de *QA Automation*, junto con un plan de acción para su implementación. Se espera que esta metodología se convierta en una herramienta valiosa para mejorar el proceso de QC en la Empresa A.

# Índice General

1.	Introducción .....	12
1.1.	Antecedentes .....	14
1.1.1.	Descripción de la empresa.....	15
1.1.2.	Descripción del problema.....	21
1.2.	Planteamiento del problema .....	22
1.3.	Justificación del proyecto.....	23
1.3.1.	Innovación.....	23
1.3.2.	Impacto.....	24
1.4.	Objetivos .....	25
1.4.1.	Objetivo general .....	25
1.4.2.	Objetivos específicos.....	25
1.5.	Alcance, supuestos y exclusiones.....	26
1.5.1.	Alcance.....	26
1.5.2.	Supuestos.....	29
1.5.3.	Exclusiones.....	30
2.	Marco teórico .....	31
2.1.	Diferencia entre aseguramiento de calidad del software y el control de calidad del software31	
2.2.	Fundamentos del proceso de pruebas.....	32
2.3.	Metodología de QA.....	33
2.4.	Estándares y marcos de referencia para pruebas de software .....	34
2.4.1.	ISTQB Certified Tester Foundation Level.....	34
2.4.2.	ISTQB Advanced Level Syllabus - Test Automation Engineer.....	38
2.4.3.	PMBOK .....	42
2.4.4.	Capability Maturity Model Integration (CMMI) .....	45
2.4.5.	Series de ISO/IEC/IEEE 29119 .....	48
3.	Marco metodológico .....	53
3.1.	Matriz de congruencia.....	53
3.2.	Tipo de investigación .....	57
3.3.	Población.....	59
3.4.	Fuentes de información .....	60

3.5.	Instrumentos y técnicas de recopilación de información.....	60
4.	Análisis de resultados.....	62
4.1.	Aspectos demográficos.....	62
4.1.1.	Personas entrevistadas.....	63
4.2.	Estado actual del departamento de QA Automation.....	64
4.2.1.	Metodología actual del departamento de QA Automation.....	64
4.2.2.	Procedimiento para la liberación del software.....	76
4.2.3.	Sistemas de métricas presentes en la metodología de QC actual.....	77
4.2.4.	Necesidades detectadas.....	77
4.2.5.	Expectativas detectadas.....	79
4.2.6.	Importancia de disponer de métricas.....	80
5.	Propuesta de solución.....	82
5.1.	Roles en metodología de QC.....	82
5.2.	Planificación de la ejecución y desarrollo de pruebas.....	83
5.3.	Ejecución de las pruebas.....	85
5.3.1.	Pruebas de Integración.....	85
5.3.2.	Pruebas de Regresión.....	86
5.3.3.	Pruebas de Humo (smoke tests).....	87
5.4.	Priorización de las pruebas.....	87
5.5.	Monitorización y control de las pruebas.....	88
5.5.1.	Nivel de calidad.....	95
5.6.	Gestión de defectos.....	107
5.7.	Gestión de la configuración del software de prueba.....	109
5.8.	Generación de Informes.....	112
5.9.	Retrospectiva.....	117
5.10.	Propuesta de plan de acción de la metodología propuesta.....	118
5.10.1.	Etapas de Plan de Acción.....	118
5.10.2.	Cronograma Propuesto.....	124
5.10.3.	Factores críticos de éxito.....	132
6.	Conclusiones y recomendaciones.....	133
6.1.	Conclusiones.....	133
6.1.1.	Beneficios de la metodología de QC propuesta.....	135

6.1.2.	Riesgos potenciales del proyecto .....	136
6.1.3.	Trabajo a futuro.....	137
6.2.	Recomendaciones .....	139
7.	Glosario.....	140
8.	Bibliografía.....	141
9.	Anexo .....	144
9.1.	Anexo 1.....	144
9.2.	Anexo 2.....	146
9.3.	Anexo 3.....	152
9.4.	Anexo 4.....	153
9.4.1.	Respuesta 1.....	154
9.4.2.	Respuesta 2.....	156
9.4.3.	Respuesta 3.....	158
9.4.4.	Respuesta 4.....	160
9.5.	Anexo 5.....	162
9.6.	Anexo 6.....	163
9.7.	Anexo 7.....	164
9.7.1.	Entrevista con Álvaro Rivera .....	164
9.7.2.	Entrevista con Ana Sanabria .....	172
9.7.3.	Entrevista con Ignacio Díaz Oreiro.....	175
9.7.4.	Entrevista con Mario Jiménez .....	178
9.7.5.	Entrevista con Luis Sanabria.....	181
9.7.6.	Entrevista con Ignacio Trejos.....	186
9.7.7.	Entrevista con Marlen Treviño.....	188
9.7.8.	Entrevista con Laura Chavarría.....	191
9.8.	Anexo 8.....	195
9.9.	Anexo 9.....	197

# Índice de Tablas

<b>Tabla 1.</b> Declaración de Alcance.....	27
<b>Tabla 2.</b> Matriz de Congruencia .....	54
<b>Tabla 3.</b> Personas entrevistadas .....	63
<b>Tabla 4.</b> Guía para la asignación de prioridad a los fallos encontrados .....	74
<b>Tabla 5.</b> Métricas propuestas para la metodología de QC.....	89
<b>Tabla 6.</b> Valores para utilizar en matriz propuesta para el cálculo de la prioridad de un fallo .....	98
<b>Tabla 7.</b> Matriz propuesta para el cálculo de la prioridad de un bug encontrado.....	99
<b>Tabla 8.</b> Valores límites para la definición del nivel de prioridad de un bug.....	100
<b>Tabla 9.</b> Ejemplo de producto con calidad alta .....	103
<b>Tabla 10.</b> Ejemplo de producto con calidad media .....	104
<b>Tabla 11.</b> Ejemplo de producto con calidad mala .....	105
<b>Tabla 12.</b> Ejemplo de producto con calidad mala .....	106
<b>Tabla 13.</b> Plantilla de informe de inicio de pruebas .....	113
<b>Tabla 14.</b> Plantilla de informe de avance de la ejecución de pruebas .....	115
<b>Tabla 15.</b> Plantilla de informe final de la ejecución de pruebas .....	116
<b>Tabla 16.</b> Cronograma propuesto para la ejecución del plan de acción .....	125

# Índice de Figuras

<b>Figura 1.</b> Organigrama de la Empresa A.....	12
<b>Figura 2.</b> Organigrama de la Empresa A enfocado en los 3 puestos de jefatura de la organización. ....	16
<b>Figura 3.</b> Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura de seguridad.....	17
<b>Figura 4.</b> Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura financiera y de operaciones .....	18
<b>Figura 5.</b> Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura de tecnología.....	19
<b>Figura 6.</b> Organigrama de la Empresa A enfocado en las áreas bajo el mando de la dirección de inteligencia de negocio, los matemáticos de juegos, los ingenieros en inteligencia de negocio y las áreas bajo el mando de la gerencia senior de ingeniería de software.....	19
<b>Figura 7.</b> Organigrama de la Empresa A enfocado en las áreas bajo el mando de la dirección de producto, el arquitecto de software, las áreas bajo el mando de la gerencia de base de datos, las áreas bajo la gerencia de SRE/DevOps, y las áreas bajo el mando de la gerencia de ingeniería de software.....	20
<b>Figura 8.</b> Diagrama de flujo del proceso actual de QC para la liberación de un producto .....	72
<b>Figura 9.</b> Ejemplo de matriz de probabilidad e impacto con esquema de puntuación .....	97
<b>Figura 10.</b> Diagrama del plan de acción para la implementación de la metodología de QC propuesta .....	128
<b>Figura 11.</b> Parte 1 del diagrama del plan de acción para la implementación de la metodología de QC propuesta.....	129
<b>Figura 12.</b> Parte 2 del diagrama del plan de acción para la implementación de la metodología de QC propuesta.....	130
<b>Figura 13.</b> Parte 3 del diagrama del plan de acción para la implementación de la metodología de QC propuesta.....	131

# 1. Introducción

En cualquier desarrollo de un proyecto, una etapa importante es la del aseguramiento de la calidad del *software*, ya que nos ayuda a detectar errores en el menor tiempo posible o retrasos en las entregas durante el desarrollo que pueden provocar el descontento del cliente y sus clientes por problemas en el producto.

Para esto es fundamental contar con una metodología de QC (Control de Calidad o “*Quality Control*” en inglés) que nos ayude a comprender el proceso, tener una mejor comprensión de los resultados de los distintos tipos de pruebas ejecutadas y tener visibilidad de la calidad del producto.

Dentro de este contexto, se encuentra el departamento de *QA Automation* (Automatización de QA) de la “Empresa A”, de ahora en adelante denominada de esta manera por motivos de confidencialidad, la cual se encarga principalmente del desarrollo de pruebas automatizadas según las necesidades de los demás departamentos y de la ejecución de dichas pruebas durante el desarrollo de un proyecto.

Dicho departamento actualmente cuenta con distintas herramientas para la ejecución de pruebas automatizadas y un catálogo de pruebas de la cuál disponen los empleados de la organización, sin embargo, este carece de una metodología de pruebas que incluyan indicadores claves de calidad que ayuden a medir el rendimiento del departamento y el monitoreo de las pruebas ejecutadas para tener una mejor visibilidad de la calidad del producto según las métricas de calidad de la empresa.

Debido a lo anterior, en el presente proyecto se pretende realizar un análisis cualitativo mediante una fase exploratoria del departamento y sus necesidades, y con base a lo anterior, se realiza la elaboración de una propuesta de una metodología de QC para el

departamento de *QA Automation* de la Empresa A que aporte valor no solamente al equipo, si no a la organización en general.

En este documento se detallan los aspectos asociados a la descripción de la empresa y el planteamiento y justificación del problema respecto a la metodología de QC actual del departamento de *QA Automation*. Se identifican los objetivos a alcanzar a partir del desarrollo de este proyecto, que se encuentran enmarcados según su respectivo alcance, supuestos y exclusiones del proyecto, para dar contexto sobre dónde se realizará el proyecto.

En relación con el desarrollo de una propuesta de metodología de QC para el departamento de automatización del aseguramiento de la calidad (*QA Automation*), se brinda un marco teórico con información necesaria para el proyecto y el marco metodológico a utilizar. Con base en lo anterior, se presenta el análisis y resultados de la aplicación del marco metodológico, la propuesta de solución, y finalmente las conclusiones y recomendación de acuerdo con lo realizado en el proyecto.

## 1.1. Antecedentes

La Empresa A surgió a partir de la idea de brindar entretenimiento al mundo; conforme esta fue creciendo, la cantidad de proyectos a desarrollar eran cada vez más, lo que como consecuencia causó que el proceso de QC fuera más extenuante. Debido a esto, se exploró la posibilidad de crear un departamento que genere pruebas automatizadas que ayuden a agilizar el proceso de QC de los proyectos, lo que dio nacimiento al departamento de *QA Automation*.

Gracias a esto, se sustituyeron pruebas que en algún momento se realizaban manualmente, sin embargo, este departamento es relativamente nuevo comparado con los demás existentes, como el de Análisis de QA Manual, por lo que no se posee una metodología de QC concreta.

Con base en lo anterior, el tema del presente proyecto nace de la falta de métricas para el monitoreo del aseguramiento de la calidad del *software* y de la falta de una metodología QA más madura, esto debido a que el departamento de Automatización de QA posee procesos para el monitoreo de los resultados de las pruebas realizadas y una herramienta que facilita la ejecución de pruebas automatizadas, sin embargo, dado a que dicha herramienta es poco utilizada y no se poseen indicadores de calidad establecidas, no hay métricas suficientes para el monitoreo apropiado de QA.

Por la falta de datos sobre la calidad del *software*, no se puede tener información detallada como la relación entre la cantidad de fallos encontrados manualmente y los encontrados con pruebas automatizadas, o qué fallos se hubieran encontrado antes de convertirse en un fallo reportado por un cliente.

Con el fin de tener más detalle sobre el contexto en el que se realizará este proyecto, a continuación, se presentan la descripción de la empresa, la descripción del problema y trabajos similares.

### **1.1.1. Descripción de la empresa**

La compañía A nace en Atlanta, Georgia, en el año 1990, de la idea de ofrecer el mejor entretenimiento para el mundo y de manera excepcional. Para el año 2007, la empresa decidió continuar creciendo y expandirse fuera de los Estados Unidos, seleccionando a Costa Rica como su segundo hogar.

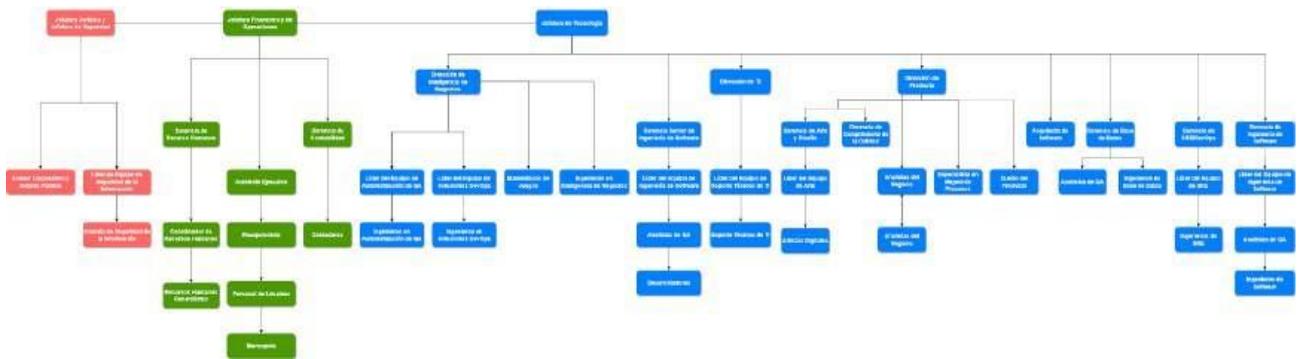
Esta es una empresa dedicada principalmente al diseño, mantenimiento y creación de videojuegos *online*, así como la atención a los clientes que adquieren y mantienen dichos juegos en línea. Este ha crecido a lo largo de los años y se han abierto varias oficinas alrededor del mundo, actualmente en Costa Rica se encuentra ubicada en Heredia y cuenta con más de 180 colaboradores.

El mercado de la Empresa A se enfoca en el mercado asiático, mercado de Filipinas específicamente, mercados con regulaciones especiales (regulados y no regulados por temas de especificaciones), mercado europeo, mercado específicamente del Reino Unido ya que se rige bajo su propio reglamento, y a futuro el mercado de Estados Unidos.

#### **1.1.1.1. Organigrama**

Uno de los pasos para comprender cómo funciona una empresa es observar su estructura interna y cómo se relaciona cada departamento con el resto de la organización. Por esta razón, y para un mayor entendimiento del documento, se presenta a continuación el organigrama de la Empresa A en la Figura 1.

**Figura 1.** Organigrama de la Empresa A.



*Nota.* En este se puede observar de forma jerárquica los puestos que componen a la organización. Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

Al analizar la estructura organizativa de la Empresa A con el fin de comprender cómo funciona su estructura interna, se pudo observar que este consiste principalmente de tres jefaturas principales, o “*Chiefs*” en inglés, los cuales son: jefatura jurídica y jefatura de seguridad o “*Chief Legal and Chief Security Officer*”, jefatura financiera y de operaciones o “*Chief Financial and Operations Officer*” y la jefatura de tecnología o “*Chief Technology Officer*”. Esto se puede observar en la Figura 2.

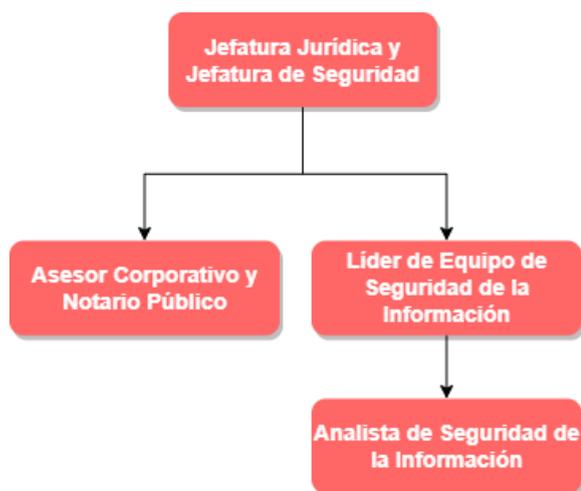
**Figura 2.** Organigrama de la Empresa A enfocado en los 3 puestos de jefatura de la organización.



*Nota.* Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

En cuanto a la parte jurídica y de seguridad, la jefatura jurídica y jefatura de seguridad tiene bajo su cargo estas dos áreas, que consisten en un asesor corporativo y notario público y el equipo de seguridad de la información, dirigido por un líder del equipo de seguridad de la información, que se encarga de analistas de seguridad de la información. Esta estructura organizativa se puede observar con más claridad en la Figura 3.

**Figura 3.** Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura de seguridad.

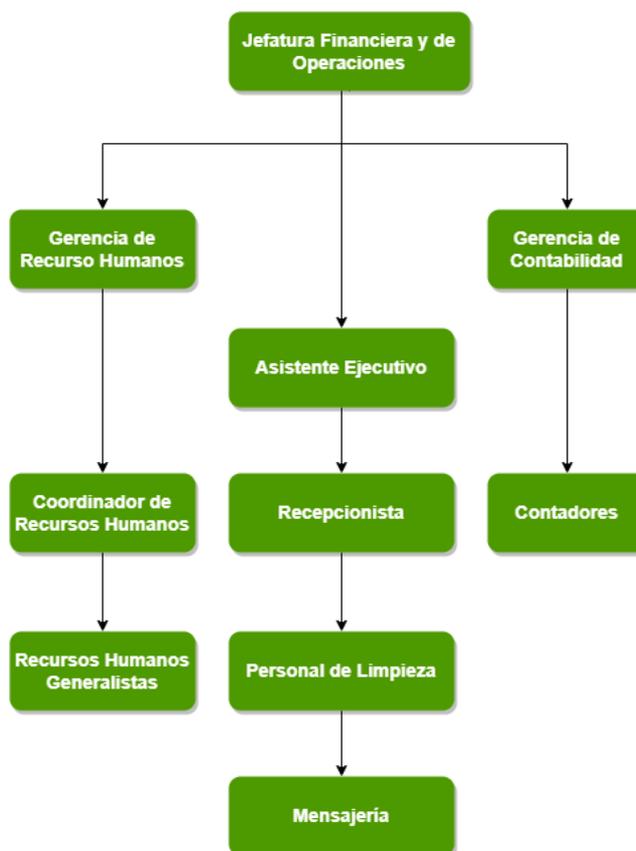


*Nota.* Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

En relación con el área de financiero y operaciones es dirigido por la jefatura financiera y de operaciones. Este tiene bajo su mando la gerencia de recursos humanos, la gerencia de contabilidad y un asistente ejecutivo. En la parte de Recursos Humanos, la gerencia de recursos humanos tiene bajo su cargo la coordinación de recursos humanos y lo que son los recursos humanos generalistas. En la parte de Contabilidad, la gerencia de contabilidad se encarga de dirigir a los contadores que tiene bajo su mando.

Respecto al asistente ejecutivo, este se encarga más en lo que tiene relación a la oficina, por lo que se encarga de dirigir al recepcionista, al personal de limpieza y la mensajería. Todo lo anterior se observa de forma más clara en la Figura 4.

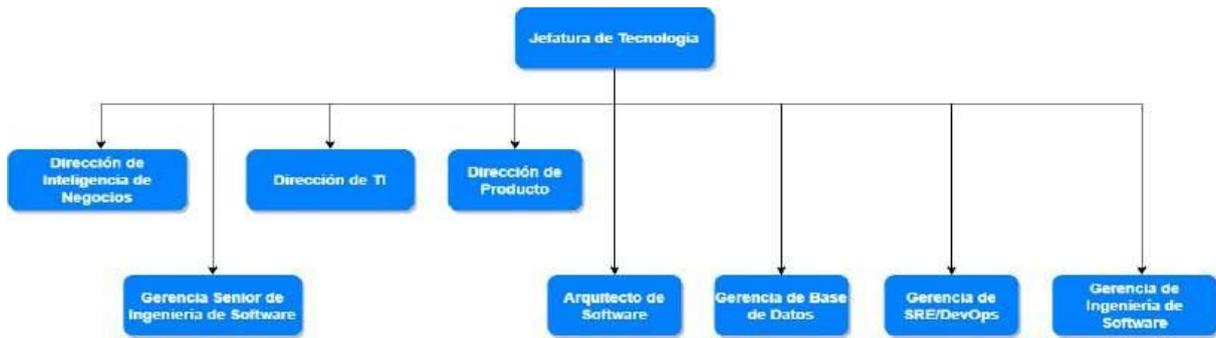
**Figura 4.** Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura financiera y de operaciones.



Nota. Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

El área de Tecnología, dirigida por la jefatura de tecnología, se divide en tres direcciones: dirección de inteligencia de negocios, dirección de TI (Tecnologías de Información) y dirección de producto. Adicionalmente, este tiene bajo su mando al arquitecto de *software* y cuatro gerencias, las cuales son: la gerencia *senior* de ingeniería de *software*, la gerencia de bases de datos, la gerencia de SRE/DevOps (*Site Reliability Engineering*) y la gerencia de ingeniería de *software*. Esto se puede observar en la Figura 5.

**Figura 5.** Organigrama de la Empresa A enfocado en las áreas bajo el mando de la jefatura de tecnología.

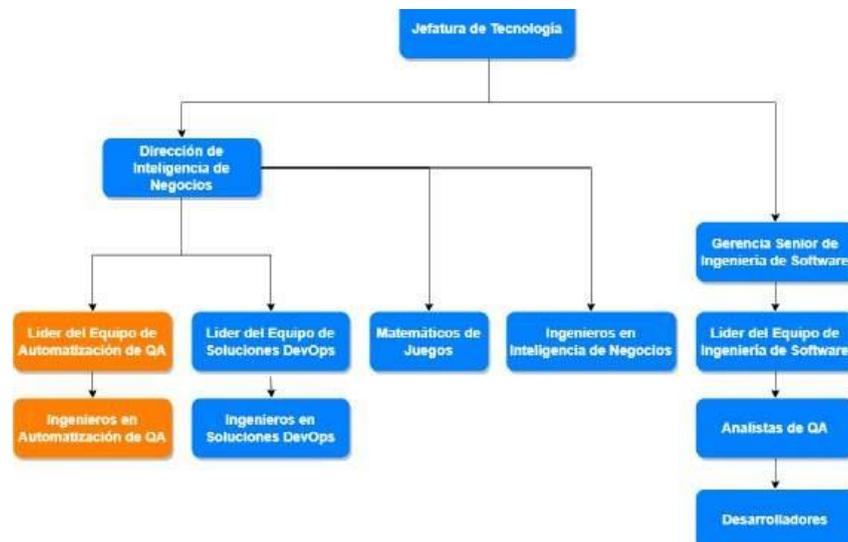


Nota. Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

Lo que es la dirección de inteligencia de negocios, este tiene bajo su cargo varios departamentos: el departamento de automatización de QA (departamento en el que se enfocará el presente proyecto destacado en color anaranjado en el siguiente diagrama), compuesto por un líder del equipo e ingenieros en automatización de QA, el departamento de *Solutions DevOps*, compuesto por un líder del equipo e ingenieros en soluciones *DevOps*, y el departamento de matemáticos de juegos. También, la dirección de inteligencia de negocios dirige lo que son los ingenieros en inteligencia de negocios.

La estructura organizativa descrita se puede observar en la Figura 6.

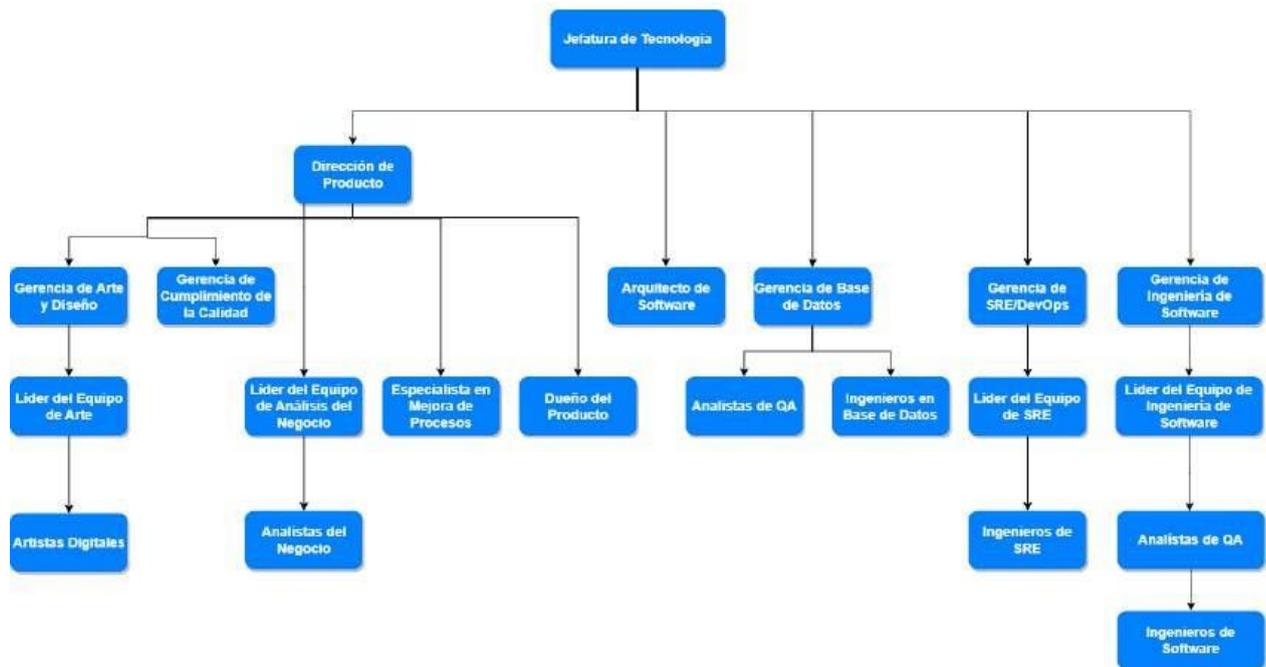
**Figura 6.** Organigrama de la Empresa A enfocado en las áreas bajo el mando de la dirección de inteligencia de negocio, los matemáticos de juegos, los ingenieros en inteligencia de negocio y las áreas bajo el mando de la gerencia senior de ingeniería de software.



Nota. Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

Respecto la dirección de producto, responden directamente a este la gerencia de cumplimiento de la calidad, el departamento de Análisis del Negocio, compuesto por el líder del equipo y analistas del negocio, un especialista en mejora de procesos, los dueños del producto, y el departamento de arte, compuesto por la gerencia de arte y diseño, el líder del equipo y artistas digitales. Esto se puede observar en la Figura 7.

**Figura 7.** Organigrama de la Empresa A enfocado en las áreas bajo el mando de la dirección de producto, el arquitecto de software, las áreas bajo el mando de la gerencia de base de datos, las áreas bajo la gerencia de SRE/DevOps, y las áreas bajo el mando de la gerencia de ingeniería de software.



Nota. Tomado de Recursos Humanos de la Empresa A, agosto 26, 2023.

### 1.1.1.2. Misión

La misión, proporcionada por el departamento de Recursos Humanos, sobre la que se desarrolla la empresa es la siguiente:

“Diseñar, desarrollar, integrar y dar soporte a los mejores productos con las mejores personas” (Recursos Humanos, agosto 26, 2023).

### **1.1.1.3. Visión**

La visión, proporcionada por departamento de Recursos Humanos, sobre la que se rige la empresa es la siguiente:

“Ser la compañía más eficiente de la industria, dando entretenimiento a las personas alrededor del mundo” (Recursos Humanos, agosto 26, 2023).

### **1.1.1.4. Valores**

La Empresa A se caracteriza por los siguientes valores:

- Hacerse cargo: Hacer lo que sea necesario para realizar un trabajo bien hecho.
- Compartir tu pasión: Emocionar e inspirar a otros.
- Trabajar juntos y ganar: Combina los talentos para lograrlo en equipo.
- El mejor lugar para trabajar: Crea diversión, compártela y celebra el éxito.

## **1.1.2. Descripción del problema**

El problema definido a continuación, parte de una oportunidad de mejora identificada en materia de la ejecución y monitoreo de pruebas automatizadas del departamento de Automatización de QA de la Empresa A.

Actualmente, el departamento cuenta con una metodología de QC poco madura; esto significa que cuentan con un proceso para el monitoreo de la ejecución de pruebas y cuentan con herramientas que pueden utilizar empleados de otros departamentos para la ejecución de las pruebas, sin embargo, no se tiene una metodología establecida para la recolección de datos utilizando estas herramientas y los procesos actuales del departamento.

Adicionalmente, debido a la falta de datos, no se poseen métricas o indicadores claves establecidos que permitan realizar un análisis en retrospectiva de proyectos en el departamento de *QA Automation*.

Esto representa una necesidad ya que, aunque se logre cumplir con las expectativas del departamento, el cual es ofrecer pruebas automatizadas para su ejecución y el monitoreo de dichas pruebas para la detección de errores durante el ciclo de desarrollo del proyecto, sin la disposición de indicadores de calidad, no se podría medir de forma más completa la calidad del producto y la eficiencia del equipo, además de la detección de oportunidades de mejora que ayuden a optimizar los procesos de QA.

## **1.2. Planteamiento del problema**

El departamento de Automatización del Aseguramiento de la Calidad (*QA Automation*) es una sección relativamente nueva en la Empresa A con alrededor de 3 años de haberse creado.

Actualmente, este ofrece distintas herramientas que pueden ser utilizadas por otros equipos o departamentos dentro de la organización con el fin de agilizar los procesos de QA, por ejemplo, la ejecución de una prueba de carga o de estrés o la ejecución de pruebas automatizadas, ambas mediante del uso de una aplicación web. No obstante, este posee un proceso de QC poco maduro e informal para lo que es la ejecución y monitoreo de las pruebas, y el seguimiento del rendimiento que tiene el departamento como tal.

Debido a lo anterior, se ve la necesidad de disponer de métricas que permitan el análisis retrospectivo de proyectos, además de poder responder preguntas tales como ¿En qué fase del proyecto se encontró un fallo?, ¿Cuál es el rendimiento del equipo?, entre otras incógnitas de esta misma índole, las cuales se abordarán con más detalle durante el desarrollo del proyecto.

Con base en lo anterior, el presente proyecto busca responder la pregunta ¿Cuál es el proceso que se debe ejecutar para realizar las pruebas y cómo se puede medir el rendimiento y monitorear su ejecución?

### **1.3. Justificación del proyecto**

En esta sección se explicará el por qué es valioso realizar este proyecto, por ejemplo, qué implicaciones pueden tener los resultados, quiénes se verán beneficiados, entre otros.

#### **1.3.1. Innovación**

El departamento de Automatización de QA actualmente no posee una metodología de QC definida y esto dificulta tareas como medir el rendimiento de la ejecución de las pruebas automatizadas, lo que podría ser de mucho provecho si se tuviera información al respecto.

Con lo planteado en este proyecto, se permitiría mejorar el control de la ejecución de las pruebas automatizadas de dicho departamento mediante el establecimiento de métricas, la forma en que estas se pueden calcular y qué tipo de pruebas ejecutar. Con la propuesta de una metodología de QC, se dispondrá de un catálogo de métricas que serán útiles para la mejora y toma de decisiones en el departamento de Automatización de QA.

En materia del tema de este proyecto (propuesta de una metodología de QC para el departamento de *QA Automation* de la Empresa A), en empresas como Ceiba Software & Arts, una empresa costarricense que se dedica al desarrollo de juegos no es común realizar pruebas automatizadas para el aseguramiento de la calidad de este producto, aunque las tecnologías para el desarrollo de juegos dispongan de herramientas para la ejecución de pruebas automatizadas (ver Anexo 2).

Esto significa que el desarrollo de este proyecto sería un gran aporte para esta área ya que ayudaría a la mejora del proceso de QC de empresas de Costa Rica, enfocados en el desarrollo de juegos, que quieran implementar y utilizar pruebas automatizadas.

### 1.3.2. Impacto

La propuesta de una metodología de QC enfocado en la definición de métricas de calidad para el departamento de *QA Automation* tendrá un gran impacto en la Empresa A ya que además de beneficiar al departamento de *QA Automation* con la información que se genere con la aplicación de esta metodología, se puede beneficiar a distintas áreas como el departamento de BI (*Business Intelligence* en inglés) ya que estos tendrían la posibilidad de responder preguntas tales como:

- ¿En qué etapa se detectó un error?
- ¿Cuál es la cantidad de errores detectados por etapa (diseño, desarrollo, entre otros)?
- ¿Nos estamos dando cuenta tarde de los errores?

Las respuestas a preguntas como las anteriores proveería información que ayudaría a los puestos de gerencia y directores a tomar decisiones que ayuden a mejorar los procesos actuales, por ejemplo, la gerencia de cumplimiento de calidad podría determinar si el producto en desarrollo cumple con los estándares de calidad de la Empresa A con el análisis de la información que realizaría el departamento de BI en conjunto con el departamento de *QA Automation* (ver Anexo 1).

Gracias a la disposición de una propuesta de metodología de QC, como la de este proyecto, en la Empresa A, se puede hacer un informe de seguimiento del proyecto (*Project Tracking Report*) que refleje el estado en el que se encuentra lo desarrollado, los riesgos y problemas detectados de forma periódica.

Con respecto al departamento de *QA Automation* como tal, este podría disponer de sus propias métricas de productividad mediante la interpretación de actividades del departamento tales como la recalificación y tabulación de los datos de los defectos reportados y no de trabajo realizado. Asimismo, se podría medir la calidad del trabajo entregado por *release*, una métrica que también ayudaría a realizar comparaciones entre los *bugs* encontrados por cada etapa de desarrollo (definición, desarrollo, integración, regresión, producción) y áreas de trabajo (equipo).

## **1.4. Objetivos**

El detalle de los objetivos tanto general como específicos que se esperan alcanzar en este proyecto, se detallan a continuación.

### **1.4.1. Objetivo general**

Proponer una metodología de QC, enfocado principalmente en la creación de métricas de calidad, que facilite la medición del rendimiento y monitoreo de la ejecución de las pruebas automatizadas del departamento de *QA Automation* de la Empresa A, utilizando las buenas prácticas recomendadas por marcos de referencia como el Syllabus de la ISTQB (*International Software Testing Qualifications Board*).

### **1.4.2. Objetivos específicos**

1. Indagar el contexto actual en el cual se desarrolla la metodología de QC del departamento de *QA Automation* de la Empresa A.
2. Analizar las mejores prácticas aplicables en la metodología de QC para el departamento de *QA Automation* de la Empresa A.

3. Desarrollar una metodología de QC que mida el rendimiento y monitoree la ejecución de las pruebas automatizadas del departamento de *QA Automation* de la Empresa A.
4. Formular un plan de acción para la implementación de las soluciones especificadas para la metodología de QC para el departamento de *QA Automation* de la Empresa A.

## 1.5. Alcance, supuestos y exclusiones

En esta sección se describe el alcance del proyecto en el cual se abarca el alcance, los supuestos y las exclusiones.

### 1.5.1. Alcance

A continuación, se presenta una tabla que detalla los entregables del proyecto junto con los criterios de aceptación correspondientes, los cuales definen el alcance del presente proyecto.

Cabe mencionar que cada uno de los entregables está relacionado a una etapa del proyecto del proyecto, estos se listan en los siguientes puntos.

- **Etapa 1: Recopilación de datos del departamento de *QA Automation*.** En esta etapa se recopila información en relación con el contexto actual del departamento de *QA Automation*
- **Etapa 2: Análisis de mejores prácticas.** Etapa para el análisis y selección de las mejores prácticas aplicables para el proyecto.
- **Etapa 3: Desarrollo de metodología de QC.** Etapa para el desarrollo de una metodología de QC con base en las necesidades del departamento de *QA Automation*.

- **Etapa 4: Formulación de plan de acción.** Etapa de formulación de un plan de acción con base en la metodología de QC desarrollada.

**Tabla 1.** Declaración de Alcance

Entregable Principal 1	Descripción	Criterio de Aceptación
1.1. Análisis del departamento de <i>QA Automation</i>	Entregable de la Etapa I - Se realiza un análisis del proceso actual y prácticas en el cual se desarrolla la metodología de QC del departamento de <i>QA Automation</i> de la Empresa A.	<ul style="list-style-type: none"> <li>• Firma de las minutas de reunión con el detalle de los acuerdos de la etapa.</li> </ul>
Sub-Entregables	Descripción	Criterio de Aceptación
1.1.2. Fase de recopilación de datos	Entregable de la Etapa 1 - Recopilación de datos mediante la aplicación de entrevistas y revisión documental, observación cualitativa y la creación de un mapa mental de necesidades y expectativas.	<ul style="list-style-type: none"> <li>• Firma de las minutas de reunión con el detalle de los acuerdos de la etapa.</li> </ul>
1.1.3. Fase de análisis	Es un entregable de la Etapa 1 - Análisis de los datos recopilados en la fase de recopilación de datos.	<ul style="list-style-type: none"> <li>• Firma de las minutas de reunión con el detalle de los acuerdos de la etapa.</li> </ul>
Entregable Principal 2	Descripción	Criterio de Aceptación
2.1. Mejores prácticas aplicables	Entregable de la Etapa 2 - Se genera un listado de las mejores prácticas aplicables en la metodología de QC, esto en materia del aseguramiento de la calidad del <i>software</i> utilizando pruebas automatizadas, que puedan aportar valor al proyecto.	<ul style="list-style-type: none"> <li>• Lista con mejores prácticas aplicables.</li> </ul>

Sub-Entregables	Descripción	Criterio de Aceptación
2.1.1. Análisis de mejores prácticas	Entregable de la Etapa 2 - Análisis de las mejores prácticas aplicables con base en el contexto actual del departamento de <i>QA Automation</i> y sus necesidades. Para esto se basará en el PMBOK, la ISTQB Syllabus, estándares de la IEEE como la IEEE 829, y estándares de la ISO como la ISO 29119-3.	<ul style="list-style-type: none"> <li>Firma del acta de aceptación del documento con las mejores prácticas.</li> </ul>

2.1.2. Selección de las mejores prácticas	Entregable de la Etapa 2 - Listado con las mejores prácticas seleccionadas según el análisis de la fase 1.2.1.	<ul style="list-style-type: none"> <li>Firma del acta de aceptación del documento con las mejores prácticas.</li> <li>Documentación con las mejores prácticas aplicables según el contexto actual y las necesidades del departamento de <i>QA Automation</i>.</li> </ul>
---	--	--

Entregable Principal 3	Descripción	Criterio de Aceptación
3.1. Desarrollo de metodología de QC	Entregable de la Etapa 3 - Con base en la información recopilada y los hallazgos encontrados, este proyecto tendrá como alcance el desarrollo de una propuesta de una metodología de QC y su implementación se verá a futuro como una oportunidad para continuar con el proyecto.	<ul style="list-style-type: none"> <li>Definición de métricas.</li> <li>Entrega de una propuesta de metodología de QC.</li> <li>Firma del acta de entrega y aceptación de la metodología de QC propuesta.</li> </ul>

Entregable Principal 4	Descripción	Criterio de Aceptación
4.1. Formulación de <i>roadmap</i>	Entregable de la Etapa 4 - Se formulará un plan de acción ( <i>roadmap</i> ) para la implementación de las soluciones especificadas para la metodología de QC para el departamento de <i>QA Automation</i> enfocado en la definición de métricas para medir el rendimiento del departamento y el monitoreo de las pruebas automatizadas.	<ul style="list-style-type: none"> <li>● Entrega de un <i>roadmap</i> de implementación con base al estado actual del departamento de <i>QA Automation</i>.</li> <li>● Firma del acta de entrega y aceptación del plan de acción propuesto.</li> </ul>

*Nota.* Cuadro con detalles sobre los entregables del proyecto y el criterio de aceptación de cada uno de estos.

