



<Gobernanza de Control de Calidad >

Versión < V2.5 >



Contenido

1.	Propósito del documento.....	3
2.	Glosario	3
3.	Procedimiento de Control de Calidad	5
4.	Casos de Pruebas.....	8
4.1.	Lineamientos para implementación de Casos de Prueba	9
	Contexto General de los Procesos de Prueba	9
4.2.	Proceso de Diseño e Implementación de Pruebas.....	10
4.3.	Especificación de Casos de Prueba	10
4.4.	Concepto de Caso de Prueba	10
4.5.	Descripción de un Caso de Prueba	10
4.6.	Nomenclatura para título de casos de prueba.....	11
4.6.1.	Objetivos	11
4.6.2.	Pautas básicas de nomenclatura.....	12
5.	Reportes de incidencias	13
5.1.	Lineamientos para informe de Incidencias en las Pruebas.....	13
5.1.1.	Contexto General de los Procesos de Prueba	13
5.2.	Proceso de Informe de Incidente de Pruebas	13
5.2.1.	Antes de crear un Incidente de Pruebas	13
5.2.2.	Luego de crear un Incidente de Pruebas (Seguimiento)	14
5.3.	Informe de Incidente.....	15
5.4.	Descripción por cada Incidente.....	15
5.5.	Nomenclatura para Título (o resumen) de incidentes	16
5.5.1.	Consideraciones	16
5.5.2.	Pautas básicas de nomenclatura.....	16
6.	Prueba Estática.....	18
7.	Checklist General de Proyectos.....	20
8.	Ambientes de Control de Calidad.....	25
8.1.	Definición	25



1. Propósito del documento

Establecer el procedimiento de aseguramiento de calidad, totalmente integrado al ciclo de vida de las pruebas a fin de garantizar la integridad y el aseguramiento de la calidad de los proyectos, actividades, etc.

2. Glosario

- **Base de Prueba:** Todos los documentos de donde los requisitos de un componente o sistema pueden ser inferidos. La documentación en la que se basan los casos de prueba. Si un documento puede ser modificado sólo por medio de un procedimiento de cambio formal, entonces la base de las pruebas se denomina base de prueba congelada.
- **Plan de Prueba:** Describe las actividades de prueba para Proyectos de Desarrollo, mantenimiento e infraestructura. La planificación depende de la política y la estrategia de prueba de la organización, los ciclos de vida de desarrollo y métodos utilizados, el alcance de la prueba, objetivos, riesgos, restricciones, criticidad, capacidad de ser probado y la disponibilidad de los recursos. La planificación de la prueba es una actividad que se realiza de forma continua a lo largo de todo el ciclo de vida del producto.
- **Estrategia de Prueba:** Descripción de alto nivel de los niveles de prueba a ser llevados a cabo y las pruebas dentro de estos niveles para una organización o programa (en uno o más proyectos).
- **Elementos de Prueba:** Elemento individual a ser probado. Normalmente hay un objeto de prueba y muchos elementos de prueba.
- **Técnicas de Diseño de Prueba:** Ayuda a identificar las condiciones de prueba, los casos de prueba y los datos de prueba.
- **Cobertura:** Grado, expresado como un porcentaje, en el que un elemento de cobertura especificado ha sido practicado por un juego de pruebas.
- **Incidente:** Cualquier evento que genere una interrupción no planificada o la reducción de calidad de algún servicio de TI que generalmente requiere investigación.
- **Stakeholders:** palabra del inglés que significa “*parte interesada*”, se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa.
- **Riesgos del producto:** Riesgo asociado al software o al sistema. Posibilidad de que el software o el sistema no satisfagan las expectativas del usuario final/cliente se conoce como riesgo del producto.
- **Riesgos de proyecto:** Riesgos asociados a la actividad de pruebas que pueden poner en peligro el ciclo del proyecto de pruebas.
- **Pruebas de confianza: (Pruebas de Humo):** Subconjunto de todos los casos de prueba definidos/planificados que cubren la funcionalidad principal de un componente o sistema, para comprobar que las funciones más cruciales de un programa funcionan antes de realizar pruebas más exhaustivas.
- **Política de Pruebas:** Describe los objetivos y metas de la organización para las pruebas.
- **Estrategia de Pruebas:** Describe los métodos de pruebas generales e independientes del proyecto de la organización.
- **Incidente:** Cualquier evento que genere una interrupción no planificada o la reducción de calidad de algún servicio de TI que generalmente requiere investigación.



