

# Plan de Calidad Para el Proyecto de Migración de Astro Millonario a la Herramienta Configurador de Productos

Jorge Enrique Gutiérrez Reyes

Fundación Universitaria Católica Lumen Gentium

*Cali, Colombia*

guti.reyes26@gmail.com

**Resumen—** Actualmente en la empresa de desarrollo de software que sirve como proveedor tecnológico a gran parte de las casas de apuestas del país, se encuentra realizando un proyecto que permite flexibilizar la creación de los productos que se comercializan en dichas empresas, ofreciendo autonomía a los clientes para que no necesariamente tengan que escalar sus requerimientos al proveedor tecnológico, sino que puedan hacerlos con la nueva herramienta por su propia cuenta y de forma ágil. Durante el desarrollo de este proyecto, es indispensable realizar un proceso de QA que permita asegurar que se satisfacen los requerimientos del cliente y que se minimicen las fallas que puedan presentarse en el producto final. En este artículo se propone un plan de calidad aplicando las mejores prácticas de la Norma ISO para aplicar durante el proceso de migración del producto Astro Millonario a la arquitectura del software configurador de productos, que permitirá que al momento de la entrega final, se tenga un impacto positivo en el cliente, el cual impulsará la implementación del producto para todas las empresas del gremio. Como resultado se obtuvo una matriz de pruebas donde se especifican cinco casos de pruebas que se hacen al producto entregado y se minimizan los riesgos de fallos en el software.

## I. INTRODUCCIÓN

Las pruebas de software se constituyen en un filtro para todo el proceso, ya que se aplican repetidamente a lo largo del desarrollo del producto, y permiten descubrir errores de forma que puedan corregirse antes de pasar a ambientes productivos.<sup>[1]</sup> Para la ejecución de las pruebas de calidad, es importante contar con un plan que permita identificar correctamente los recursos necesarios para ejecutar las pruebas, los componentes a probar, los casos de prueba y que incluya un cronograma de actividades, de forma que se puedan reducir de forma eficaz los errores y defectos encontrados en los ambientes productivos.

Según la norma UNE-EN ISO 9000:2005, sistemas de gestión de la calidad, fundamentos y vocabulario, se define Plan de calidad como documento que especifica qué procedimientos y recursos asociados deben aplicarse, quién debe aplicarlos y cuándo deben aplicarse a un proyecto, producto, proceso o contrato específico.<sup>[1]</sup>

Este proceso de QA es muy importante en un proyecto y es tenido en cuenta principalmente en la fase final del desarrollo donde se asegura que el producto a entregar cumple con los requerimientos funcionales y no funcionales definidos durante la etapa de recolección de requerimientos. De igual forma, asegura que se reduzca de forma importante la cantidad de fallas que puedan liberarse al ambiente productivo del cliente.

Las revisiones del software son un “filtro” para el proceso del software. Es decir, se aplican en varios puntos durante la ingeniería de software y sirven para descubrir errores y defectos a fin de poder eliminarlos. Las revisiones del software “purifican” los productos del trabajo de la ingeniería de software, incluso los modelos de requerimientos y diseño, código y datos de prueba.<sup>[1]</sup>

Durante la ejecución del proyecto, el analista de desarrollo realiza entregas pequeñas y tempranas para que el analista de calidad realice una iteración de pruebas que permita verificar la funcionalidad del producto entregado. La participación del analista de calidad en el inicio del proyecto, permite verificar desde la documentación, cómo la historia de usuario y los casos de uso cumplen con los requerimientos definidos con el cliente.

Antes de dar inicio al desarrollo se realiza un documento de pruebas de unidad donde se especifican las pruebas que se deben tener en cuenta para ejecutarlas sobre el producto elaborado. Igualmente, el analista de calidad elaborará durante el desarrollo los documentos que permitan verificar que se cumpla con todas las pruebas definidas para la validación del producto.