### 1.5.2. Supuestos

Para el desarrollo de este proyecto se establecen los siguientes supuestos.

- Disponibilidad del departamento de *QA Automation*, así como los conocimientos y experiencia de su personal.
- Existen los vistos buenos correspondientes para el acceso a las plataformas necesarias para la recopilación de datos.
- Acceso a la información que se genera actualmente en el departamento de QA con respecto a los procesos actuales para el aseguramiento de la calidad del *software* tales como métricas, horas invertidas por el equipo de trabajo, repositorio de *bugs*, clasificación de errores, etapa en el que se encontró el error, entre otros.

### 1.5.3. Exclusiones

Para el desarrollo de este proyecto, se establecen las siguientes exclusiones con base en los objetivos específicos del proyecto.

- Ya que según el área existen otros departamentos de QA, el proyecto se enfocará en el contexto en el cual se desarrolla la metodología de QC del departamento de *QA Automation*.
- Las metodologías o mejores prácticas se limitarán a las aplicables en lo que es asegurar la calidad mediante la automatización de pruebas, con la aplicación del PMBOK, IEEE y el Syllabus de la ISTQB. El resto no están dentro del alcance.
- El desarrollo de la metodología de QC no será implementado, sino que se limitará a ser una propuesta como una oportunidad de mejora para el departamento de *QA Automation*.
- Al igual que con el desarrollo de la metodología de QC, la formulación del plan de acción para la implementación de las soluciones específicas para la metodología de QC para el departamento de *QA Automation* se contempla como una oportunidad para continuar con el proyecto en el futuro.

## 2. Marco teórico

Con el fin de obtener información detallada necesaria para este proyecto, en esta sección se proporcionan fundamentos necesarios para entender temas relacionados al aseguramiento de la calidad del *software* (QA) y brindar contexto sobre el tema a abarcar.

En este capítulo, se abordan tanto conceptos como modelos y datos importantes, los cuales permiten establecer conexiones entre el conocimiento existente en el área y el contexto en el que se desenvuelve el proyecto, en este caso, un departamento de automatización del aseguramiento de la calidad de una empresa. Según la información recopilada, se generará una propuesta metodología de QC considerando las necesidades y situación actual del departamento.

### 2.1. Diferencia entre aseguramiento de calidad del software y el control de calidad del software

El aseguramiento de la calidad del *software* y el control de la calidad del software son dos conceptos que pueden llegar a confundirse, sin embargo, estos se centran en aspectos diferentes.

Según Fiorenza I. (2024), el aseguramiento de la calidad tiene un enfoque preventivo para la gestión de la calidad y se centra en la definición de procesos, estándares y procedimientos con el fin de prevenir defectos y problemas de calidad desde el inicio del ciclo de desarrollo del *software*, además, este abarca actividades que implican la elaboración de planes de calidad que establecen objetivos, estrategias, recursos y procesos para el control de calidad durante todo el ciclo de vida del proyecto.

Por otro lado, el control de calidad tiene un enfoque retrospectivo para la gestión de la calidad y se centra en verificar los productos de *software* para identificar defectos y

problemas de calidad después de que se haya desarrollado el *software*. Sus principales actividades implican pruebas, revisiones, inspecciones y verificación de documentos para identificar defectos en el *software* y sus componentes.

## 2.2. Fundamentos del proceso de pruebas

Como parte de las definiciones para entender los puntos a abarcar en este proyecto, hay que conocer sobre los fundamentos del proceso de pruebas, que abarca varios puntos.

El primer punto sería la definición de “prueba de *software*”; este es definido por la ISTQB (2018) como una forma de evaluar la calidad del *software* y de reducir el riesgo de fallo del *software* en la operación. Esto es importante ya que el *software* puede no funcionar apropiadamente y ocasionar consecuencias como la pérdida de dinero, tiempo o reputación de una empresa. Cabe mencionar que las pruebas no solamente consisten en la ejecución de pruebas, sino que también implica la validación, que es verificar si el sistema cumple con las necesidades de los usuarios.

Teniendo conocimiento sobre en qué consisten las pruebas de *software*, es de relevancia tener conocimiento de los objetivos que buscan cumplir. Estos se listan a continuación con base en los propuesto por la ISTQB (2018):

- Evaluar productos de trabajo, tales como requisitos, historias de usuario y código fuente.
- Verificar si se cumplen todos los requisitos especificados.
- Validar si el objeto de prueba está completo y funciona como esperan los usuarios y otras partes interesadas.
- Fomentar la confianza en el nivel de calidad del objeto de prueba.
- Prevenir defectos.

- Encontrar fallos y defectos.
- Proporcionar información suficiente a las partes interesadas que les permitan tomar decisiones informadas, especialmente con respecto al nivel de calidad del objeto de prueba.
- Reducir el nivel de riesgo de calidad inadecuada del *software*, por ejemplo, fallos no detectados previamente que ocurran en la operación.
- Cumplir con los requisitos o normas contractuales, legales o reglamentarios, y/o verificar el cumplimiento del objeto de prueba con dichos requisitos o normas.

Cabe mencionar que los objetivos de las pruebas pueden cambiar dependiendo del contexto del sistema que se esté probando.

Al tener una idea de qué son las pruebas y sus principales objetivos, se puede desarrollar una propuesta de metodología de QC adaptada a las necesidades de la empresa y el equipo de automatización del aseguramiento de la calidad para alcanzar los objetivos descritos.

### **2.3. Metodología de QA**

Como parte de las definiciones relevantes de este proyecto está el concepto de “Metodología de QA”, el cual se explica de distintas formas. Según Kyrylo I. (s.f.), las metodologías de QA ayudan a describir las acciones que toman los equipos para poder organizar y optimizar el proceso de planificación, diseño, seguimiento y optimización del control de calidad. Este también explica que las metodologías para el aseguramiento de la calidad y para el desarrollo usualmente son similares ya que los equipos utilizan enfoques similares para todos los procesos de ingeniería.

Otra definición es de Bose S. (2023), que explica que las metodologías de QA son importantes, ya que definen claramente los requisitos y dan a los evaluadores una comprensión profunda de las funciones y plan para progresar.

Ambas descripciones de lo que es una metodología de QA nos ayuda a comprender lo que se espera proponer en este proyecto para poder ofrecer una oportunidad de mejora en el proceso de QC del departamento de Automatización del Aseguramiento de la Calidad del *Software* de la Empresa A.

## **2.4. Estándares y marcos de referencia para pruebas de software**

En la automatización de pruebas, hay varios estándares y marcos de referencia reconocidos y seguidos por profesionales en la industria del desarrollo de *software* que abarcan los temas relacionados con el aseguramiento de la calidad del *software*. Gracias a la disposición de los estándares y marcos de referencia, se puede contar con guías que nos ayuden a conocer sobre las buenas prácticas utilizadas en la industria para así poder reducir la brecha que existe entre la situación actual del negocio y lo que proponen dichos documentos.

A continuación, se describen los estándares y marcos de trabajo referentes a las pruebas de *software* que se utilizarán como referencia en este proyecto.

### **2.4.1. ISTQB Certified Tester Foundation Level**

La Junta Internacional de Calificaciones de Pruebas de *Software* (2018), autor de la *ISTQB Tester Foundation Level (International Software Testing Qualifications Board)* describe este marco de referencia como una guía que proporciona conocimientos sobre las pruebas esenciales que se pueden poner en práctica, asimismo, explica las terminologías y conceptos que se utilizan en el área de las pruebas de *software*.

A continuación, se detallan las métricas que propone este marco de referencia para el análisis de la calidad de los productos, cómo se gestiona la información de las pruebas y los principales modelos de ciclo de vida de desarrollo de *software* que sugiere.

#### **2.4.1.1. Métricas de QA**

La monitorización de las pruebas es una tarea que forma parte de una metodología de QC; este se encarga de recopilar la información necesaria para así poder brindar retroalimentación y visibilidad sobre el estado de la prueba. Para poder obtener dicha información, es necesario establecer métricas.

Según la *ISTQB Tester Foundation Level (2018)*, las métricas se pueden recopilar durante y al final de las actividades de prueba que se van a evaluar. Las más comunes se listan a continuación:

- Avance contra el cronograma y presupuesto planificados.
- Calidad actual del objeto de prueba.
- Adecuación del enfoque de prueba.
- Eficacia de las actividades de prueba con respecto a los objetivos.

Respecto a las métricas de prueba comunes, dicho marco de referencia plantea:

- Porcentaje de trabajo planificado realizado en la preparación de casos de prueba (o porcentaje de casos de prueba planificados implementados).
- Porcentaje del trabajo planificado realizado en la preparación del entorno de prueba.
- Ejecución de casos de prueba (por ejemplo, número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasan/fallan y/o condiciones de prueba que pasan/fallan).

- Información de defectos (p. ej., densidad de defectos, defectos encontrados y corregidos, tasa de fallos y resultados de las pruebas de confirmación).
- Cobertura de pruebas de requisitos, historias de usuario, criterios de aceptación, riesgos o código.
- Finalización de tareas, asignación y uso de recursos, y esfuerzo.
- El costo de la prueba, incluido el costo comparado con el beneficio de encontrar el próximo defecto o el costo comparado con el beneficio de ejecutar la siguiente prueba.

#### **2.4.1.2. *Gestión de la información de las pruebas***

A la hora de gestionar información de las pruebas, es necesario gestionar la información según su propósito. Al gestionar la información, se logra resumir y comunicar datos de la actividad de las pruebas. La *ISTQB Tester Foundation Level* (2018) propone varios tipos de informes.

Uno de los informes se realiza durante la monitorización y control de las pruebas. Este se emite por el jefe de pruebas y comúnmente incluye los siguientes puntos.

- Estado de las actividades de prueba y el avance en comparación con el plan de prueba.
- Factores que impiden el avance.
- Pruebas programadas para el próximo periodo de información.
- La calidad del objeto de prueba.

Cuando se alcanzan los criterios de salida, en otras palabras, los criterios que determinan si es posible dar por finalizado el ciclo de pruebas, el jefe de pruebas emite un informe con el resumen de pruebas, el cual consiste en un resumen de las pruebas realizadas. Este informe puede incluir los siguientes puntos.

- Resumen de las pruebas realizadas.
- Información sobre lo que ocurrió durante un periodo de prueba.
- Desviación del plan, incluidas las desviaciones en el cronograma, la duración o el esfuerzo de las actividades de prueba.
- Estado de prueba y calidad del producto con respecto a los criterios de salida o definición de hecho.
- Factores que han bloqueado o continúan bloqueando el avance.
- Riesgos residuales.

#### 2.4.1.3. *Modelos de ciclo de vida de desarrollo de software*

La ISTQB *Tester Foundation Level* explica que un modelo de ciclo de vida de desarrollo de *software* describe los tipos de actividad que se realizan en cada etapa de un proyecto, y cómo estas actividades se relacionan entre sí de manera lógica y cronológica.

En el desarrollo de *software*, según la ISTQB (2018), existen diferentes modelos de ciclo de vida, sin embargo, independientemente del modelo que se utilice, las actividades de pruebas deben comenzar en las primeras etapas del ciclo de vida. Según este marco de referencia, los modelos más comunes son: modelo de desarrollo secuencial y modelo de desarrollo iterativo e incremental; estos se detallan a continuación.

- Modelos de desarrollo secuencial: Este modelo describe el proceso de desarrollo de *software* como un flujo de actividades lineal y secuencial, esto significa que cualquier fase en el proceso de desarrollo comienza cuando se complete la fase anterior. Un ejemplo de este modelo es el “Modelo Cascada”, donde las actividades de prueba solo ocurren después de que se hayan completado las actividades de desarrollo.

- Modelos de desarrollo iterativo e incremental: Este modelo implica el establecimiento de requisitos, diseño y prueba de un sistema en partes. Esto significa que el desarrollo del *software* aumenta gradualmente en una serie de ciclos. Un ejemplo de este modelo es el “Modelo V”, el cual integra el proceso de desarrollo implementando el principio de prueba temprana.

#### **2.4.2. ISTQB Advanced Level Syllabus - Test Automation Engineer**

Según la *International Software Testing Qualifications Board* (2016) o ISTQB por sus siglas, el *Syllabus* para un *Test Automation Engineer* (TAE) es un documento que describe las tareas de un ingeniero de automatización de pruebas las cuales consisten en el diseño, desarrollo y mantenimiento de soluciones de automatización de pruebas.

Este marco de referencia se centra en los conceptos, métodos, herramientas y procesos necesarios para la automatización de pruebas funcionales dinámicas, además de la gestión de las pruebas automatizadas, la gestión de la configuración, la gestión de los defectos, los procesos de desarrollo de *software* y el aseguramiento de la calidad.

Según la ISTQB (2016), un ingeniero de automatización de pruebas es alguien que posee un amplio conocimiento de las pruebas en general, y asimismo, conocimiento en el área de la automatización de pruebas. Las tareas más comunes de este son:

- El uso de herramientas de *software* diseñadas para controlar y configurar las condiciones previas de la prueba.
- Ejecución de pruebas.
- Comparación de los resultados reales con los resultados previstos.

Este marco también explica que los objetivos de la automatización de pruebas incluyen:

- Mejora de la eficiencia de las pruebas.
- Proporcionar una cobertura de funciones más amplia.
- Reducir el coste total de la prueba.
- Realizar pruebas que los probadores manuales no pueden.
- Acortar el período de ejecución de la prueba.
- Aumentar la frecuencia de las pruebas/reducir el tiempo necesario para los ciclos de pruebas.

Como parte de las secciones más relevantes en relación con el tema de este proyecto, a continuación, se detallan las métricas que propone este marco de referencia para el análisis de la calidad de los productos y cómo se registran las soluciones de automatización de pruebas y del sistema bajo prueba.

#### ***2.4.2.1. Informes y métricas de automatización de pruebas***

Las métricas y los informes son una herramienta importante a la hora de llevar seguimiento de una tarea; en el caso de la ejecución de pruebas automatizadas las métricas, según la ISTQB (2016), se pueden utilizar para el monitoreo de la estrategia de automatización de pruebas, su efectividad y eficiencia, asimismo, el administrador e ingeniero de automatización de pruebas dar seguimiento del proceso con base en los objetivos establecidos y el impacto que pueden tener los cambios realizados.

Según este marco de referencia, las soluciones de automatización de pruebas se pueden dividir en dos grupos, los cuales se detallan a continuación.

#### **2.4.2.1.1. Métricas Externas**

Estas son aquellas que se utilizan para medir el impacto de las pruebas automatizadas en otras actividades (en particular las actividades de prueba).

Generalmente incluyen las siguientes medidas.

- Beneficio de la automatización.
- Esfuerzo para construir pruebas automatizadas.
- Esfuerzo para analizar incidentes de pruebas automatizadas.
- Esfuerzo por mantener pruebas automatizadas.
- Relación de fallas a defectos.
- Tiempo requerido para la ejecución de las pruebas automatizadas.
- Número de casos de prueba automatizados.
- Número de resultados de aprobación y falla.
- Número de resultados falsos fallidos y falsos aprobados.
- Cobertura de código.

#### **2.4.2.1.2. Métricas Internas**

Estas métricas son aquellas utilizadas para medir la eficacia y eficiencia de las pruebas automatizadas respecto al cumplimiento de sus objetivos. Generalmente incluyen las siguientes medidas.

- Métricas de herramientas de ejecución de código.
- Densidad de defectos del código de automatización.
- Velocidad y eficiencia de los componentes de la solución para la ejecución de pruebas automatizadas.

#### **2.4.2.2. Registro de la solución de automatización de pruebas y del software bajo prueba**

El registro es de relevancia en una solución de automatización de pruebas, ya que es una fuente que se utiliza con frecuencia y puede usarse para analizar problemas potenciales que se puedan presentar. Algunos de los puntos que puede incluir un registro son los siguientes.

- Qué caso de prueba está actualmente en ejecución, incluyendo su hora de inicio y de finalización.
- El estado de ejecución del caso de prueba. Esto debe de estar presente tanto en el archivo de registros, como a través de un panel para que su interpretación sea fácil.
- Detalles del registro de prueba a alto nivel, por ejemplo, los pasos que se ejecutaron, y el tiempo que se tardó.

Respecto al registro del sistema bajo prueba, la ISTQB (2016) propone los siguientes puntos.

- Cuando el sistema bajo prueba identifica un problema, se debe registrar toda la información necesaria para el análisis de un problema, por ejemplo, la fecha y hora en el que se encontró, el mensaje de error, entre otros.
- El sistema bajo prueba puede registrar todas las interacciones del usuario para reproducir un problema si se da.
- La información de la configuración debe registrarse en un archivo. Este contiene información como la versión del *software*, la configuración del sistema que se está probando, entre otros.

Es importante mencionar que toda la información de registro de la solución de pruebas y del sistema bajo prueba debería poder buscarse fácilmente, por ejemplo, utilizando la hora y fecha en la que se realizó el registro.

#### **2.4.2.3. Informes de automatización de pruebas**

Los registros mencionados en la sección anterior, por sí solos, no proporcionan una buena descripción general del resultado obtenido. Es necesario poder disponer de un informe que nos dé un resumen conciso de las pruebas ejecutadas para el sistema bajo prueba.

Los informes, dentro del contexto de la ejecución de pruebas automatizadas, generalmente poseen la siguiente información.

- Un resumen que ofrece una descripción general del resultado de la ejecución.
- El sistema bajo prueba.
- El entorno en que se ejecutaron las pruebas.
- Qué pruebas fallaron y el motivo del error, para facilitar la resolución de problemas.

Los informes que se generen deben de estar disponibles para los interesados en los resultados de la ejecución de las pruebas.

#### **2.4.3. PMBOK**

El PMBOK, siglas de *Project Management Body of Knowledge*; o Cuerpo de Conocimiento de Gestión de Proyectos, en español, es uno de los marcos de referencia más utilizado y conocido. Según Rodrigues (2021), este es un documento creado por el PMI (*Project Management Institute*) que contiene procesos, prácticas recomendadas, terminologías y directrices para una gestión de proyectos exitosa.

El PMBOK se empezó a publicar a partir de los años 80 del siglo pasado como un artículo de divulgación y se transformó en un manual conforme pasó el tiempo, esto gracias a que se alimentó de la experiencia valiosa de otros gestores profesionales.

Este marco de referencia es tan popular, que existe una certificación para los gestores de proyecto que tengan interés en actualizarse, asimismo, algunas empresas ofrecen cursos para poder realizar el examen para adquirir la certificación.

Para efectos de este proyecto, se utilizará la séptima edición del PMBOK.

#### **2.4.3.1. *Objetivo del PMBOK***

El marco metodológico PMBOK es más que una metodología, este comparte buenas prácticas y lineamientos que permiten dar buen seguimiento a los proyectos y a sus etapas; esto según lo que explica Rodrigues (2021).

El PMBOK tiene como objetivo poder ayudar a las personas que quieran iniciar en el área de la gestión de proyectos al explicar los procesos clave que los ayude a adaptarse a la metodología. El mismo autor explica que la edición más reciente de este marco de referencia se enfoca en la generación de valor en los negocios a través de los proyectos y que más allá de los entregables, se busca alcanzar los resultados.

#### **2.4.3.2. *Fases de un proyecto según el PMBOK (sexta edición o anteriores)***

El ciclo de vida de un proyecto se divide en distintas fases para poder tener un inicio y un fin, en lugar de desarrollarse de forma aleatoria. El PMBOK divide en 5 fases el ciclo de vida en ciclo de vida para la gestión de un proyecto. Estos se detallan a continuación.

- **Iniciación:** En esta fase nace el proyecto. Al iniciar un proyecto se define en qué consistirá el proyecto y se expone la justificación de tal manera que la

empresa y los interesados lo entiendan. El proyecto se aprueba para su iniciación cuando se confirma su viabilidad y el su valor.

- **Planificación:** Con la planificación del proyecto los equipos podrán saber qué dirección seguir para obtener los resultados finales, asimismo, se presentan las acciones a tomar para prevenir los escenarios negativos que se puedan dar y se establecen las acciones a realizar para terminar el proyecto en el tiempo y presupuesto estipulado.
- **Ejecución:** En la fase de ejecución de un proyecto, el proyecto se implementa. Para esto los involucrados se reúnen para explicar las responsabilidades, objetivos y cómo la compleción de una tarea puede afectar; con esto se busca que las personas entiendan la importancia de la tarea que deben realizar y se enteren de a quién referirse si algún obstáculo se presenta.
- **Seguimiento y control:** Esta fase o paso, generalmente, se da de forma simultánea con la ejecución. Esto permite poder actuar en el caso de que se presente un problema e ir optimizando o mejorando un proceso. Gracias al seguimiento y control, se puede prever los obstáculos o detectar oportunidades.
- **Cierre:** El cierre ocurre cuando es posible presentar al cliente, o a la directiva interesada, informes, resultados o entregables que se establecieron desde el inicio. En esta etapa los equipos analizan lo desarrollado, cómo se afrontaron los retos y qué aprendieron en el proyecto para poder aplicarlo en futuros proyectos.

#### **2.4.4. Capability Maturity Model Integration (CMMI)**

El CMMI, o *Capability Maturity Model Integration*, según *CMMI Institute* (2010), es un marco de referencia ampliamente utilizado para mejorar los procesos de desarrollo de productos y servicios en organizaciones. Con base en el contexto de este proyecto, se utilizará la sección *Process and Product Quality Assurance* (PPQA), o Garantía de Calidad de Proceso y Productos en español, cuyo propósito es proporcionar al personal y a la gerencia una visión objetiva de los procesos y los productos de trabajo asociados.

La sección *Process and Product Quality Assurance* del CMMI se centra en garantizar la calidad tanto de los procesos como de los productos en todas las etapas del ciclo de vida de desarrollo, asimismo, ayuda respaldar la entrega de los productos con alta calidad al permitir proporcionar al personal y a los gerentes de distintos niveles una visibilidad adecuada, retroalimentación sobre los procesos, y productos asociados durante el ciclo de vida de un proyecto.

La CMMI (2010) explica que, si el aseguramiento de la calidad está integrado en el proceso, quienes realicen actividades de aseguramiento de la calidad deben recibir capacitación en este tema, y menciona que las actividades de aseguramiento de calidad deben comenzar en las primeras fases de un proyecto para establecer planes, procesos, estándares y procedimientos que agregan valor al proyecto y satisfarán los requisitos de este, además de las políticas organizacionales.

En el caso de que se identifique algún problema de incumplimiento, la CMMI (2010) recomienda abordarlo primero en el proyecto de ser posible, si no, se recomienda elevarlo a un nivel apropiado de gestión para su resolución.

El PPQA consiste en las siguientes actividades:

- Evaluar objetivamente los procesos realizados y los productos de trabajo en comparación con la descripción, los estándares, y los procedimientos del proceso aplicables. Este consiste en evaluar regularmente los procesos y productos para identificar áreas de mejora y oportunidades de optimización.
- Identificar y documentar problemas de incumplimiento.
- Proporcionar retroalimentación al personal y a los gerentes de proyectos sobre los resultados de las actividades de garantía de calidad. Esto ayuda a garantizar la transparencia y la retroalimentación adecuada para la mejora continua.
- Garantizar que se aborden los problemas de incumplimiento. Para esto es necesario verificar que los procesos y productos cumplan con los estándares, procedimientos y requisitos definidos.

El PPQA en el CMMI se enfoca en establecer procesos efectivos para poder garantizar la calidad tanto de los procesos como de los productos, esto mediante la planificación, monitoreo, verificación y mejora continua, y como resultado, se lograría alcanzar niveles más altos de madurez y excelencia en la entrega de productos y su nivel de calidad.

Este marco de referencia, específicamente la sección de PPQA, ofrece varias mejores prácticas según las distintas actividades que establece como parte del PPQA, estas ayudan a garantizar que se implementen los procesos planificados; estas se detallan a continuación.

#### **2.4.4.1. Mejores prácticas por objetivo**

PPQA en el CMMI establece las mejores prácticas por objetivos que son necesarias para garantizar que los procesos y productos cumplan con los estándares y requisitos establecidos. Estas se detallan a continuación:

##### **2.4.4.1.1. Evaluar objetivamente los procesos y productos de trabajo**

Este objetivo trata sobre evaluar objetivamente el cumplimiento del proceso realizado y los productos de trabajo asociados con las descripciones de procesos, estándares y procedimientos aplicables (CMMI, 2010).

Este objetivo, según lo propuesto por el *CMMI Institute* (2010), cuenta con varias prácticas específicas. El primero consiste en evaluar objetivamente los procesos de trabajo elegidos; se busca evaluar los procesos realizados seleccionados comparándolos con las descripciones, estándares y procedimientos aplicables.

La segunda práctica específica que propone el *CMMI Institute* (2010) consiste en evaluar objetivamente los productos de trabajo seleccionados en relación con las descripciones, estándares y procedimientos de procesos aplicables, por ejemplo, productos de trabajo como informes de evaluación, informes de incumplimiento, o acciones correctivas.

##### **2.4.4.1.2. Proporcionar una visión objetiva**

Este objetivo consiste en que los problemas de incumplimiento se deben de rastrear y comunicar objetivamente para poder garantizar su resolución.

Este, según lo propuesto por el *CMMI Institute* (2010), cuenta con varias prácticas específicas. La primera práctica es la comunicación y resolución de incumplimientos, el cual trata sobre la comunicación de problemas de calidad y garantizar la resolución de

problemas de incumplimiento con el personal y los gerentes. El incumplimiento se puede identificar gracias a las evaluaciones ya que ayudan a identificar la falta de cumplimiento de normas, descripciones de procesos o procedimientos aplicables.

La segunda práctica que propone el PPQA del CMMI es establecer y mantener registros de las actividades de control de calidad. Un ejemplo de esto es el registro de las actividades de aseguramiento de la calidad del proceso y del producto con suficiente detalle para que se conozca el estado y los resultados.

#### **2.4.5. Series de ISO/IEC/IEEE 29119**

Un estándar de relevancia a utilizar como guía en este proyecto es la serie de la ISO/IEC/IEEE 29119, el cual descrito por la *International Organization for Standardization* (2024), o ISO por sus siglas en inglés, como un estándar cuyo propósito es definir un conjunto de estándares acordados internacionalmente para pruebas de *software* y que puede ser utilizado por cualquier organización al realizar dicho tipo de pruebas.

Esta organización también explica que los estándares que componen la IEEE 29119 se basan en un conjunto de procesos de pruebas que cubren las pruebas a nivel organizacional, de gestión y de pruebas dinámicas. Actualmente esta serie está compuesta por 5 partes que ayudan a proporcionar una guía integral para todas las etapas del proceso de pruebas de *software*, desde la planificación hasta la ejecución y la gestión. A continuación, se presenta un breve resumen sobre dichas partes a modo de contexto.

##### **2.4.5.1. Parte 1: Conceptos y definiciones (IEEE 29119-1)**

Dado que la serie del estándar IEEE 29119, tiene como propósito definir un conjunto de estándares acordados internacionalmente para pruebas de *software* que puedan ser utilizados por cualquier organización. La parte 1 de este estándar, IEEE

29119-1, introduce conceptos generales, esto según la *International Organization for Standardization* (2022).

Esta parte establece una base común de comprensión al definir y explicar los términos y conceptos fundamentales utilizados en el campo de las pruebas de *software*. Incluye definiciones de conceptos como caso de prueba, plan de prueba, estrategia de pruebas, artefactos de pruebas, entre otros. Proporciona una terminología estandarizada para facilitar la comunicación y la comprensión entre los profesionales de pruebas.

A continuación, se lista los conceptos que abarcar esta parte de la serie:

- Descripción del papel de las pruebas de *software* en la gestión de la calidad, y como parte de la verificación y validación.
- Se define la implementación de las pruebas, tanto estáticas como dinámicas.
- Se explica la impracticabilidad de las pruebas exhaustivas y la necesidad de realizar muestreos.
- Se describe la importancia de la base de pruebas y el oráculo de prueba.
- Se presentan los beneficios de la independencia de las pruebas.
- Se describen los planes de pruebas y las estrategias dentro del contexto de las pruebas basadas en riesgos.
- Se describen los niveles de prueba, los tipos de prueba y las técnicas de diseño de prueba (y sus medidas correspondientes).
- Se presentan marcos de prueba, métricas de prueba, documentación de prueba, gestión de la configuración y soporte de herramientas.
- Se describe el desempeño del diseño y ejecución de pruebas basado en el uso de un modelo de prueba.

#### **2.4.5.2. Parte 2: Procesos de prueba (IEEE 29119-2)**

Al tener conocimiento de los conceptos y el vocabulario que respalda esta serie de estándares gracias a la IEEE 29119-1, podemos tener un mejor entendimiento de los temas que se abordan en la segunda parte.

Con base en lo que explica la *International Organization for Standardization* (2013), la parte 2 de la IEEE 29119-2 sigue un enfoque de prueba basado en riesgos. Este enfoque ayuda al desarrollo de estrategias y la gestión de las pruebas, ya que permite priorizar las pruebas y centrarse en las características y atributos de calidad más importantes.

En la IEEE 29119-2, según la ISO (2013), describe en detalle los procesos involucrados en la planificación, diseño, ejecución y evaluación de pruebas de *software*. Esto incluye la identificación de requisitos de pruebas, la elaboración de estrategias y planes de pruebas, la especificación de casos de prueba, la ejecución de pruebas, la evaluación de los resultados y la gestión de defectos. Proporciona pautas paso a paso para llevar a cabo un proceso de pruebas completo y sistemático.

#### **2.4.5.3. Parte 3: Documentación de pruebas (IEEE 29119-3)**

La parte 3 de la IEEE 29119 es la documentación de pruebas. Según la *International Organization for Standardization* (2021), esta serie del estándar define las plantillas y ejemplos de documentación de pruebas que se producen durante el proceso de prueba.

La ISO, en esta parte de la serie, se enfoca en crear y mantener la documentación relacionada con las actividades de pruebas de *software*, incluyendo la elaboración de planes de pruebas, informes de resultados, matrices de trazabilidad, casos de prueba, entre otros documentos. Esto tiene como fin garantizar que la documentación de pruebas sea

clara, completa y coherente, lo que facilita la comprensión y el seguimiento del progreso de las pruebas.

#### **2.4.5.4. Parte 4: Técnicas de prueba (IEEE 29119-4)**

La parte 4 de la serie del estándar IEEE 29119 (IEEE 29119-4) habla sobre las técnicas de diseño de pruebas que se pueden utilizar para derivar casos de prueba que generen evidencia de que se han cumplido los requisitos del elemento bajo prueba o que muestre que hay defectos presentes a la hora de ejecutar la prueba.

La *International Organization for Standardization* (2021), las técnicas de diseño de prueba de *software* que se definen en la parte 4 de la IEEE 29119 se pueden utilizar dentro de cualquier ciclo de vida y para cualquier producto.

Según lo presentado por la ISO, esta parte de la serie detalla una variedad de técnicas y enfoques para diseñar y ejecutar pruebas de *software* efectivas. Esto puede incluir técnicas de diseño de casos de prueba, como partición de equivalencia, análisis de valores límite, pruebas de decisión, pruebas de estado, entre otras. Proporciona orientación sobre cómo seleccionar y aplicar las técnicas de pruebas más apropiadas para maximizar la cobertura y la detección de defectos en el *software*.

#### **2.4.5.5. Parte 5: Gestión de prueba (IEEE 29119-5)**

Según la *International Organization for Standardization* (2016), la parte 5 de la serie de IEEE 29119, la IEEE 29119-5, explica los conceptos principales y la aplicación de las pruebas basadas en palabras claves (*Keyword-Driven Testing*<sup>1</sup>). Además, define los atributos de los marcos diseñados para admitir pruebas pasadas en palabras claves.

---

<sup>1</sup> Pruebas utilizando casos de prueba compuestos a partir de palabras clave.

La IEEE 29119-5 describe una implementación específica del diseño de pruebas y los procesos de implementación detallados en la parte 2 de la serie (IEEE 29119-2), en particular derivar casos de pruebas, ensamblar conjuntos de pruebas y derivar procedimientos de prueba.

Esta parte de la serie asume la aplicación de la parte 4 de la misma serie (IEEE 29119-4) para el diseño de los casos de prueba que se van a describir con base en lo propuesto en esta parte (IEEE 29119-5).

La IEEE 29119-5, en síntesis, se centra en la gestión y supervisión del proceso de pruebas de *software*. Esto incluye la gestión de recursos de pruebas, como personal, herramientas y entornos de pruebas, la gestión de la configuración de pruebas, la gestión de la calidad de las pruebas y la mejora continua del proceso de pruebas, asimismo, proporciona directrices para asegurar que el proceso de pruebas se realice de manera eficiente, dentro del presupuesto y con los niveles de calidad esperados.

### **3. Marco metodológico**

En esta sección se presentan las metodologías utilizadas en este proyecto, basado en los entregables establecidos y la matriz de congruencia que se puede observar en la Tabla 1.

#### **3.1. Matriz de congruencia**

A continuación, se presenta una matriz de congruencia como parte de la definición de la metodología que se siguió para alcanzar los objetivos indicados.

**Tabla 2.** Matriz de Congruencia

<i>Problema</i>	<i>Objetivos Específicos</i>	<i>Unidad de Análisis</i>	<i>Variables</i>	<i>Conceptualización de la variable</i>	<i>Operacionalización (Instrumentación)</i>
¿Cómo se puede medir el rendimiento y monitorear la ejecución de las pruebas automatizadas para tener una mejor visibilidad de la calidad del producto según las métricas de calidad de la empresa?	Indagar el contexto actual en el cual se desarrolla la metodología de QC del departamento de <i>QA Automation</i> de la Empresa A.	Departamento de <i>QA Automation</i>  Gerencia de Cumplimiento de la Calidad	Equipo de trabajo	Colaboradores del departamento de <i>QA Automation</i> .	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● Observación cualitativa</li> </ul>
			Expectativas	Estas son las expectativas que tienen el líder del departamento de <i>QA Automation</i> , la dirección de inteligencia del negocio y la gerencia de cumplimiento de la calidad respecto a este proyecto.	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● Observación cualitativa</li> <li>● Mapa mental de necesidades y expectativas</li> </ul>
			Necesidades	Estas son las necesidades que presenta el equipo de <i>QA Automation</i> con base en la metodología de QC actual que utilizan.	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● Observación cualitativa</li> <li>● Mapa mental de necesidades y expectativas</li> </ul>

		Limitaciones	Aspectos de proyecto que no podrán cubrirse o que son inalcanzables.	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● Observación cualitativa</li> <li>● Mapa mental de necesidades y expectativas</li> </ul>
Analizar las mejores prácticas aplicables en la metodología de QC para el departamento de <i>QA Automation</i> de la Empresa A.	PMBOK ISTQB <i>Syllabus</i> IEEE 829 ISO 29119-3	Metodologías o marcos de referencia	Documentación en la que se basará el proyecto para su desarrollo. La metodología, según la RAE (2023), es un conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.	<ul style="list-style-type: none"> <li>● Revisión documental</li> </ul>
Desarrollar una metodología de QC que mida el rendimiento y monitoree la ejecución de las pruebas automatizadas del departamento de <i>QA Automation</i> de la Empresa A.	Departamento de <i>QA Automation</i> Dirección de Inteligencia de Negocios	Métricas o indicadores de calidad	Son los instrumentos utilizados para medir y evaluar la calidad de los productos y servicios. Los indicadores de calidad son una medida estandarizada, cuyo objetivo es evaluar el desempeño de productos, servicios, procesos y personas en una empresa (Silva, 2022).	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> </ul>

		Desafíos	Estas son las complicaciones que pueden aparecer a la hora de desarrollar el proyecto.	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> </ul>
<p>Formular un plan de acción para la implementación de las soluciones especificadas para la metodología de QC para el departamento de <i>QA Automation</i> de la Empresa A.</p>	<p>Departamento de <i>QA Automation</i></p> <p>Dirección de Inteligencia de Negocios</p> <p>Gerencia de Cumplimiento de la Calidad</p>	<p><i>Roadmap</i> o plan de acción</p>	<p>Plan de acción que sirve como guía para la implementación de la metodología del proyecto y detalla los pasos a seguir para alcanzarla. Según Martins (2022), este también se define como una vista en perspectiva de alto nivel de todas las entregas, los logros clave y los objetivos en general del proyecto.</p>	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● <i>Roadmap</i></li> </ul>
		<p>Tiempo propuesto para la ejecución del plan de acción</p>	<p>Esta es la estimación de los tiempos para la ejecución de cada tarea establecida en el plan de acción.</p>	<ul style="list-style-type: none"> <li>● Entrevista</li> <li>● Revisión documental</li> <li>● <i>Roadmap</i></li> </ul>

*Nota. Matriz de congruencia del proyecto en el cual abarca el problema, los objetivos específicos, la unidad de análisis, las variables, la conceptualización de las variables, y la operacionalización (instrumentación).*

## 3.2. Tipo de investigación

Dado que el objetivo de este proyecto es desarrollar una propuesta de una metodología de QC para el equipo de *QA Automation* de la Empresa A, enfocado principalmente en la creación de métricas de calidad, que facilite la medición del rendimiento y monitoreo de la ejecución de las pruebas automatizadas, utilizando las buenas prácticas recomendadas por normas como el PMBOK (*Project Management Body of Knowledge*) y el Syllabus de la ISTQB (*International Software Testing Qualifications Board*), se seleccionó el enfoque cualitativo como tipo de investigación a utilizar.