- **Anomalía:** Cualquier condición que se desvíe de las expectativas basadas en las especificaciones de requisitos, documentos de diseño, documentos de usuario, estándares, etc., o de la percepción o experiencia de alguien. Las anomalías pueden ser encontradas durante, aunque no se limitan sólo a, revisiones, proceso de pruebas, análisis, compilación, o uso de productos de software o documentación aplicable.
- **Error:** Acción humana que produce un resultado incorrecto.
- **Defecto:** Imperfección en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas. Ejemplo: una sentencia o una definición de datos incorrectas. Si se localiza un defecto durante una ejecución puede causar un fallo en el componente o sistema.
- **Fallo:** Desviación del componente o del sistema respecto de prestación, servicio o resultado esperado.
- **Elemento de acción:** tareas que se derivan de los requerimientos, que luego se asignan a un individuo o equipo y para ser completados.

3. Referencias

- ISO/IEC/IEEE 29119-3:2013(E).
- ISTQB® Foundation Level, Versión 2018 V3.1.
- Manual de Gobierno y Control de Tecnologías de Información (MGCTI), Resolución SB.SG N 00124/2017.



4. Procedimiento de Control de Calidad

Delineamientos para las tareas que se llevarán a cabo en el departamento de Control de Calidad.

(*) Los responsables de cada actividad serán definidos en la Estrategia "XXX"

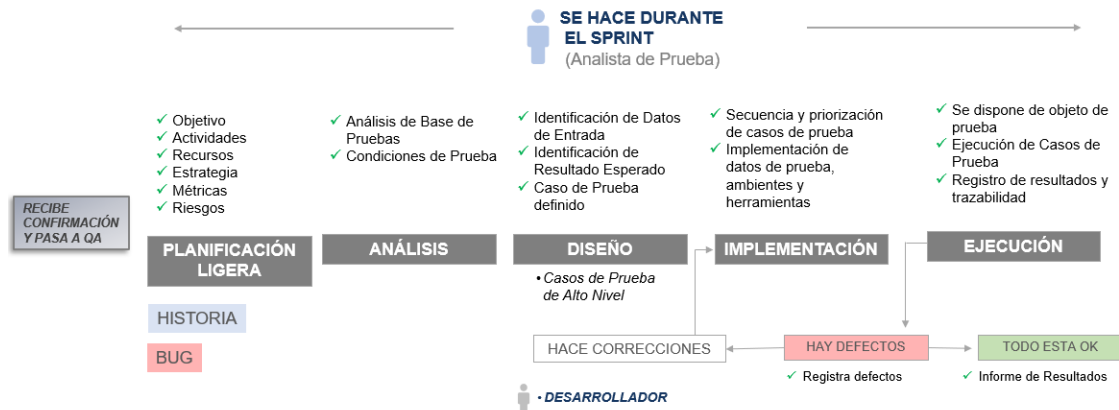
ACTIVIDAD	TAREAS DE PRUEBA SUGERIDAS	ENTRADA	SALIDA
Planificación	<ul style="list-style-type: none">Definir objetivos de la prueba.Evaluar de Riesgos del producto.Establecer alcance de las pruebas: elementos que serán probados y cuáles no, criterios de entrada, salida, suspensión, reanudación.Establecer estimaciones de las pruebas: Desglose de tareas, ciclo de vida de las pruebas, hitos.Desarrollar un plan de pruebas: calendario de pruebas, recursos/habilidades necesarias, participación de stakeholders, riesgos de proyecto de prueba.	<ul style="list-style-type: none">Política de PruebasEstrategia de PruebasIndicadores de Desempeño	<ul style="list-style-type: none">Plan de Pruebas (inicial)
Análisis	<ul style="list-style-type: none">Analizar la Base de Pruebas: requisitos que especifiquen el comportamiento funcional y no funcional deseado del componente o sistema.Identificar y priorizar las condiciones de prueba.Identificar posibles defectos	<ul style="list-style-type: none">Plan de Pruebas (inicial)Especificación funcional y de diseño	<ul style="list-style-type: none">Plan de pruebas (actualizado)Condiciones de Prueba
Diseño	<ul style="list-style-type: none">Identificar, Priorizar y Documentar los Casos de Prueba.Identificar los Datos de Prueba específicosMantener la trazabilidad hacia los requisitos que facilite la monitorización de la cobertura	<ul style="list-style-type: none">Plan de pruebas (actualizado)Condiciones de Prueba	<ul style="list-style-type: none">Casos de Prueba