Varios estudios de la industria indican que las actividades de diseño introducen de 50 a 65 por ciento de todos los errores (y en realidad de todos los defectos) durante el proceso del software. Sin embargo, las técnicas de revisión han demostrado tener una eficacia de hasta 75 por ciento [Jon86] para descubrir fallas del diseño.<sup>[1]</sup>

Como objetivo el Plan de Pruebas de Software se elabora para determinar los componentes que se van a probar en la

liberación del Release, para realizar el proceso de Validación y Verificación de los requerimientos e incidentes.

El Plan de Pruebas se aplica para obtener errores, defectos o fallas en el sistema, y así realizar las correcciones pertinentes según el caso para asegurar la calidad del producto que será entregado al cliente.

En el presente artículo se plantea un plan de calidad para realizar el proceso de pruebas al producto Astro millonario, el cual está siendo migrado actualmente a una nueva arquitectura, la cual involucra una herramienta llamada configurador de productos y que permite que los usuarios finales realicen modificaciones a los productos que comercializan actualmente y creen nuevos productos sin depender del proveedor de software. Finalmente se muestran los resultados de realizar una primera iteración del plan de calidad planteado, junto con los documentos asociados a este proceso.

## II. PLANTEAMIENTO DEL PROBLEMA

Las empresas de juegos de apuestas y azar actualmente envían requerimientos de nuevos desarrollos a su proveedor tecnológico solicitando la creación y/o configuración de nuevos productos de chance para comercializarlos luego en los puntos de venta.

Con el fin de reducir la cantidad de requerimientos recibidos, se decidió crear el software configurador de productos, el cual permite a las empresas de apuestas crear sus propios productos de chance sin necesidad de enviar peticiones de nuevos desarrollos al proveedor tecnológico.

El software configurador de productos tiene una arquitectura diferente a la de los sistemas que se utilizan actualmente en las empresas, por lo cual se hace necesario que los productos que están siendo comercializados en la actualidad, sean migrados a la nueva arquitectura para permitir que las empresas hagan uso de una sola plataforma.

Astro millonario es uno de los productos de chance con mayores niveles de venta en la mayoría de las casas de apuestas, por lo cual su migración es de vital importancia para dar un impulso positivo a la implementación del software configurador de productos, para esto, es indispensable que el producto llegue al cliente con la menor cantidad de fallas posible, por lo cual se diseñó un plan de pruebas que permita asegurar el correcto funcionamiento del producto astro millonario bajo la nueva arquitectura.

## III. OBJETIVOS

### Objetivo General

Definir el plan de pruebas a aplicar para el proceso de

venta de la primera versión del producto Astro millonario, migrado a la nueva arquitectura de configurador de productos.

### Objetivos Específicos

- Identificar los tipos de prueba que se deben aplicar para el producto Astro millonario, una vez migrado a la arquitectura de configurador de productos.
- Definir los recursos, físicos, humanos y lógicos que serán necesarios para llevar a cabo las pruebas.
- Establecer los casos de pruebas necesarios para cubrir la mayor parte de funcionalidades del proceso de venta del producto Astro millonario.
- Plantear el cronograma necesario para llevar a cabo las pruebas del proceso de venta del producto Astro millonario.
- Ejecutar una primera iteración del plan de pruebas planteado y generar la respectiva documentación.

## IV. MARCO TEORICO

### Calidad del Software

La calidad de software es el “grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida, dichas características de cierta manera garantizan que el cliente cuente con un sistema confiable, lo cual aumenta su satisfacción frente a la funcionalidad y eficiencia del sistema construido.” [\[2\]](#)

Se define también como un “Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan”. [\[1\]](#)

### Modelo de Calidad de Software

Los modelos de calidad de software se definen como “documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad”. [\[2\]](#)

### Modelo ISO 9126

El modelo ISO 9126 está basado en el modelo de McCall, el cual está dirigido a desarrolladores, aseguradores de calidad, evaluadores, analistas y en general a quienes estén involucrados en el proceso de desarrollo de software. [\[2\]](#)

Este modelo establece 6 factores clave para la evaluación de la calidad del software, los cuales se definen a continuación: [1]

**Funcionalidad:** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

**Confiabilidad:** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.

**Usabilidad:** Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.

**Eficiencia:** Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.

**Facilidad de recibir mantenimiento:** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.

**Portabilidad:** Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible

Por su parte Roger Pressman, establece 4 atributos que se deben perseguir y que están directamente relacionados con la interfaz del software [1]:

**Intuitiva:** Grado en el que la interfaz sigue patrones esperados de uso, de modo que hasta un novato la pueda utilizar sin mucha capacitación.