La elección del enfoque cualitativo para este proyecto se basa en lo establecido por Hernández, Fernández, y Baptista (2010), quienes señalan que este tipo de enfoque utiliza la recolección de datos sin medición numérica para descubrir o afinar preguntas de una investigación.

Las principales características del enfoque cualitativo que posee este proyecto, según Hernández et al, (2010, p. 7), son las siguientes.

1. Se desarrolla una teoría coherente con los datos, de acuerdo con lo que se observa (denominada “teoría fundamentada”). Se basan más en una lógica y proceso inductivo (explorar y describir, y luego generar perspectivas teóricas).
2. El enfoque se basa en métodos de recolección de datos *no* estandarizados ni completamente predeterminados. No se efectúa una medición numérica, por lo cual el análisis no es estadístico. La recolección de los datos consiste en obtener las perspectivas y puntos de vista de los participantes (sus emociones, prioridades, experiencias, significados y otros aspectos subjetivos).

3. El investigador cualitativo utiliza técnicas para recolectar datos, como la observación no estructurada, entrevistas abiertas, revisión de documentos, discusión en grupo, evaluación de experiencias personales, registro de historias de vida, e interacción e introspección con grupos o comunidades.
4. El enfoque cualitativo evalúa el desarrollo natural de los sucesos, es decir, no hay manipulación ni estimulación con respecto a la realidad.
5. El investigador se introduce en las experiencias de los participantes y construye el conocimiento, siempre consciente de que es parte del fenómeno estudiado.

Para el desarrollo de este proyecto es necesario la aplicación de las características detalladas anteriormente. En el caso del punto 1, este se evidencia a la hora de tener que realizar una observación cualitativa del proceso de QC que realiza actualmente el equipo de *QA Automation*.

El punto número 2 se aplica en el desarrollo del proyecto ya que los datos que se vayan a obtener y generar no tienen la necesidad de ser medidos, asimismo, dichos datos se recolectarán mediante la aplicación de preguntas que ayuden a recabar información relacionada al tema. El punto número 3 menciona herramientas que se utilizarán para la recolección de los datos necesario para el desarrollo del proyecto.

Respecto al punto 4 y 5, para este proyecto será necesario evaluar el desarrollo natural de los sucesos sin intervenir y se tendrá que vivir la experiencia de los participantes, esto con el fin de tener un mejor entendimiento del contexto en el que se desarrolla el proyecto.

### 3.3. Población

Conforme el proyecto se desarrolla, este dependerá de diferentes personas y grupos que tienen interés y les causa un impacto en el proyecto. Se les aplicarán los instrumentos y las técnicas de recopilación de información para poner información relevante para el proyecto.

A continuación, se detalla la información de dicha población, así como una descripción de su rol en el presente proyecto de graduación.

- Líder del Equipo de *QA Automation*: Su función respecto al proyecto es brindar la orientación necesaria respecto a las necesidades del proyecto y procurará brindar todos los conocimientos necesarios.
- Colaboradores del equipo de *QA Automation*: Este sector de la Empresa A, compuesto por profesionales con distintos grados de experiencia, será parte de una serie de actividades, tales como encuestas o entrevistas necesarias para el desarrollo del proyecto.
- Dirección de BI: Dado que este es una de las personas a las que le es relevante las métricas, en este caso las de QA, este podrá proveer comentarios y/o oportunidades de mejora al proyecto.
- Gerencia de cumplimiento de la calidad: Dado que este se encarga de gestionar todo lo relacionado a la evaluación de la calidad de cada *release*, su rol es de relevancia en este proyecto ya que se le solicitaría comentarios u oportunidades de mejora con respecto a lo desarrollado en el proyecto. Este también tendrá participación en las encuestas o entrevistas a realizar.

- Dueños de empresas de juegos en Costa Rica y expertos en Aseguramiento de la Calidad del *Software*: Ya que estos laboran en un contexto similar a la empresa del presente proyecto, su función es brindar información que se pueda utilizar como fuente de información para este proyecto; esto se realizará mediante la aplicación de una encuesta.

### **3.4. Fuentes de información**

Las fuentes de información de este proyecto son varios, estos se detallan a continuación.

- Normas como el PMBOK (*Project Management Body of Knowledge*) y el Syllabus de la ISTQB (*International Software Testing Qualifications Board*), los cuales serán analizados con el fin de saber cuál se adapta mejor a las necesidades y expectativas del proyecto.
- Se tomarán en consideración las guías internas del Departamento de *QA Automation*, la Dirección de Inteligencia de Negocios, y la Gerencia de Cumplimiento de la Calidad.
- Se incluye cualquier entrevista, encuesta o documento adicional que sea de elaboración propia.

### **3.5. Instrumentos y técnicas de recopilación de información**

Las técnicas por emplear para la recopilación de información para su utilización en el desarrollo del proyecto se detallan a continuación.

- Entrevista: Este se utilizará con el fin de poder recabar los datos necesarios para el desarrollo del proyecto tales como las expectativas y necesidades.

- Revisión documental: Esto se realizará a la hora de analizar ya sean documentos de la organización que son necesarios para el desarrollo del proyecto o a la hora de identificar y analizar investigaciones de trabajos similares o normas que se puedan utilizar para el desarrollo del proyecto.
- Observación cualitativa: Esta técnica se utilizará para tener un mejor conocimiento de las tareas que se realizan en el día a día del departamento de *QA Automation* y cómo estos trabajan actualmente, esto con el fin de conocer el estado actual del equipo.

## 4. Análisis de resultados

Como parte del desarrollo de este proyecto, es necesario la recolección de datos, esto con el uso de instrumentos como entrevistas o encuestas, los cuales están detallados en la sección de “Marco Metodológico” de este documento.

La intención de esta sección es proveer los insumos necesarios para conocer el entorno en el que se desenvuelve el proyecto para, posteriormente, generar una propuesta de metodología que se considere las necesidades y requerimientos del departamento de *QA Automation*, y que a su vez permita encontrar oportunidades de mejora y aporte el mayor valor posible.

La información recopilada fue extraída de distintas formas; se realizó una encuesta al equipo de automatización del aseguramiento de la calidad de la Empresa A por medio de un formulario digital que busca conocer las necesidades y estado actual del departamento con base en lo que hayan respondido los miembros de equipo. Otra actividad realizada fue la ejecución de entrevistas a expertos en el tema del aseguramiento de la calidad con distintos perfiles y experiencia. Las encuestas utilizadas para la obtención de la información utilizada en esta sección se pueden consultar en el Anexo del documento.

### 4.1. Aspectos demográficos

Para la recolección de datos a utilizar en este proyecto, se solicitó la colaboración de personas con distintos perfiles.

- La primera fuente son los miembros del equipo de *QA Automation*, ya que estos permiten conocer la metodología actual que se utiliza.

La segunda fuente es la Gerencia de Cumplimiento de la Calidad, el cual brindó información sobre el proceso actual de QA, de liberación del producto, entre otros datos de relevancia de la empresa.

La tercera fuente es la colaboración de distintos expertos en el tema del aseguramiento de la calidad del *software*, quienes ayudan a conocer las mejores prácticas que se usan hoy día en la industria.

#### 4.1.1. Personas entrevistadas

Con relación a las entrevistas realizadas para la recopilación de datos, se realizaron varias reuniones con las siguientes personas.

**Tabla 3.** *Personas entrevistadas*

<i>Nombre</i>	<i>Puesto</i>	<i>Empresa en donde labora</i>
Álvaro Rivera	CEO	ncubo Ideas
Ana Sanabria	<i>Compliance Specialist</i>	Infinite
Equipo de <i>QA Automation</i>	<i>QA Automation Engineer</i>	Empresa A
Ignacio Díaz Oreiro	Docente	Universidad de Costa Rica
Ignacio Trejos Zelaya	Docente	Cenfotec
Laura Chavarría	Máster en Administración de la Tecnología de la Información	<i>Freelancer</i>
Luis Sanabria	Ingeniero en Seguridad Informática	Millicom
Marlen Treviño	Docente	Tecnológico de Costa Rica
Mario Jiménez	Director de TI	Newfire Global Partners
Rodolfo Mora	CEO	Ceiba Software & Arts
██████████	Gerencia de Cumplimiento de la Calidad	Empresa A

*Nota. Personas a las que se entrevistaron o aplicaron una encuesta para la obtención de datos para el desarrollo del proyecto.*

Estos expertos poseen experiencia y conocimientos en relación con el aseguramiento de la calidad de *software* que se utiliza en la industria, lo que nos brinda una guía para el desarrollo de una propuesta de metodología de QC.

## **4.2. Estado actual del departamento de QA Automation**

Parte importante de este proyecto es tener una comprensión clara del estado actual del departamento de *QA Automation* de la Empresa A ya que, al poder entender su metodología actual, las métricas que se utilizan actualmente, entre otros, se podrá detectar la brecha existente entre las mejores prácticas de distintos marcos de referencia.

Para lograr conocer el estado actual de dicho departamento, se utilizó la información recopilada por medio de la aplicación de encuestas al equipo de *QA Automation* (ver Anexo 4), en esta sección se presenta el estado actual del departamento. A continuación, se presentará, en las siguientes subsecciones: la metodología actual, las métricas que se utilizan actualmente, las necesidades y expectativas que se tienen, y la importancia del uso de métricas en el proceso del departamento de automatización del aseguramiento de la calidad.

Para conocer en detalle cómo se reportan los *bugs* y el proceso de liberación de *software*, se entrevistó a la Gerencia de Cumplimiento de la Calidad. Dicha información también se detalla en las siguientes subsecciones.

### **4.2.1. Metodología actual del departamento de QA Automation**

Al aplicar la encuesta adjunta en el Anexo 4 del documento, se pudo identificar distintas secciones de la metodología de QC actual. A modo de contexto, el departamento de *QA Automation* realiza distintas tareas, las cuales hemos clasificado en “tareas frecuentes” y “tareas principales”.

Cabe mencionar que la metodología que se describe a continuación, específicamente la sección 4.2.1.4, la sección 4.2.1.5, y la sección 4.2.1.6 describen la aplicación de las pruebas automatizadas en el proceso de liberación de un producto, esto debido a que es este es el proceso en el que se ejercita con más frecuencia y sirve como caso base para la aplicación de la metodología de QC propuesta. Adicionalmente a las pruebas utilizadas para evaluar el producto a liberar, se menciona el set de pruebas utilizado para evaluar el estado de un producto conforme se van añadiendo nuevas funcionalidades o juegos (pruebas de humo).

#### **4.2.1.1. Tareas principales y recurrentes**

Las tareas principales consisten en las asignaciones de mayor relevancia que realiza el departamento de *QA Automation*, o sea, las actividades que pueden brindar valor y afectar a otros departamentos de la empresa. Estas se listan a continuación.

- Análisis e identificación de procesos, relacionados al aseguramiento de la calidad del producto, que se pueden automatizar.
- Desarrollo de herramientas que ayuden a facilitar la validación del producto.
- Desarrollo e implementación de procedimientos para la automatización de pruebas, para elevar la calidad en la entrega de productos desarrollados por los departamentos de la empresa.
- Transmitir el conocimiento necesario a los colaboradores de la empresa para que hagan uso de las herramientas que provee el departamento de automatización del aseguramiento de la calidad del *software*.
- Análisis y migración de pruebas manuales repetitivas y de contenido invariable en un período prolongado.

- Análisis y detección de fallos en funcionalidades mediante la ejecución de pruebas automatizadas.

El departamento de *QA Automation* también cuenta con actividades recurrentes que son parte de las funciones que debe cumplir. Este se encarga de la ejecución de pruebas de caja negra enfocadas en la interfaz gráfica del producto, pruebas de carga, pruebas de producto, y la ejecución de pruebas de regresión de los APIs, sistemas de administración y gestión de juegos.

#### **4.2.1.2. Desarrollo de pruebas automatizadas**

Para el desarrollo de las pruebas automatizadas, el departamento de *QA Automation* utiliza las tecnologías Selenium junto con Node.js como herramienta de prueba o herramienta para la ejecución de las pruebas. Al desarrollar y agregar nuevas pruebas automatizadas, se verifican los pasos que realizan y se validan según la prueba manual que reemplazará para verificar que revise los resultados esperados.

Las pruebas que necesitan mantenimiento, por ejemplo, por cambios en el caso de prueba manual, y la elección de pruebas manuales que se pueden desarrollar, se manejan con base en la metodología ágil *Scrum*. La planificación se realiza en unidades de trabajo (*sprint*) de cada 30 días y se asignan puntos a las tareas para estimar el esfuerzo que se le tiene que invertir. Estos se listan a continuación:

- **1 punto:** Medio día para realizar la tarea.
- **2 puntos:** 1 día para realizar la tarea.
- **3 puntos:** 1 día o 1 día y medio para realizar la tarea.
- **5 puntos:** 2 a 3 días para realizar la tarea.
- **8 puntos:** Una semana de tiempo para realizar la tarea.

#### **4.2.1.3. Enfoque del equipo**

Respecto al enfoque que posee el equipo de automatización del aseguramiento de la calidad a la hora de revisar los reportes con los resultados de la ejecución de pruebas de integración y regresión, este consiste en los siguientes puntos.

- Las pruebas automatizadas ejecutadas deben de terminar correctamente, o sea, este se tuvo que haber ejecutado y haber mostrado un resultado apropiado (ya sea un fallo o un resultado exitoso) y no un fallo relacionado al desarrollo inapropiado de la prueba.
- Qué inconsistencias se presentan en los fallos que se encuentran con los resultados de la ejecución de las pruebas automatizadas.
- Cantidad de fallos reportados después de la ejecución del conjunto de pruebas automatizadas.
- Revisión de las secciones más utilizadas o críticas del producto.

#### **4.2.1.4. Proceso de preparación de los ambientes de pruebas**

El proceso de preparación de los ambientes de prueba, en este caso los ambientes para las pruebas de regresión y las pruebas de integración, se da en periodos de un mes, eso debido a que, durante el mes, el producto se encuentra en desarrollo.

Con base en la información proveída por la Gerencia de Cumplimiento de la Calidad (ver Anexo 8), este proceso inicia en la fecha de *Code Freeze*, el cual consiste en que todos los equipos envían todos los cambios realizados al *branch* del repositorio del *release* a liberar. En este momento, el equipo de *Solutions* toma el paquete con el producto desarrollado y realiza el debido proceso para la generación de los binarios necesarios, empaquetar todo y aplicar a los ambientes de integración por ser el primer tipo de pruebas a ejecutar durante el proceso de QC.

Cabe mencionar que los ingenieros en calidad del *software* no necesitan realizar ninguna acción relacionada a la preparación del ambiente de pruebas, estos solo poseen permisos para la manipulación de los ambientes de prueba con base en el nivel de experiencia que tienen en la empresa, en este caso *senior*, para investigar y poder hacer un análisis forense de ser necesario. Lo usual es que, aunque sea algo simple, el equipo de *Solutions* instale los ambientes y se encargue de la resolución de problemas de estos.

A diferencia del ambiente para la ejecución de las pruebas de integración, el ambiente para las pruebas de regresión se instala después de la ejecución de las pruebas de integración, el cual tiene una duración de 1 o 2 días.

Cuando el ambiente de integración o regresión tiene instalado el producto a liberar, las pruebas automatizadas se ejecutarán automáticamente.

#### **4.2.1.5. Proceso de QC**

Una vez que se realiza el *Code Freeze* de lo desarrollado, el equipo de *Solutions* de la empresa genera el paquete con todo lo que se va a liberar en el respectivo periodo y aplica el paquete en los ambientes de prueba, en este caso, los ambientes para las pruebas de integración.

Al aplicar el paquete a liberar en los ambientes de prueba, el conjunto de pruebas automatizadas preestablecidas se ejecuta de forma automática. Las pruebas se ejecutan utilizando una herramienta desarrollada por el departamento de *QA Automation*. Este se encarga de encolar y ejecutar el conjunto de pruebas seleccionado, posteriormente, una vez finalizada la ejecución de las pruebas, se hace la revisión de los resultados.

Durante este proceso, los miembros del equipo de *QA Automation* se encargan de revisar los resultados generados por las pruebas, en otras palabras, se trata de identificar y clasificar los fallos que se encuentren en el producto. De ser necesario, con base en el

nivel de experiencia del miembro del equipo, este puede tener acceso a los servidores en donde se aplicó el paquete a revisar para poder realizar un análisis forense de lo que está sucediendo.

Al finalizar la ejecución de las pruebas de integración, el equipo de *Solution* generará otro paquete con los parches o arreglos realizados y los instala en los ambientes de prueba de regresión para realizar el mismo proceso de QC descrito.

El conjunto de pruebas a ejecutar puede variar dependiendo del enfoque y el tipo de prueba que se vaya a ejecutar, por ejemplo, pruebas dedicadas a verificar la estabilidad de un juego, probar el sistema de administración, o realizar pruebas de regresión antes de la liberación de un producto para asegurar que este alcanza los objetivos de calidad de la empresa.

Dicho conjunto de pruebas proviene de un amplio catálogo de pruebas que han automatizado el equipo de automatización del aseguramiento de la calidad de la empresa. Este catálogo cuenta con 345 pruebas distintas que pueden utilizarse ya sea para ejecutar un grupo de pruebas en específico, pero enfocando en una sección o funcionalidad específica del producto, o en este caso, para realizar pruebas de regresión, integración, y/o pruebas de humo. A continuación, se detallan las pruebas que se ejecutan en el proceso de QC.

#### **4.2.1.5.1. Pruebas de integración**

Actualmente, las pruebas de integración incluyen 140 pruebas automatizadas. Estas pruebas están diseñadas para evaluar aspectos críticos del sistema y que los diferentes componentes del producto interactúan bien entre sí; dichas pruebas incluyendo:

- Funcionalidades especiales de los juegos.
- Verificación de la carga y jugabilidad del juego.

- Gestión de jugadores a través del módulo de administración.
- Generación y revisión de reportes de los juegos.
- Configuración y personalización de los juegos.
- Operaciones especiales relacionadas con los juegos.

La ejecución de las pruebas de integración puede durar una hora; esto varía según la cantidad de juegos que se deseen revisar antes de su liberación ya que, al contar con una cantidad amplia de juegos a revisar, se tendrá que cargar y jugar cada uno de estos. Otro factor que puede afectar el tiempo es el estado del ambiente de prueba esto debido a que puede haber secciones o componentes que no carguen o que tarden mucho en cargar.

#### **4.2.1.5.2. Pruebas de regresión**

El conjunto de pruebas de regresión consta de 179 casos automatizados, diseñados para garantizar la estabilidad y funcionalidad del sistema tras cualquier cambio o actualización. Estas pruebas se enfocan en evaluar los siguientes aspectos:

- Integridad y funcionamiento de las APIs.
- Verificación de la carga y jugabilidad del juego.
- Funcionalidades especiales de los juegos.
- Gestión de jugadores en la plataforma.
- Generación y precisión de los reportes.
- Configuraciones avanzadas y personalizadas.
- Operaciones del sistema de administración.

La ejecución de las pruebas de regresión puede durar una hora y media; esto también varía según la cantidad de juegos que se deseen revisar antes de su liberación ya que, al contar con una cantidad amplia de juegos a revisar, se tendrá que cargar y jugar cada uno de estos.

Al igual que con las pruebas de integración, otro factor que puede afectar el tiempo es el estado del ambiente de prueba ya que puede haber secciones o componentes que no carguen o que tarden mucho en cargar, asimismo, en las pruebas de regresión el tiempo es más extenso porque abarca una mayor cantidad de pruebas que revisan los productos a profundidad.

#### **4.2.1.5.3. Pruebas de humo**

Las pruebas de humo o *smoke tests* cuentan con un total de 27 pruebas automatizadas, diseñadas para verificar la estabilidad básica de los sistemas (sistema de administración, los juegos, y el *lobby* de los juegos) y garantizar que los componentes críticos funcionen correctamente. Estas pruebas se dividen en:

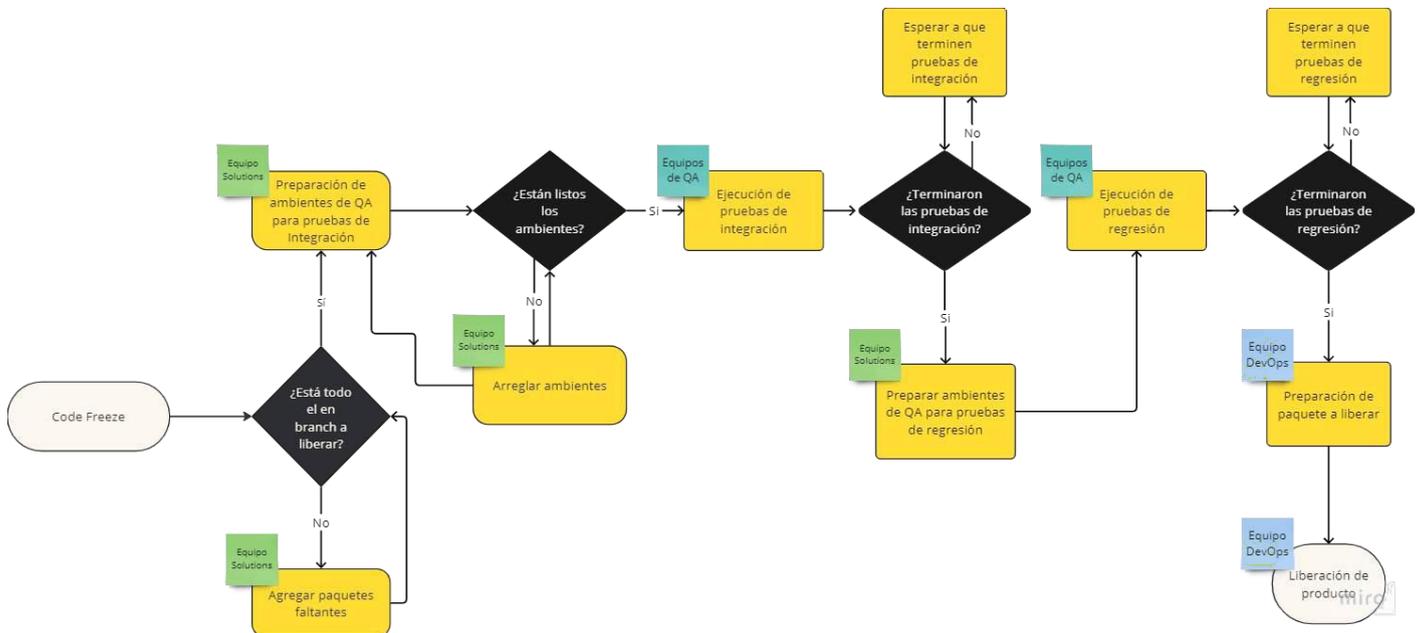
- *Backend smoke tests* (2 pruebas): Evalúan la integridad de la base de datos, el servidor, y los sistemas de reportes.
- *Client smoke tests* (12 pruebas): Estas se enfocan en verificar la funcionalidad del juego.

Las pruebas de humo constituyen un conjunto esencial de pruebas seleccionadas para asegurar que las funciones principales operen de manera adecuada. Estas pruebas son críticas y deben superar cada ejecución; cualquier falla indica un problema significativo en el sistema. Los aspectos evaluados incluyen:

- Verificación de la carga y accesibilidad de los juegos.
- Jugabilidad básica de los juegos.
- Autenticación en el sistema de administración.
- Acceso y autenticación en el lobby de los juegos.

En función de lo explicado y detallado en las subsecciones de esta sección, se presenta a continuación una gráfica que ilustra y resume visualmente la información discutida.

**Figura 8.** Diagrama de flujo del proceso actual de QC para la liberación de un producto



#### 4.2.1.6. Reporte de fallos

Cuando se presenta un fallo en una prueba, se debe valorar si existen fallos externos o algún agente que pueda afectar el resultado de la prueba, por ejemplo, el entorno en el que se realizó, el tiempo de respuesta, los componentes, entre otros. Los fallos se reportan de la siguiente manera.

- Si estos son de ambiente o son errores propiamente del producto, estos se reportan a la Gerencia de Cumplimiento de la Calidad y se genera un *ticket* utilizando la herramienta Jira. A dichos *tickets* se les añade:
  - Fase del proceso de QC en el que se encontró el error
  - Severidad
  - Pasos para reproducir el fallo.
  - Descripción del fallo.

- Ambiente en el que se ejecutó la prueba.
- Si estos son errores de la prueba ejecutada, se analiza cuál es la razón y se reporta mediante la generación de un *ticket* en Jira en el proyecto del equipo de *QA Automation*. Dichos *tickets* contienen una descripción sobre el error encontrado para su análisis y solución.
- Si el fallo encontrado es del producto en liberación, este se reporta en el proyecto que se está probando.

A la hora de generar un *bug* en *Jira*, la prioridad es actualmente asignada por la Gerencia de Cumplimiento de la Calidad o por un *QA Senior*; usualmente se utilizan los valores “*low*”, “*minor*”, “*major*”, o “*critical*” para describir la prioridad que tienen.

Para poder decidir qué prioridad hay que asignarle se utiliza una guía que posee todas las reglas que ayudan a categorizar el fallo.

Según lo explicado por la Gerencia de Cumplimiento de la Calidad, en esta organización la prioridad va de la mano con la severidad del fallo que se haya encontrado, esto debido a que la prioridad es qué tan rápido quiero que se resuelva el fallo y la severidad es el impacto que está teniendo el cliente.

Para la asignación de la prioridad de los fallos encontrados, se utiliza la siguiente guía proveída por la Gerencia de Cumplimiento de la Calidad.

**Tabla 4.** Guía para la asignación de prioridad a los fallos encontrados

<i>Prioridad</i>	<i>Descripción</i>	<i>Fallo durante pruebas</i>	<i>Fallo en producción</i>	<i>Ejemplo</i>
<i>Emergency</i> (Emergencia)	El problema bloqueará el progreso.	<ul style="list-style-type: none"> <li>- Funcionalidad principal inutilizable</li> <li>- Las pruebas no pueden continuar</li> <li>- El producto no puede ser entregado</li> </ul>	<ul style="list-style-type: none"> <li>- El jugador/operador no puede utilizar la función.</li> <li>- Será un DEFCON.</li> </ul>	<ul style="list-style-type: none"> <li>- La aplicación falla.</li> <li>- Cálculos incorrectos.</li> </ul>
<i>Critical</i> (Crítico)	Grave problema que podría bloquear el progreso.	<ul style="list-style-type: none"> <li>- Funcionalidad principal fallando</li> <li>- Casos de prueba positivos que fallan</li> <li>- Se necesita una solución alternativa para continuar con las pruebas.</li> <li>- El producto no puede ser entregado.</li> </ul>	<ul style="list-style-type: none"> <li>- La funcionalidad principal no hace lo que se espera que haga.</li> <li>- Usar esta función será difícil o inestable.</li> <li>- Probablemente sería un DEFCON.</li> </ul>	<ul style="list-style-type: none"> <li>- El juego se detiene esporádicamente por situaciones inusuales.</li> <li>- Mensajes de error, que podrían omitirse.</li> <li>- Se muestran valores incorrectos.</li> </ul>
<i>Major</i> (Importante)	Tiene el potencial de afectar el progreso.	<ul style="list-style-type: none"> <li>- La prueba negativa de la funcionalidad principal está fallando.</li> <li>- La funcionalidad secundaria falla.</li> <li>- Se necesita una solución alternativa para continuar con las pruebas.</li> <li>- El producto</li> </ul>	<ul style="list-style-type: none"> <li>- Alta afectación a la funcionalidad del operador o jugador fuera del embudo de valor.</li> <li>- Generará <i>tickets</i> fuera de MCA y escalaciones para el equipo de desarrollo.</li> </ul>	<ul style="list-style-type: none"> <li>- Mensajes erróneos o confusos.</li> <li>- Los componentes de la página están mal asignados, no es posible leerlos ni hacer click en ellos.</li> </ul>

no debe entregarse.

<i>Medium</i> (Medio)	Problemas menores o defectos que se pueden solucionar fácilmente.	<ul style="list-style-type: none"> <li>- Menor afectación a la funcionalidad del operador/jugador que no está en el embudo de valor.</li> <li>- El producto no está listo para ser entregado. Si es así, los defectos se solucionarían en caliente.</li> </ul>	<ul style="list-style-type: none"> <li>- Menor afectación a la funcionalidad del operador o jugador que no está en el embudo de valor.</li> <li>- Probablemente no haya un impacto negativo en la experiencia del jugador/operador al utilizar el producto.</li> </ul>	<ul style="list-style-type: none"> <li>- La animación del juego no coincide con los eventos reales.</li> <li>- Los componentes de la página están mal asignados, son difíciles de leer o de hacer click.</li> </ul>
<i>Low</i> (Bajo)	Problema trivial con poco o ningún impacto en el progreso o la usabilidad.	<ul style="list-style-type: none"> <li>- Es bueno tenerlo arreglado.</li> <li>- Problema cosmético o mejora.</li> <li>- El producto se puede entregar con el acuerdo del propietario del producto.</li> </ul>	<ul style="list-style-type: none"> <li>- Rara vez o difícil de tener por los jugadores.</li> <li>- Impacto bajo o nulo para los usuarios finales.</li> <li>- No afectará negativamente la experiencia del jugador/operador al usar el producto.</li> </ul>	<ul style="list-style-type: none"> <li>- Defecto cosmético.</li> <li>- Artefactos.</li> <li>- Los componentes de la página están mal asignados, pero son fáciles de leer o de hacer click.</li> </ul>

*Nota.* Guía para la asignación de prioridad a los fallos encontrados con la descripción de cada tipo de prioridad, características de estas, ya sea que se haya encontrado el fallo durante la ejecución de pruebas o en producción, y ejemplos.

Las prioridades descritas en la tabla anterior son influenciadas por la severidad, esto debido a que, por ejemplo, si el impacto es muy alto, la prioridad también va a ser alta.

#### **4.2.2. Procedimiento para la liberación del software**

El proceso para la liberación del *software* se da en periodos de 1 mes. Durante el mes, los equipos de desarrollo se encuentran generando el producto y realizando todas las tareas asignadas hasta la fecha de *Code Freeze*.

Al llegar al día de *Code Freeze*, todo lo desarrollado debe de estar en el *branch* principal del repositorio del cual se va a generar el paquete para el producto a liberar. Dicho paquete lo produce el equipo de *Solutions* y es aplicado a los ambientes de integración y regresión para el proceso de *testing* (pruebas de regresión e integración).

Después de que el *software* pasó por el proceso de *testing* y se han reportado los defectos encontrados, se evalúa si los *bugs* encontrados se pueden arreglar de una vez o si se delega para que sea arreglado como un *hotfix* posteriormente dependiendo del nivel de impacto que tenga, esto debido a que la regeneración del paquete a liberar podría hacer más ruido de lo necesario en el proceso de liberación, por ejemplo, la regeneración del paquete a liberar puede desestabilizar el proceso de liberación y causar ruido al detener el proceso de QC al realizar un cambio repentino en el producto que se está probando.

Una vez que el paquete está listo, este mismo paquete se mueve a una carpeta; esta carpeta es utilizada por el equipo de DevOps para instalar el producto desarrollado en los ambientes de producción. Dicho proceso utilizado por el equipo de DevOps es el mismo utilizado por el equipo de *Solutions* para la instalación de los paquetes a liberar en los ambientes de prueba.

### 4.2.3. Sistemas de métricas presentes en la metodología de QC actual

Según la información obtenida con una encuesta a los miembros del departamento de automatización de QA, se encontraron las siguientes métricas utilizadas por el departamento.

- Resultados de las pruebas: Este se refiere al reporte generado después de la ejecución de una prueba en donde se muestra si la prueba terminó de forma exitosa o si este falló.
- Cantidad de pruebas: Este se refiere a la cantidad de pruebas ejecutadas.

### 4.2.4. Necesidades detectadas

Al entrevistar a los miembros del departamento de *QA Automation*, se pudo detectar varias necesidades en el equipo por la falta de una metodología de QC formal, los cuales se enumeran a continuación.

- Tiempos de entrega.
- Cantidad de *bugs* encontrados por *release*.
- Cómo utilizar correctamente los resultados de las métricas para optimizar procesos.
- Medir la calidad del producto entregado.
- Recurrencia de los problemas, o sea, qué tantas veces se comete el mismo error.

Asimismo, los miembros del departamento expresan que sería de utilidad poder categorizar las pruebas. Estos sugirieron las siguientes categorías:

- Pruebas de regresión.
- Pruebas de integración.

- Pruebas de humo (*smoke test*).
- Pruebas de API.
- Pruebas del sistema de administración.
- Pruebas del sistema que gestiona los juegos.
- Pruebas para juegos en específico.

Otro punto de relevancia es el poder clasificar los *bugs* o fallos que se encuentren a la hora de ejecutar un conjunto de pruebas utilizando la herramienta desarrollada por el departamento. A continuación, se listan las categorías que, según los miembros del equipo, serían de más utilidad para el análisis de datos.

- Por *bugs*:
  - *Bugs que son parte del producto que se liberará.*
  - *Bugs de hotfix.*
  - *Bugs de pruebas que se vuelven a ejecutar después de que se arregló un bug reportado (retesting).*
  - *Bugs encontrados en producción.*
- Por componente:
  - Sistema de administración.
  - Sistema para la gestión de los juegos.
  - Un juego en específico.
- Por etapa:
  - Etapa de desarrollo.
  - Etapa de pruebas de integración.
  - Etapa de pruebas de regresión.
  - En producción.

#### 4.2.5. Expectativas detectadas

Al hablar con los miembros del departamento de automatización del aseguramiento de la calidad, se consultó qué información les sería de relevancia a la hora de analizar el reporte de resultados de las ejecuciones de las pruebas. Con base en las respuestas, se plantearon varias expectativas que se desea alcanzar con la disponibilidad de una metodología formal de QA.

Se busca poder conocer el estado de las actividades de prueba y el avance en comparación con el plan de prueba a ejecutar, esto para poder llevar seguimiento del estado de la ejecución de las pruebas durante una prueba de regresión o una prueba de integración.

Asimismo, se busca conocer las pruebas que se podrían correr en el próximo periodo en que se libere el producto. Al poder tener conocimiento de qué pruebas podrían cubrir la revisión de un componente o qué fallos se hubieran podido capturar por medio de la ejecución de una prueba, se puede armar un conjunto de pruebas para la próxima ejecución.

Otro punto importante que monitorear son los factores que se pueden presentar, durante el proceso de QC con la ejecución de pruebas automatizadas, que impidan su avance. Esto es información de relevancia que se debe hacerse conocer y documentar para poder darles solución o prevenirlas en futuras ejecuciones.

Desde el contexto del departamento de *QA Automation*, es valioso tener conocimiento del estado actual de la herramienta utilizada para la ejecución de pruebas; es por esto que se espera poder analizar y recopilar información sobre este con el objetivo de poder atender todos los fallos que este pueda presentar y poder mantener las pruebas automatizadas lo más actualizadas posible.

Todos estos datos se pueden utilizar para la generación de un informe sobre la ejecución de las pruebas para presentar con el fin de informar a las personas interesadas sobre los resultados obtenidos.

#### **4.2.6. Importancia de disponer de métricas**

Uno de los enfoques principales de la metodología de QC a proponer es la generación de métricas para el departamento de *QA Automation*. Es por esto que se preguntó a los miembros del departamento qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados o no ejecutados, la cantidad de pruebas fallidas o exitosas, entre otros.

Con base en las respuestas obtenidas, se pudo notar el impacto e importancia que tendría el poder disponer de métricas para su análisis. Estos expresan que el beneficio sería alto, ya que permitiría analizar el estado actual de todos los procesos automáticos que posee el departamento, además, esto permitiría generar un informe de QA más robusto y completo que muestre el nivel de calidad del producto probado.

Una métrica que ha sido identificada como significativa consiste en el conteo de errores detectados durante las pruebas de integración y regresión, así como la distinción entre los fallos atribuibles a la herramienta de ejecución de pruebas. Esta métrica es de suma relevancia, ya que permite demostrar el valor que la herramienta aporta a la empresa. Además, nos brinda una guía clara sobre qué aspectos deben recibir prioridad, convirtiéndola en un componente fundamental del proceso de pruebas.

Es deseable disponer del conteo de errores detectados durante las pruebas de integración y regresión por su capacidad para evaluar la eficacia de las pruebas automatizadas en la detección temprana de problemas antes de la fase de liberación del producto.

Una métrica adicional que destaca la necesidad de contar con métricas pertinentes es la tasa de falsos positivos. Conocer este dato resulta fundamental para optimizar el proceso de desarrollo de pruebas y garantizar la idoneidad de nuestras pruebas. Este análisis nos permite revisar minuciosamente las pruebas existentes y ajustarlas para evitar la generación de resultados incorrectos, contribuyendo así a la mejora continua del proceso de prueba. Además, la medición de la tasa de falsos positivos ofrece una evaluación de la confiabilidad del *framework* utilizado en las pruebas automatizadas.

## 5. Propuesta de solución

Como resultado del análisis exhaustivo de marcos de referencia aplicables y las entrevistas realizadas con los expertos del área, se propone el desarrollo de una metodología de QC basada en las mejores prácticas. En esta sección se presenta una herramienta con las mejores prácticas seleccionadas y un plan de acción detallado para su implementación en el departamento de *QA Automation*.

Se ha incluido un acta de aceptación del documento (ver Anexo 9), la cual valida su contenido y formaliza la aprobación de la propuesta. Esta acta garantiza que las mejores prácticas identificadas son aplicables al contexto actual del departamento y que el plan de acción planteado es adecuado para su implementación exitosa.