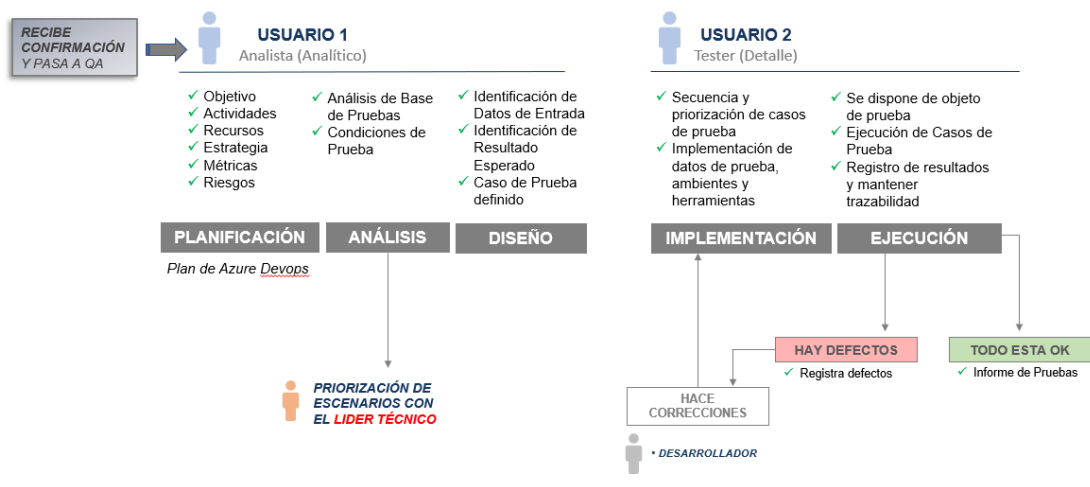
Implementación	<ul style="list-style-type: none">• Desarrollar procedimientos de la prueba: se combinan casos de prueba en un orden particular.• Crear datos de prueba específicos: se almacenan y gestionan en el entorno de pruebas• (prueba de humo) de Especificar las pruebas de confianza finir una lista de verificaciones a ejecutar usando criterios de entrada del plan• Desarrollar cronograma de ejecución• Configuración de entorno, datos de prueba y herramientas independientes a producción.	<ul style="list-style-type: none">• Casos de Prueba	<ul style="list-style-type: none">• Procedimiento de prueba• Entorno y datos de pruebas
Ejecución	<ul style="list-style-type: none">• Realizar pruebas de confianza (humo)• Ejecutar los casos de prueba: Comparar los resultados esperados vs ejecutados, realizar pruebas de confirmación y de regresión (si aplica)• Reportar incidencias de pruebas• Registrar las pruebas	<ul style="list-style-type: none">• Procedimiento de prueba• Entorno y datos de prueba• Diseño de Estándares para Informe de Incidentes de Pruebas• Sistema para probar	<ul style="list-style-type: none">• Registro de Pruebas• Informe de Incidentes de Pruebas