**Eficiencia:** Grado en el que es posible localizar o iniciar las operaciones y la información.

**Robustez:** Grado en el que el software maneja entradas erróneas de datos o en el que se presenta interacción inapropiada por parte del usuario.

**Riqueza:** Grado en el que la interfaz provee un conjunto abundante de características.

## Plan de Calidad de Software

Un plan de calidad de software se define como un “documento que especifica qué procedimientos y recursos deberían aplicarse, quién debe aplicarlos y cuándo deberían aplicarse a un proyecto, proceso, producto o contrato específico, de manera de poder alcanzar los objetivos de la calidad.” [3]

### Tipos de Pruebas

A continuación se describen algunos tipos de pruebas [4]:

**Funcionales:** “Las pruebas funcionales se enfocan en validar la correcta implementación de las necesidades del cliente”. Durante estas pruebas, se ingresan datos al sistema y se comparan los datos de salida con los resultados esperados, esto, basado en los requerimientos.

**No Funcionales:** “Las pruebas no funcionales revisan las características implícitas del sistema, lo que las hace difícil de validar.” Estas pruebas se refieren a características que el software debe tener, pero que no son expresadas por el cliente.

Existen diferentes tipos de pruebas no funcionales:

**Rendimiento:** Rapidez con la cual un sistema ejecuta una determinada función.

**Estrés:** Someter a la aplicación a una carga mucho mayor a la esperada y evaluar la capacidad del sistema de seguir su operación después de este tipo de demanda.

**Fiabilidad:** El sistema debe ser confiable a los usuarios para que puedan operar sin preocuparse de sus datos o de la mala operación de un sistema.

**Mantenibilidad:** El código debe cumplir con ciertas normas y estándares para evitar que se encuentre de forma compleja, el código siempre debe estar comentado para cada función realizada.

**Portabilidad:** Se deben asegurar las características del sistema; en qué sistema operativo operará, cuál será el navegador de preferencia, etcétera.

Técnicas de Prueba [5]

**Estáticas:** “Este tipo de técnicas son aquellas que no ejecuta la aplicación. Se llevan a cabo a nivel de especificaciones. No ejecutan código pero si realizarán un análisis estático del código”

**Dinámicas:** Estas técnicas son las que se realizan ejecutando la aplicación. Existen varios tipos de pruebas dinámicas, las cuales se describen a continuación:

**Pruebas de Caja Blanca:** La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez. [1]

**Pruebas de caja negra:** Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación. [1]

**No conformidad:** Según la norma ISO 9000, una no conformidad, se define como el “Incumplimiento de un requisito” [6]

## QA

El término QA se refiere a Quality Assurance, es decir al aseguramiento de la calidad. Esto puede ser aplicado a productos y servicios que sean manufacturados o prestados. Sin embargo un aseguramiento de la calidad va más allá de una supervisión de procesos como se pudiera pensar. [7]

Un QA técnico verifica la calidad de un sistema en varias etapas [7]:

**Pruebas de integración:** Pruebas que garantizan que los componentes funcionan bien juntos. Estos son realizados por developers y verificados por el QA. Estas pruebas son basadas en las pruebas unitarias y llevadas al siguiente nivel.

**Pruebas de aceptación:** Se testean escenarios de aplicaciones de acuerdo a perfiles de usuarios. Los niveles de pruebas incluyen funcionalidad, seguridad y performance. Estas pruebas están diseñadas para comprobar que la funcionalidad del sistema sea lo que se requirió.

## Principio de las pruebas

Principio 1: Las pruebas demuestran la presencia de defectos, no su ausencia.

Principio 2: Las pruebas exhaustivas son imposibles.

Principio 3: Las pruebas tempranas ahorran tiempo, esfuerzo y costo.

Principio 4: Los defectos se agrupan.

Principio 5: Cuidado con la paradoja del pesticida.