### 5.1. Roles en metodología de QC

Dado a que actualmente se utiliza la metodología ágil “*Scrum*” con el fin de poder responder a los cambios que puedan surgir al *software* que se necesita probar mediante las pruebas automatizadas, los roles para esta metodología se basan en los utilizados por *Scrum*. Estos se detallan a continuación.

- ***Product owner***: Este rol es el representante de los clientes que usan el *software*. En este contexto, los clientes consisten en los líderes y/o gerencias de los diferentes departamentos que componen a la empresa, y el *software* a utilizar son las herramientas que ofrece el departamento de *QA Automation* para la ejecución de casos de prueba de forma automatizada y pruebas de carga.

- **Líder del equipo de *QA Automation*:** Este es la persona que se encarga de liderar el equipo y se asegura de que se cumplan las reglas y que se siga el proceso establecido por la metodología de QC.
- **Equipo de desarrollo de pruebas:** El equipo de desarrollo de pruebas consiste en los ingenieros en automatización de QA que componen el equipo de *QA Automation*. Estos se encargan de convertir la lista de requerimientos en funcionalidades del *software*.

## 5.2. Planificación de la ejecución y desarrollo de pruebas

La planificación de la ejecución y desarrollo de pruebas, al igual que la sección sobre los roles de la metodología, se basa en el marco de referencia *Scrum*, esto también con el fin de poder adaptarse a las necesidades y requerimientos que vayan a liberar los equipos de desarrolladores, por ejemplo, la adición de una nueva funcionalidad o el cambio de enfoque y prioridad sobre las tareas que se están trabajando actualmente.

El departamento de *QA Automation* seguirá utilizando una unidad de trabajo (*sprint*) de cada 30 días para trabajar en las tareas asignadas para cada miembro del equipo durante ese periodo. Este tiempo es ideal ya que brinda tiempo para trabajar en las asignaciones y no se pierde el enfoque rápidamente si se utilizara una unidad de trabajo menor.

Al finalizar los 30 días para trabajar en las tareas, se realizará la planificación y desarrollo del conjunto de pruebas para las pruebas de integración y pruebas de regresión del siguiente *sprint*. Con base en la explicación sobre el proceso de QC brindado por la Gerencia de Cumplimiento de la Calidad, se sugiere que un representante del equipo de *QA Automation* participe en la reunión para la definición de la estrategia de prueba por

proyecto; esta estrategia de prueba es generada con base en una plantilla definida por la Gerencia de Cumplimiento de la Calidad con base en el marco de referencia ISTQB.

Dicha reunión tiene como fin definir y entender la estrategia de prueba que se aplicará en los proyectos, asimismo, cuenta con la participación de personas de diferentes equipos para poder recopilar diferentes puntos de vista de distintos revisores para así evitar que estos se sesguen.

La Gerencia de Cumplimiento de la Calidad también explicó que se planea agregar una sección dentro del documento de estrategia de prueba que abarque las pruebas automatizadas necesarias (ver Anexo 5). Esta sugerencia brinda grandes oportunidades y beneficios ya que se puede dar a conocer las herramientas que ofrece el departamento de *QA Automation*, se conoce las pruebas necesarias a ejecutar para un proyecto, y se puede ahorrar tiempo con la ejecución de pruebas automatizadas para que los analistas de QA puedan invertir el tiempo en la evaluación de otros casos de prueba.

Haciendo una revisión del formato que sigue el documento de dicha estrategia de pruebas para los distintos productos, se pudo observar que este posee información de valor para el departamento de *QA Automation*. Con esto se puede conocer los objetivos de prueba y definir mejor las pruebas a utilizar para poder alcanzar los objetivos planteados, asimismo, este describe las características a las que hay que ponerle atención a la hora de su revisión.

Otra sección de relevancia del documento con la estrategia de prueba es la sección con las pruebas de integración y las pruebas de regresión que se plantean realizar. Con esta información se puede seleccionar y/o generar un conjunto de pruebas ideal que abarque la mayoría de las pruebas a ejecutar.

## 5.3. Ejecución de las pruebas

Una vez que se ha generado el plan de pruebas de los proyectos, está la ejecución de las pruebas automatizadas. Esto, dentro del contexto del departamento de automatización del aseguramiento de la calidad de la Empresa A, se enfocará principalmente en la ejecución de las pruebas de integración, las pruebas de regresión de cada *sprint*, y las pruebas de humo (*smoke tests*). La ejecución de dichas pruebas se detalla a continuación.

### 5.3.1. Pruebas de Integración

Las pruebas de integración se centran en las interacciones entre componentes o sistemas y brindan confianza en que los cambios no han afectado a las interfaces, componentes o sistemas existentes.

La ejecución de estas pruebas se divide en dos fases, las cuales son la fase de preparación del ambiente y la fase de la ejecución de las pruebas. Estas se detallan a continuación.

#### 5.3.1.1. *Fase 1: Preparación del ambiente de pruebas*

La primera fase es la preparación de los ambientes de prueba con el “*release candidato*” a validar, el cual es realizado por otro departamento; en esta fase se hace “*merge*” de los “*hotfixes*” y nuevas funcionalidades desarrolladas en los ambientes de prueba.

#### 5.3.1.2. *Fase 2: Ejecución de las pruebas*

Una vez que se termina de instalar los elementos a probar, el conjunto de pruebas para las pruebas de integración se ejecuta automáticamente utilizando la herramienta *Teamcity*. Dado que ya existe un catálogo amplio de pruebas existente, se propone utilizar

el mismo conjunto de pruebas seleccionadas para la fase de ejecución de las pruebas de integración del proceso de QC para la liberación de un producto. Este conjunto puede modificarse de ser necesario, por ejemplo, que se necesita probar una parte del producto con más profundidad debido a un cambio que se ha implementado.

### **5.3.2. Pruebas de Regresión**

Una vez que se valida la interacción entre componentes o sistemas con la ejecución de pruebas de integración, se ejecutan las pruebas de regresión con el fin de asegurar que los cambios o mejoras realizadas no hayan introducido errores o impactado de forma negativa en las funciones existentes, además de garantizar la estabilidad del producto.

La ejecución de estas pruebas se divide en dos fases, las cuales son la fase de preparación del ambiente y la fase de la ejecución de las pruebas. Estas se detallan a continuación.

#### **5.3.2.1. Fase 1: Preparación del ambiente de pruebas**

En esta fase, al igual que con las pruebas de integración, se hace “merge” de los “hotfixes” que se hayan realizado durante las pruebas de integración.

#### **5.3.2.2. Fase 2: Ejecución de las pruebas**

Una vez que se instalan todos los “hotfixes” y elementos a probar, el conjunto de pruebas de regresión se ejecuta de forma automática utilizando la herramienta *Teamcity*. Al igual que con las pruebas de integración, se propone utilizar las pruebas existentes del catálogo de pruebas existente para la fase de ejecución de las pruebas de regresión del proceso de QC para la liberación de un producto. Este conjunto puede modificarse de ser

necesario, por ejemplo, que se necesita probar una parte del producto con más profundidad debido a un cambio que se ha implementado.

### **5.3.3. Pruebas de Humo (smoke tests)**

Las pruebas de humo, a comparación con las pruebas de integración y de regresión, se ejecutan diariamente de forma automática utilizando la herramienta *Teamcity* en los ambientes de prueba más estables. El conjunto de pruebas seleccionados, como el ambiente de prueba, son los más estables y menos propensos a sufrir cambios, para comprobar el funcionamiento básico del producto. Este conjunto de pruebas se mantendrá de momento.

## **5.4. Priorización de las pruebas**

La priorización de las pruebas a ejecutar es una tarea importante dentro de una metodología de QC. Según la ISTQB (2024), se sugiere que una vez que los casos de prueba se hayan automatizado, o desarrollado, y se tenga un conjunto de pruebas, se pueden organizar estos conjuntos de prueba en un programa de ejecución de pruebas que defina el orden en el que se ejecutarán.

Con base en lo anterior, se sugiere priorizar los casos de prueba automatizados con priorización basada en cobertura, según el contexto del negocio y las estrategias de priorización sugeridas por el marco de referencia mencionado.

Esta estrategia de priorización consiste en ordenar la ejecución de las pruebas con base en su cobertura, con esta estrategia los casos de prueba que logra mayor cobertura se ejecutan primero y cada caso de prueba posterior es el que logra la mayor cobertura adicional. Un ejemplo de esto, presente en la empresa en la que se desarrolla el proyecto, es la ejecución de una prueba que se encarga de revisar que sean accesibles y que carguen

bien todas las secciones del sistema de administración; esto permite revisar la estabilidad del sistema, y su configuración, antes de revisar funcionalidades más específicas.

Con base en lo propuesto por Browserstack (2024), una plataforma basada en la nube que se utiliza para probar sitios web y aplicaciones móviles en diferentes navegadores y sistemas operativos, se sugiere utilizar los siguientes niveles de prioridad.

- **Prioridad 1:** Estos son los casos de prueba que deben ejecutarse y abarca los casos de prueba críticos en los que las posibilidades de que una funcionalidad se vea afectada por las nuevas funcionalidades implementadas.
- **Prioridad 2:** Estos son los casos de prueba no muy críticos que podrían ejecutarse con suficiente tiempo y sirven para realizar una doble verificación antes de liberar el producto.
- **Prioridad 3:** Los casos de prueba con este nivel de prioridad no son importantes para probar antes de la versión actual del producto. Estos se pueden probar después de que la nueva versión del producto se haya liberado.
- **Prioridad 4:** Estos abarca los casos de prueba que tienen un impacto insignificante.

## **5.5. Monitorización y control de las pruebas**

El propósito de la monitorización de las pruebas es poder recopilar información para analizar, además, dar retroalimentación y visibilidad sobre las actividades de prueba realizadas.

Según la ISTQB, la información que se debe monitorear se puede recopilar de forma manual o automática, en ese caso, los datos son recopilados de forma automatizada utilizando la herramienta para la administración de proyectos Jira. Esta información se

debe utilizar para la evaluación del avance de la ejecución de las pruebas y medir si se cumplen los criterios de salida de las pruebas.

Con base en las necesidades, expectativas, y situación actual del departamento de *QA Automation*, se proponen las siguientes métricas.

**Tabla 5.** Métricas propuestas para la metodología de *QC*

<i>Categoría</i>	<i>Métrica</i>	<i>Descripción</i>	<i>¿Se calcula actualmente?</i>	<i>Fórmula</i>
Durante la ejecución de pruebas	Número de casos de prueba exitosos	Esta es la cantidad de casos de prueba que han terminado exitosamente en la ejecución de un conjunto de pruebas.	Sí. Este se calcula actualmente utilizando Jira y Teamcity para la ejecución de las pruebas automatizadas y la generación del reporte con los resultados de las ejecuciones.  Este, según la ISTQB-CT-TAE, es información que forma parte de un reporte de la ejecución de pruebas automatizadas.	a = Total de pruebas ejecutadas  b = Cantidad de pruebas fallidas  c = Cantidad de pruebas abortadas (no se pudieron iniciar)  Formula:  a - b - c
	Número de casos de prueba fallidos	Esta es la cantidad de casos de prueba que han fallado en la ejecución de un conjunto de pruebas.	Sí. Este se calcula actualmente utilizando Jira y Teamcity para la ejecución de las pruebas automatizadas y la generación del reporte con los resultados de las ejecuciones.  Este, según la ISTQB-CT-TAE,	a = Total de pruebas ejecutadas  b = Cantidad de pruebas exitosas  c = Cantidad de pruebas abortadas (no se pudieron iniciar)  Formula:  a - b - c

			es información que forma parte de un reporte de la ejecución de pruebas automatizadas.	
	Cantidad total de pruebas que se han ejecutado	Esta es la cantidad de casos de prueba que se han ejecutado con base en un conjunto de pruebas seleccionado.	Sí. Este se calcula actualmente utilizando Jira y Teamcity para la ejecución de las pruebas automatizadas y la generación del reporte con los resultados de las ejecuciones.	<p>a = Cantidad de pruebas exitosas</p> <p>b = Cantidad de pruebas fallidas</p> <p>c = Cantidad de pruebas abortadas (no se pudieron iniciar)</p> <p>Formula:</p> $a + b + c$
Información sobre defectos	Defectos encontrados en las pruebas automatizadas por <i>release</i>	Esta es la cantidad de defectos encontrados en las pruebas que se han ejecutado en una versión liberada de las pruebas. Los fallos encontrados son de la herramienta de prueba y no del producto que se está probando.	No. Actualmente solo se reportan los fallos como <i>tickets</i> de tipo <i>bug</i> , sin embargo, para poder identificar en qué <i>release</i> se encontró el fallo, se puede añadir un tag con la versión del <i>release</i> en el que se encontró.	Se pueden filtrar los defectos en Jira utilizando los siguientes datos: <ul style="list-style-type: none"> <li>- Proyecto en Jira del equipo de <i>QA Automation</i>.</li> <li>- <i>Ticket</i> de tipo <i>bug</i>.</li> <li>- Tag con el <i>release</i> que se desea buscar.</li> </ul>
	Defectos encontrados en el producto por <i>release</i>	Esta es la cantidad de defectos que se han encontrado durante la ejecución de pruebas de regresión e integración para el producto que se va a liberar.	No. Actualmente estos se reportan como <i>bugs</i> de otro proyecto dentro de Jira. A estos se les adjunta la versión del <i>release</i> que afecta el fallo encontrado.	Se pueden filtrar los defectos en Jira utilizando los siguientes datos: <ul style="list-style-type: none"> <li>- Proyecto en el que se está ejecutando las pruebas.</li> <li>- <i>Ticket</i> de tipo <i>bug</i>.</li> <li>- Tag con el <i>release</i> que se desea buscar.</li> </ul>

	Defectos encontrados en una etapa	Estos son los defectos que se encuentran en una etapa en específico del proceso de QC, por ejemplo, en la etapa de pruebas de integración o en la etapa de pruebas de regresión.	No. Actualmente en los <i>tickets</i> de Jira existe una sección en la que se adjunta la etapa del proceso de QC en la que se encontró un fallo en el producto que se está probando.	Se pueden filtrar todos los fallos que tengan las siguientes características: <ul style="list-style-type: none"> <li>- Proyecto de Jira para la ejecución de pruebas.</li> <li>- <i>Release</i> específico que se quiere utilizar.</li> <li>- Etapa que se quiere buscar.</li> </ul>
	Cantidad de falsos positivos	Según la ISTQB-CT-TAE, los falsos positivos ocurren cuando hubo una falla en el <i>software</i> bajo prueba, pero la automatización de la prueba no lo identificó, por lo que se informó un resultado de paso, lo que puede resultar en un defecto potencial que no se pudo detectar.	No. Se pueden reportar errores de configuración del ambiente o de prueba y adjuntarlos como parte del proyecto de Jira en donde se están ejecutando las pruebas junto con una etiqueta que lo identifique como un falso positivo.	Se pueden filtrar todos los fallos que tengan las siguientes características: <ul style="list-style-type: none"> <li>- Proyecto de Jira para la ejecución de pruebas.</li> <li>- <i>Release</i> específico que se quiere utilizar.</li> <li>- Etiqueta utilizada para identificar los falsos positivos.</li> <li>- Fuente del error: Error de configuración, Error del ambiente, Error de la prueba automatizada.</li> </ul>
Después de la ejecución de pruebas	Nivel de calidad	Este es el nivel de calidad del producto que se le entrega al equipo de QA con base en los <i>bugs</i> reportados y el estándar de calidad establecido por la	No. Actualmente no se tiene una métrica que diga la calidad que tiene el producto; se tienen datos que se puede utilizar para esto, pero no el reporte.	Se puede utilizar una matriz basada en la matriz de probabilidad e impacto propuesta en el PMBOK como instrumento para la recolección y análisis

		<p>organización. Este se mide de forma iterativa para poder llevar seguimiento de la calidad del producto a liberar y asegurar que este se entregue con una calidad alta.</p>		<p>de datos.</p> <p>Según este marco de referencia, la matriz de probabilidad e impacto es una cuadrícula que mapea la probabilidad de ocurrencia de cada riesgo y su impacto en los objetivos del proyecto si ese riesgo ocurre.</p> <p>Se sugiere utilizar una matriz similar, pero con la severidad del fallo en lugar del impacto del riesgo, y la prioridad del <i>bug</i> en lugar de la probabilidad de ocurrencia de un riesgo.</p> <p>Esto se detalla en la sección 5.5.1 del documento.</p>
<p>Cantidad de pruebas por categoría</p>	<p>Pruebas de regresión</p>	<p>Esta es la cantidad de pruebas automatizadas que forman parte del conjunto de pruebas que se ejecutan durante las pruebas de regresión.</p>	<p>Sí. A la hora de crear las pruebas con el <i>software</i> de prueba, se agregan etiquetas para la identificación de las pruebas que se corren en regresión.</p> <p>Estas etiquetas también se presentan en el <i>ticket</i> de Jira de cada prueba.</p>	<p>Contar la cantidad de pruebas que poseen la etiqueta “<i>regression</i>”.</p>
	<p>Pruebas de integración</p>	<p>Esta es la cantidad de pruebas automatizadas que forman parte del</p>	<p>Sí. A la hora de crear las pruebas con el <i>software</i> de prueba, se agregan</p>	<p>Contar la cantidad de pruebas que poseen la etiqueta “<i>ci</i>” (letras para “<i>Continuous</i></p>

conjunto de pruebas que se ejecutan durante las pruebas de integración.

etiquetas para la identificación de las pruebas que se corren en integración.

*Integration*”).

Estas etiquetas también se presentan en el *ticket* de Jira de cada prueba.

Pruebas de humo (*smoke tests*)

Esta es la cantidad de pruebas automatizadas que se ejecutan como parte de las pruebas de humo.

Sí. A la hora de crear las pruebas con el *software* de prueba, se agregan etiquetas para la identificación de las de humo.

Contar la cantidad de pruebas que poseen la etiqueta “*smokeTest*”.

Estas etiquetas también se presentan en el *ticket* de Jira de cada prueba.

Pruebas de API

Esta es la cantidad de pruebas automatizadas que se utilizan para la prueba de los APIs del producto.

Sí. A la hora de crear las pruebas con el *software* de prueba, se agregan etiquetas para la identificación de las pruebas de API.

Contar la cantidad de pruebas que poseen la etiqueta “*cmsEndpoints*”.

Estas etiquetas también se presentan en el *ticket* de Jira de cada prueba.

Actualmente se utiliza Jira para crear una colección de pruebas que ejecuta sólo las pruebas de API.

	<p>Pruebas del sistema de administración</p>	<p>Esta es la cantidad de pruebas automatizadas que se corren en el sistema de administración tales como configuración de parámetros, administración de jugadores, y gestión de los juegos.</p>	<p>Sí. A la hora de crear las pruebas con el <i>software</i> de prueba, se agregan etiquetas para la identificación de las pruebas que se corren en el sistema de administración.</p> <p>Estas etiquetas también se presentan en el <i>ticket</i> de Jira de cada prueba.</p> <p>Actualmente se utiliza Jira para crear una colección de pruebas que ejecuta sólo las pruebas específicas para el sistema de administración.</p>	<p>Contar la cantidad de pruebas que poseen la etiqueta “<i>admin</i>”.</p>
	<p>Pruebas del sistema que gestiona los juegos</p>	<p>Esta es la cantidad de pruebas automatizadas que se corren en el sistema que gestiona los juegos.</p>	<p>Sí. A la hora de crear las pruebas con el <i>software</i> de prueba, se agregan etiquetas para la identificación de las pruebas que se corren en el sistema que gestiona los juegos.</p> <p>Estas etiquetas también se presentan en el <i>ticket</i> de Jira de cada prueba.</p> <p>Actualmente se utiliza Jira para crear una colección de pruebas que ejecuta sólo las pruebas específicas para el sistema de gestión de juegos.</p>	<p>Contar la cantidad de pruebas que poseen la etiqueta respectiva para el sistema que gestiona los juegos, por ejemplo, “<i>lobby</i>”.</p>

	Pruebas para juegos en específico	Esta es la cantidad de pruebas automatizadas que se corren para la validación de un juego en específico.	Sí. A la hora de crear las pruebas con el <i>software</i> de prueba, se agregan etiquetas para la identificación de las pruebas que se pueden correr para un juego en específico.  Estas etiquetas también se presentan en el <i>ticket</i> de Jira de cada prueba.	Contar la cantidad de pruebas que poseen la etiqueta respectiva para la ejecución de pruebas para un juego en específico, por ejemplo, “specificGame”.
--	-----------------------------------	--	---	--

Nota. Métricas propuestas para la metodología de QC categoría, descripción, se puede calcular actualmente, fórmula

Con base en la información que se muestra en la tabla 5, se puede observar que para esta metodología se proponen 15 métricas, donde 10 ya se calculan, pero no se lleva seguimiento, y 5 son métricas nuevas que no se calculan.

Se propone llevar seguimiento de estas métricas por medio de *dashboards* (representaciones gráficas) de la información, además de generar un informe mensual con un análisis sobre el estado de las métricas actualmente y el estado de estas en el mes anterior, y compartir el informe con las partes interesadas, por ejemplo, la gerencia de cumplimiento de la calidad o la dirección de inteligencia de negocios.

### 5.5.1. Nivel de calidad

Actualmente la empresa no posee una metodología formal para definición de la calidad de un producto liberado, sin embargo, si se posee datos para realizar el análisis. Debido a esto, en esta sección se propone una metodología para el cálculo del nivel de calidad.

Para dicho cálculo, se utilizará una matriz de probabilidad de riesgo adaptada para lo que es “calidad”, y con base en los resultados que se logren obtener con la aplicación de dicha matriz con los *bugs* encontrados, se definirán una serie de reglas que ayudarán a determinar la calidad del producto.

Esta métrica no mide la calidad del producto después de salir a producción, sino que sirve para medir el producto que se le entrega a QA. Dependiendo del contexto, esta métrica se puede aplicar de forma iterativa para poder llevar seguimiento de la calidad que están entregando los desarrolladores, ya sea un proyecto complejo como los *release*, hasta una funcionalidad en específico, en otras palabras, puede aplicarse a distintos niveles de granularidad. A continuación, se detallan ambos elementos (matriz y reglas).

#### **5.5.1.1. *Matriz de prioridad de un bug***

Al analizar los datos disponibles en la herramienta para la gestión de proyectos Jira y con base en la entrevista realizada a la Gerencia de Cumplimiento de Calidad, se propone la utilización de una matriz inspirada en la matriz de probabilidad de ocurrencia de un riesgo, el cual, según el PMBOK 2016, es una matriz de probabilidad e impacto que utiliza definiciones positivas de impacto para las oportunidades y definiciones de impacto negativo para las amenazas. Este se muestra en la siguiente figura.

Figura 9. Ejemplo de matriz de probabilidad e impacto con esquema de puntuación.

		Threats					Opportunities						
Probability	Very High 0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05	Probability	Very High 0.90
	High 0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04		High 0.70
	Medium 0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03		Medium 0.50
	Low 0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02		Low 0.30
	Very Low 0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01		Very Low 0.10
		Very Low 0.05	Low 0.10	Moderate 0.20	High 0.40	Very High 0.80	Very High 0.80	High 0.40	Moderate 0.20	Low 0.10	Very Low 0.05		
Negative Impact						Positive Impact							

Nota. Adaptado de PMBOK Guide – Sixth Edition + Agile Practice Guide (p. 408), por Project Management Institute, 2017, Project Management Institute.

La matriz propuesta para este contexto es una matriz que ayuda a determinar la prioridad de un fallo encontrado, ya sea nivel *low* (bajo), *minor* (medio), *major* (mayor) o *critical* (crítico). Para esto se sugiere utilizar el impacto o severidad del fallo encontrado y su frecuencia de ocurrencia como dimensiones de la matriz, asimismo, se otorgará un valor numérico a cada una de las variables para realizar el respectivo cálculo.

El valor de las variables de las dimensiones a utilizar para la matriz propuesta se basa en las utilizadas en la matriz de probabilidad e impacto de un riesgo del marco de referencia PMBOK 2016 y su descripción se basan en los niveles de prioridad definidos por la empresa. Estos se detallan a continuación.

**Tabla 6.** Valores para utilizar en matriz propuesta para el cálculo de la prioridad de un fallo

<i>Categoría</i>	<i>Nivel</i>	<i>Valor</i>	<i>Descripción</i>
Frecuencia de ocurrencia	Muy bajo	0.01	La probabilidad de que ocurra el fallo es demasiado baja, casi nulo. Esto significa que, de la cantidad de datos de prueba, se da entre el 1% y 29% de las veces.
	Bajo	0.30	La probabilidad de que ocurra es baja, aunque puede presentarse. Significa que, de la cantidad de datos de prueba, se da entre el 30% y el 49% de las veces.
	Medio	0.50	El fallo puede materializarse en cualquier momento, o sea que, de la cantidad de datos de prueba, se da entre el 50% y el 69% de las veces.
	Alto	0.70	El fallo suele presentarse, tiene una alta probabilidad de ocurrencia. Esto significa que, de la cantidad de datos de prueba, se el 70% de las veces o más.
Severidad	Bajo	0.10	El fallo es un problema trivial con poco o ningún impacto en el progreso o la usabilidad.
	Medio	0.20	Problemas menores o defectos que se pueden solucionar fácilmente.
	Alto	0.40	Tiene el potencial de afectar el progreso.
	<b>Crítico</b>	0.80	Grave problema que podría bloquear el progreso.

*Nota.* Esta tabla describe los niveles de frecuencia de ocurrencia y severidad de los fallos, muestra la categoría al que pertenece, y el valor que representa.

Con el uso de estos valores numéricos para la severidad de un riesgo y su frecuencia de ocurrencia, se pueden multiplicar para dar una puntuación que nos ayudará a determinar la prioridad del fallo.

En la siguiente tabla se puede observar la tabla propuesta para el cálculo del nivel de prioridad de un *bug*. En el eje horizontal (eje x) se tiene el nivel de severidad o impacto del fallo encontrado junto a su valor numérico utilizado para el cálculo de la prioridad. En el eje vertical (eje y) se tiene el nivel de ocurrencia del fallo, es decir, la frecuencia en la que sucede un fallo.

**Tabla 7.** Matriz propuesta para el cálculo de la prioridad de un *bug* encontrado

<i>Ocurrencia</i>				
Alto = 0.70	0.07	0.14	0.28	0.56
Medio = 0.50	0.05	0.10	0.20	0.40
Bajo = 0.30	0.03	0.06	0.12	0.24
Muy Bajo = 0.01	0.001	0.002	0.004	0.008
	Bajo = 0.10	Medio = 0.20	Alto = 0.40	Crítico = 0.80
	<i>Severidad / Impacto</i>			

*Nota.* Esta tabla ayuda a calcular el valor de la prioridad de un *bug* según su nivel de frecuencia de ocurrencia y la severidad o impacto que este tenga.

Los valores límite definidos para determinar el nivel de prioridad y las características que lo identifican se basan en los niveles de prioridad utilizados actualmente por la empresa. Estos se detallan a continuación:

**Tabla 8.** Valores límites para la definición del nivel de prioridad de un bug

<i>Nivel</i>	<i>Mayor o Igual</i>	<i>Menor</i>
Bajo ( <i>Low</i> )	0.001	0.05
Medio ( <i>Minor</i> )	0.05	0.14
Alto ( <i>Major</i> )	0.14	0.28
Crítico ( <i>Critical</i> )	0.28	0.80

*Nota.* Esta tabla muestra el valor mínimo y el valor máximo para la clasificación de la prioridad del bug según el valor obtenido en la matriz propuesta para el cálculo de la prioridad del bug.

El rango para definir el nivel de prioridad de un bug es calculado con base en la tabla propuesta para el cálculo del nivel de prioridad de un bug. El bug debe ser mayor o igual que el menor valor que se encuentra dentro del área del color correspondiente a cada nivel, por ejemplo, un bug de prioridad baja debe ser mayor o igual que 0.001 (el menor valor que se encuentra dentro del área verde).

El valor máximo es el número menor que se encuentra dentro del área del color del siguiente nivel menos 1. En el caso del área verde, como ejemplo, la prioridad debe ser menor a 0.05.

#### **5.5.1.2. Definición de nivel de calidad de un producto en desarrollo**

Para la definición del nivel de calidad del producto a liberar por la Empresa A, se propone el uso de la matriz descrita en la sección 5.5.1.1 del documento y el uso de las siguientes restricciones o reglas propuestas para la asignación del nivel de calidad. Cabe mencionar que esto puede ser modificado a futuro según las necesidades.

Estas reglas se aplicarán de forma recursiva, esto quiere decir que cada vez que se entregue al equipo de QA un producto para que este sea revisado, se definirá el nivel de calidad de este con la aplicación de la matriz y reglas propuestas con el fin de poder llevar seguimiento del nivel de calidad del producto en desarrollo.

Para esto se propone 4 niveles de calidad: Alta, Media, Baja y Muy Baja. Cada uno posee las siguientes condiciones.

Definiremos "m" como la media de la cantidad de *bugs* reportados en proyectos anteriores de la misma naturaleza, o en su defecto un valor definido por un experto.

- Calidad Alta
  - Para este nivel de calidad, se deben cumplir todas las siguientes condiciones.
  - Condición 1: Todos los *bugs* son de prioridad Baja (*low*).
  - Condición 2: La cantidad de *bugs* es menor o igual a "2m" (2 veces la media).
  - Condición 3: La suma del valor de las prioridades debe ser menor a 0.05.
    - Donde 0.05 representa el menor valor para un *bug* con prioridad media según la matriz propuesta.
  
- Calidad Media
  - Para este nivel de calidad, se deben cumplir todas las siguientes condiciones.
  - Condición 1: Todos los *bugs* son de prioridad menor o igual que la prioridad Media (*minor*).
  - Condición 2: La cantidad de *bugs* es menor o igual a "2m" (2 veces la media).
  - Condición 3: La suma del valor de las prioridades debe ser menor a 0.14.
    - Donde 0.14 representa el menor valor para un *bug* con prioridad alta según la matriz propuesta.

- Calidad Baja
  - Para este nivel de calidad, se deben cumplir todas las siguientes condiciones.
  - Condición 1: Todos los *bugs* son de prioridad menor o igual que la prioridad Alta (*major*).
  - Condición 2: La cantidad de *bugs* es menor o igual a “2m” (2 veces la media).
  - Condición 3: La suma del valor de las prioridades debe ser menor a 0.28.
    - Donde 0.28 representa el menor valor para un *bug* con prioridad crítica según la matriz propuesta.
  
- Calidad Muy Baja
  - Para este nivel de calidad, se debe cumplir al menos una de las siguientes condiciones.
  - Condición 1: Existe al menos 1 *bug* de prioridad Crítica (*critical*).
  - Condición 2: La cantidad de *bugs* es mayor a “2m” (2 veces la media).
  - Condición 3: La suma del valor de las prioridades debe ser mayor o igual 0.28.
    - Donde 0.28 representa el menor valor para un *bug* con prioridad crítica según la matriz propuesta.

Cabe mencionar que esta métrica mide la calidad interna, es decir, el producto entregado por el desarrollador, no el producto que se entrega al cliente después de la ejecución de las pruebas de regresión e integración, y de la aplicación de las correcciones necesarias.

### 5.5.1.3. Ejemplo de aplicación de métrica de nivel de calidad

La aplicación de la métrica para medir el nivel de calidad se puede aplicar en productos de distinta granularidad, desde el desarrollo de una funcionalidad nueva, hasta el desarrollo de un producto como tal.

A modo de ejemplo, utilizaremos como contexto el desarrollo de una nueva funcionalidad para un producto existente y “m” (la media de *bugs* reportados en proyectos anteriores de la misma naturaleza) será 5. A continuación, se presenta un ejemplo de un producto con calidad alta, un producto con calidad media, y un producto con calidad baja.

#### 5.5.1.3.1. Ejemplo de producto con calidad alta

En este primer ejemplo, lo primero que se debe de hacer con los fallos encontrados es utilizar la matriz propuesta para definir el nivel de prioridad de cada *bug*. Al aplicar dicha matriz, los resultados lucirían de la siguiente manera.

**Tabla 9.** Ejemplo de producto con calidad alta

Bug	Nivel de ocurrencia	Nivel de impacto	Prioridad
Bug 1	Bajo (0.30)	Bajo (0.10)	Baja (0.03)
Bug 2	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 3	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 4	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 5	Muy Bajo (0.01)	Alto (0.40)	Baja (0.004)
<b>Total</b>			0.037

*Nota.* En esta tabla se muestra el cálculo de la prioridad de los *bugs* encontrados en un producto con alta calidad.

En la tabla anterior se encontraron 5 *bugs* de baja prioridad. Al poder contar con el nivel de prioridad de cada *bug*, se procede a consultar las condiciones que nos ayudan a definir el nivel de calidad del producto (ver sección 5.5.1.2).

Con base en las reglas propuestas, este ejemplo cumple con las condiciones de un producto con calidad “Alta” ya que todos los *bugs* reportados son de prioridad baja, la cantidad de *bugs* reportados es menor o igual que 2m (10), y la suma del valor de las prioridades (0.037) es menor a 0.05.

### 5.5.1.3.2. Ejemplo de producto con calidad media

Al igual que el ejemplo anterior, se define la prioridad de cada *bug* encontrado utilizando la matriz propuesta. Al aplicar dicha matriz, los resultados lucirían de la siguiente manera.

**Tabla 10.** Ejemplo de producto con calidad media

Bug	Nivel de ocurrencia	Nivel de impacto	Prioridad
Bug 1	Bajo (0.30)	Bajo (0.10)	Baja (0.03)
Bug 2	Bajo (0.30)	Medio (0.20)	Media (0.06)
Bug 3	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 4	Muy Bajo (0.01)	Medio (0.20)	Baja (0.002)
Bug 5	Bajo (0.30)	Bajo (0.10)	Baja (0.03)
<b>Total</b>			0.123

*Nota.* En esta tabla se muestra el cálculo de la prioridad de los *bugs* encontrados en un producto con calidad media.

En este caso se encontraron 4 *bugs* de prioridad baja y 1 *bug* de prioridad media. Con base en los datos obtenidos, se consultan las condiciones que nos ayudan a definir el nivel de calidad del producto (ver sección 5.5.1.2). De acuerdo reglas, este ejemplo cumple con las condiciones del nivel de calidad “Media”.

- Todos los *bugs* tienen una prioridad menor o igual que la prioridad Media.
- La cantidad de *bugs* reportados es menor o igual que 2m (10).
- La suma de los valores de las prioridades es inferior a 0.14; en este caso, el total fue 0.123.

### 5.5.1.3.3. Ejemplo de producto con calidad baja

Para determinar la calidad del producto de este ejemplo, calcularemos la prioridad de cada *bug* utilizando la matriz propuesta para definir el nivel de prioridad de cada *bug*. Al aplicar dicha matriz, los resultados lucirían de la siguiente manera.

**Tabla 11.** Ejemplo de producto con calidad mala

Bug	Nivel de ocurrencia	Nivel de impacto	Prioridad
Bug 1	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 2	Muy Bajo (0.01)	Bajo (0.10)	Baja (0.001)
Bug 3	Medio (0.50)	Bajo (0.10)	Media (0.05)
Bug 4	Bajo (0.30)	Bajo (0.10)	Baja (0.03)
Bug 5	Medio (0.50)	Medio (0.20)	Media (0.10)
Bug 6	Alto (0.70)	Medio (0.20)	Alta (0.14)
<b>Total</b>			<b>0.272</b>

*Nota.* En esta tabla se muestra el cálculo de la prioridad de los *bugs* encontrados en un producto con calidad mala.

En este ejemplo se encontró 3 *bugs* de prioridad baja, 2 *bugs* de prioridad media, y 1 *bug* de prioridad alta. Al revisar las condiciones para determinar el nivel de calidad del producto, este ejemplo cumple con las reglas del nivel de calidad “Baja”; a continuación, se listan los puntos que se cumplen.

- Todos los *bugs* reportados tienen una prioridad menor o igual que la prioridad Alta.
- El número de *bugs* reportados es menor o igual que 2m (10), habiéndose identificado 6 *bugs* en este caso.
- La suma de los valores de las prioridades es inferior a 0.28; en este caso, el total fue 0.272.

#### 5.5.1.3.4. Ejemplo de producto con calidad muy baja

Para determinar la calidad del producto de este ejemplo, calcularemos la prioridad de cada *bug* utilizando la matriz propuesta para definir el nivel de prioridad de cada *bug*. Al aplicar dicha matriz, los resultados lucirían de la siguiente manera.

**Tabla 12.** Ejemplo de producto con calidad mala

Bug	Nivel de ocurrencia	Nivel de impacto	Prioridad
Bug 1	Bajo (0.30)	Bajo (0.10)	Baja (0.03)
Bug 2	Medio (0.50)	Alto (0.40)	Alta (0.20)
Bug 5	Bajo (0.30)	Alto (0.40)	Media (0.12)
Bug 7	Medio (0.50)	Bajo (0.10)	Media (0.05)
Bug 9	Medio (0.50)	Medio (0.20)	Media (0.10)
Bug 10	Medio (0.50)	Crítico (0.80)	Crítica (0.40)
<b>Total</b>			<b>0.9</b>

*Nota.* En esta tabla se muestra el cálculo de la prioridad de los *bugs* encontrados en un producto con calidad mala.

En este ejemplo se encontró 1 *bug* de prioridad baja, 3 *bugs* de prioridad media, 1 *bug* de prioridad alta, y 1 *bug* de prioridad crítica. Al revisar las condiciones definidas para determinar la calidad del producto, este ejemplo cumple con las reglas del nivel de calidad “Muy Baja”; a continuación, se listan los puntos que se cumplen.

- Se ha detectado al menos un *bug* con prioridad crítica. La presencia de un solo *bug* crítico clasifica automáticamente el nivel de calidad del producto como "Muy Bajo".
- Aunque ya se cumple una de las condiciones anteriores, también se cumple que la suma de los valores de las prioridades es igual o superior a 0.28, dado que el total en este caso fue 0.9.

Con los ejemplos anteriores, se puede observar cómo se puede calcular el nivel de un producto sin importar el tamaño que tenga, como se menciona anteriormente. Con la aplicación de esto, se puede llevar el nivel de calidad a alto al poder tener conocimiento del nivel de calidad actual de un producto y cómo éste fluctúa con los cambios que se le apliquen.