ESCENARIO CÉLULA



ESCENARIO FÁBRICA





4.1. Pruebas Exploratorias

En caso de que hay presión significativa con respecto al tiempo de finalización de las pruebas, se utiliza un tipo de prueba no exhaustiva de forma superficial llamada **prueba exploratoria**. Mediante un contrato (correo con dirigido al responsable técnico/interesado) que especifica que se realizará de forma dinámica pruebas informales (no predefinidas) durante la ejecución.

Al tener naturaleza de ser superficial el objetivo de este tipo de pruebas es reducir algún riesgo de bugs críticos en el circuito básico establecido según la definición de la documentación y/o la historia de usuario.



4.2. Smoke Test

Es una revisión rápida de un producto de software para comprobar que funciona y no tiene defectos evidentes que interrumpan la operación básica del mismo. Son pruebas que pretenden hacer una evaluación inicial de la calidad de un producto de software previo a una recepción formal para el equipo de pruebas (quien ejecutará una batería completa de comprobaciones). En pocas palabras, esta prueba es un chequeo de salud normal de una aplicación antes de llevarla a prueba en profundidad.

El Smoke test es lo primero que se realiza al recibir el desarrollo en control de Calidad. Se realiza previamente a la etapa de planificación del Proceso de Control de Calidad.

Además de las revisiones hechas, el analista/tester encargado debe especificar que pruebas se van a realizar como validador de la salud de objeto de prueba.

Checklist de Smoke Test

Concepto	Cumple
Pruebas estáticas exitosa (Ver punto 7).	SI/NO
Prueba exitosa definida por QA	SI/NO

Si el Checklist del Smoke Test tiene 100% de aprobación se considera que el desarrollo está apto para ser probados por el equipo de pruebas. De caso contrario, el proyecto vuelve al equipo de desarrollo para hacer los arreglos pertinentes.

5. Casos de Pruebas

Reutilización de casos de pruebas de alto nivel en proyectos futuros, con perspectivas a la Automatización.

Definir los casos de prueba derivados de los elementos de cobertura de la prueba. El número de casos de prueba derivados de los elementos de cobertura de la prueba dependerá del criterio de cobertura de la prueba definido en el plan de pruebas.

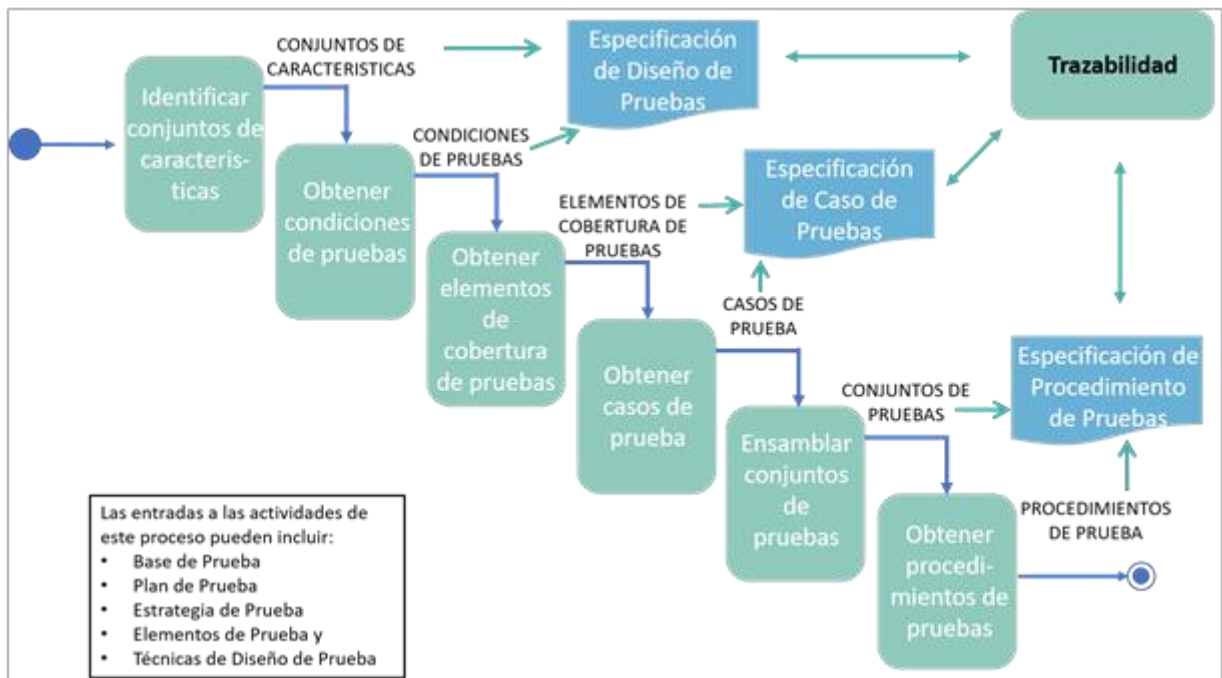
5.1. Lineamientos para implementación de Casos de Prueba