Principio 6: La prueba depende del contexto.

Principio 7: Falacia de la ausencia de errores. [8]

## Ciclo de vida del software.

Es el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y retiro del sistema. Se definen las distintas fases intermedias que se requieren para validar el desarrollo de un software, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, se asegura de que los métodos utilizados son apropiados. [9]

## V. MÉTODO

Antes de iniciar la definición del plan de calidad a aplicar al módulo de Astro Millonario, se realizó un estudio sobre el procedimiento que actualmente se lleva a cabo en el área de SQA de la empresa, durante el cual se identificaron las actividades y artefactos que componen el proceso de calidad en la actualidad.

Durante dicho estudio se encontró que existe una serie de documentos que el analista debe usar como guía para asegurar que el proceso de SQA se realice adecuadamente según los estándares definidos por la empresa.

Dichos documentos están agrupados en 3 categorías, las cuales se detallan a continuación:

1. **Instructivos:** En estos documentos se define la forma en la cual el analista debe realizar un proceso definido. Actualmente se cuenta con los siguientes

instructivos:

- **Diseño Requerimientos Prueba:** Este instructivo indica al analista los aspectos que debe tener en cuenta para construir un buen diseño de casos de prueba que permitan dar el mayor cubrimiento a las pruebas que se deben realizar para certificar el correcto funcionamiento de un desarrollo.
- **Empaquetar Software:** Este documento describe paso a paso las actividades que el analista debe realizar para generar los empaquetados o entregas que se realizan al cliente final. Se describe en detalle la documentación que se debe generar y la forma correcta de organizar los archivos y carpetas que componen el empaquetado.

2. **Templates o plantillas:** Estos documentos son formatos que el analista debe diligenciar con los resultados de las actividades que realiza durante el proceso de SQA. Las plantillas que se utilizan actualmente se explican a continuación:

- **Matriz de pruebas:** En este documento el analista debe identificar cada uno de los casos de prueba que se van a ejecutar durante el proceso, en este, debe identificar el componente a probar, el proceso al cual corresponde, el caso de uso, los resultados esperados y el tipo de prueba que se aplicará para cada caso.
- **Informe de pruebas:** El informe de pruebas resume los hallazgos de todo el proceso de pruebas realizado. Se deben listar los requerimientos probados y los defectos o no conformidades encontradas. Además, contiene una sección de indicadores y gráficos porcentuales por cada tipo de no conformidad definida en el proceso (Bloqueantes, Funcionales, Presentación).
- **Ficha Técnica:** En la ficha técnica de producto, el analista debe registrar los requerimientos técnicos del producto que se está entregando al cliente, tales como sistemas operativos compatibles, navegadores, versiones compatibles de java etc.
- **Notas de Release:** En este documento el analista debe registrar los cambios, funcionalidades, y correcciones realizadas a los componentes de software entregados al cliente.
- **Pruebas de Carga:** En este formato, el analista debe registrar las evidencias de los resultados obtenidos durante las pruebas de carga y especificar el total de las muestras, total de usuarios que se utilizaron durante la prueba,

cuantos ciclos se realizaron y cuantas transacciones fueron efectivas.

- **Plantilla de Entrega:** El objetivo de esta plantilla es registrar cada uno de los requerimientos que se liberan en un empaquetado, en él se especifica la fecha de entrega, el analista que realiza las pruebas, el cliente y una breve descripción de la solución entregada.

3. **Guías y procedimientos:** Estos documentos proporcionan un paso a paso al analista sobre las actividades que debe realizar durante el proceso de pruebas. Actualmente se cuenta con un documento, el cual se explica a continuación:

- **Pruebas de Software:** Este documento describe en detalle cada una de las actividades que se deben realizar para completar el proceso de software, el rol que debe realizar la actividad y los documentos que se deben generar durante cada actividad.

Para la aplicación del plan que se va a plantear se van a utilizar los documentos que se utilizan actualmente.