## 5.6. Gestión de defectos

Durante la ejecución de las pruebas, uno de los objetivos principales es encontrar defectos y registrarlos de forma apropiada. La gestión de los defectos puede variar, según la ISTQB, la forma en la que se registran los defectos puede depender del contexto del componente o sistema que se está probando, el nivel de prueba y el modelo de ciclo de vida de desarrollo de *software*.

Al contar con defectos que bien registrados, se pueden obtener varios beneficios, por ejemplo, se informar a los desarrolladores y analistas de QA sobre cualquier evento adverso ocurrido, asimismo, se puede tener guía de cómo reproducir el problema, aislarlo de ser necesario, o corregir el problema.

Tener conocimiento de los defectos reportados también permite informar a la Gerencia de Cumplimiento de la Calidad, y cualquier otro interesado, rastrear la calidad del producto y el impacto de estos.

Según el marco de referencia ISTQB, se sugieren estos puntos como datos que debe poseer un defecto al reportarse.

- Un identificador.
- Título del defecto encontrado.
- Descripción del defecto encontrado.

- Fecha en la que se reportó el defecto.
- Fase del proceso de QC en el que se observó el defecto, por ejemplo, en las pruebas de regresión. Esto en el caso de ser un error que no es de la prueba automatizada.
- Pasos para reproducir el defecto junto a evidencias (capturas de pantalla, mensajes de error, entre otros).
- Resultados reales y resultados esperados.
- Grado de gravedad del defecto. El grado del defecto se maneja diferente según el origen de este. Si este es un defecto encontrado en las pruebas automatizadas, esta es definida por el equipo de *QA Automation*, si este es encontrado en el producto, el nivel de gravedad es definido por la Gerencia de Cumplimiento de Calidad.
- Prioridad del arreglo. Al igual que con el nivel de gravedad, este se define según el origen del fallo. Si este es un defecto encontrado en las pruebas automatizadas, esta es definida por el equipo de *QA Automation*, si este es encontrado en el producto, la prioridad es definida por la Gerencia de Cumplimiento de Calidad.
- Estado del informe de defectos, en este caso: abierto, cerrado, en progreso, en revisión, no se arreglará, reabierto, listo.
- Conclusiones, recomendaciones y aprobaciones.
- Historial de cambios, como la secuencia de acciones tomadas por los miembros del equipo del proyecto con respecto al defecto para aislarlo, repararlo y confirmarlo como arreglado.
- Referencia al caso de prueba que reveló el problema.
- Versión del producto en el que se encontró el defecto.

Cabe mencionar que el nivel de prioridad y de severidad de un defecto encontrado en las pruebas automatizadas actualmente no es definido por el equipo de *QA Automation*, sino que esto se realiza ocasionalmente.

En cuanto a los defectos que son encontrados en el producto bajo prueba, la severidad de un defecto encontrado se discute en una reunión con la gerencia de distintos departamentos o equipos con el fin de contemplar si el producto va a ser liberado con los defectos encontrados, o si se va a atrasar su liberación.

Actualmente es posible disponer de estos datos al contar con el uso de herramientas para la gestión de proyectos, en este caso Jira. Esta herramienta nos permite administrar automáticamente los defectos, por ejemplo, con la asignación de un identificador, la actualización del estado del defecto reportado, llevar seguimiento de los cambios realizados, entre otros.

## **5.7. Gestión de la configuración del software de prueba**

La gestión apropiada de la configuración ayudará a establecer y mantener la integridad de los ambientes de prueba utilizados por el departamento de automatización del aseguramiento de la calidad y el *software* de prueba (herramienta usada para desarrollar pruebas) en el ciclo de vida del proyecto. Con la gestión apropiada de la configuración se va a lograr la trazabilidad de los elementos de configuración.

Basándonos en el contexto del departamento de *QA Automation*, es de relevancia administrar la configuración del producto que se va a probar, los ambientes de prueba que se van a utilizar en las pruebas de regresión, integración y de humo, también llevar seguimiento del *software* de prueba utilizado para la ejecución y desarrollo de las pruebas automatizadas, y los casos de prueba automatizados.

Con base en lo anterior y lo propuesto por la ISTQB, se sugieren los siguientes puntos para la gestión de la configuración como parte de la metodología de QC propuesta por cada elemento de configuración de relevancia.

- Producto que se va a probar
  - La instalación del producto que se va a probar se realiza por el departamento de *Solution*. El manejo de versiones de cada producto que se libera se maneja con base en el nombre del *branch* en el que se ha hecho *merge* todos los cambios.
  - Para consultar la versión de los ambientes, se puede revisar el sistema de administración.
- Ambientes de prueba
  - Se sugiere tener documentados los navegadores web soportados, las versiones soportadas de los navegadores web, y los sistemas operativos de los dispositivos móviles, con el fin de poner tener una fuente con la cual poder confirmar que un fallo no tiene que ver con la configuración del ambiente de prueba.
  - Este debe actualizarse cuando una versión o navegador web ya no se soporta.
- *Software* de prueba
  - Para el *software* de prueba, el equipo de *QA Automation* posee una documentación en el cual se describen las librerías utilizadas y sus respectivas versiones, además de los pasos necesarios para configurar la herramienta para su uso.
- Casos de prueba automatizados
  - Todos los casos de prueba se identifican de forma única.

- Actualmente el departamento maneja las pruebas por medio de un identificador único relacionado a un *ticket* de Jira. Las pruebas poseen el mismo identificador que se genera a la hora de crear un *ticket* de tipo “*test case*”.
- Se controlan las versiones.
  - Hoy día en el departamento de *QA Automation* no se lleva seguimiento de la versión de *release* en la que se desarrolló la prueba. Con el fin de poder ubicar la última vez en que se modificó la prueba, se propone añadir una etiqueta en el *ticket* de Jira que está relacionado con la prueba que contenga la versión en la que se desarrolló o modificó la prueba; con esto se puede revisar la antigüedad de la prueba a comparación de las funcionalidades que se han liberado.
- Se rastrean los cambios.
  - Los cambios actualmente se rastrean por medio de los *commits* realizados en el repositorio de *Bitbucket* que contiene el proyecto; cada *commit* realizado contiene el identificador del *ticket* de Jira al que está ligado el cambio. Esto junto a lo propuesto en el punto anterior (adición de etiqueta con la versión del *release* en la que se desarrolló la prueba), pueden mejorar la trazabilidad de los cambios realizados.
- Los casos de prueba automatizados están ligados al caso de prueba manual.
  - Actualmente el equipo de automatización del aseguramiento de la calidad liga el *ticket* del caso de prueba manual con el *ticket*

del caso de prueba automatizado. Esto permite tener conocimiento de lo que se intenta realizar mediante la prueba automatizada.

## **5.8. Generación de Informes**

La generación de informes es una fase importante de esta metodología de QC. La ejecución de las pruebas automatizadas genera información de relevancia que puede compartirse con los demás equipos de QA para poder dar visibilidad del estado actual de la ejecución de dichas pruebas, asimismo, con un informe que ayude a presentar de forma clara la información obtenida, se puede ayudar a la Gerencia de Cumplimiento de la Calidad a tomar decisiones.

Para esta metodología se propone que los informes, dentro del contexto del departamento de automatización del aseguramiento de la calidad, se realicen una vez al mes, esto con el fin de que se alinee con las fechas en las que se ejecutan las pruebas de integración y de regresión del producto a liberar, lo cual sucede cada mes.

Además, se sugiere crear 3 informes en distintas etapas de la ejecución de las pruebas de integración y/o regresión. Estos se listan a continuación.

- Se generará un informe sobre el inicio de la ejecución de las pruebas automatizadas. Este resume el conjunto de pruebas que se van a ejecutar.
- Se generará un informe sobre el avance de la ejecución de las pruebas, ya sea de regresión o de integración. La ejecución y análisis de los resultados de estas pruebas tienen una duración de 1 día, por lo que este informe se estaría compartiendo a mediodía cuando ya se lleva un avance con el análisis de los resultados.
- Se generará un informe con el resultado final sobre la ejecución de las pruebas.

Con base en los propuesto por la ISTQB, se proponen los siguientes puntos como datos esenciales que deberían de tener los informes.

- Informe inicial
  - Para el informe inicial es necesario brindar un resumen de las pruebas a realizar con el fin de que los interesados sepan la cobertura de las pruebas automatizadas a ejecutar.
  - Este debe de tener la fecha y hora de inicio de las pruebas.
  - Este debe de tener la versión del producto que se estará probando.
  - Este también debe de tener un listado de los ambientes de prueba que se van a utilizar para la ejecución de las pruebas.
  - A continuación, se presenta un ejemplo de cómo podría lucir la estructura del informe de inicio.

**Tabla 13.** *Plantilla de informe de inicio de pruebas*

Informe de Inicio - [Nombre del Tipo de Prueba]			
Versión del producto:			
Fecha y hora de inicio:		Fecha y hora de fin:	
Cantidad total de pruebas a ejecutar:			
Ambientes de prueba:			
Resumen de pruebas a ejecutar:			

- Informe de avance
  - Para el informe de avance, se debe mostrar el estado de las actividades de prueba, en este caso, la cantidad de pruebas que se han ejecutado de forma exitosa, cuántas faltan de ejecutarse y cuáles han fallado.
  - Este debe de tener la fecha y hora de inicio de las pruebas.
  - Este debe de tener la versión del producto que se está probando.
  - Este también debe de tener un listado de los ambientes de prueba que se están utilizando.
  - Este debe listar los factores que impidan la correcta ejecución y avance de las pruebas automatizadas, por ejemplo, errores de configuración del ambiente de prueba.
  - Este debe mostrar las desviaciones del plan, incluidas las desviaciones en el cronograma, por ejemplo, la ejecución de forma manual de las pruebas automatizadas por un error de configuración en el ambiente que no permite la ejecución de las pruebas.
  - A continuación, se presenta un ejemplo de cómo podría lucir la estructura del informe de avance.

**Tabla 14.** *Plantilla de informe de avance de la ejecución de pruebas*

Informe de Avance - [Nombre del Tipo de Prueba]			
Versión del producto:			
Fecha y hora de inicio:		Fecha y hora de fin:	
Cantidad total de pruebas a ejecutar:		Cantidad de pruebas no ejecutadas	
Cantidad de pruebas exitosas:		Cantidad de pruebas fallidas:	
Ambientes de prueba:			
Factores que impiden la ejecución de pruebas:			
Desviaciones del plan:			

- Informe final
  - Este informe provee un resumen de las pruebas realizadas y lo ocurrido durante el periodo de pruebas de integración o las pruebas de regresión en este caso.
  - Este debe de tener la fecha y hora de inicio de las pruebas.
  - Este debe de tener la versión del producto que se probó.
  - Este también debe de tener un listado de los ambientes de prueba que se están utilizando.
  - Este debe incluir las desviaciones del plan, incluidas las desviaciones en el cronograma.
  - Este debe incluir la cantidad de pruebas ejecutadas, la cantidad de pruebas exitosas y la cantidad de pruebas fallidas.

- Factores que han afectado la ejecución de las pruebas automatizadas.
- Este informe debe proveer un listado de los *bugs* encontrados y reportados.
- A continuación, se presenta un ejemplo de cómo podría lucir la estructura del informe final.

**Tabla 15.** *Plantilla de informe final de la ejecución de pruebas*

Informe Final - [Nombre del Tipo de Prueba]			
Versión del producto:			
Fecha y hora de inicio:		Fecha y hora de fin:	
Cantidad total de pruebas a ejecutar:			
Cantidad de pruebas exitosas:		Cantidad de pruebas fallidas:	
Ambientes de prueba:			
<i>Bugs</i> reportados:			
Factores que afectaron la ejecución de pruebas:			
Desviaciones del plan:			

Al finalizar la generación de los informes, estos se pueden analizar para poder determinar qué pruebas se pueden tomar en consideración para el próximo periodo de pruebas. Esto se realizaría tomando en cuenta las partes que no se pudieron cubrir con las pruebas automatizadas que se ejecutaron, los fallos reportados, entre otros factores.

## 5.9. Retrospectiva

En un proyecto ágil, la aplicación de una retrospectiva es de importancia. Según MacNeil C. (2024) una retrospectiva forma parte del modelo ágil *Scrum* y se realiza al finalizar cada *sprint*. MacNeil C. (2024) explica que este consiste en realizar una reunión con los miembros del equipo para reflexionar sobre qué salió bien y qué podría mejorarse en el próximo *sprint*.

Como parte de la propuesta, se sugiere realizar una retrospectiva al finalizar cada *sprint*, en este caso, al finalizar los 30 días para trabajar en las tareas asignadas. Esto, junto con lo recomendado en el PPQA del CMMI, puede ayudar a detectar oportunidades de mejora y áreas de mejora.

Para esto, según *CMMI Institute* (2010), se tiene que evaluar los procesos realizados con base en los estándares, y los procedimientos de los procesos existentes. Si se encuentra un problema relacionado al incumplimiento de los procesos de QA, según el *CMMI Institute* (2010), se sugiere comunicar los problemas de calidad encontrados, ya sea con el personal o con los gerentes, con el fin de garantizar la resolución del problema.

## **5.10. Propuesta de plan de acción de la metodología propuesta**

Contar con una metodología de QC ayuda a tener una metodología formal para el departamento de automatización de QA, sin embargo, este se puede adaptar a otros departamentos u organizaciones, además, ayuda llevar control de las actividades de QA que se realizan.

Con un plan de acción para la metodología de QC propuesta, se podrá contemplar el tiempo necesario para adaptarlo al departamento de *QA Automation* para que este sea acogido en su totalidad, además, en este se listan las tareas necesarias para lograr obtener el resultado deseado, el cual es incorporar la metodología de QC propuesta y hacer modificaciones de ser necesario.

En vista de que la metodología propuesta no es un elemento del que la institución dependa inmediatamente para cumplir con sus funciones, se recomienda comenzar con proyectos pequeños y de poca complejidad únicamente, para poder realizar los ajustes necesarios y así poder ir implementándolo de manera progresiva en proyecto de mayor dificultad y tamaño, esto con el fin de no causar un gran impacto.

A continuación, se detalla las etapas que componen el plan de acción propuesto.

### **5.10.1. Etapas de Plan de Acción**

Como parte de este proyecto, se propone un plan de acción que nos servirá como herramienta para establecer una guía a seguir para la implementación de la metodología de QC propuesta. Este plan de acción está compuesto por etapas que nos ayudarán a alcanzar los objetivos, asimismo, cada etapa tiene un conjunto de tareas y actividades.

El objetivo de esto es poder ejecutar un plan piloto de la metodología de QC propuesta y que, inicialmente, sea ejecutado por el departamento de automatización de aseguramiento de la calidad, todo esto en un periodo de 6 meses como estimación.

Este plan de acción está compuesto por varias etapas que ayudarán a alcanzar los objetivos, asimismo, cada etapa tiene un conjunto de tareas y actividades. Dichas etapas se detallan a continuación.

#### **5.10.1.1. Etapa 1: Inducción**

Como parte del plan de acción, es importante realizar una inducción sobre la metodología de QC propuesta con el fin de que los involucrados, en este caso departamento de *QA Automation*, comprendan lo que se planea realizar y obtener con la aplicación de esta metodología. En este caso se aplicará inicialmente con el departamento de *QA Automation*; con ellos se detallarán las etapas, actividades y entregables involucrados, así como los cambios respecto a la metodología original.

En esta etapa se aplicará el plan piloto con base en la metodología propuesta. Esto consistirá en tomar la propuesta, calendarizar las acciones necesarias y ejecutarla sobre los proyectos seleccionados para así poder monitorear el proceso de QC. Cabe mencionar que, de ser necesario, la metodología se puede modificar para que se adapte a las necesidades del proyecto.

Para la inducción, se estima un periodo de 2 semanas, y se recomienda realizar una reunión con los miembros del equipo de automatización del aseguramiento de la calidad para poder garantizar la comprensión de todos, resolver las dudas en el caso de que se presenten, y definir el plan piloto a seguir.

### **5.10.1.2. Etapa 2: Implementación de la metodología**

Seguido de la etapa de inducción, se procede con la etapa de implementación. En esta etapa se ajustan, añaden y/o modifican los elementos necesarios, en este caso la herramienta para el manejo de proyectos Jira, para poder recopilar las métricas propuestas en la metodología de QC desarrollada, además, se ajustarán los elementos necesarios, tales como los informes propuestos, al contexto actual del departamento de *QA Automation*.

Dichos ajustes se realizarán con base en la retroalimentación brindada en la etapa anterior con el fin de que el plan piloto se pueda ejecutar lo mejor posible para obtener resultados certeros sobre la ejecución de la metodología propuesta.

Para esta etapa se estima una duración de 1 mes ya que se necesitan realizar varias tareas, entre ellas, realizar las correcciones de la metodología según los comentarios realizados por el equipo de *QA Automation*, revisar las métricas actuales para añadir y/o ajustar las métricas propuestas en este proyecto, además de realizar una revisión final de los cambios realizados y cómo estos serán utilizados.

### **5.10.1.3. Etapa 3: Capacitación de las personas involucradas**

Como parte del plan de acción para la implementación de la metodología propuesta, es necesario brindar capacitación a las personas involucradas, en este caso, los miembros del equipo de *QA Automation* y la Gerencia de Cumplimiento de la Calidad de la empresa. Este consistirá en dar una explicación de cómo funciona la metodología, los beneficios que se pueden obtener, y qué recursos son necesarios para poder ponerlo en práctica. A continuación, se listan los recursos principales para la implementación de esta metodología propuesta.

- Jira: Es necesario la disponibilidad de la herramienta de administración de proyectos Jira ya que en este se gestionan los vasos de pruebas automatizados.
- Equipo de QA Automation: Es indispensable la disponibilidad de por lo menos dos miembros del equipo de automatización del aseguramiento de la calidad para poder ejecutar la metodología de QC propuesta e irla ajustando con base en la retroalimentación o puntos de mejora que se observen.
- Herramienta de ejecución de pruebas automatizadas: Es necesario el uso de la herramienta desarrollada por el equipo de QA Automation para la ejecución de las pruebas automatizadas.

Además de poder garantizar que los involucrados ejecuten apropiadamente la metodología, también se podrá obtener retroalimentación sobre esta. Con esto se podrá analizar las mejoras propuestas e implementarlas en la etapa 3 del plan de acción.

Para esta etapa se estima una duración de 2 semanas donde una será dedicado a capacitar a los miembros del equipo de QA Automation y otra será para la capacitación de la Gerencia de Cumplimiento de la Calidad.

#### **5.10.1.4. Etapa 4: Ejecución de la metodología - Departamento de QA Automation**

Después de implementar y validar la correctitud de todos los elementos necesarios para la ejecución de esta metodología, se puede empezar con la ejecución de la metodología de QC propuesta. Esta metodología se aplicará cada vez que se realicen pruebas de integración y regresión de los productos a liberar, esto significa que se realizará cada mes.

La ejecución de esto se limitará inicialmente solo a los fallos encontrados por el equipo de QA Automation en el producto como tal y los fallos encontrados en la herramienta para la ejecución de pruebas. En cada una de las ejecuciones de las pruebas

de integración y regresión para la liberación de un producto, se aplicará la metodología propuesta para llevar seguimiento del desempeño del equipo en relación con la ejecución de las pruebas automatizadas y se revisará la calidad del producto con base en los fallos encontrados por la herramienta de prueba.

En esta etapa del plan de acción, se va a iniciar con la ejecución de la metodología, se realizará una breve reunión de arranque con el objetivo de asegurar que todos los miembros del equipo estén alineados en cuanto a el enfoque de la ejecución de las pruebas automatizadas y sus resultados, asimismo, la preparación de los ambientes de prueba será realizado por el equipo de *Solutions*.

Una vez que estén listos los ambientes, se realiza la ejecución de las pruebas de integración, la monitorización de estas, el reporte de los fallos que se encuentren y su priorización, posteriormente se realiza el mismo proceso para las pruebas de regresión, y finalmente se calcula la calidad del producto con base en los *bugs* encontrados.

Para esta etapa se estima 1 mes de plazo con el fin de realizar los ajustes necesarios, de forma gradual en las primeras ejecuciones de la metodología propuesta.

#### ***5.10.1.5. Etapa 5: Evaluación y reforma de la metodología propuesta - Departamento de QA Automation***

Después de cada ejecución de la metodología de QC, es decir, cada vez que se realizan las pruebas de integración y de regresión como parte del proceso de QC para la liberación de un producto. Al monitorear la ejecución de dichas pruebas, se realizará una evaluación de los resultados de las métricas y de las pruebas para revisar si son congruentes respecto a los resultados actuales de las pruebas ejecutadas.

En esta etapa se evalúa lo que se realizó de forma correcta y lo que se podría mejorar en la metodología de QC propuesta. Una vez que se obtiene la retroalimentación, se harán los ajustes propuestos en la metodología.

Con esto se logrará incentivar la mejora continua del proceso de QC y de la metodología de QC propuesta, y se podrá mejorar los procesos, además de garantizar la eficiencia en las actividades.

Al igual que en la etapa 3, este se ejecutará por 1 mes después de la ejecución de la metodología de QC, asimismo, se sugiere realizar una reunión con los miembros del equipo de *QA Automation* para planificar cómo abordar los ajustes a realizar a la metodología.

#### ***5.10.1.6. Etapa 6: Traspaso de conocimiento a 1 equipo de QA del área de desarrollo de juegos***

Es fundamental llevar a cabo el traspaso de conocimiento, que es una etapa clave dentro del plan de acción. Esta fase abarca diversas actividades que se desarrollan en etapas anteriores y se aplicará a un equipo de QA del área de desarrollo de juegos. En esta etapa se debe realizar la capacitación de los analistas de QA del equipo de desarrollo de juegos seleccionado, mismas actividades que abarca la etapa 3.

Posteriormente, al igual que en la etapa 4, se lleva a cabo la ejecución de la metodología, que incluye la reunión de arranque para la ejecución de pruebas, la preparación de los ambientes de prueba, la ejecución de las pruebas, la monitorización del proceso, la priorización de los *bugs* identificados en las pruebas de regresión, y el cálculo de la calidad del producto basado en los *bugs* encontrados, ya sea que estos estén resueltos o no.

Finalmente, con base en la etapa 5, se procede con el análisis de la ejecución de la metodología de QC, la evaluación de resultados, la obtención de retroalimentación y la modificación de la metodología de QC propuesta.

Para esta etapa se estima 2 meses de tiempo para ajustar la metodología al equipo de QA del departamento de desarrollo de juegos y para realizar las reformas respectivas con base en los puntos de mejora que se detecten después de la primera ejecución.

#### ***5.10.1.7. Etapa 7: Traspaso de conocimiento a los demás equipos de QA del área de desarrollo de juegos***

Después de completar esta etapa de traspaso de conocimiento descrito en la etapa 6 del plan de acción, se procede con la etapa 7, el cual consiste en el traspaso de conocimiento a los demás equipos de QA del departamento de desarrollo de juegos, el cual está compuesto por distintos equipos de desarrollo. Esta nueva etapa seguirá los mismos pasos establecidos, incorporando los ajustes realizados a la metodología durante la fase anterior.

Al igual que la etapa 6, se estima una duración de 2 meses para el traspaso de conocimiento al demás equipo de QA del área de desarrollo de juegos.

### **5.10.2. Cronograma Propuesto**

Con base en las etapas propuestas como parte del plan de acción para la implementación de la metodología de QC propuesta, a continuación, se muestra una tabla con el cronograma que se sugiere seguir.

**Tabla 16.** Cronograma propuesto para la ejecución del plan de acción

<i>Etapa</i>	<i>Actividad</i>	<i>Plazo</i>	<i>Plazo Total</i>
<b>Etapa 1:</b> Inducción	Reunión inicial de inducción sobre la metodología.	1 semana	2 semanas
	Inicio de la aplicación del Plan Piloto.	1 semana	
<b>Etapa 2:</b> Implementación de la metodología	Análisis de retroalimentación recibida.	1 semana	1 mes
	Implementación de elementos necesarios de la metodología.	3 semanas	
	Evaluación de elementos implementados.	1 semana	
<b>Etapa 3:</b> Capacitación de las personas involucradas	Capacitación a los miembros del departamento de <i>QA Automation</i> .	1 semana	2 semanas
	Capacitación a la Gerencia de Cumplimiento de la Calidad.	1 semana	
<b>Etapa 4:</b> Ejecución de la metodología por <i>release</i>	Reunión de arranque de ejecución de pruebas.	1 semana	1 mes
	Preparación de los ambientes de prueba.	1 semana	
	Ejecución de pruebas de integración.	2 días	
	Monitorización de la ejecución de las pruebas de integración.	2 días	
	Priorización de los <i>bugs</i> encontrados en las pruebas de integración.	1 día	

	Ejecución de pruebas de regresión.	2 días	
	Monitorización de la ejecución de las pruebas de regresión.	2 días	
	Priorización de los <i>bugs</i> encontrados en las pruebas de regresión.	1 día	
	Cálculo de la calidad del producto con base en los <i>bugs</i> encontrados (ya sea que estén resueltos o no).	1 día	
<b>Etapa 5:</b> Evaluación y reforma de la metodología propuesta	Análisis de la ejecución de la metodología de QC y evaluación de resultados.	1 semana	
	Obtención de retroalimentación.	1 semana	1 mes
	Modificación de la metodología de QC propuesta.	2 semanas	
<b>Etapa 6:</b> Traspaso de conocimiento a 1 equipo de QA del área de desarrollo de juegos	Capacitación de las personas involucradas.	2 semanas	2 meses
	Ejecución de la metodología.	3 semanas	
	Evaluación y reforma de la metodología propuesta.	3 semanas	
<b>Etapa 7:</b> Traspaso de conocimiento a los demás equipos de QA del área de desarrollo de juegos	Capacitación de las personas involucradas.	2 semanas	2 meses
	Ejecución de la metodología.	3 semanas	
	Evaluación y reforma de la metodología propuesta.	3 semanas	

---

---

	<b>Total</b>	8 meses
--	--------------	---------

---

Cabe mencionar que los tiempos propuestos para cada etapa del plan de acción pueden ser ajustados posteriormente según las necesidades y el contexto de la organización.

A continuación, se muestra un diagrama del plan de acción propuesto.

Figura 10. Diagrama del plan de acción para la implementación de la metodología de QC propuesta

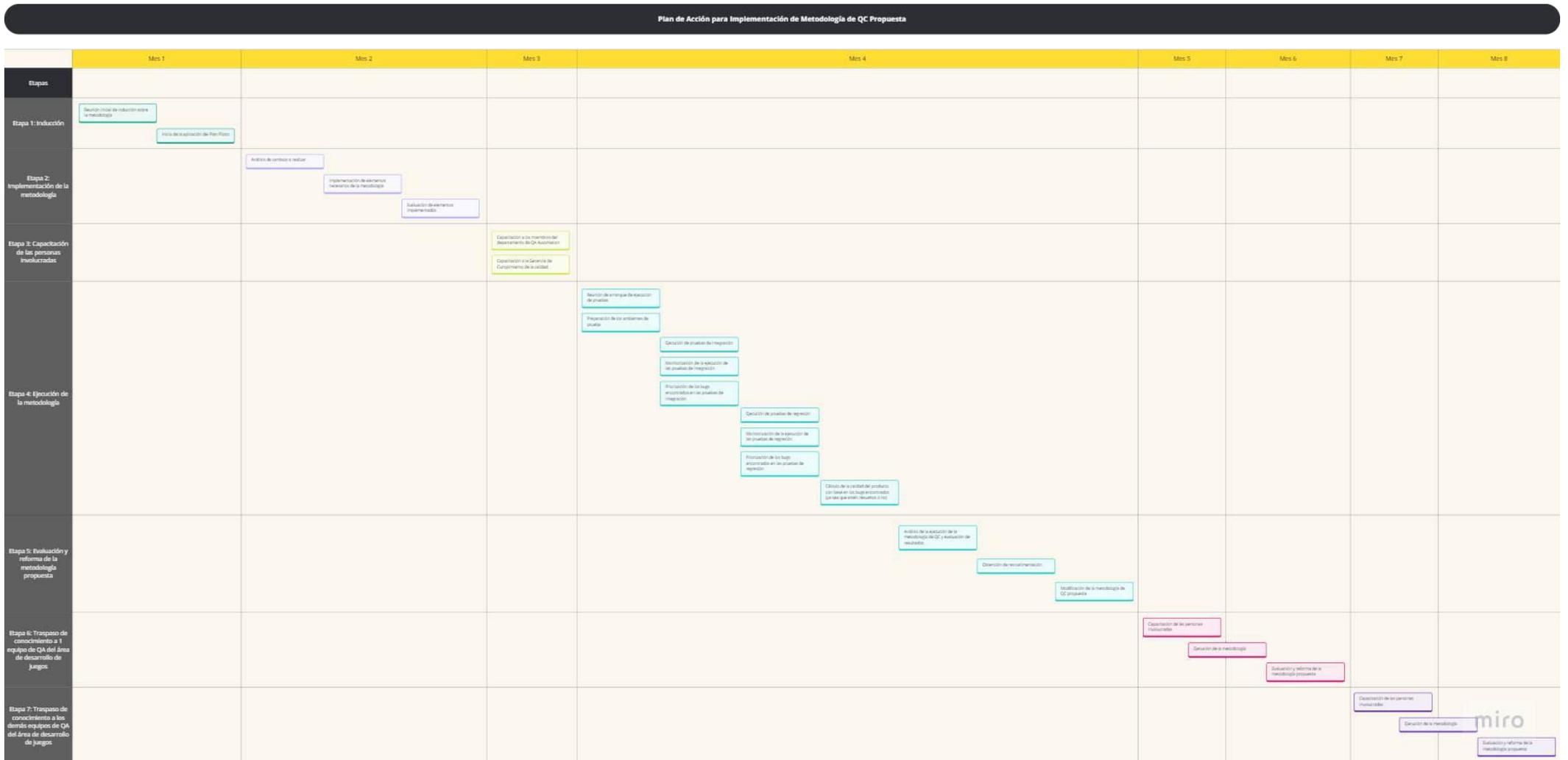


Figura 13. Parte 1 del diagrama del plan de acción para la implementación de la metodología de QC propuesta

**Plan de Acción para Implementación de Metodología de QC Propuesta - Parte 1**

	Mes 1	Mes 2	Mes 3
<b>Etapas</b>			
<b>Etapa 1: Inducción</b>	<p>Reunión inicial de inducción sobre la metodología</p> <p>Inicio de la aplicación del Plan Piloto</p>		
<b>Etapa 2: Implementación de la metodología</b>		<p>Análisis de cambios a realizar</p> <p>Implementación de elementos necesarios de la metodología</p> <p>Evaluación de elementos implementados</p>	
<b>Etapa 3: Capacitación de las personas involucradas</b>			<p>Capacitación a los miembros del departamento de QA Automation</p> <p>Capacitación a la Gerencia de Cumplimiento de la calidad</p>

**Figura 16.** Parte 2 del diagrama del plan de acción para la implementación de la metodología de QC propuesta

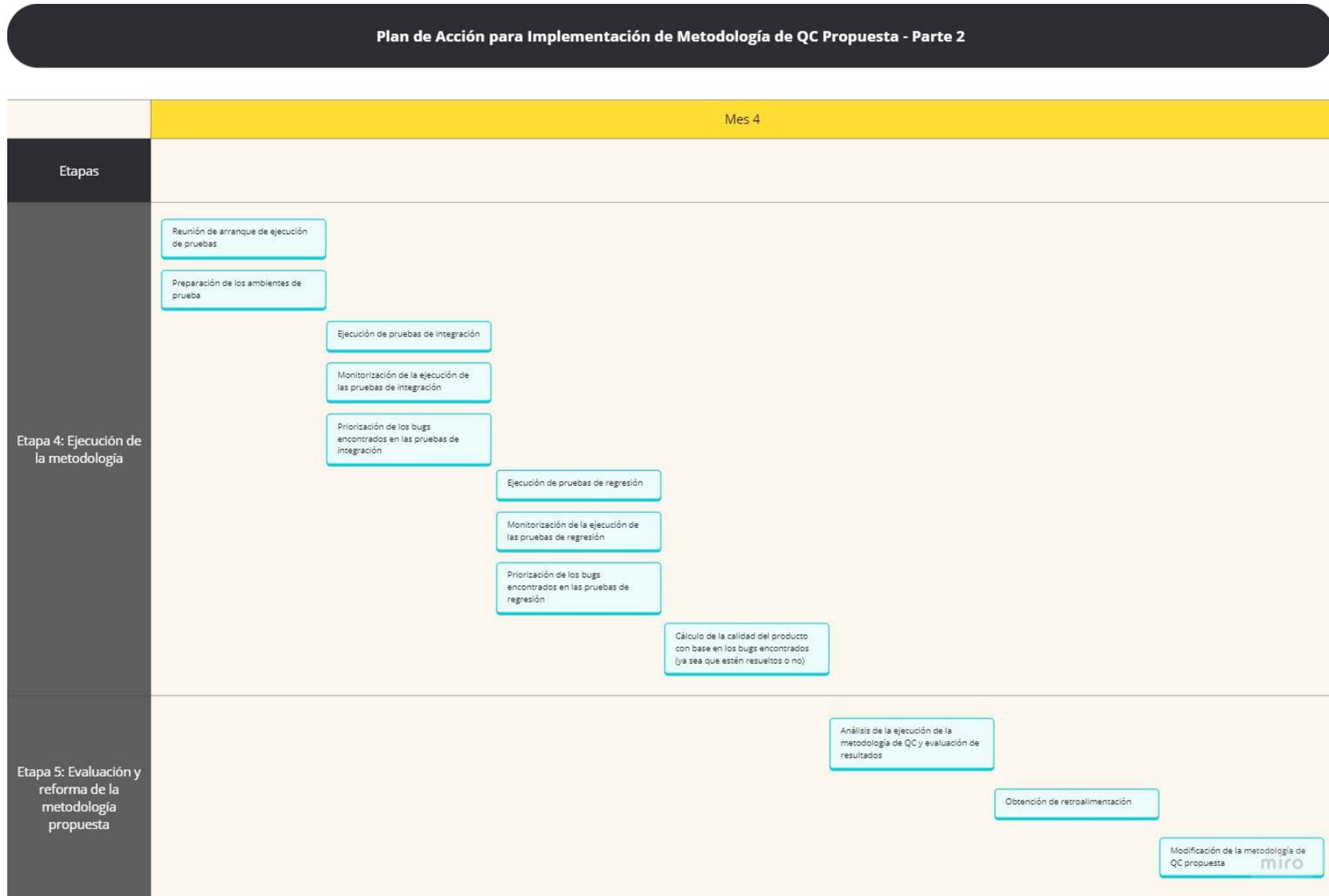


Figura 19. Parte 3 del diagrama del plan de acción para la implementación de la metodología de QC propuesta

**Plan de Acción para Implementación de Metodología de QC Propuesta - Parte 3**

	Mes 5	Mes 6	Mes 7	Mes 8
Etapas				
Etapa 6: Traspaso de conocimiento a 1 equipo de QA del área de desarrollo de juegos	<p>Capacitación de las personas involucradas</p>	<p>Ejecución de la metodología</p> <p>Evaluación y reforma de la metodología propuesta</p>		
Etapa 7: Traspaso de conocimiento a los demás equipos de QA del área de desarrollo de juegos			<p>Capacitación de las personas involucradas</p> <p>Ejecución de la metodología</p>	<p>Evaluación y reforma de la metodología propuesta <i>miro</i></p>

### 5.10.3. Factores críticos de éxito

Después de definir las etapas del plan de acción para implementar la metodología de QC propuesta, además de su aplicación, hay que considerar algunos puntos que nos ayudarán a identificar si la implementación ha sido exitosa. A continuación, se detallan los factores críticos de éxito.

- Implementación de la metodología de QC propuesta para el departamento de *QA Automation*. Este es un factor importante ya que se espera que se haya implementado dicha metodología en su totalidad ya que con esto se busca poder monitorear los procesos de QA del departamento de *QA Automation*, además de encontrar oportunidades de mejora para el mismo.
- Los *stakeholders* logran tener mejor visibilidad del estado actual de la ejecución de las pruebas, se logra conocer la calidad del producto, y se tiene disponibilidad de múltiples métricas que se pueden utilizar para el análisis de datos.
- Con los datos generales, será posible generar gráficos que resuma los datos para que estos sean compartidos con los *stakeholders*.
- Un factor de éxito también es poder obtener retroalimentación constante cada vez que se ejecuta la metodología, o de una forma periódica, para así poder mejorar continuamente.
- En relación con la obtención de retroalimentación, se considera un factor de éxito que los miembros del equipo de *QA Automation* busquen la mejora continua de los procesos de QA actuales mediante el análisis de los datos generados y la mejora de la metodología de QC propuesta.

## 6. Conclusiones y recomendaciones

En esta sección se presentan las conclusiones y recomendaciones obtenidas una vez desarrollada la propuesta de solución. Con base en los resultados obtenidos, se discuten las principales fortalezas y debilidades de la metodología propuesta, así como los posibles riesgos en su implementación. Además, se discuten las implicaciones prácticas de las herramientas y riesgos del proyecto, para finalmente presentar recomendaciones para futuras investigaciones, mejoras en la gestión de proyectos y trabajos pendientes

### 6.1. Conclusiones

A lo largo del desarrollo de este proyecto, se realizó una investigación sobre distintos marcos de referencia, como parte del marco teórico, que se podían utilizar y adaptar al contexto actual de la empresa en la que se realizó el proyecto para poder desarrollar la propuesta de una metodología de QC para el departamento de automatización del aseguramiento de la calidad.

También se recolectaron datos y se analizaron para poder obtener el estado actual del departamento. Con base en el marco teórico y la información recolectada de sobre el proceso actual de QA del equipo de *QA Automation*, se desarrolló una propuesta de metodología de QC junto con su plan de acción para su implementación.

Dicho plan de acción permite implementar la metodología de QC propuesta en un tiempo mínimo de 8 meses dividido en 7 etapas. Este plan de acción permitirá realizar los ajustes necesarios para que se ajuste a la situación actual de la Empresa A empezando con un grupo pequeño, en este caso el equipo de *QA Automation*, y después con un grupo más grande como los analistas de QA del equipo de desarrollo de juegos.

Gracias al desarrollo de este proyecto, como parte de los resultados, se elaboró una documentación detallada sobre el proceso actual de QA en el departamento de *QA Automation* para la utilización de las pruebas automatizadas que se han desarrollado (ver sección 4.2.1) gracias a la aplicación de entrevistas a los miembros del departamento a la Gerencia de Cumplimiento de la Calidad.

Esto no solamente significa que se generó el primer documento formal que describe la metodología de QC del departamento, sino que nos ayudó a tener mejor visibilidad de los puntos de mejora a incluir en la propuesta y ayudó a brindar una mejor comprensión de cómo funciona.

Con base en el análisis realizado, para esta metodología se propuso la ejecución de los mismos tipos de pruebas que se utilizan actualmente: pruebas de regresión, de integración y de humo. Esto de acuerdo con las pruebas que se aplican actualmente por el departamento de *QA Automation* y el uso de las pruebas automatizadas por otros departamentos.

- El uso de las pruebas de integración son el primer conjunto de pruebas que se ejecuta cuando comienza el periodo de liberación de un producto. La ejecución de este tipo de pruebas permite agilizar la revisión de las interfaces e interacciones entre componentes integrados (ISTQB *Glossary*, s.f.), lo que permite a los analistas de QA de otros departamentos enfocarse en otras tareas de prioridad.
- El uso de las pruebas de regresión son el segundo conjunto de pruebas que se ejecutan cuando comienza el periodo de liberación de un producto, La ejecución de estas pruebas ayudan a poder detectar si se han introducido o descubierto defectos en zonas no modificadas del *software* (ISTQB *Glossary*, s.f.).

- Las pruebas de humo son de gran utilidad ya que los analistas de QA lo utilizan constantemente durante el desarrollo de un producto para asegurar que los cambios introducidos no afectan las funcionalidades principales de un componente o sistema (ISTQB *Glossary*, s.f.).

Otro punto importante que resaltar es que logró desarrollar una propuesta de metodología de QC basada en la información recopilada al aplicar entrevistas a distintos expertos (ver Anexo 7) en el tema del aseguramiento de la calidad y la aplicación de pruebas automatizadas, y las mejores prácticas propuestas por distintos marcos de referencia, por ejemplo, el PMBOK y la ISTQB. Esto ayudó a poder desarrollar una metodología que se acerque a la forma de trabajar de otras industrias en relación con el QA.

A partir del trabajo realizado, se destaca que es importante mantener actualizada la metodología de QC del departamento para garantizar la mejora continua del proceso y generar el mayor valor posible para el equipo.

### **6.1.1. Beneficios de la metodología de QC propuesta**

Al desarrollar la propuesta de metodología de QC, se pudo notar algunos beneficios que se podrían obtener con su aplicación. Estos se listan a continuación:

- Al poder disponer de una metodología formal de QA como esta, se puede tener un mejor seguimiento de la ejecución de las pruebas, ya sean las de regresión, las de integración, o un conjunto de pruebas en específico, ya que con esto se puede generar una base de datos que se puede analizar.
- Con la generación de métricas gracias a la aplicación de esta metodología, se puede evaluar de forma más objetiva al poder tener la posibilidad de conocer

datos como la cobertura actual de las pruebas, tener medidas estáticas, entre otros.

- Esto también permite tener un acceso más fácil a la información sobre las pruebas dado que, por ejemplo, se pueden generar gráficos para la representación de los datos.
- Otro beneficio de relevancia y de gran utilidad que tiene esta metodología propuesta es que permite revisar la calidad de los productos de forma granular, esto significa que los productos a evaluar se pueden ajustar y aplicar a productos grandes como el desarrollo de un juego, y hasta el desarrollo de una nueva funcionalidad.
- El poder disponer con las métricas propuestas en la metodología, se podrá tener un mejor seguimiento del rendimiento del equipo de *QA Automation* a la hora de ejecutar pruebas automatizadas.
  - Dichas métricas nos ayudarán a poder categorizar las pruebas por tipo de prueba y ver cuáles dan resultados que no son correctos y que necesitan ser trabajados, por ejemplo, pruebas que den falsos positivos o pruebas que fallan constantemente, aunque todo esté configurado correctamente, asimismo, se podrá visualizar si la mayoría de los errores son presentes por errores de configuración o si son del *software* como tal.

### **6.1.2. Riesgos potenciales del proyecto**

Algunos de los riesgos potenciales que se podrían dar con la utilización de la metodología de QC propuesta como herramienta de apoyo para la ejecución de pruebas son las siguientes.

- El tiempo, costo y esfuerzo para la introducción inicial de la metodología de QC propuesta puede llegar a ser subestimado ya que esta etapa puede involucrar la capacitación del personal y temas relacionados a la gestión del cambio.
- El tiempo y el esfuerzo necesarios para lograr beneficios significativos y continuos de la herramienta pueden ser subestimados, por ejemplo, puede darse la necesidad de realizar cambios en el proceso de prueba o puede dedicarse más recursos de los esperados para mejorar la forma en la que se utiliza la metodología de QC propuesta.

### **6.1.3. Trabajo a futuro**

Con base en el análisis y desarrollo de la metodología de QC propuesta para el departamento de *QA Automation* de la Empresa A, se proponen varios trabajos a futuro que se pueden realizar como parte de las mejoras a la herramienta.

#### **6.1.3.1. Incorporación de IA para asignación de prioridad a los bugs**

Un trabajo a futuro es la incorporación de IA (Inteligencia Artificial) para validar y asignar el nivel de prioridad (alta, baja, media, entre otros) de un *bug* encontrado durante la ejecución de pruebas. Para ello, se plantea como propuesta inicial usar Jira para implementar una automatización que desencadene el proceso para calcular el nivel de prioridad de los *bugs*, y con esos datos generados, poder calcular automáticamente el nivel de calidad de un producto.

#### **6.1.3.2. Implementación de “falsos negativos” como métrica**

Con relación a las métricas propuestas, se propone como trabajo a futuro la implementación de “falsos negativos” como parte de las métricas de la metodología ya que al ser la primera metodología de prueba formal que tiene el departamento de *QA*

*Automation*, es necesario priorizar la cantidad de métricas que desean implementar dado que esto significa un gran cambio en la organización.

Adicionalmente, se optó por añadir dicha métrica en un trabajo a futuro con base en lo comentado por el experto Luis Sanabria (ver Anexo 9.7.5), quien explica que esta métrica es importante, pero tiene un alcance limitado; esto debido a que el público meta es la supervisión directa del equipo de automatización de pruebas para controlar la calidad de las pruebas automatizadas, sin embargo, podría ser confuso para otros puestos de jefatura y/o equipos. Esto significa que se necesita realizar un análisis sobre la importancia y beneficios que puede brindar esta métrica.

#### **6.1.3.3. *Generación automática de informes***

Como parte de la metodología de QC propuesta, se propuso la generación de tres informes en distintas etapas durante la ejecución de las pruebas de integración y/o regresión, los cuales son: Informe de Inicio de Ejecución de Pruebas Automatizadas, Informe de Avance de Ejecución, e Informe de Resultados Finales

Inicialmente, estos informes serán elaborados de forma manual con el fin de poder ajustarlo según las necesidades que se vayan detectando en cada iteración en la que se aplique la metodología. Sin embargo, se considera como un trabajo a futuro la automatización de la generación de dichos informes, utilizando herramientas como Jira, la cual es actualmente la herramienta de administración de proyectos utilizada en la empresa y usada para el reporte de errores.

#### **6.1.3.4. *Ejecución de pruebas automatizadas en la noche***

Un trabajo a futuro útil es poder automatizar la ejecución de pruebas de humo, regresión, o integración en las noches. Esto ayudaría a ahorrar tiempo en la preparación de los ambientes y en la ejecución de pruebas, esto debido a que los ambientes se podrán

probar a primera hora apenas se inicie la jornada laboral. Otra ventaja es que la ejecución de pruebas en la noche también permite tener resultados más fiables debido a que los ambientes no serán afectados por alguna otra persona que esté usando el ambiente de pruebas.

## **6.2. Recomendaciones**

Como parte de las recomendaciones para la implementación de esta metodología de QC, es fundamental mantener el documento que detalla la metodología de QC, en este caso del departamento de *QA Automation*, lo más completo y actualizado posible. Este documento no solo especifica la metodología, sino que también sirve para establecerla como una guía accesible para los colaboradores de la empresa. De esta manera, tanto el equipo de automatización de aseguramiento de calidad como miembros de otros departamentos podrán consultarlo y adaptarlo a sus propios procesos de QA.

Adicionalmente al punto anterior es necesario llevar un buen manejo de las versiones del documento para poder tener conocimiento de qué tan desactualizado está con base en el estado actual de la metodología de QC y/o la herramienta para la ejecución de pruebas automatizadas.

Se recomienda también llevar control de todos los informes que se generen con la aplicación de esta metodología. Esto permitirá poder tener trazabilidad entre los fallos que se reportan en la ejecución de las pruebas, ya sea de integración y/o integración, y las pruebas automatizadas, lo que facilitará llevar seguimiento del comportamiento de la metodología.

## 7. Glosario

A continuación, se presenta una lista de vocablos con sus respectivos significados, cuyo objetivo es lograr un adecuado entendimiento de este documento.

- 1. Bugs:** Una falla en un componente o sistema que puede causar que el componente o sistema no realice su función requerida, por ejemplo, una declaración o definición de datos incorrecta. Un defecto, si se encuentra durante la ejecución, puede causar una falla del componente o sistema. (ISTQB, 2023)
- 2. DevOps:** Es un proceso de desarrollo de *software* y un cambio de la cultura empresarial que acelera la entrega de *software* de alta calidad mediante la automatización e integración de los esfuerzos de los equipos de desarrollo y operaciones de TI (Tecnología de Información). (IBM, 2019)
- 3. Hotfix:** Este es el lanzamiento rápido de un parche de *software* que normalmente se aplica a un sistema activo sin necesidad de reiniciar. Las revisiones solucionan errores críticos o vulnerabilidades de seguridad que requieren atención inmediata. (Devaraj, 2023)
- 4. Merge:** Es un comando de Git que se utiliza para combinar dos ramas de un repositorio. Esto significa que Git encuentra una confirmación base en común y crea una “confirmación de fusión” nueva que combina los cambios de cada secuencia de confirmación de fusión puesta en cola. (Atlassian, 2024)
- 5. Release:** Es un conjunto de cambios que actualizan o agregan nuevas funciones a un producto o servicio de *software*. (Dodd, 2023)

## 8. Bibliografía

Atlassian (2024). *Git merge*. Atlassian. Recuperado el 21 de julio de 2024 de

<https://www.atlassian.com/es/git/tutorials/using-branches/git-merge>

Bose S. (20 de junio de 2023). *How to set up QA processes from scratch*. BrowserStack.

<https://www.browserstack.com/guide/setup-qa-process>

Browserstack. (2024). *What is Test Case Prioritization*. [https://www.browserstack.com/test-](https://www.browserstack.com/test-management/features/test-run-management/what-is-test-case-prioritization#:~:text=based%20test%20prioritization-)

[management/features/test-run-management/what-is-test-case-](https://www.browserstack.com/test-management/features/test-run-management/what-is-test-case-prioritization#:~:text=based%20test%20prioritization-)

[prioritization#:~:text=based%20test%20prioritization-](https://www.browserstack.com/test-management/features/test-run-management/what-is-test-case-prioritization#:~:text=based%20test%20prioritization-)

[.Coverage%2Dbased%20prioritization,more%20likely%20to%20contain%20defects.](https://www.browserstack.com/test-management/features/test-run-management/what-is-test-case-prioritization#:~:text=based%20test%20prioritization-)

CMMI Institute. (2010). *CMMI for Development, Version 1.3*. Carnegie Mellon University.

[https://insights.sei.cmu.edu/documents/853/2010\\_005\\_001\\_15287.pdf](https://insights.sei.cmu.edu/documents/853/2010_005_001_15287.pdf)

Devaraj, K. (27 de diciembre de 2023). *What is a Hotfix? | Benefits , Challenges & How to*

*Test?* Testsigma. Recuperado el 21 de julio de 2024 de

<https://testsigma.com/blog/what-is-hotfix/>

Dodd, R. (febrero de 2023). *Guide: What Is a Software Release?* LaunchDarkly.

<https://launchdarkly.com/blog/guide-what-is-a-software-release/>

Fiorenza I. (24 de marzo de 2024). *Control de Calidad en software qué es y qué hace*. W&B

Asset Studio. Recuperado el 1 de septiembre de 2024 de

[https://www.wbassetstudio.com/blog/control-de-calidad-en-software-que-es-y-que-](https://www.wbassetstudio.com/blog/control-de-calidad-en-software-que-es-y-que-hace/)

[hace/](https://www.wbassetstudio.com/blog/control-de-calidad-en-software-que-es-y-que-hace/)

IBM. (2019). *¿Qué es DevOps?* IBM. <https://www.ibm.com/mx-es/topics/devops>

International Organization for Standardization. (2024). *ISO/IEC/IEEE 29119 series*.

Recuperado el 26 de marzo de 2024 de

<https://committee.iso.org/sites/jtc1sc7/home/projects/flagship-standards/isoiecieee-29119-series.html>

International Organization for Standardization Glossary. (2023). Bug. ISTQB Glossary.

<https://istqb->

[glossary.page/bug/#:~:text=A%20flaw%20in%20a%20component,of%20the%20component%20or%20system](https://istqb-glossary.page/bug/#:~:text=A%20flaw%20in%20a%20component,of%20the%20component%20or%20system).

International Organization for Standardization Glossary. (2024). *Prueba de Integración de*

*Componentes*. Recuperado el 6 de Agosto de 2024 de

[https://glossary.istqb.org/es\\_ES/term/prueba-de-integracion-de-componentes](https://glossary.istqb.org/es_ES/term/prueba-de-integracion-de-componentes).

International Organization for Standardization Glossary. (s.f.). *Prueba de humo*. Recuperado

el 6 de Agosto de 2024 de [https://glossary.istqb.org/es\\_ES/term/prueba-de-humo](https://glossary.istqb.org/es_ES/term/prueba-de-humo)

International Organization for Standardization Glossary. (s.f.). *Prueba de Regresión*.

Recuperado el 6 de Agosto de 2024 de [https://glossary.istqb.org/es\\_ES/term/prueba-de-regresion](https://glossary.istqb.org/es_ES/term/prueba-de-regresion)

ISO/IEC, IEEE. (2016). *ISO/IEC/IEEE 29119-5*. Institute of Electrical and Electronics

Engineers, Inc. <https://ieeexplore-ieee->

[org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9591574](https://ieeexplore-ieee-org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9591574)

ISO/IEC, IEEE. (2021). *ISO/IEC/IEEE 29119-3*. Institute of Electrical and Electronics

Engineers, Inc. <https://ieeexplore-ieee->

[org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=6588543](https://ieeexplore-ieee-org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=6588543)

ISO/IEC, IEEE. (2021) *ISO/IEC/IEEE 29119-4*. Institute of Electrical and Electronics Engineers, Inc. <https://ieeexplore-ieee->

[org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9591577](https://ieeexplore-ieee-org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9591577)

ISO/IEC, IEEE. (2022). *ISO/IEC/IEEE 29119-1*. Institute of Electrical and Electronics Engineers, Inc. <https://ieeexplore-ieee->

[org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=7750539](https://ieeexplore-ieee-org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=7750539)

ISO/IEC, IEEE. (2022). *ISO/IEC/IEEE 29119-1*. Institute of Electrical and Electronics Engineers, Inc. <https://ieeexplore-ieee->

[org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9698145](https://ieeexplore-ieee-org.ezproxy.itcr.ac.cr/stamp/stamp.jsp?tp=&arnumber=9698145)

Kyrylo, I. (s.f.). *All There Is To Know About The Best Testing And QA Methodologies*.

<https://jelvix.com/blog/qa-testing->

[methodologies#:~:text=What%20is%20a%20QA%20methodology,approaches%20for%20all%20engineering%20process.](https://jelvix.com/blog/qa-testing-methodologies#:~:text=What%20is%20a%20QA%20methodology,approaches%20for%20all%20engineering%20process.)

MacNeil C. (7 de febrero de 2024). *Guía de retrospectiva Scrum: qué es y cómo hacerla*.

Asana. <https://asana.com/es/resources/sprint-retrospective>

Martins, J. (14 de noviembre de 2022). *Qué es un Project roadmap y cómo crear uno paso a paso*. Asana. <https://asana.com/es/resources/project-roadmap>

Real Academia Española. (2023). Metodología. En Diccionario de la lengua española.

Recuperado el 20 de octubre, 2023, de <https://dle.rae.es/>

Rodrigues, N. (20 de enero de 2023). *PMBOK: qué es, para qué sirve, fases y herramientas*.

HubSpot. Recuperado el 26 marzo de 2024 de <https://blog.hubspot.es/sales/que-es-pmbok>

Silva, D. D. (27 de junio de 2022). *Blog de Zendesk*.

<https://www.zendesk.com.mx/blog/indicadores-calidad-empresa/>

# 9. Anexo

## 9.1. Anexo 1

### **Meeting Notes**

**Project:** QA Methodology Proposal for the QA Automation Department

**Date:** August 29, 2023

**Consecutive:** 001

**Attendees:**

- [REDACTED] – Director of Business Intelligence
- Carmen Mok Zheng – Junior QA Automation Engineer

**Starting time:** 3:50 p.m.

**End Time:** 4:00 p.m.

The meeting was held in the company where the project will be carried out, with the participation of Carmen Mok Zheng and Marco Varela, the Director of Business Intelligence.

The following topics were discussed:

1. The following question was asked to the director of business intelligence:

“If you had metrics regarding the QA Automation team, which question would you like to be able to answer with them?”

2. It was answered the following points:

- a. It would be good to be able to create a "Project tracking report" with which you can see:
  - i. The errors and defects.
  - ii. At what stage a defect was detected (design stage, development, QA, beta, among others).
  - iii. Number of errors detected per stage.
  - iv. If defects are increasing or decreasing.
  - v. If there were more defects per game this month or not.
  - vi. The difference between the number of defects according to the team and the game that the team is working on.
  - vii. Defects by origin, stage or cause (design, development or QA problem).
  - viii. Different methodologies between teams
- b. It would be good to be able to analyze the performance of the methodology used between teams.
- c. How many bugs are we detecting in regression?
- d. Why are some bugs going into production?
- e. What is the percentage of test cases in regression that are automated?

Agreements	Responsible	Deadline
The information obtained in this meeting will be used for the development of the project.	Carmen Mok Zheng	June, 2024

Signatures:



Director of Business Intelligence

Handwritten signature of Carmen Mok Zheng in black ink.

Carmen Mok Zheng  
Junior QA Automation Engineer

## 9.2. Anexo 2

### Minuta de reunión

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 22 de septiembre del 2023

**Consecutivo:** 002

**Convocados:**

- Adriana Álvarez Figueroa – Profesora Asesora
- Rodolfo Mora Zamora - CEO de Ceiba Software & Arts

**Hora de Inicio:** 01:00 p.m.

**Hora de Finalización:** 02:30 p.m.

La reunión se realizó de forma presencial, con la participación de Adriana Álvarez Figueroa y CEO de Ceiba Software & Arts.

En este se trataron los siguientes puntos:

- La profesora Adriana comienza preguntando si se utiliza alguna herramienta especial para lo que es la administración de proyectos y el reporte de *bugs* en la ejecución de pruebas para los juegos. Para esto, el CEO Rodolfo responde que se utilizan herramientas tradicionales de administración de proyectos como Jira para reportar *bugs* y explica que para "*play testing*" se utiliza la misma herramienta.
- Rodolfo también explica que las pruebas de los juegos en desarrollo son igual a las pruebas de "*unit testing*" manual en un sistema, lo cual consiste en usar el sistema y asegurarse de que lo que debe funcionar, funcione. Rodolfo también menciona que la gran diferencia con el *testing* de los juegos es que hay más cosas que pueden fallar, por ejemplo, simulación de físicas o más elementos móviles, por lo que se tienen que probar aún más elementos del producto en desarrollo.
- Rodolfo explica las siguientes tipos de pruebas que se ejecutan en su empresa:
  - *Compatibility Testing*
    - Esto se hace cuando se tiene que probar algo con un dispositivo fuera de lo común. En PC (computadora personal), uno utiliza un dispositivo con un rendimiento lo más bajo posible y a partir de ahí, se ve si los juegos funcionan desde ese dispositivo como base, hasta otro con mejores características para observar su rendimiento. Hay demasiadas variables que considerar a la hora de realizar este tipo de pruebas, por ejemplo, tarjetas de video, diferentes posibilidades en operadores para renderizado, video, entre otros.
    - Este tipo de pruebas también se realizan con juegos que sean en dispositivos móviles, por lo que hay que probarlo con sistemas vigentes

- de sistemas operativos, buscar la gama de dispositivos que vayan a funcionar, entre otros.
- El *compatibility testing* es importante cuando la empresa trabaja con distintas consolas, por ejemplo, el Switch. Las consolas pueden ser muy restringidas y tener bibliotecas que no funcionen correctamente. En el caso de las computadoras, todo puede funcionar, pero en las consolas es diferente.
  - Normalmente si la empresa tiene los *DevKits* (utilizado para simular la capacidad de una consola), el proceso de *porting* (pasar el juegos a la consola), consiste en hacer *build*.
  - Para el *compatibility testing*, se dedica mes o mes y medio por juego y por consola.
- *Performance testing*
    - Hay herramientas especiales para esto. La herramienta a utilizar depende de dónde se vaya a utilizar el juego, factor que puede variar. Por ejemplo, un *engine*.
    - Si este tipo de prueba es realizada en un ambiente web, este dispone de herramientas de *performance*.
    - En el desarrollo de juegos existe lo que son los *engines*, que son un como *framework* que consiste en un conjunto de bibliotecas que tienen funciones genéricas que se van a utilizar en un videojuego, por ejemplo, funciones de renderizado, de física, de *input*, entre otros. El *engine* trae herramientas para poder programar encima de lo mencionado, se puede utilizar su API, y trae un editor donde se puede trabajar más cómodamente. Dicho editor funciona parecido a un IDE, sin embargo, el editor de un *engine* tiene más *features* (características o funciones) porque trabajan con la importación de archivos especializados, entre otros.
    - Los *engines* disponen de herramientas para *performance*, que son más especializadas porque dependen mucho del API que está corriendo por debajo.
- Adriana pregunta si en la empresa se realizan pruebas automatizadas. A esto, Rodolfo responde que en su empresa no. Además añade que la realización de pruebas automatizadas no es común y que desconoce de alguien que las realice, no obstante, explica que si hacen y que los mismos *engines* tienen herramientas para lo que es *Automated Testing* (Automatización de pruebas) que se pueden utilizar.
  - Adriana pregunta ¿Aproximadamente de qué tamaño son los equipos de QA? El CEO Rodolfo explica que los tamaños de los equipos de QA son pequeños. Estos son aproximadamente de 2-3 personas por juego en una empresa. Siempre se trata que sea un equipo diferente del de desarrollo y el de diseño, en especial en juegos ya que estos

tienen muchos elementos de diseño, por lo que no es conveniente que sean los mismos diseñadores que hagan la revisión del producto.

- Rodolfo menciona que Unity es un *engine* muy popular.
- Adriana pregunta si hay alguna serie de pasos o algo que se considere estándar para las pruebas en juegos. Rodolfo responde que eso ya es algo muy específico del juego; cada consola tiene un manual con las especificaciones y los tipos de pruebas que se tienen que hacer, por lo que se sigue el manual que el proveedor de la consola da. Para web no hay algo en específico.
- Adriana pregunta qué es un *profiler*. Rodolfo explica que un *profiler* (en Unity) es un componente del *engine* que hace que mientras el juego corra, se logre analizar todas las fuentes del consumo de recursos del juego y genere un perfil de rendimiento. Además, menciona que esto puede decir cuál es el *frame rate* (la tasa de refrescado en el que está corriendo el juego) o qué está impactando los procesos de renderizado, *scripts* propios, el sistema de físicas, el motor de animación que trae el *engine*, el recolector de basura, entre otros. También explica que con la ayuda de esta herramienta, se puede aislar cuando algo se está comportando incorrectamente, por ejemplo, ver el consumo de un skin en particular.
- Adriana pregunta si todos los *frameworks* que se utilizan para el desarrollo de juegos tienen una herramienta similar para monitorear todos los elementos mencionados. El CEO Rodolfo respondió que las empresas más maduras sí y explica que actualmente en las industrias más maduras hay tres *engines* que se consideran como las más importantes: "Unity", "Unreal" y "Gamelab". Adicionalmente, hay un *engine* que está agarrando más fuerza es "Godot", con el cual posee herramientas de *profiling*, y hay otros *engines* más comerciales que no son tan fuertes, pero que sí son maduros y disponen de herramientas como la mencionada anteriormente. Rodolfo también explica que para las aplicaciones web, estos mismos tienen herramientas para ver el consumo y realizar *profiling*.
- Adriana pregunta si las empresas tienen una serie de pasos definidos. Rodolfo responde que sí y explica que normalmente estos son proyectos "ad hoc". Adriana, con base en la respuesta brindada por Rodolfo, comenta que para estos tipos de proyectos, se selecciona qué es lo más conveniente de las metodologías y posteriormente se adecúa o tropicaliza a la empresa.
- Adriana pregunta cuáles son las metodologías en las que se basan para el QA de los juegos. A esto, Rodolfo responde que las empresas usualmente sí tienen una serie de pasos definidos, pero en el caso de su empresa, ellos tercerizan el proceso de QA.
- Rodolfo comenta que empresas como Fair Play Labs tienen procesos de QA más maduros ya que trabajan en proyectos que se consideran más grandes. Con base en el comentario realizado, Adriana pregunta si el *testing* de esta Fair Play Labs se realiza aquí. Rodolfo responde que sí y explica que Fair Play Labs tiene un proceso de QA mucho más maduro y maneja muchos proyectos para los cuales se hacen *testing* interno dependiendo del cliente con el que se esté trabajando.

- Adriana pregunta cuánto en promedio se tarda haciendo *testing* de un juego por cada juego que sacan. Rodolfo responde que esto es complicado de estimar ya que, en el caso de su empresa, se tuvo contratiempos con la sociedad comercial con la que estaba trabajando para la tercerización del proceso de QA y *testing* para las consolas, sin embargo, se puede decir que se debería de durar 2 meses.
- Adriana pregunta ¿Aproximadamente cuántas líneas de código tiene un juego? ¿Es relevante? El CEO Rodolfo responde que la cantidad de líneas de código no se mide, sino que se mide la cantidad de tiempo de desarrollo, que en este caso fue de 2 años y medio con 4 personas.
- Adriana pregunta al CEO Rodolfo qué métricas de QA le interesan. Rodolfo responde lo siguiente:
  - *Frame rate*.
    - Esta es la tasa de refrescado.
    - El estándar actual es de 60 frames por segundo en adelante, lo que se considera como lo esperable para juegos de alto rendimiento, por ejemplo, para playstation.
    - Para juegos de bajo rendimiento, la reacción del jugador no es tan importante, por lo que se trabaja con 30 frames por segundo, por ejemplo, para la consola switch, ya que es una consola con precisiones muy bajas.
    - Esta métrica depende de la naturaleza del juego y qué tan importante es la reacción del jugador.
    - No siempre se trabaja con esos dos rangos.
    - En un juego como "*Mortal Kombat*" donde la precisión de los *inputs* es importante, se tiene que correr en una tasa de refrescado fija.
    - En juegos en donde la velocidad del *input* es muy importante, se suele poner un tope máximo y se trabaja en un rango de 5 *frames* (entre 60 - 65).
    - La tasa de refrescado gobierna la mayoría de las variables de tiempo dentro de un juego, sin embargo, esto significa que hay muchos elementos que se tienen que normalizar para poder utilizar un "reloj normal". Existe el problema de que estos son en realidad cálculos que dependen de la tasa de refrescado un ejemplo de esto es se da con Unity, donde se le llama "el delta", que es el tiempo que tardó el último *frame rate* en cargarse. Cuando la tasa de refrescado baja, el delta y la conversión de tiempo comienza a dar errores porque empieza a introducir errores de redondeo en cálculo de números flotantes.
  - *Frame drop*.
    - Esto es cuántas veces el *frame rate* cae por algo.
    - Hay operaciones que son muy caras en memoria, por ejemplo, la instanciación de un objeto nuevo. Esto causa *drops* en el *frame rate*. Si

- empiezan a haber *drops* muy seguidos, eso impacta todo el rendimiento del juego.
- *Batches o Render passes.*
    - Esto es cuántas veces el motor de renderizado tiene que pasar para poder generar un *frame*. Entre más veces tenga que pasar, más lento es el *frame rate*.
    - El *frame rate* se utiliza como base, pero cuando se ocupa modificar, se necesita poder observar en otros factores, por ejemplo, ¿Dónde tengo *frame drops*? ¿Cuántos *render passes* tengo?
    - Lo que gobierna todo es en general el *frame rate*.
  - *Frames per second*
  - Adriana consulta si la importancia del rango del *input* se relaciona con la velocidad de respuesta al utilizar un dispositivo. Rodolfo responde que sí.
  - Adriana pregunta que hay métricas respecto a funcionalidad, por ejemplo, cuántos errores hay en un juego de cierto tamaño, esto debido a que en el desarrollo de *software* se mide cuántos errores se inyectan, por ejemplo, en cada 100 líneas de código. Rodolfo responde que en su empresa se mide por nivel y no por cantidad de líneas de código, y explica que su nivel de tolerancia es muy baja.
  - Adriana pregunta si hay métricas respecto a funcionalidad. Rodolfo responde lo siguiente.
    - Si hay métricas respecto a funcionalidad.
    - Estos se miden por nivel, pero la tolerancia es muy baja, en otras palabras, no puede haber errores como si fuera un sistema crítico.
    - Los errores se clasifican por intensidad de la siguiente manera.
      - Errores gráficos: estos son errores un poquito más tolerables, pero normalmente solo se permite la presencia de 1 o 2 por nivel.
      - Errores críticos: se permiten 0 errores, lo que significa que hay que arreglarlos.
        - Su término técnico es "*game breaking*", el cual significa que ya no se puede jugar.
        - Para estos hay varios tipos:
          - *Crashes*: el juego "crasheó".
          - *Soft locks*: Es cuando sucede algo que no permite al jugador avanzar en el juego y no hay excepción o algo a nivel de código que signifique que hay un error.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Adriana Álvarez Figueroa	Junio, 2024

**Firmas:**



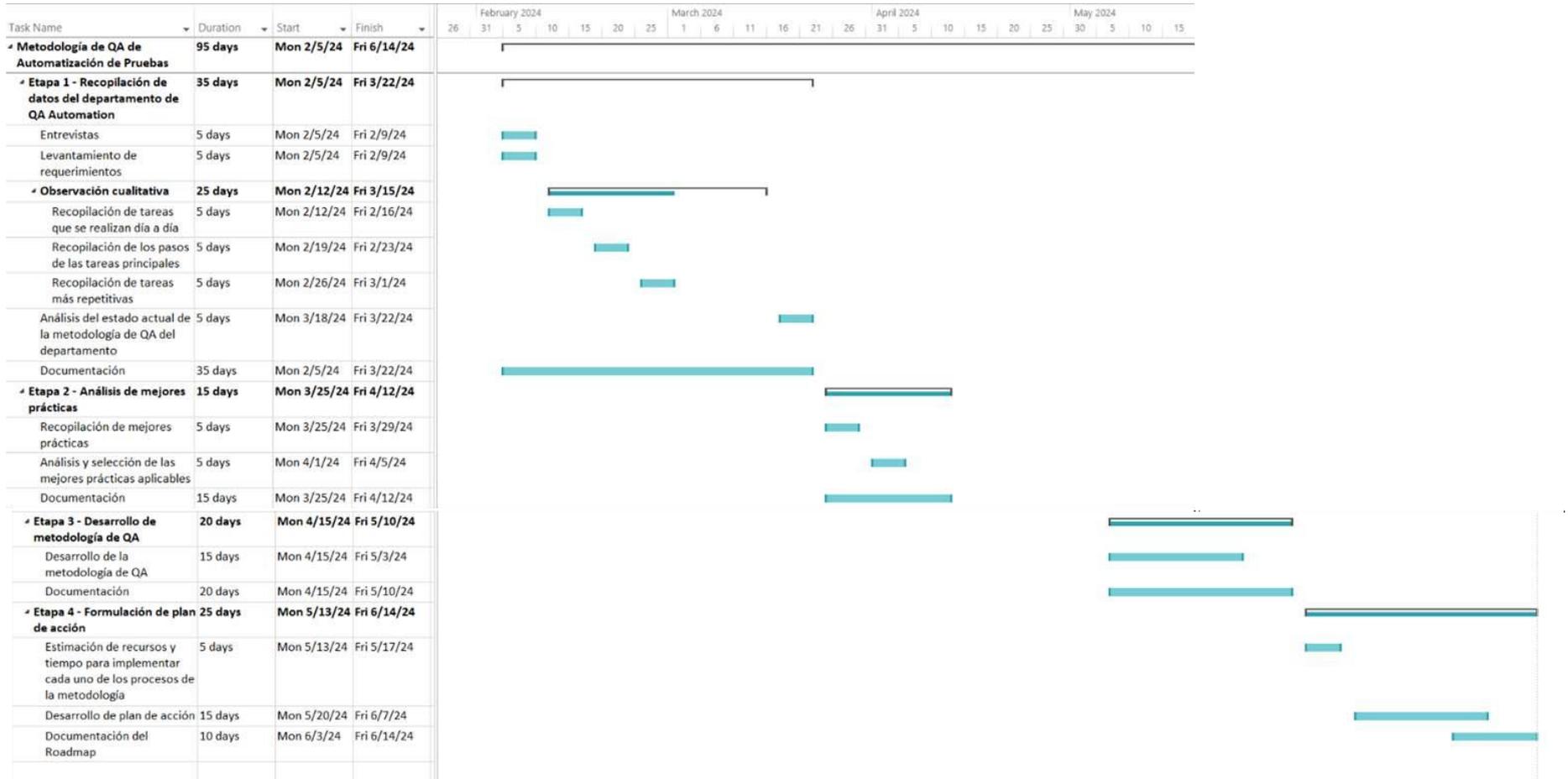
Rodolfo Mora Zamora  
CEO de Ceiba Software & Arts

Digitally signed by  
ADRIANA DE LOS ANGELES ALVAREZ FIGUEROA (FIRMA)  
Date: 2023.11.06 16:44:25 -06'00'

Adriana Álvarez Figueroa  
Profesora Asesora

### 9.3. Anexo 3

#### Cronograma y Diagrama Gantt del proyecto



## 9.4. Anexo 4

Resultado de la siguiente encuesta.

1. ¿Qué preguntas les gustaría responder con el uso de métricas?
2. ¿En qué se enfoca al momento de revisar los resultados de las pruebas de integración y regresión?
3. ¿Cuáles tareas considera que son las principales en este equipo (QA Automation)?
4. ¿Cuáles son las pruebas automatizadas que se hacen?
5. Explique el proceso de QA utilizando las pruebas automatizadas
6. ¿Qué categorías utilizaría para la clasificación de las pruebas?
7. ¿Qué categorías utilizaría para la clasificación de los bugs que se encuentran?
8. ¿Actualmente se tienen o utilizan métricas?
9. ¿Cuáles son las métricas que se utilizan actualmente?
10. ¿Qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasa/fallan y/o condiciones de pruebas que pasan/fallan?
11. ¿Qué opina sobre la posibilidad de saber cuántos bugs fueron encontrados en las pruebas de integración y regresión, y cuántos son de la herramienta para la ejecución de pruebas automatizadas?
12. ¿Ve valor en tener conocimiento sobre la cantidad de falsos positivos?
13. ¿Qué información ve relevante para un informe?

### 9.4.1. Respuesta 1

1. ¿Qué preguntas le gustaría responder con el uso de métricas?
  - ¿Cuáles son los elementos más frecuentes?
  - ¿Quiénes son los que frecuentan estos elementos?
  - ¿Cuál es la frecuencia con la que se consultan los elementos?
2. ¿En qué se enfoca al momento de revisar los resultados de las pruebas de integración y regresión?

Personalmente no hago esa actividad pero en caso de hacerlo revisaría que las pruebas terminen correctamente y en caso de no hacerlo entonces buscaría las inconsistencias que pudiesen haber causado un problema.

3. ¿Cuáles tareas considera que son las principales en este equipo (QA Automation)?

Identificar procesos que podrían llegar a automatizarse, coordinar con las personas que los aplican y eventualmente dar una alternativa automatizada a dicho proceso, de manera que este pueda correr de manera automática y que el usuario pueda dedicar su tiempo a otras actividades no automatizables.

4. ¿Cuáles son las pruebas automatizadas que se hacen?

N/A

5. Explique el proceso de QA utilizando las pruebas automatizadas

Durante el proceso de desarrollo se corren las pruebas unitarias, si estas pasan el código pasa a QA, se hacen pruebas de integración y regresión, durante estas fases se corren las pruebas automatizadas correspondientes y las manuales se ejecutan; se revisa que ambas estén correctas y el proceso continúa.

6. ¿Qué categorías utilizaría para la clasificación de las pruebas?

N/A

7. ¿Qué categorías utilizaría para la clasificación de los bugs que se encuentran?

N/A

8. ¿Actualmente se tienen o utilizan métricas?

N/A

9. ¿Cuáles son las métricas que se utilizan actualmente?

N/A

10. ¿Qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasa/fallan y/o condiciones de pruebas que pasan/fallan?

N/A

11. ¿Qué opina sobre la posibilidad de saber cuántos bugs fueron encontrados en las pruebas de integración y regresión, y cuántos son de la herramienta para la ejecución de pruebas automatizadas?

N/A

12. ¿Ve valor en tener conocimiento sobre la cantidad de falsos positivos?

N/A

13 ¿Qué información ve relevante para un informe?

N/A

## 9.4.2. Respuesta 2

1. ¿Qué preguntas les gustaría responder con el uso de métricas?
  - Tiempos de entrega.
  - Calidad del producto entregado
  - Cantidad de bugs encontrados por release.
2. ¿En qué se enfoca al momento de revisar los resultados de las pruebas de integración y regresión?

La cantidad de fallos reportados.

3. ¿Cuáles tareas considera que son las principales en este equipo (QA Automation)?

Creación de herramientas y pruebas automáticas que aumenten la calidad en la entrega de los productos de los demás departamentos de la empresa.

4. ¿Cuáles son las pruebas automatizadas que se hacen?

Pruebas de caja negra, enfocadas en UI. Pruebas de carga.