### Descripción del proyecto

En este proyecto se busca que el producto de apuestas azar Astro Millonario, sea migrado a la herramienta configurador de productos, el cual es una herramienta que permite que las empresas de apuestas puedan crear sus nuevos productos sin necesidad de enviar una solicitud a la empresa desarrolladora. Teniendo en cuenta que para futuros desarrollos se va a utilizar la herramienta de configurador de productos, es necesario que los productos ya existentes como Astro Millonario sean migrados a la nueva plataforma y que conserven su funcionalidad.

Las pruebas de software son ejecutadas al producto desarrollado en la empresa, previo a la instalación del producto en el ambiente del cliente o la entrega de cualquier artefacto que se obtenga como parte del desarrollo del mismo. Constituyen una etapa dentro del ciclo de vida del software y se ejecutan en paralelo al proceso de desarrollo; se aplican a todos los artefactos que constituyen entregables al cliente.

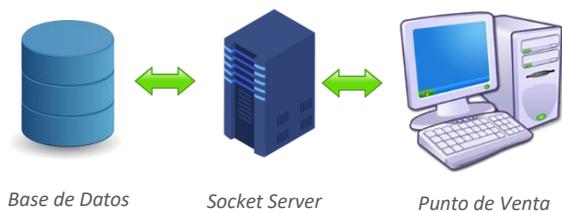
Se realizan en este nivel pruebas dinámicas y estáticas según el tipo de artefacto.

### Recursos que intervienen en la prueba

#### *Entorno de pruebas:*

Construir el entorno de pruebas apropiado garantiza la

correcta ejecución de las mismas, de ahí que en esta prueba se debe simular el entorno real donde será desplegada la aplicación. En la siguiente figura, se muestra la arquitectura del ambiente productivo, la cual será usada como modelo para la ejecución del plan de pruebas (ver figura 1).



**Figura 1:** Entorno de pruebas

La figura anterior se muestra la iteración entre los componentes involucrados en el proceso venta, durante este proceso, desde el punto de venta se envía una trama al socket server, el cual es un componente en java que procesa todas las tramas recibidas para convertirlas en peticiones que son enviadas hacia la base de datos. Una vez la base de datos procesa la información, crea una respuesta y la envía al socket server el cual la procesa y la convierte en una trama de respuesta que es enviada al punto de venta. Finalmente, el punto de venta recibe la trama de respuesta muestra el mensaje de venta exitosa al vendedor.

<b>Responsable del plan de recursos:</b>	<b>Analista de Calidad</b>		
<b>Objetivos:</b>	Determinar todo aquello que es necesario para ejecutar y finalizar satisfactoriamente las pruebas de software		
<b>Observaciones:</b>	Se deben agregar cuantos recursos se necesiten para la ejecución de las pruebas, en caso de no aparecer en la lista se debe ingresar de acuerdo a la clasificación dada: Recursos Lógicos, Recursos Físicos y Recursos Humanos.		
<b>Recursos lógicos</b>			
No. Ítem	Recurso	Marque	
1	Disponibilidad de conexión a la base de datos de calidad.	X	
2	Disponibilidad de permisos para acceso y consulta a la base de datos de calidad.	X	
3	Acceso permanente a los servidores donde se encuentra desplegada la aplicación de cliente administrativo.	X	
<b>Recursos Físicos</b>			
No.	Cantidad	Recuso	Marque

Ítem			
1	1	Computador y base de datos de calidad	X
<b>Recursos Humanos</b>			
No. Ítem	Cantidad	Recuso	Marque
1	1	Ingeniero de calidad.	X
2	1	Ingeniero de desarrollo.	X
3	1	Ingeniero de implementación	X

**Tabla 1.** Plan de recursos.

### Cronograma de la prueba

A continuación, se define el cronograma planificado en el que se recogen las fechas (Inicio y fin) para llevar a cabo cada actividad del proceso de prueba. El cronograma real será registrado en el documento “Cronograma de la prueba”.