5. Explique el proceso de QA utilizando las pruebas automatizadas

Una vez terminada una prueba nueva, debe de verificarse que los pasos que realiza son los correctos, así como los resultados entregados.

Se debe valorar si existen fallos externos o algún agente que pueda afectar el resultado de la prueba. (Entorno, tiempos de repuestas, componentes)

6. ¿Qué categorías utilizaría para la clasificación de las pruebas?

- Pruebas de regresión
- Pruebas de integración
- Pruebas de Humo (Smoke Test)
- Pruebas de API;

7. ¿Qué categorías utilizaría para la clasificación de los bugs que se encuentran?

- *Bugs de release*
- *Bugs de hotfix*
- *Bugs de retesting*

8. ¿Actualmente se tienen o utilizan métricas?

No

9. ¿Cuáles son las métricas que se utilizan actualmente?

N/A

10. ¿Qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasa/fallan y/o condiciones de pruebas que pasan/fallan?

Alto.

Permitiría analizar el estado actual de todos nuestros procesos automáticos.

11. ¿Qué opina sobre la posibilidad de saber cuántos bugs fueron encontrados en las pruebas de integración y regresión, y cuántos son de la herramienta para la ejecución de pruebas automatizadas?

Permitiría demostrar el valor que posee la herramienta en la empresa.

12. ¿Ve valor en tener conocimiento sobre la cantidad de falsos positivos?

Si, para mejorar el proceso de desarrollo de pruebas y evaluar que lo que estamos probando es lo correcto.

13. ¿Qué información ve relevante para un informe?

Estado de las actividades de prueba y el avance en comparación con el plan de prueba; Pruebas programadas para el próximo período; Factores que impiden el avance;

### 9.4.3. Respuesta 3

1. ¿Qué preguntas les gustaría responder con el uso de métricas?

Cómo utilizar correctamente los resultados de las métricas para optimizar procesos.

2. ¿En qué se enfoca al momento de revisar los resultados de las pruebas de integración y regresión?

Inicialmente existe un enfoque en las secciones más utilizadas o críticas del producto.

3. ¿Cuáles tareas considera que son las principales en este equipo (QA Automation)?

- Analizar y migrar pruebas manuales.
- Generar nuevas herramientas que ayuden a la validación de productos.

4. ¿Cuáles son las pruebas automatizadas que se hacen?

Pruebas de producto y carga.

5. Explique el proceso de QA utilizando las pruebas automatizadas

Se seleccionan las pruebas requeridas y se conjuntan en un *test execution*.

Luego se corren en bloque utilizando una herramienta que dispara las pruebas y por último se hace una revisión de resultados.

6. ¿Qué categorías utilizaría para la clasificación de las pruebas?

- Pruebas del admin
- Pruebas del lobby
- Pruebas de API;

7. ¿Qué categorías utilizaría para la clasificación de los bugs que se encuentran?

- Productivos.
- Release.

8. ¿Actualmente se tienen o utilizan métricas?

Si

9. ¿Cuáles son las métricas que se utilizan actualmente?

- Resultados de pruebas.
- Cantidad de tests.

10. ¿Qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasa/fallan y/o condiciones de pruebas que pasan/fallan?

Todos los puntos anteriores son parte de un correcto informe de QA.

11. ¿Qué opina sobre la posibilidad de saber cuántos bugs fueron encontrados en las pruebas de integración y regresión, y cuántos son de la herramienta para la ejecución de pruebas automatizadas?

Es parte fundamental del proceso de pruebas.

Esto nos indica que se debe priorizar.

12. ¿Ve valor en tener conocimiento sobre la cantidad de falsos positivos?

Claro. Con esto se puede revisar los test y reestructuralos de forma tal que no genere estos falsos positivos.

13. ¿Qué información ve relevante para un informe?

Estado de las actividades de prueba y el avance en comparación con el plan de prueba; Factores que impiden el avance;

#### 9.4.4. Respuesta 4

1. ¿Qué preguntas les gustaría responder con el uso de métricas?

Medir calidad del release y recurrencia de problemas (que tantas veces se comete el mismo error).

2. ¿En qué se enfoca al momento de revisar los resultados de las pruebas de integración y regresión?

Detectar errores en funcionalidad que no ha debería de haber cambiado

3. ¿Cuáles tareas considera que son las principales en este equipo (QA Automation)?

Detectar errores en funcionalidad que no debería de cambiar.

4. ¿Cuáles son las pruebas automatizadas que se hacen?

Regresión del Lobby, APIs and Admin.

5. Explique el proceso de QA utilizando las pruebas automatizadas

Pruebas automatizadas se ejecutan después de la instalación del release candidato y los resultados son revisados de 2 a 3 horas después.

6. ¿Qué categorías utilizaría para la clasificación de las pruebas?

- Pruebas del admin
- Pruebas del lobby
- Pruebas de API
- Juegos

7. ¿Qué categorías utilizaría para la clasificación de los bugs que se encuentran?

- Componente: Admin, Lobby, Juegos, etc
- Etapa: Desarrollo, Integration, Regression, Production

8. ¿Actualmente se tienen o utilizan métricas?

No

9. ¿Cuáles son las métricas que se utilizan actualmente?

N/A

10. ¿Qué tan relevante ve el poder disponer de métricas como el número de casos de prueba ejecutados/no ejecutados, casos de prueba que pasa/fallan y/o condiciones de pruebas que pasan/fallan?

Es deseable.

11. ¿Qué opina sobre la posibilidad de saber cuántos bugs fueron encontrados en las pruebas de integración y regresión, y cuántos son de la herramienta para la ejecución de pruebas automatizadas?

Es deseable (puede utilizarse para medir utilidad de pruebas automatizadas para atajar problemas antes de la liberación del paquete)

12. ¿Ve valor en tener conocimiento sobre la cantidad de falsos positivos?

Es deseable (puede utilizarse para medir confiabilidad del framework)

13. ¿Qué información ve relevante para un informe?

Factores que impiden el avance;Estado Actual del tool ;

## **9.5. Anexo 5**

### **Formato de documento de estrategia de prueba**

[Título del proyecto]

#### **1. Descripción del proyecto**

Esta sección del documento brinda una descripción del producto o proyecto a desarrollar. Este abarca puntos tales como el nombre del proyecto, sus principales componentes, y cómo se espera que funcione.

#### **2. Partes interesadas (Stakeholder)**

Este presenta los individuos o grupos que tienen interés en las actividades y resultados del producto.

#### **3. Objetivos de la prueba**

En esta sección del documento de estrategia de prueba se listan los principales objetivos de la ejecución de las pruebas para el aseguramiento de la calidad del producto, asimismo, se muestran los tipos de pruebas necesarias y a ejecutar.

#### **4. Riesgo del producto**

En esta sección se listan los posibles riesgos que se pueden presentar durante la ejecución de pruebas.

## 9.6. Anexo 6

### Respuesta 1

1. ¿En qué empresa labora?

Ceiba Software & Arts

2. ¿Cuál metodología de QA utiliza? (Selección múltiple)

Nosotros tercerizamos el QA

3. ¿Qué métricas utilizan? (Selección múltiple)

Número de defectos encontrados en un entregable

4. ¿Qué tipo de pruebas realizan para asegurar la calidad del producto? (Selección múltiple)

Pruebas de Playtesting, Pruebas unitarias, Pruebas de rendimiento

5. ¿Qué herramientas utilizan para la gestión de las pruebas? (Selección múltiple)

Herramientas especializadas para videojuegos (Unity Profiler)

6. ¿Realizan pruebas automatizadas?

No

7. ¿Qué tecnologías utilizan para las pruebas automatizadas? (Selección múltiple)

N/A

## 9.7. Anexo 7

### 9.7.1. Entrevista con Álvaro Rivera

#### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 8 de abril del 2024

**Consecutivo:** 004

**Convocados:**

- Álvaro Rivera - CEO ncubo Ideas
- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica

**Hora de Inicio:** 04:00 p.m.

**Hora de Finalización:** 06:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de Carmen Mok Zheng y el CEO de ncubo Ideas Álvaro Rivera.

En este se trataron los siguientes puntos:

- La entrevista se inició con una pregunta de Carmen Mok dirigida a Álvaro Rivera sobre su ocupación y empleador actual. Álvaro Rivera proporcionó una descripción detallada de sus actividades laborales, indicando que se dedica al desarrollo y arquitectura de software. Mencionó su experiencia de varios años en aseguramiento de la calidad (QA), abarcando roles desde *tester* hasta gerente de calidad, con un enfoque en automatización y pruebas de lenguajes de programación.
- Álvaro Rivera compartió información sobre la fundación de su empresa "*Testing Software*", la cual se especializa exclusivamente en pruebas de *software*, aprovechando la falta de conocimiento general sobre este campo en aquel entonces. Describió el amplio espectro de pruebas realizadas, que incluían tanto pantallas como aplicaciones, con clientes principalmente en empresas de tecnología en Estados Unidos.
- En cuanto a su trayectoria actual, Álvaro Rivera mencionó que lleva aproximadamente 8 años dedicado al desarrollo de *e-commerce*, desempeñando roles tanto de arquitecto como de desarrollador en este ámbito específico.
- Carmen Mok procedió a indagar sobre la experiencia de Álvaro Rivera en el uso de pruebas automatizadas. Álvaro Rivera confirmó esta afirmación, mencionando que cuenta con aproximadamente 20 años de experiencia en esta área.

- Álvaro Rivera explicó que la automatización es fundamental en el proceso de desarrollo de software y la describió como "la cereza del pastel". Explicó que la automatización permite ciertas capacidades que agilizan el proceso de desarrollo y prueba, como la detección de fallos en regresiones y la capacidad de realizar cambios radicales en el código de manera ágil.
- Asimismo, Álvaro Rivera advirtió sobre el error común de creer que la automatización resolverá todos los problemas, enfatizando la importancia de tener un proceso de pruebas manual sólido antes de incorporar la automatización. Explicó que la automatización no solucionará problemas fundamentales si el proceso de desarrollo y prueba manual aún presenta deficiencias.
- Álvaro Rivera destacó que antes de implementar la automatización, es necesario abordar las fallas y deficiencias existentes en el proceso de desarrollo y prueba manual. Subrayó la importancia de tener un plan de pruebas adecuado y una buena cobertura de pruebas antes de considerar la automatización como un complemento al proceso.
- Finalmente, Álvaro Rivera mencionó que existen diferentes tipos de automatización, cada uno con sus propias características y aplicaciones específicas, lo que subraya la importancia de comprender las necesidades y requisitos del proyecto antes de decidir qué tipo de automatización implementar.
- Esta adición proporciona más detalles sobre la experiencia y conocimientos de Álvaro Rivera en el campo de las pruebas automatizadas, así como su perspectiva sobre su importancia y aplicación en el desarrollo de software.
- Carmen Mok consultó a Álvaro Rivera sobre las tecnologías utilizadas para las pruebas automatizadas y si su elección dependía del proyecto en cuestión. Álvaro Rivera respondió afirmativamente, explicando que la selección de tecnologías depende de varios factores, como la estrategia establecida, los recursos disponibles y el estado del proyecto.
- Álvaro Rivera enfatizó la importancia de generar una estrategia adecuada para las pruebas automatizadas, destacando que esta puede variar según el punto de partida del proyecto. Ya sea que se esté comenzando un nuevo sistema o mejorando uno existente, la estrategia debe adaptarse a las circunstancias específicas.
- El entrevistado mencionó la importancia de tener una base de datos para las pruebas, especialmente si hay una gran cantidad de datos preexistentes. Explicó que la automatización de las pruebas sigue dependiendo de los escenarios específicos que se

estén buscando, como la presencia de servicios REST, capas de aplicación, procedimientos almacenados en la base de datos o interfaces de usuario (UI).

- Álvaro Rivera solicitó más detalles sobre el tipo de aplicación que se está discutiendo, como si se trata de juegos o páginas web. Carmen Mok confirmó que se enfocan principalmente en pruebas para juegos y páginas web.
- Álvaro Rivera continuó explicando que tanto las páginas web como los juegos tienen una lógica de negocios específica sobre cómo se exponen al usuario, lo que puede variar según la plataforma en la que se ejecuten, ya sea en un celular, una computadora o mediante comunicación con un servidor.
- Álvaro solicitó más información respecto a si los juegos se encuentran en los dispositivos móviles o computadoras, o si estos se comunican con un servidor.
- Carmen Mok detalló que los juegos se integran dentro de una página web, donde los usuarios acceden a un catálogo de juegos. Estos juegos pueden ser integrados en la página web o llamados a través de APIs.
- Álvaro Rivera destacó que, en el caso de juegos divididos en capas, existen dos alternativas para las pruebas: probar la interfaz de usuario (UI) o probar tanto la UI como la lógica de negocio. Explicó que los juegos a menudo requieren cambios en la UI, lo que puede invalidar la automatización si se centra exclusivamente en la UI. Por lo tanto, sugirió enfocarse en automatizar la lógica de negocio, que tiende a ser más estable y constante en el tiempo.
- Álvaro Rivera compartió información sobre las actividades de su equipo, que se dedica actualmente al *e-commerce*. Destacó que se enfocan en probar manualmente las pantallas mientras automatizan la comunicación con el backend, donde residen los REST APIs y la lógica de negocios. Estas pruebas incluyen aspectos como rendimiento, seguridad y funcionalidad.
- Álvaro Rivera explicó que hay muchas herramientas disponibles para la automatización de pruebas, algunas licenciadas y otras de código abierto, como Selenium. Recomendó evaluar diferentes herramientas y seleccionar aquellas que se ajusten mejor a las necesidades del proyecto.
- Álvaro Rivera continuó explicando su enfoque en la automatización de pruebas de lógica de negocio, utilizando pruebas de unidad, seguridad, robustez, rendimiento y funcionalidad. Destacó la importancia de asegurar una cobertura completa de pruebas

para todos los servicios, incluyendo la identificación y automatización de casos de prueba faltantes.

- El entrevistado enfatizó la importancia de priorizar los casos de prueba y automatizar aquellos que son más críticos para el negocio. Explicó que la automatización no debe limitarse a los casos de prueba ya establecidos, sino que debe adaptarse a medida que se desarrollan nuevas funcionalidades y requisitos.
- Álvaro Rivera aclaró que el papel del equipo de QA es certificar la calidad del producto y no simplemente encontrar defectos en el trabajo de los desarrolladores. Explicó que la automatización de pruebas varía según el lenguaje de programación utilizado y recomendó evaluar diferentes herramientas según las necesidades específicas del proyecto.
- Álvaro Rivera concluyó destacando que la automatización de pruebas no es una solución para reducir costos, sino una inversión en la calidad del producto. Mencionó que las empresas de alto nivel en calidad pueden invertir recursos en la automatización de pruebas como parte de su enfoque en la excelencia del producto.
- Álvaro Rivera finalizó mencionando que no utilizaría una sola herramienta para la automatización, ya que cada tipo de prueba puede requerir una herramienta diferente, como una para automatizar pruebas de rendimiento y otra para pruebas de servicios REST.
- Esto proporciona una visión detallada de la estrategia y el enfoque de Álvaro Rivera en la automatización de pruebas, así como sus recomendaciones para seleccionar herramientas y priorizar casos de prueba.
- Álvaro Rivera explicó que, en el caso de su equipo de *e-commerce*, se realizan pruebas de rendimiento específicamente en los servicios relacionados con compras y cierres de transacciones, mientras que otros servicios REST no se someten a pruebas de rendimiento automatizadas, sino que se monitorean en producción utilizando una herramienta de rendimiento de aplicaciones (APM).
- El entrevistado destacó que se automatizan pruebas de seguridad y robustez para todos los servicios, con el objetivo de garantizar que no ocurran inyecciones de código o caídas del servidor debido a fallos en los servicios. Estas pruebas se realizan paramétricamente y se enfocan en aspectos críticos de seguridad y funcionalidad.
- Álvaro Rivera compartió su experiencia sobre la dificultad de mantener la automatización de pruebas, especialmente cuando las decisiones de la gerencia pueden

afectar la validez de los casos de prueba automatizados. Explicó que es común encontrarse en situaciones donde los cambios en las políticas de la empresa afectan la relevancia de los casos de prueba, lo que requiere una revisión y actualización constante de la automatización.

- El entrevistado resaltó que, a pesar de los desafíos, la automatización de pruebas proporciona a las empresas la capacidad de ser ágiles y responder rápidamente a los cambios en el código y los requisitos del negocio. Explicó que la automatización permite realizar cambios significativos en el software de manera estable y rápida, lo que no sería posible con pruebas manuales.
- Álvaro Rivera concluyó que la utilidad de la automatización de pruebas depende del contexto y la agilidad de la organización. En entornos donde los recursos son escasos o la organización no es ágil, la automatización puede tener un valor limitado. Sugirió que, en tales casos, se prioricen las pruebas manuales en áreas críticas como la interfaz de usuario, mientras que se automatizan las pruebas de servicios REST.
- Esta discusión proporciona una visión detallada de los desafíos y beneficios de la automatización de pruebas, así como recomendaciones sobre cómo abordarla de manera efectiva según el contexto organizacional y los recursos disponibles.
- Carmen Mok resumió la conversación hasta ese momento, destacando que no hay una única tecnología que sea la solución perfecta para las pruebas automatizadas, sino que depende del contexto y las herramientas utilizadas por la organización. Utilizó como ejemplo el desarrollo de una aplicación en Java, sugiriendo que sería mejor crear pruebas unitarias en Java en este caso.
- Álvaro Rivera estuvo de acuerdo con esta afirmación.
- Carmen expresó su interés en la idea de que los desarrolladores también podrían realizar pruebas unitarias, lo que ayudaría a evitar la aparición de errores que luego serían detectados por los equipos de QA.
- Álvaro explicó que aunque los desarrolladores podrían realizar pruebas unitarias como parte del desarrollo orientado a pruebas (*test-driven development*), esto no exime a los equipos de QA de su responsabilidad en el proceso de aseguramiento de la calidad. Destacó la importancia de implementar prácticas de desarrollo orientadas a pruebas como parte integral del ciclo de desarrollo.

- Álvaro también mencionó que los testers podrían utilizar tecnologías como .NET o Java para realizar su propia automatización de pruebas, dependiendo de las necesidades específicas del proyecto.
- Finalmente, Álvaro señaló que el rol de la automatización de pruebas normalmente recae en los equipos de QA, aunque los ingenieros de QA poseen habilidades similares a las de los desarrolladores.
- Álvaro destacó la importancia de que los desarrolladores implementen pruebas unitarias como parte del proceso de desarrollo, enfatizando que es una responsabilidad fundamental. Señaló que aunque algunos desarrolladores podrían no sentirse cómodos con esta práctica, es crucial para garantizar la calidad del código.
- Carmen planteó una pregunta sobre la metodología Test Driven Development (Desarrollo Orientado a Pruebas) y si los testers podrían realizar pruebas unitarias.
- Álvaro aclaró que la responsabilidad de realizar pruebas unitarias recae en los desarrolladores, ya que estas pruebas buscan demostrar que comprenden lo que están programando. Explicó que en algunos casos, las pruebas unitarias y el código pueden enviarse juntos en un solo *commit*, aunque idealmente deberían ser separados.
- Carmen Mok planteó una consulta sobre el concepto de "tomar una foto a la base de datos de producción" para realizar pruebas, preguntando si se refería a hacer una copia duplicada de esa base de datos.
- Álvaro también explicó el concepto de tomar una "foto" de la base de datos de producción para realizar pruebas, indicando que se trata de crear una copia estática de la base de datos en un estado inicial conocido. Esta copia se utiliza para realizar pruebas, y siempre se parte de esta versión inicial para mantener la consistencia en las pruebas, incluso cuando se realizan cambios en la base de datos durante el desarrollo del software.
- Carmen planteó una pregunta sobre el tipo de pruebas que se realizan para garantizar la calidad del producto y si esto varía según el propósito de la prueba.
- Álvaro respondió destacando el valor profesional de un QA (Asegurador de Calidad) y la importancia de ser crítico con lo que se está revisando. Explicó que algunos testers se limitan a informar sobre fallos obvios, como errores ortográficos o caídas del sistema, lo cual consideró como algo básico. Sin embargo, señaló que para garantizar verdaderamente la calidad del producto, el tester debe ser muy crítico y centrarse en aspectos más profundos de la aplicación.

- Álvaro enfatizó que el QA debe ser crítico con el proceso en su totalidad y no limitarse a reportar sólo problemas triviales. Esta actitud crítica es fundamental para identificar y resolver problemas que podrían pasar desapercibidos de otra manera.
- Carmen planteó otra pregunta sobre las métricas que Álvaro considera indispensables en el proceso de QA, específicamente en el contexto de la automatización.
- Álvaro compartió su enfoque para abordar las incidencias o "pulgas" que llegan a producción. Explicó que cuando detecta un problema, retrasa su resolución para investigar su origen, incluyendo la línea de código correspondiente en el repositorio y quién fue responsable de probar esa área. Este proceso de "post-mortem" le permite comprender por qué ocurrió la incidencia y tomar medidas correctivas en el proceso.
- Además, Álvaro mencionó la importancia de establecer políticas para la automatización de pruebas, como la priorización de casos de prueba de alta prioridad para la automatización. También destacó la idea de algunas corrientes que sugieren que si la automatización no detecta ciertos errores, entonces la automatización no es efectiva y podría ser eliminada.
- Finalmente, Álvaro enfatizó la importancia de encontrar un equilibrio entre automatizar lo más tarde posible para garantizar la estabilidad del sistema, pero lo suficientemente temprano para que las funcionalidades estén maduras. Este equilibrio implica considerar el calendario del proyecto y los recursos disponibles.
- Carmen planteó la cuestión de las métricas, preguntando sobre la opinión de Álvaro en relación con indicadores como el número de defectos encontrados en un entregable o el tiempo necesario para ejecutar casos de prueba de regresión.
- Álvaro explicó que durante el ciclo de desarrollo, el equipo de QA no es responsable de las incidencias que puedan surgir. Su responsabilidad es asegurarse de que el *software* no tenga defectos antes de su liberación. Por lo tanto, considera que el número de defectos encontrados no es una métrica relevante para el equipo de QA, ya que su objetivo principal es garantizar la calidad del producto, sin embargo, esta información puede ser importante para el líder de desarrollo.
- Carmen indaga sobre las herramientas que podrían utilizarse para gestionar los resultados de las pruebas, como la generación de informes sobre las pruebas ejecutadas, mencionando ejemplos como Jira.
- Álvaro compartió su experiencia, mencionando que utiliza *Team Services*, un servicio integrado con .Net que mapea todo el ciclo de desarrollo. Destacó la importancia de

herramientas como Jira y Microsoft Team Services para gestionar el ciclo de vida de las incidencias, ya que proporcionan una estructura clara y buenas prácticas para el seguimiento de los problemas y tareas.

- Álvaro recomendó seguir las buenas prácticas y adaptarse al flujo de trabajo establecido por estas herramientas en lugar de crear uno propio, ya que las herramientas están diseñadas pensando en ese flujo de trabajo específico y su uso adecuado facilita la gestión eficiente de las incidencias y tareas.
- Carmen también planteó otra pregunta relacionada con la experiencia de Álvaro: ¿Qué metodologías de QA considera más comunes?
- Álvaro respondió que, en su experiencia de 35 años en computación, ha observado que la metodología ágil es la más común. Destacó que, tanto en Agile como en otras metodologías, el enfoque principal es la interacción de las personas dentro del equipo. Sin embargo, señaló que lo que está en constante cambio son las herramientas disponibles, pero el objetivo principal sigue siendo ser ágil en el desarrollo y prueba de software.
- Carmen también expresó su interés en contar con algún marco de referencia para su investigación.
- Álvaro recomendó asistir a *workshops* sobre QA como una forma de obtener más información y conocer las prácticas utilizadas en otros lugares. Además, sugirió realizar *post-mortems* después de cada *milestone* en el proceso de desarrollo. Estos análisis permiten identificar errores y proponer mejoras continuas en el proceso. Álvaro destacó que esta actividad generalmente la lidera el *QA Manager*, quien estandariza las prácticas de QA en la empresa.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Junio, 2024

**Firmas:**

ALVARO  
ANTONIO RIVERA  
BELLIDO (FIRMA)

Álvaro Rivera  
CEO de ncubo Ideas



Carmen María Mok Zheng  
Estudiante del TEC

## 9.7.2. Entrevista con Ana Sanabria

### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 17 de abril del 2024

**Consecutivo:** 005

**Convocados:**

- Ana Sanabria - *Compliance Specialist* de la empresa *Infinite*
- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica

**Hora de Inicio:** 04:30 p.m.

**Hora de Finalización:** 06:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Microsoft Teams, con la participación de Carmen Mok Zheng y la Ingeniera Ana Sanabria.

En este se trataron los siguientes puntos:

- Carmen Mok inició la entrevista preguntando a Ana Sanabria sobre su ocupación y empresa actual.
- Ana Sanabria, ingeniera en computación graduada del Tecnológico de Costa Rica, detalló su trayectoria profesional de más de 30 años. Comenzó como programadora desarrolladora, luego se trasladó al departamento de control de calidad y trabajó en el área de QA durante 11 años en Walmart. Posteriormente, se unió a Fiserv Financial Services, donde se desempeñó en el departamento de PQM, enfocado en la mejora de procesos en el desarrollo de software. Ana describió su experiencia en el desarrollo de capacitaciones técnicas internas, abarcando temas como ciberseguridad, Scrum, liderazgo, *DevOps* y *Analytics*.
- Recientemente, fue asignada al área de *Risk and Compliance* de *Infinite*, una empresa que adquirió la operación de Fiserv en Costa Rica.
- Ana explicó su papel actual como proveedora de servicios para Fiserv a través de *Infinite*, centrada en auditorías rigurosas requeridas por la empresa cotizada en bolsa en Estados Unidos.
- Además, compartió su experiencia en gestión de *help desk* y obtención de certificaciones como *Green Belt*, destacando su capacidad para integrar la mejora de procesos en el desarrollo de *software*.

- Ana mencionó el proceso de evaluación en CMMI que su equipo pasó, siendo la primera empresa en América Central en ser evaluada oficialmente en nivel 3 en 2014 y reevaluada en 2017.
- Finalmente, Ana presentó las buenas prácticas y medidas de desempeño de los procesos implementados en una charla en el Club de Investigación Tecnológica.
- Carmen Mok preguntó a Ana Sanabria sobre su opinión respecto a los marcos de referencia para proyectos de desarrollo de software, mencionando específicamente el CMMI.
- Ana Sanabria enfatizó la importancia de comprender que los marcos de referencia proporcionan un marco general y no dictan el cómo deben implementarse los procesos.
- Destacó la robustez del CMMI al abarcar tanto el desarrollo como los servicios, incluyendo el manejo de proveedores.
- Ana advirtió sobre el error de intentar adaptar la organización a un marco de referencia en lugar de adaptar el marco a las necesidades y la cultura de la organización.
- Recomendó evaluar los marcos de referencia en función de las necesidades específicas de la organización y considerar su integralidad en la evaluación y mejora de procesos.
- Ana sugirió medir la inyección de defectos entre fases del desarrollo, priorizando la detección temprana de errores para evitar costos mayores en etapas posteriores.
- Explicó la importancia de clasificar los errores por tipo, origen e impacto, así como desarrollar una cultura interna de detección y mejora de errores en todas las etapas del proceso de desarrollo.
- Ana destacó la necesidad de medir el tiempo ahorrado y los beneficios obtenidos mediante la automatización de pruebas, así como promover una cultura de análisis y mejora continua en el equipo.
- Recomendó la implementación de fases para medir, analizar y mejorar procesos, y mencionó la relevancia de *frameworks* como *Six Sigma* para este propósito.
- Subrayó la importancia de justificar las mediciones mediante beneficios económicos tangibles, como la reducción de costos, la eficiencia y la mejora en la calidad del producto final.
- Carmen Mok preguntó sobre la influencia del contexto organizacional y la cultura de los empleados en la metodología de QA.
- Ana Sanabria explicó que la elección de una metodología depende del tipo de producto y cultura organizacional, advirtiendo sobre la rigidez en la adopción de un modelo sin evaluación constante de su eficacia.
- Mencionó la importancia de herramientas como Jira para gestionar tareas y acuerdos, así como la gestión de riesgos en proyectos.

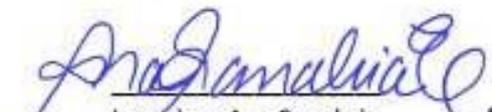
- Se discutió sobre los tipos de pruebas realizadas para asegurar la calidad del producto, destacando que dependen del proyecto y su criticidad, incluyendo inspecciones de código, pruebas unitarias, de integración, de esfuerzo y de penetración.
- Ana mencionó el uso de Selenium para pruebas automatizadas.
- Carmen consultó sobre la viabilidad de una metodología de QA centrada en el uso de métricas.
- Ana expresó su apoyo a la propuesta, enfatizando la importancia de tener un proceso manual sólido antes de la automatización.

Acciones a Seguir:

- Carmen Mok acordó considerar las recomendaciones de Ana Sanabria para el desarrollo de la metodología de QA, incluyendo la evaluación constante de su eficacia y la implementación gradual de métricas.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Junio, 2024

Firmas:

  
Ingeniera Ana Sanabria

  
Carmen María Mok Zheng  
Estudiante del TEC

### 9.7.3. Entrevista con Ignacio Díaz Oreiro

#### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 9 de abril del 2024

**Consecutivo:** 006

**Convocados:**

- Ignacio Díaz Oreiro - Profesor de la Universidad de Costa Rica
- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica

**Hora de Inicio:** 04:30 p.m.

**Hora de Finalización:** 06:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de Carmen Mok Zheng y el profesor Ignacio Díaz.

En esta se trataron los siguientes puntos:

- Ignacio consultó a Carmen sobre el tipo de pruebas que estaba considerando realizar, preguntando si se enfocarían en pruebas funcionales, de unidad o de sistema, y si había algún énfasis particular.
- Carmen explicó que actualmente realizan pruebas de API y pruebas relacionadas con la interfaz de usuario de páginas web, debido a que su empresa se dedica a la creación de videojuegos.
- Ignacio destacó la importancia de clasificar las pruebas según el nivel (unidad, integración, sistema) y el tipo (funcionales, de desempeño, etc.).
- Sugirió la necesidad de establecer un marco conceptual de calidad para definir los criterios de calidad desde el punto de vista del usuario y de los constructores de software.
- Se mencionó el estándar ISO 25010 de la ISO como un marco que aborda la calidad desde la perspectiva interna y externa de la construcción del software.
- Se discutió el concepto de Aseguramiento de la Calidad (QA) y su relación con pruebas y calidad en general.
- Ignacio señaló la importancia de definir qué pruebas se desean automatizar y mencionó la necesidad de abordar el área de desempeño, incluyendo pruebas de carga y estrés.
- Se discutió sobre la comparación de resultados en pruebas de desempeño y la necesidad de un entorno controlado para su ejecución.

- Carmen preguntó a Ignacio sobre su experiencia laboral y en qué empresa trabaja actualmente.
- Ignacio compartió que se dedica a la docencia desde hace seis años y que previamente trabajó durante 27 años en la industria de software. De esos 27 años, aproximadamente 15 estuvo en el área de pruebas y 5 años en el área de calidad, específicamente en "Administración de la configuración".
- Explicó que trabajó principalmente en una empresa llamada Artinsoft, que posteriormente cambió su nombre a Mobilize.net. En esta empresa, realizaba pruebas en diferentes sistemas, especialmente pruebas funcionales. También mencionó su experiencia como líder de equipo de pruebas y cómo se llevaba a cabo el proceso de pruebas en la empresa.
- Ignacio mencionó que en la empresa donde trabajaba no se utilizaba una metodología específica de pruebas, pero que Microsoft Solution Framework (MSF) era una metodología comúnmente utilizada por Microsoft, con la que estaba familiarizado.
- Se destacó que, debido a la duración de los proyectos con clientes externos y a la inestabilidad inicial del software, no se automatizaban muchas pruebas y la mayoría se realizaban manualmente. Sin embargo, las herramientas internas (migradores de software) desarrolladas por la empresa para su función primordial, sí contaban con muchos tipos de pruebas (funcionales, de desempeño, entre otros) en sus diferentes niveles, y se aplican al incluir código nuevo (*check-in test*) y también como parte de la metodología MSF que requiere la ejecución del conjunto total de pruebas (miles) al final del día o el fin de semana (*full-test runs*).
- Carmen y Ignacio conversaron sobre la flexibilidad de las metodologías de aseguramiento de calidad (QA) y cómo estas dependen en gran medida del contexto de la organización.
- Se destacó que, en muchos casos, las empresas no invierten lo suficiente en QA debido a la falta de claridad sobre su utilidad y el deseo de cumplir con los plazos de entrega.
- Ignacio mencionó que en las metodologías ágiles, es común que los mismos desarrolladores realicen las pruebas, aunque algunos equipos pueden tener personal especializado en pruebas.
- Se discutieron varios marcos de referencia recomendados para investigar en el ámbito de QA, incluyendo ISO 25010, CMMI, PMBOK y ISTQB.
- Se destacó que la elección del marco de referencia depende de las necesidades y el contexto específico de la empresa.
- Ignacio proporcionó una lista de métricas importantes en el proceso de QA, que incluyen casos de prueba, defectos, tendencias, defectos activos, resueltos y cerrados, así como reactivaciones de defectos.

- Se mencionó la importancia de considerar métricas específicas para pruebas de rendimiento, como tiempo, recursos y accesos a bases de datos.
- Carmen consultó a Ignacio sobre los tipos de pruebas necesarias para asegurar la calidad del producto.
- Ignacio explicó que, idealmente, se deberían realizar todas las pruebas posibles, pero los recursos limitados pueden afectar la cantidad y el tipo de pruebas que se realicen.
- Se mencionaron pruebas como pruebas de humo, pruebas de extremo a extremo, pruebas de check-in, pruebas de sistema, pruebas de regresión y pruebas de confirmación.
- Carmen preguntó sobre herramientas para la gestión de pruebas y la generación de reportes de resultados.
- Ignacio explicó que la elección de herramientas depende de la empresa y la metodología utilizada, y mencionó herramientas como Jira y plugins gratuitos.
- Carmen consultó a Ignacio sobre su experiencia en pruebas automatizadas y las tecnologías utilizadas.
- Se discutieron herramientas como J-Unit, Selenium y Cypress IO, y se destacó que la elección de la tecnología depende del lenguaje de programación y el tipo de prueba.
- Carmen solicitó la opinión de Ignacio sobre su proyecto de desarrollar una propuesta de metodología de QA centrada en pruebas automatizadas.
- Ignacio expresó que la propuesta sería útil, pero recomendó enfocarse en las características específicas de la empresa y no automatizar todo indiscriminadamente.
- Se resaltó la importancia de considerar el contexto y los entornos de prueba al automatizar pruebas.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Agosto, 2024

**Firmas:**

IGNACIO DIAZ OREIRO (FIRMA)  
 Firmado digitalmente por IGNACIO DIAZ OREIRO (FIRMA)  
 Fecha: 2024.08.04 11:21:25 -0500

Ignacio Díaz Oreiro  
 Profesor de la Universidad de Costa Rica



Carmen María Mok Zheng  
 Estudiante del TEC

## 9.7.4. Entrevista con Mario Jiménez

### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 2 de abril del 2024

**Consecutivo:** 007

**Convocados:**

- Adriana Álvarez Figueroa - Profesora Asesora
- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica
- Mario Jimenez - Director Global de IT en Newfire Global Partners

**Hora de Inicio:** 04:30 p.m.

**Hora de Finalización:** 06:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de la estudiante Carmen Mok Zheng, la profesora asesora Adriana Álvarez, y Mario Jimenez, Director Global de IT en Newfire Global Partners.

En este se trataron los siguientes puntos:

- Carmen comenzó con una breve explicación sobre el propósito de la entrevista y el proyecto a realizar con la información que se obtenga. Posteriormente preguntó al Director Mario Jimenez por su experiencia laboral.
- Mario Jimenez explica que actualmente es el Director Global de IT en Newfire Global Partners y reporta al CTO de la compañía. La empresa se especializa en servicios profesionales de desarrollo de software, principalmente para la industria de la salud. Newfire Global Partners cuenta con clientes en EE.UU., Latinoamérica y Europa. La empresa tiene operaciones en EE.UU., Costa Rica, Ucrania y Croacia, y desarrolladores en 20 países. Los desarrolladores trabajan desde sus casas, contratados en función de la disponibilidad de talento especializado.
- Respecto a la experiencia profesional de Mario, este comenta que trabajó 11 años en Walmart, dedicando más de siete años a la ingeniería de sistemas y aseguramiento de calidad. Fue Director del Área de Ingeniería en Walmart, implementando mejoras en prácticas de ingeniería y creando el área de aseguramiento de calidad. Participó en el startup de Telefónica en Costa Rica durante dos años, trabajando en software y comunicaciones, y entre 2013 y 2020, trabajó en Fiserv (Financial Services), como Director de Excelencia Operativa, responsable del aseguramiento de calidad, Business Excellence y procesos.

- Carmen pregunta a Mario cuáles son las metodologías de QA más comunes en la industria.
- Mario Jiménez explica que el Testing Maturity Model Integration (TMMI) es un marco similar al CMMI que evalúa el nivel de madurez en prácticas de testing. Sugiere que la calidad de aseguramiento (QA) es más amplia que el testing, abarcando prácticas más comprensivas.
- Mario recomienda no adherirse a un solo modelo, sino verlos como herramientas en una caja, extrayendo lo necesario en cada situación, asimismo, identifica tres elementos principales para crear un marco metodológico adaptado: Test Driven Development (TDD), TMMI y CMMI.
- Menciona la importancia del Capability Maturity Model Integration (CMMI) en áreas de práctica relacionadas con la calidad y comenta sobre la evolución del CMMI a lo largo del tiempo, desde la versión 1.0 hasta la versión 3.0, siendo adquirido posteriormente por ISACA.
- Adriana Álvarez propone discutir los marcos de referencia utilizados en el anteproyecto para ver si aplican.
- Carmen Mok menciona que está utilizando ISTQB Foundation, ISTQB Automation Engineer, PMBOK y uno de la IEEE.
- Mario Jiménez comenta sobre la eficacia y limitaciones de estos marcos.
- Recomienda TMMI y CMMI para prácticas de calidad, debido a su amplio alcance y uso común en la industria.
- Adriana Álvarez destaca la importancia de incluir aspectos de prevención en el proceso y la necesidad de considerar los departamentos de QA y desarrollo por separado.
- Mario Jiménez explica su enfoque en la prevención de errores desde el desarrollo.
- Adriana Álvarez discute la relevancia de este enfoque para el departamento de QA.
- Mario Jiménez menciona la importancia de abordar la calidad de manera holística, desde la planificación hasta el delivery.
- Propone la inclusión de CMMI y prácticas específicas de QA en el proyecto.
- Carmen Mok consulta sobre las métricas comunes en el proceso de QA, según la experiencia de Mario Jiménez.
- Mario Jiménez destaca la importancia de las métricas de pre y post entrega de defectos, enfatizando la densidad de defectos por esfuerzo.
- Propone considerar también la cobertura de código y el seguimiento del total de casos de prueba en la planificación.
- Adriana Álvarez pregunta sobre estándares o marcos de referencia para métricas sugeridas.
- Mario Jiménez sugiere buscar en el Software Productivity Research y ofrece compartir un documento con métricas variadas, incluyendo métricas de testing.

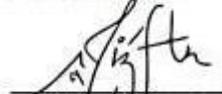
- Carmen Mok consulta sobre los tipos de pruebas utilizadas para asegurar la calidad del producto.
- Mario Jimenez enfatiza la importancia de una guía de decisión basada en el contexto y la naturaleza de los cambios para determinar qué pruebas realizar.
- Destaca la necesidad de pruebas funcionales y técnicas, además de considerar criterios de aceptación tanto del usuario como técnicos.
- Carmen Mok pregunta sobre las herramientas más utilizadas para la gestión de pruebas.
- Mario Jimenez menciona Azure DevOps y Jira como las herramientas populares para el seguimiento de pruebas, con experiencia particular en Azure DevOps.
- Carmen Mok pregunta sobre la experiencia de Mario Jimenez con pruebas automatizadas.
- Mario Jimenez menciona que no tiene experiencia personal en automatización, pero ofrece investigar y compartir información sobre las herramientas más populares.
- Carmen Mok consulta sobre la visión de Mario Jimenez respecto a una propuesta de metodología enfocada en pruebas automatizadas.
- Mario Jimenez enfatiza la importancia del respaldo de la alta dirección y la necesidad de consecuencias por la no adherencia al proceso.
- Adriana Álvarez explica que la implementación de la metodología propuesta está fuera del alcance del proyecto de graduación de Carmen debido a la resistencia al cambio y la complejidad de la implementación.
- Mario Jimenez apoya esta decisión desde una perspectiva práctica.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Agosto, 2024

**Firmas:**

ADRIANA DE LOS ANGELES ALVAREZ FIGUEROA (FIRMA)  
Digitally signed by ADRIANA DE LOS ANGELES ALVAREZ FIGUEROA (FIRMA)  
 Date: 2024.08.07 10:44:19 -0500

Adriana Álvarez Figueroa  
 Profesora Asesora



Mario Jimenez  
 Director Global de IT en Newfire Global Partners



Carmen María Mok Zheng  
 Estudiante del TEC

## 9.7.5. Entrevista con Luis Sanabria

### **Minuta de Entrevista**

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 10 de abril del 2024

**Consecutivo:** 008

**Convocados:**

- Adriana Álvarez Figueroa - Profesora Asesora
- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica
- Luis Sanabria - Ingeniero en Seguridad de Tecnologías de Información

**Hora de Inicio:** 04:30 p.m.

**Hora de Finalización:** 06:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de la estudiante Carmen Mok Zheng, la profesora asesora Adriana Álvarez, y Luis Sanabria, Ingeniero en Seguridad de Tecnologías de Información.

En este se trataron los siguientes puntos:

- Carmen Mok se presenta y brinda una breve introducción sobre el propósito de la entrevista. Carmen explica que el proyecto consiste en el desarrollo de una metodología adecuada enfocada en métricas y pruebas automatizadas para un departamento de QA Automation sin metodología formal.
- Carmen también explica que los objetivos del proyecto es poder medir el rendimiento del equipo, hacer seguimiento a las pruebas y realizar análisis post mortem de fallos.
- Carmen Mok pregunta sobre la experiencia y roles de Luis en aseguramiento de calidad.
- Luis explica que ha trabajado en aseguramiento de calidad por aproximadamente 8 a 10 años hasta 2009-2010; trabajó en aseguramiento de calidad de procesos y proyectos, y luego en seguridad informática. En su experiencia, se enfocó en la automatización de pruebas y en asegurar que el producto se entregara a tiempo y sin errores críticos. En el departamento de calidad, se priorizaba evitar interrupciones del negocio con nuevas versiones del software. Las pruebas eran manuales y se usaba HP Quality Center para el seguimiento de pruebas.
  - Procesos involucrados: inspección de requerimientos, determinación de pruebas necesarias, ejecución de pruebas funcionales, de rendimiento, de seguridad, y regresión.
  - Automatización de pruebas de regresión y creación de planes de prueba para nuevas funcionalidades.

- Identificación y manejo de defectos críticos antes de la liberación formal del software.
- Luis tiene experiencia en automatización de pruebas y gestión de calidad en proyectos de software, además de conocimiento en herramientas de seguimiento de pruebas y creación de planes de prueba.
- Carmen pregunta sobre las metodologías de QA más comunes como cascada, modelo V y ágil.
- Luis responde explica los puntos a modo de respuesta a la pregunta realizada:
  - No se usaba una metodología específica; el proceso se desarrollaba según la necesidad.
  - La empresa facilitó certificaciones de ingeniería en calidad a los miembros del departamento.
  - Se utilizaban ideas de entrenamientos y algunos procesos de CMMI nivel 2 y Six Sigma.
  - Luis ha trabajado con metodologías ágiles. En metodologías ágiles, es crucial asegurar que la fase de calidad tenga espacio en los sprints.
  - Importancia de la negociación con el equipo ágil para incluir aseguramiento de calidad en el proceso.
  - La implementación ágil sin considerar QA puede resultar en insuficiente tiempo para pruebas y productos defectuosos.
  - Luis también explica que el proceso era evolutivo y no seguían Agile al pie de la letra. La implementación de Agile siempre debe evolucionar según las necesidades.
- Carmen pregunta por recomendaciones de marcos de referencia para investigación y uso de pruebas automatizadas en QA.
- Luis explica que no existe un marco específico que pueda recomendar. El desarrollo que hicieron fue ad hoc. Mencionó procesos importantes del trabajo, pero no frameworks específicos de QA.
- Luis detalla que ha trabajado con Scrum e ITIL, pero no son áreas específicamente de QA. ITIL define el proceso de liberación a producción y la necesidad de pruebas, pero no cubre cómo se llega a la conclusión de que el software está listo para producción.
- Carmen pregunta qué métricas no podrían faltar en el proceso de QA.
- Luis Sanabria explica que las métricas son cruciales porque son usadas por la gerencia para evaluar el desempeño. El detalla lo siguiente:
  - La métrica principal en automatización es la cobertura del alcance.
  - Descripción del proceso:
    - Definición del alcance del software y las funcionalidades a probar.

- Ejemplo con un software de punto de venta: ciclo de liberaciones mensuales o bimensuales, donde cada liberación añade funcionalidades.
    - Comparación entre pruebas manuales y automatizadas:
      - Manual: 120 pruebas manuales pueden tomar dos a tres semanas para un tester.
      - Automatización: la transición gradual de pruebas manuales a automatizadas, incrementando la cobertura automatizada mes a mes.
    - Comunicación a la gerencia para mostrar la cobertura de pruebas automatizadas como porcentaje del plan de pruebas total.
      - Ejemplo de incremento: inicial sin automatización, ejecución manual; después de un mes, automatización de 10 pruebas, reducción del tiempo de pruebas manuales.
      - Objetivo: demostrar ahorro de tiempo y recursos, aunque la cobertura y la evolución del software cambien constantemente.
- La profesora Adriana Álvarez pregunta sobre el estándar sobre el porcentaje de pruebas automatizadas en la industria.
- Luis comenta que no existe un número estándar, ya que es muy variable dependiendo de la evolución del software. En fases iniciales de implementación, la funcionalidad puede cambiar rápidamente, afectando el porcentaje de cobertura, por ejemplo, en fases de mantenimiento, el porcentaje de cobertura puede ser alto, pero en fases de evolución, puede variar ampliamente.
- Luis destaca que la cobertura debe ser convertida en términos económicos o de tiempo para justificar la inversión, por ejemplo, ahorro de tiempo de pruebas manuales comparado con la inversión en automatización, o la importancia de mostrar a la gerencia el impacto económico y de tiempo de la automatización.
- La profesora Adriana pregunta si hay un porcentaje específico que debería disminuir.
- Luis explica que esto es muy específico de cada proyecto ya que diferentes tipos de software dentro de la misma compañía pueden tener distintos niveles de automatización. Se necesita definir métricas específicas para cada proyecto, considerando el tiempo y recursos disponibles.
- Luis menciona que una métrica adicional es el porcentaje de pruebas fallidas debido a defectos o ajustes necesarios en los scripts.
- La profesora Adriana pregunta sobre la existencia de un estándar de porcentaje de pruebas automatizadas en la industria.
- Luis explica que no hay un estándar fijo debido a la variabilidad en la evolución del software y las diferentes etapas del ciclo de vida del software.

- Luis también destaca la importancia de convertir la cobertura de pruebas en una métrica de tiempo ahorrado en pruebas manuales para justificar la inversión en automatización.
- Luis explica sobre la selección de pruebas a automatizar basándose en la frecuencia de uso y probabilidad de cambios en las funcionalidades, en otras palabras, es vital seleccionar las pruebas correctas para automatizar, enfocándose en funcionalidades críticas y frecuentemente usadas.
- La profesora Adriana menciona la importancia de encontrar un balance tipo Pareto, automatizando el 20% de los procesos que generen el 80% de los beneficios.
- Luis Pérez explica la diferencia entre defectos en nuevas funcionalidades y defectos de regresión, y cómo la automatización ayuda a identificar estos últimos. La automatización es más efectiva en detectar defectos de regresión, mientras que las pruebas manuales son necesarias para nuevas funcionalidades.
- Adriana Álvarez comenta sobre la relevancia de priorizar pruebas en funcionalidades que tienden a cambiar más y son más usadas.
- Luis Pérez sugiere que las métricas deben ser comunicadas de forma que permitan decisiones informadas, indicando la cobertura de pruebas automatizadas y su impacto en la detección de defectos.
- Carmen resalta la importancia de identificar en qué etapa se encontraron los defectos, ya sea en regresión o integración.
- Se concluye que es esencial proporcionar datos claros y útiles al management para decisiones estratégicas sobre la automatización.
- Luis Pérez describe la diferencia entre pruebas unitarias y funcionales, y cómo se mide el éxito en cada tipo de prueba, especialmente en el contexto de integración continua. Las pruebas unitarias requieren alta cobertura y monitoreo continuo para evitar fallos en el build, mientras que las pruebas funcionales deben enfocarse en la efectividad de la automatización en detectar defectos de regresión.
- Carmen pregunta sobre los tipos de pruebas que se realizan para asegurar la calidad del producto.
- Luis detalla los tipos de pruebas realizadas: UAT, pruebas de regresión, pruebas de integración y pruebas unitarias. La automatización ayuda principalmente en pruebas unitarias y de integración. Las pruebas de aceptación y beta son más difíciles de automatizar ya que requieren intervención humana.
- Carmen pregunta sobre herramientas recomendadas para la gestión de pruebas y reportes.
- Luis menciona herramientas como las de HP, que aunque son caras, son ampliamente usadas. También menciona JUnit para pruebas unitarias y herramientas open-source de Apache para pruebas funcionales. Destaca la necesidad de una herramienta para el seguimiento de defectos. También explica que existen diversas herramientas para la

gestión de pruebas y defectos, pero la elección depende del presupuesto y las necesidades específicas del proyecto.

- Carmen pregunta sobre la importancia de una metodología de QA enfocada en la automatización y el uso de métricas.
- Luis reafirma que las métricas son fundamentales para justificar el costo de la automatización y deben incluir el ahorro en horas de pruebas manuales. Es crucial realizar un seguimiento detallado del tiempo dedicado a pruebas manuales y automatizadas para medir el impacto y justificar la inversión.
- Carmen consulta si los "falsos negativos" se consideran como una métrica importante para el proceso de QA.
- Luis explica que en su opinión estas son una métrica importante pero con un alcance limitado ya que su público meta sería la supervisión directa del equipo de automatización de pruebas para controlar la calidad de las pruebas automatizadas y este podría llegar a ser confuso para las gerencias superiores u otros equipos ya que es un concepto difícil de explicar para quienes no están relacionados con los procesos de ejecución de pruebas.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Septiembre, 2024

**Firmas:**

\_\_\_\_\_  
Adriana Álvarez Figueroa  
Profesora Asesora

\_\_\_\_\_  
Carmen María Mok Zheng  
Estudiante del TEC

\_\_\_\_\_  
Luis Sanabria  
Ingeniero en Seguridad de Tecnologías de Información

## 9.7.6. Entrevista con Ignacio Trejos

### Minuta de reunión

**Proyecto:** Propuesta de Metodología de QA para el Departamento de QA Automation

**Fecha:** 8 de Marzo del 2024

**Consecutivo:** 009

**Convocados:**

- Carmen María Mok Zheng – Estudiante de Maestría del Tecnológico de Costa Rica
- Ignacio Trejos Zelaya - Co-Fundador y Profesor+Investigador de Cenfotec

**Hora de Inicio:** 03:30 p.m.

**Hora de Finalización:** 04:30 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Microsoft Teams, con la participación de Carmen María Mok Zheng y el profesor Ignacio Trejos Zelaya.

En este se trataron los siguientes puntos:

- Carmen Mok se presenta y brinda una breve introducción sobre el propósito de la entrevista. Carmen explica que el proyecto consiste en el desarrollo de una metodología enfocada en métricas y pruebas automatizadas, para un departamento de QA Automation sin metodología formal, con potencial valor para otras empresas que buscan utilizar la automatización de QA.
- Carmen preguntó si Ignacio podía recomendar a alguien para contactar y preguntar sobre tipos de pruebas de automatización realizadas en sus empresas.
- Ignacio aclaró que su experiencia práctica en aseguramiento de calidad del software es limitada, siendo mayormente académica y sistemática.
- A pesar de su limitada experiencia práctica, Ignacio se ofreció a proporcionar recomendaciones de personas involucradas en aseguramiento de calidad y pruebas de software, aunque advirtió sobre la posible reticencia de estas personas debido a conflictos de interés con sus empleadores.
- Ignacio mencionó varios posibles contactos con vasto conocimiento en pruebas de software.
- Ignacio se comprometió a enviar estos contactos y a ayudar con información sobre estándares y marcos de referencia en QA.
- Carmen pregunta qué tan común es que una empresa utilice metodologías de QA enfocadas en la automatización en empresas.
- El profesor Ignacio responde que las pruebas automatizadas (automated testing) son parte del control de calidad. Es esencial diseñar cuándo y cómo se realizan, quiénes son responsables y cómo se gestionan los datos generados.
- Carmen explica que la automatización ayuda a reemplazar tareas repetitivas, permitiendo que los QA se enfoquen en nuevas pruebas.

- Ignacio añade que el diseño de automatización debe incluir actividades manuales y la gestión de datos, asegurando que los scripts de automatización sean eficientes y ordenados.
- Ignacio aconseja documentar el proceso actual y, a la luz de los estándares de calidad, identificar oportunidades de mejora, además de revisar estándares como ISTQB y IEEE para guiar el proceso.
- El profesor Ignacio también recomienda usar el estándar IEEE 730 y revisar el ISTQB Certified Tester Foundation Level Syllabus.
- Carmen consulta cuál es la proporción de testers en relación a los developers.
- Ignacio responde que en empresas grandes, la proporción puede variar. Ejemplos mencionados incluyen el Banco Central y Microsoft.
- Ignacio también añade que la automatización debe estar justificada por costo-beneficio y oportunidad, mejorando la calidad y tiempo de entrega. Menor tiempo de pruebas y mayor confianza en la calidad del producto.
- Carmen consulta si existe un estándar en la cantidad de fallos permitidos.
- Ignacio explica que no hay un estándar específico y que esto depende de la organización y la industria. Hay organizaciones que recogen y reportan datos sobre métricas de software.
- Ignacio recomienda revisar libros sobre automatización y procesos de negocio. Buscar información y benchmarks en proyectos Open Source y grandes organizaciones como Google y Microsoft.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Septiembre, 2024

**Firmas:**

IGNACIO  
TREJOS  
ZELAYA  
(FIRMA)

Firmado digitalmente  
por IGNACIO TREJOS  
ZELAYA (FIRMA)  
Fecha: 2024.08.28  
20:55:41 -06'00'

Ignacio Trejos Zelaya  
Co-Fundador y Profesor+Investigador de Cenfotec



Carmen María Mok Zheng  
Estudiante del TEC

## 9.7.7. Entrevista con Marlen Treviño

### **Minuta de Entrevista**

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 12 de abril del 2024

**Consecutivo:** 010

**Convocados:**

- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica
- Marlen Treviño - Profesora del Tecnológico de Costa Rica

**Hora de Inicio:** 08:00 a.m.

**Hora de Finalización:** 09:30 a.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de la estudiante Carmen Mok Zheng y Marlen Treviño, Profesora del Tecnológico de Costa Rica.

En este se trataron los siguientes puntos:

- Carmen Mok se presenta y brinda una breve introducción sobre el propósito de la entrevista. Carmen explica que el proyecto consiste en el desarrollo de una metodología adecuada enfocada en métricas y pruebas automatizadas para un departamento de QA Automation sin metodología formal, con potencial valor para otras empresas que buscan utilizar la automatización de QA.
- Carmen pregunta por la experiencia laboral de Marlen.
- Marlen mencionó su trabajo en el Instituto Tecnológico de Costa Rica desde 2007 como docente e investigadora, enseñando cursos sobre bases de datos, sistemas de información geográfica y aseguramiento de la calidad del software. También señaló su certificación ISTQB Foundation level y avanzada.
- Carmen pregunta qué metodologías de QA son más comunes.
- Marlen explica que en los últimos años ha existido una preferencia con el uso de metodologías ágiles en las organizaciones modernas debido a la necesidad de respuestas rápidas. Marlen enfatizó que la elección de la metodología depende tanto de la organización como del proyecto específico.
- Carmen pregunta por recomendaciones de marcos de referencia para el tema de esta investigación.
- Marlen recomendó el estándar ISTQB como una referencia clave en pruebas, mencionando también los estándares IEEE y su relevancia en el ámbito de la automatización de pruebas.
- Carmen pregunta por métricas que considera que no pueden faltar en el proceso de QA.

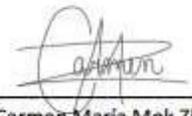
- Marlen identificó varias métricas importantes en el proceso de QA, como el número de defectos encontrados, el número de casos de prueba ejecutados, y la complejidad ciclométrica de los algoritmos.
- Marlen también mencionó la importancia de métricas como el número de pruebas automatizadas, el tiempo de ejecución de cada prueba, el número de defectos, y el número de casos de prueba exitosos y fallidos.
- Carmen pregunta qué tipos de pruebas se deben realizar para asegurar la calidad de un producto.
- Marlen explicó los cinco niveles de pruebas definidos por el ISTQB: pruebas unitarias, de integración, de sistema, de aceptación y de mantenimiento. También mencionó los tipos de pruebas estáticas y dinámicas.
- Carmen pregunta si el uso de las pruebas, por ejemplo, las pruebas unitarias, pruebas de regresión, y las pruebas de integración, dependen mucho del momento o situación en el que se va a revisar un producto.
- Marlen aclaró que las pruebas de regresión se utilizan cuando se hace un cambio en una aplicación existente para asegurarse de que los cambios no introduzcan nuevos defectos.
- Carmen pregunta qué herramientas recomienda para la gestión de las pruebas o reporte de los resultados de las pruebas.
- Marlen recomendó varias herramientas para diferentes aspectos del proceso de pruebas, incluyendo TestLink y Jira para la gestión de pruebas, RMTrack y Mantis Bug Tracker para el seguimiento de defectos, y Selenium y Postman para la ejecución de pruebas automatizadas.
- Marlen destacó que la elección de herramientas también depende del presupuesto de la organización, ya que hay opciones tanto gratuitas como de pago.
- Carmen preguntó por la experiencia de Marlen utilizando pruebas automatizadas.
- Marlen explicó que, aunque no tiene experiencia profesional directa en pruebas automatizadas, ha trabajado con ellas en un contexto académico. En sus cursos de calidad, dedica tiempo a la automatización de pruebas unitarias con frameworks como Java, Angular, Karma, y Jasmine. También mencionó el uso de Protractor y Cypress para pruebas de caja negra y Apache JMeter para pruebas de rendimiento.
- Carmen pregunta qué tecnologías recomienda para la implementación de pruebas automatizadas.
- Marlen señaló que las tecnologías recomendadas para automatización dependen del tipo de prueba y del lenguaje de programación utilizado en el desarrollo. Por ejemplo, JUnit se utiliza para pruebas unitarias en Java, mientras que las pruebas de caja negra pueden usar frameworks como Protractor o Cypress.
- Carmen pregunta por la opinión de Marlen sobre esta investigación.
- Marlen comentó que le parece interesante la idea de desarrollar una metodología de QA enfocada en la ejecución de pruebas automatizadas, especialmente en organizaciones que actualmente no cuentan con una metodología formal. Mencionó que esto podría ser beneficioso no solo para la organización en cuestión, sino también para otras empresas que puedan adoptar y enriquecer la metodología.

- Marlen compartió su experiencia en un proyecto de realidad aumentada y señaló que las pruebas en aplicaciones de juegos tienen características únicas. Subrayó la importancia de la aceptación del usuario y la posibilidad de que una metodología formalizada podría ser muy útil para empresas desarrolladoras de juegos en Costa Rica.
- Marlen indicó que generalmente, cada empresa define internamente lo que considera una cantidad aceptable de fallos. La importancia de los defectos depende de su severidad y prioridad, y es crucial considerar estos factores al determinar si un software está listo para su liberación.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Septiembre, 2024

**Firmas:**  Firmado digitalmente por  
MARLEN TREVIÑO  
VILLALOBOS (FIRMA)  
Fecha: 2024.08.07  
1233256 -06702

Marlen Treviño  
Profesora del Tecnológico de Costa Rica

  
Carmen María Mok Zheng  
Estudiante del TEC

## 9.7.8. Entrevista con Laura Chavarría

### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 06 de abril del 2024

**Consecutivo:** 011

**Convocados:**

- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica
- Laura Chavarría - MATI (Master en Administración de la Tecnología de la Información)

**Hora de Inicio:** 05:00 p.m.

**Hora de Finalización:** 06:30 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Google Meet, con la participación de la estudiante Carmen Mok Zheng y Laura Chavarría, Master en Administración y Tecnología de la Información.

En este se trataron los siguientes puntos:

- Carmen Mok se presenta y brinda una breve introducción sobre el propósito de la entrevista. Carmen explica que el proyecto consiste en el desarrollo de una metodología adecuada enfocada en métricas y pruebas automatizadas para un departamento de QA Automation sin metodología formal, con potencial valor para otras empresas que buscan utilizar la automatización de QA.
- Laura confirmó que había leído y estudiado el documento previo a la entrevista, subrayando la importancia de comprender el problema antes de dar retroalimentación.
- Se discutió la orientación a métricas del proyecto de Carmen y Laura proporcionó ejemplos y metodología relevante.
- Laura compartió su experiencia trabajando en QA, destacando un proyecto significativo en Costa Rica y la importancia del aseguramiento de calidad.
- Se mencionaron los desafíos en la implementación de sistemas de QA en empresas, y Laura comentó sobre su experiencia en testing y su evolución profesional.
- Laura habló sobre su experiencia personal y profesional, incluyendo su transición a la consultoría y docencia, así como la influencia de su familia en su carrera.
- Carmen explicó que su proyecto se enfoca en identificar la brecha entre la situación actual del departamento de QA y las buenas prácticas para proponer una metodología adecuada.
- Carmen mencionó que busca mejorar la calidad del proceso y que conoce poco sobre marcos de referencia específicos.
- Carmen pregunta qué metodologías de QA considera más comunes en la industria.
- Laura explicó que existen dos enfoques principales: tradicionales y ágiles.

- Laura mencionó la metodología ISTQB y el modelo en V, destacando su estructura desde la concepción hasta el final del proyecto, estableciendo criterios de calidad en cada etapa.
- Laura destacó la importancia de definir criterios de aceptación desde el inicio, incluyendo la participación de expertos y stakeholders relevantes.
- Laura explicó que en cada etapa del proyecto se establecen criterios de calidad que deben ser cumplidos antes de avanzar, lo que asegura que al final el producto cumpla con los estándares deseados.
- Laura resaltó la diferencia entre las metodologías tradicionales y ágiles, mencionando que las ágiles suelen trabajar con prototipos, lo que puede ser más costoso si no se gestionan adecuadamente.
- Laura mencionó que la metodología de desarrollo de la empresa debe ser considerada al implementar QA, ya que un enfoque tradicional como el modelo en V no sería adecuado en un entorno ágil.
- Laura sugirió que Carmen debe adaptar su propuesta de calidad a la metodología ágil que utiliza su empresa, integrando QA de manera que no cause retrasos en el desarrollo.
- Laura mencionó que es fundamental convencer a la empresa de la importancia de QA, mostrando cómo puede mejorar la calidad y reducir el retrabajo.
- Laura explicó que QA debe hacer visible la calidad, cuantificando errores y asegurando que los productos cumplan con los estándares deseados.
- Laura recomendó que Carmen estudie la metodología de desarrollo de su empresa y adapte la propuesta de QA a dicha metodología.
- Laura sugirió que Carmen se enfoque en hacer visible la calidad, mostrando métricas y resultados que demuestren la importancia de QA.
- Laura enfatiza la importancia de estudiar y aplicar metodologías que se alineen con el desarrollo y los participantes del proyecto. También señala que los Project Managers (PMs) a menudo no valoran el QA porque puede retrasar los proyectos.
- Laura explica que la calidad puede detener un proyecto, especialmente si hay métricas que demuestran que no se está cumpliendo con los estándares necesarios, y recomienda evidenciar problemas con datos concretos para evitar la liberación de productos defectuosos.
- Carmen reconoce la responsabilidad que implica el trabajo de QA.
- Laura comparte su experiencia personal y profesional, subrayando que QA es un trabajo de gran responsabilidad que puede afectar la vida personal.
- Laura recomienda cursos de habilidades blandas para mejorar la interacción con equipos y manejar mejor las emociones y situaciones difíciles.
- Carmen consulta sobre las métricas esenciales en el proceso de QA, mencionando ejemplos como el número de defectos y el tiempo de ejecución de casos.
- Laura enumera las métricas clave:
  - **Tipo y cantidad de defectos:** Clasificación por severidad, tipo (procedimiento, especificación, diseño, documentación, etc.).
  - **Severidad del defecto:** Impacto en el proyecto (severidad 1, 2, 3).
  - **Retrabajo:** Horas implicadas en corregir defectos.

- **Etapas del proyecto donde se encuentran defectos:** Defectos en fases tempranas versus posteriores.
  - **Defectos encontrados antes y después de la liberación del producto.**
- Laura resalta la importancia de tener un test plan numerado para rastrear y reportar resultados específicos.
- Carmen pregunta sobre la utilidad de realizar un postmortem para aprender de los proyectos.
- Laura explica que la herramienta de Análisis de Ambigüedades no es para la etapa de post mortem del proyecto y que esta herramienta es muy útil en las primeras etapas del proyecto, específicamente en la "Etapa de Análisis del Proyecto".
- También menciona que la herramienta de Análisis de Ambigüedad ayuda a determinar si la especificación del proyecto es adecuada para continuar o si hay ambigüedades que necesitan ser resueltas.
- Laura indica que los criterios de aceptación se deben definir en la etapa de análisis del proyecto y explica que estos criterios deben verificarse en las últimas etapas, al momento de aceptar el proyecto.
- Laura añade que el cliente o usuario es quien define y verifica el cumplimiento de estos criterios.
- Laura recomienda empezar con fases de implementación y numerar todas las pruebas para tener un control detallado de los resultados y las métricas.
- Laura sugiere iniciar con métricas básicas y adaptar herramientas disponibles a las necesidades específicas del proyecto y la empresa.
- Laura se ofrece a investigar y compartir ejemplos adicionales de métricas basadas en su experiencia.
- Carmen agradece las recomendaciones y confirma que su empresa ya maneja identificadores de Jira para las pruebas automatizadas y no automatizadas.
- Carmen pregunta sobre los tipos de pruebas que se realizan para asegurar la calidad del producto y cuáles son las más comunes o si dependen del contexto.
- Laura explica que no hay una prueba más importante que otra. Los proyectos pueden incluir pruebas unitarias, de integración y de usuario.
- Laura recomienda conocer todos los tipos de pruebas disponibles y elegir las más adecuadas para cada proyecto.
- También menciona que algunas pruebas pueden no haber sido necesarias anteriormente pero podrían ser relevantes según el proyecto actual y sugiere investigar y conocer la teoría sobre los diferentes tipos de pruebas.
- Carmen consulta sobre herramientas utilizadas para la gestión de pruebas y generación de reportes.
- Laura menciona que a menudo se desarrollan soluciones a medida para la gestión de resultados de QA y explica que muchas organizaciones utilizan Excel debido a sus características avanzadas.
- Laura comenta sobre herramientas para llevar el inventario de componentes y administrar la configuración, que pueden incluir el mapeo de servidores y objetos de desarrollo reutilizables.

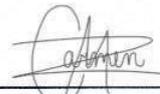
- Carmen explica que su empresa utiliza JIRA para la administración del ciclo de vida del proyecto y pregunta sobre la experiencia de Laura con pruebas automatizadas.
- Laura indica que no tiene experiencia directa con pruebas automatizadas.
- Carmen solicita la opinión de Laura sobre la implementación de una metodología enfocada en pruebas automatizadas y sus beneficios para las organizaciones.
- Laura afirma que una metodología de pruebas automatizadas es beneficiosa, destacando que hay una necesidad evidente de mejorar el proceso de QA.
- Laura explica que las métricas son esenciales para medir el éxito y que la automatización es una extensión natural del aseguramiento de calidad.
- Laura recomienda investigar si la herramienta actual está siendo utilizada al máximo de su capacidad.
- Laura pregunta sobre la existencia de un ambiente de pruebas separado del ambiente de desarrollo en la empresa de Carmen.
- Carmen indica que tienen ambientes de QA, pero no uno específico para automatización, lo que causa problemas debido a cambios constantes en la configuración por parte del equipo de QA de otros departamentos.
- Laura reconoce la importancia de tener un ambiente de pruebas separado para evitar problemas de configuración y sugiere avanzar hacia un mayor nivel de madurez en la gestión de QA.
- Carmen pregunta sobre el tamaño típico de los equipos de QA y si esto depende de la empresa y el proyecto.
- Laura explica que el tamaño del equipo de QA depende de la empresa y la cantidad de proyectos.
- Laura menciona que un QA puede estar asignado a varios proyectos y que la complejidad del proceso determina la necesidad de personal adicional.
- Laura recomienda ajustar el tamaño del equipo según la metodología y la fase del proyecto.
- Carmen agradece a Laura por su tiempo y sus respuestas, confirmando que todas sus preguntas han sido abordadas satisfactoriamente. Carmen también expresa su aprecio por la valiosa información proporcionada por Laura.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Septiembre, 2024

**Firmas:**



Laura Chavarría  
Master en Administración  
de la Tecnología de la Información



Carmen María Mok Zheng  
Estudiante del TEC

## 9.8. Anexo 8

### Minuta de Entrevista

**Proyecto:** Propuesta de Metodología de QA para el departamento de QA Automation

**Fecha:** 23 de mayo del 2024

**Consecutivo:** 012

**Convocados:**

- Carmen Mok Zheng - Estudiante de Maestría del Tecnológico de Costa Rica
- [REDACTED] - Gerencia de Cumplimiento de la Calidad

**Hora de Inicio:** 01:00 p.m.

**Hora de Finalización:** 02:00 p.m.

La reunión se realizó de forma virtual por medio de la plataforma Microsoft Teams, con la participación de la estudiante Carmen Mok Zheng y Johnny Marín, Gerencia de Cumplimiento de la Calidad de la empresa en donde se realiza el proyecto.

En este se trataron los siguientes puntos:

- Johnny Marín explicó que Jira está configurado de manera genérica, y diferentes proyectos utilizan distintas prioridades. Para los bugs, se emplean comúnmente las prioridades "low," "medium," "major," y "critical."
- Johnny Marín mencionó la existencia de una guía para decidir qué prioridad asignar a un bug, y explica que no se utiliza severidad en la actualidad.
- Carmen Mok propuso la idea de incluir la severidad como un campo adicional, similar a una matriz de gestión de riesgos, para ayudar a evaluar la calidad del producto.
- Johnny Marín señaló que, aunque la severidad podría ser una opción, podría generar incertidumbre debido a la falta de visibilidad del impacto.
- Johnny Marín explicó que la prioridad de un bug está influenciada por su severidad y frecuencia de ocurrencia. Comentó que una alta severidad normalmente lleva a una alta prioridad, aunque la frecuencia también juega un papel importante.
- Se discutió la posibilidad de utilizar una matriz que combine severidad y frecuencia para determinar la prioridad de un bug, lo que podría facilitar el proceso de asignación.
- Carmen Mok preguntó sobre la práctica de realizar reuniones durante el 80-90% de la regresión para revisar bugs críticos con desarrolladores y product owners.
- Johnny Marín aclaró que no se sigue un número fijo de bugs para estas reuniones; lo que determina la necesidad de revisión es la severidad del bug, aunque esta no siempre se encuentra documentada directamente en el sistema.

- Se discutió cómo la prioridad de un bug está influenciada por su severidad. Johnny Marin explicó que, aunque la prioridad se utiliza para definir la urgencia con la que se debe resolver un bug, esta generalmente está influenciada por la severidad. Sin embargo, aclaró que la cantidad de bugs no es un factor decisivo.
- Carmen Mok propuso la idea de establecer una guía o reglas para medir la calidad del producto basada en la severidad y prioridad de los bugs.
- Johnny Marin sugirió que una posible métrica podría involucrar la asignación de valores cuantitativos a los bugs, basados en su prioridad e impacto, y luego sumar esos valores para obtener una indicación de la calidad.
- Johnny Marin explicó que el proceso de deployment es realizado por el equipo de DevOps. Describió cómo el proceso incluye un code freeze, la generación de paquetes por parte de Solutions, y la revisión de los mismos durante las pruebas de integración y regresión.
- Se discutió cómo el impacto de un bug puede estar relacionado con áreas críticas del producto. Johnny Marin explicó que la prioridad de un bug es determinada por una combinación de su severidad y frecuencia de ocurrencia, sugiriendo que una matriz que considere ambos factores podría ser útil para medir la prioridad.
- Carmen Mok propuso la idea de utilizar una matriz que considere la severidad y la frecuencia para determinar la prioridad, lo cual podría ayudar a evaluar la calidad del release.
- Johnny Marin apoyó esta idea, indicando que, aunque actualmente se mide la calidad principalmente por la cantidad de defectos, un enfoque más cuantitativo podría proporcionar una mejor visión de la calidad del producto.
- Carmen Mok continuará desarrollando su propuesta de tesis, que incluirá una matriz cuantitativa para evaluar la calidad del producto. Se compromete a compartir los documentos finales con Johnny Marin para recibir su retroalimentación antes de su presentación final.

Acuerdos	Responsable	Fecha límite
La información obtenida en esta reunión será utilizada para el desarrollo del proyecto.	Carmen María Mok Zheng	Septiembre, 2024

Firmas:



Gerencia de Cumplimiento  
de la Calidad

Carmen María Mok Zheng  
Estudiante del TEC

## 9.9. Anexo 9

### Acta de Aceptación del Documento

Fecha: 30 de Septiembre, 2024

Proyecto: Propuesta de Metodología de QA para el departamento de Automatización de QA

Responsable:

- Solicitante: Carmen María Mok Zheng
  - Elaborado por: Carmen María Mok Zheng
  - Aprobado por: [REDACTED]
- 

**Objetivo del Documento:**

El presente documento tiene como objetivo presentar el análisis de las mejores prácticas identificadas a partir de diversos marcos de referencia y entrevistas realizadas con expertos del área. Dichas mejores prácticas han sido seleccionadas con base en su aplicabilidad dentro del contexto actual del departamento de QA Automation, y forman parte de una propuesta de metodología de QA. Asimismo, se incluye un plan de acción para la implementación de estas prácticas.

---

**Resumen del Contenido:**

**1. Análisis de Marcos de Referencia:**

- Descripción de los marcos de referencia analizados
  - i. ISTQB Certified Tester Foundation Level
  - ii. ISTQB Advanced Level Syllabus - Test Automation Engineer
  - iii. PMBOK
  - iv. CMMI
  - v. Series de ISO/IEC/IEEE 29119
- Identificación de las mejores prácticas aplicables a QA Automation.

**2. Análisis del estado actual del departamento de QA Automation:**

- Descripción del estado actual del departamento de QA Automation
  - i. Metodología actual
  - ii. Procedimiento para la liberación del software
  - iii. Sistemas de métricas presentes en la metodología de QA actual

- iv. Necesidades detectadas
- v. Expectativas detectadas

**3. Propuesta de Metodología de QA:**

- o Definición de la metodología sugerida para el departamento de QA Automation.
- o Detalle de los pasos y procedimientos a seguir para su adopción.
- o Mejores prácticas seleccionadas en función de las necesidades y retos del departamento de QA Automation.

**4. Plan de Acción:**

- o Propuesta de un cronograma para la implementación de las mejores prácticas seleccionadas.
- o Identificación de recursos y responsables para llevar a cabo la implementación.
- o Indicadores de éxito para medir el impacto de la metodología propuesta.

---

**Conclusión:**

Los firmantes de esta acta expresan su conformidad con el contenido del documento y la propuesta de metodología de QA. Se acepta su validez para la implementación de las mejores prácticas seleccionadas, bajo los lineamientos establecidos en el plan de acción.

---

**Firmas:**

Johnny Marín

Cargo: Gerencia de Cumplimiento de la Calidad

Firma: 