<b>Calidad Pruebas Funcionales</b>					
<b>Total Calidad Pruebas Funcionales: En horas</b>					
No.	Descripción	Duración Tiempo Estimado	Duración Tiempo Real	Inicio	Terminado
1	Total horas de Ticket	29 Horas	23,85 Horas	21/12/19	26/12/19
	Capacitación a Calidad	2 Horas	2 Horas	21/12/19	21/12/19
	Preparación Ambiente de Pruebas	4 Horas	6 Horas	21/12/19	23/12/19
	Ejecución Pruebas	16 Horas	10,35 Horas	23/12/19	24/12/19
	Lectura Documentación	3 Horas	2,41 Horas	24/12/19	26/12/19
	Generación Documentación	4 Horas	3 Horas	26/12/19	26/12/19
<b>Calidad Pruebas de Sistema</b>					
<b>Total Calidad Pruebas de Sistema: En horas</b>					
No.	Descripción	Duración Tiempo Estimado	Duración Tiempo Real	Inicio	Terminado
1	Ejecución de Pruebas de operación de negocio y funcionalidades del sistema	3 horas	3 horas	26/12/19	26/12/19

**Tabla 2.** Cronograma.

### A. Mediciones

Para tener en cuenta las mediciones, se obtiene información del documento de Estimacion\_actividades\_SQA el cual muestra los tiempos estimados y el tiempo real ejecutado por los analistas de desarrollo y SQA, al igual que los colaboradores reportan

las actividades realizadas en el día en la herramienta FengOffice que permiten ver las tareas que se están ejecutando y el tiempo que llevan de ejecutado. También las no conformidades reportadas en la herramienta Mantis nos permite llevar mediciones sobre el costo que lleva el reproceso de un proyecto al corregir las no conformidades por parte de desarrollo y las iteraciones que el analista de SQA debe realizar para verificar el producto.

## VI. RESULTADOS Y DISCUSIÓN

Como resultado del trabajo realizado se obtuvo la documentación de calidad diligenciada, correspondiente al desarrollo de la migración del producto de Astro al configurador de productos, empezando por la elaboración de la matriz de pruebas donde se indica lo que se revisó en la documentación del desarrollador y su correcto diligenciamiento, En la columna *id. Caso de uso* se indica el caso de uso relacionado con la prueba que se va a realizar, en caso que la prueba sea genérica y no esté relacionada directamente con un caso de uso, se debe colocar 999, que es el caso de uso definido para estas pruebas.

En la siguiente tabla también se especifican los casos de pruebas para la validación del producto Astro migrado a la nueva arquitectura, como se había indicado antes, el principal objetivo es validar que la venta del producto no se vea afectada durante la migración (ver Tabla 3).

MATRIZ DE PRUEBAS	
Proyecto	MIGRACION DE ASTRO AL CONFIGURADOR DE PRODUCTOS
Iteración	1
Documentación	
Proceso	Revisión de documentación y componentes
Componente a probar	Componentes de software entregados
Id. Caso de prueba	CP 1
Id. Caso de uso	CU 999
Caso de prueba	Se requiere revisar los componentes entregados.
Resultado esperado	Revisar: - Todos los componentes se encuentren versionados - Estén versionados con el formato correcto (Versión:) - Estén en formato SQL
Tipo de Prueba	Estáticas
Observaciones	OK
Id. Caso de uso	CU 999

Id. Caso de prueba	CP 2
Id. Caso de uso	CU 999
Caso de prueba	Se requiere revisar la configuración del ambiente con los componentes entregados.
Resultado esperado	- Los paquetes / scripts de base de datos no generen errores al ejecutarlos. - La actualización de los componentes de software

	funcionen - El / Los proyectos inicien correctamente.
Tipo de Prueba	Estáticas
Observaciones	OK

Componente a probar	Documentación
Id. Caso de prueba	CP 3
Id. Caso de uso	CU 999
Caso de prueba	Revisión de documentación entregada por desarrollador:  -Historia de usuario -Revisiones: código y diseño -Especificación técnica -Pruebas de unidad
Resultado esperado	Revisar: - En la documentación se encuentre lo solicitado por el cliente - La información de los documentos es clara, concisa y detallada. - Formato correcto. - La historia de usuario debe estar aprobada.
Tipo de Prueba	Estáticas
Observaciones	OK

Aplicativo de venta de chance ACTUAL	
Proceso	Venta de producto Astro
Componente a probar	Aplicativo de chance
Id. Caso de prueba	CP 4
Id. Caso de uso	CU 999
Caso de prueba	Se requiere realizar la venta del producto Astro millonario en la arquitectura actual.
Resultado esperado	Verificar que: - El sistema permita realizar la venta del producto Astro e imprima el comprobante de venta
Tipo de Prueba	Funcionales positivas
Observaciones	Se verifica en la forma "Formularios" que el formulario haya sido vendido.
Aplicativo de venta de chance SOLUCION	
Proceso	Venta de producto Astro
Componente a probar	Aplicativo de chance
Id. Caso de prueba	CP 5
Id. Caso de uso	CU 999
Caso de prueba	Se requiere realizar la venta del producto Astro millonario migrado al configurador de productos.
Resultado esperado	Verificar que: - El sistema permita realizar la venta del producto Astro e imprima el comprobante de venta
Tipo de Prueba	Funcionales positivas
Observaciones	Se verifica en la forma "Formularios" que el formulario haya sido vendido.

**Tabla 3.** Matriz de pruebas.

## VII. CONCLUSIONES

Llevando un plan de calidad adecuado, se minimizan los riesgos de fallos en el software cuando se entrega al cliente,

además de entregar un producto de calidad, también se entrega un producto a tiempo cumpliendo con los requerimientos solicitados inicialmente.

El planteamiento del cronograma de actividades dentro del plan de calidad, permitió hacer seguimiento a los avances de las pruebas y conocer el estado de la ejecución del plan durante toda su duración. Además, permitió tener una estimación más real del tiempo requerido para completar todo el proceso de pruebas.

La interacción constante entre el desarrollador y el analista de QA durante la aplicación del plan de calidad, permite disminuir los tiempos de entrega del producto final, lo cual se puede evidenciar en el cronograma (tabla 2) donde el tiempo estimado era de 29 horas y el tiempo ejecutado fue de 23,85 horas, debido a que la comunicación continua durante el desarrollo facilita la corrección de las fallas encontradas.

## VIII. RECOMENDACIONES

Se espera poder continuar con la aplicación del plan de calidad planteado, en futuros proyectos de la empresa proveedora de tecnología, con el fin de ajustar paulatinamente dicho plan a las características de la empresa y de los productos desarrollados, de forma que sea posible mejorar continuamente la calidad de los productos entregados al cliente final.

## IX. BIBLIOGRAFIA

- [1] R. S. Pressman, *Ingeniería del software: un enfoque práctico*. 2013.
- [2] Universidad Pedagógica y Tecnológica de Colombia, Tunja - Colombia, M. Callejas-Cuervo, A. C. Alarcón-Aldana, Universidad Pedagógica y Tecnológica de Colombia, Tunja - Colombia, A. M. Álvarez-Carreño, y Universidad Pedagógica y Tecnológica de Colombia, Tunja - Colombia, «Modelos de calidad del software, un estado del arte», *ENTRAMADO*, vol. 13, n.º 1, pp. 236-250, 2017, doi: 10.18041/entramado.2017v13n1.25125.
- [3] A. I. Álvarez y M. López, «Elaboración de planes de la calidad en proyectos de software», p. 12.
- [4] Cindy Campos Chiu, «Las pruebas en el desarrollo de software».
- [5] J. M. S. Peño, «Pruebas de Software. Fundamentos y Técnicas.», p. 130.
- [6] Instituto colombiano de normas técnicas y certificación ICONTEC, «Norma técnica colombiana NTC ISO 9000». 15-oct-2015.
- [7] ¿Qué es un QA?», ¿Qué es un QA? [En línea]. Disponible en: <https://django.mx/blog/entrada/que-es-un-qa/>. [Accedido: 05-ene-2020].
- [8] Y. Trujillo Casañola y A. Marín Díaz, «Calidad en aplicaciones Web y móviles», La Habana - Cuba UCI, oct-2019.
- [9] Ciclo de vida del software», Ciclo de vida del software. [En línea]. Disponible en: [https://www.ecured.cu/Ciclo\\_de\\_vida\\_del\\_software](https://www.ecured.cu/Ciclo_de_vida_del_software). [Accedido: 05-ene-2020].