Contexto General de los Procesos de Prueba





5.2. Proceso de Diseño e Implementación de Pruebas



5.3. Especificación de Casos de Prueba

La Especificación de casos de prueba describe casos de prueba y los elementos de cobertura de prueba que el caso de prueba ejercita. Los elementos de cobertura de prueba (ej. *Un valor limite*) son derivados de las condiciones de prueba usando una técnica de diseño de prueba.

5.4. Concepto de Caso de Prueba

Conjunto de:

- Precondición del caso de prueba,
- Entradas (incluidas acciones, cuando corresponda) y
- Resultados esperados

desarrollados para impulsar la ejecución de un elemento de prueba a fin de cumplir con los objetivos de la prueba, incluida:

- la implementación correcta,
- la identificación de errores,
- la verificación de la calidad y
- otra información valiosa.

5.5. Descripción de un Caso de Prueba

Elemento	Descripción
Título	Ver Anexo A nomenclatura para título de caso de prueba



Identificador Único	Proporcionado para que cada caso de prueba pueda ser identifica de manera única (Ejemplo: TC001, TC002). Las herramientas de gestión de prueba generan identificadores automáticamente.
Fecha	Tiempo (dd/mm/aaaa) en el que se lleva a cabo el caso de prueba.
Objetivo	Brevemente describe el foco del caso de prueba. (Podría ser el mismo título)
Responsable	Persona encargada de llevar a cabo la prueba
Prioridad	Proporcionado para que cada caso de prueba pueda ser ejercitado delante de otros (con baja prioridad). Por ejemplo, si queremos que los Casos de Prueba asociados con alta frecuencia de transacciones sean ejecutados antes de otras transacciones.
Dependencia	Identificados del Caso de Prueba del cual depende.
Trazabilidad	Proporcionado para poder ver cual elemento de cobertura de prueba está ejercitando este caso de prueba (podría referirse a la Condición de Prueba o al Requisito/Historia de Usuario)
Pre - condiciones	Especifica el estado requerido del entorno de prueba (incluyendo la configuración de datos de prueba) para que la prueba se ejecute. Esto es frecuentemente especificado para un grupo de casos de prueba.
ESCENARIO	
ítem	Número de caso de prueba.
Descripción	Breve resumen del caso de prueba
Entradas	Describe los datos de entrada (puede incluir las acciones que se necesitan realizar para ejecutar el caso de prueba. Ejemplo: "Ingresar el precio de venta de \$50", "Presionar el boto Calculo de Interés"
Resultado Esperado	Describe el comportamiento esperado de las salidas resultantes de ejecutar el caso de prueba. Por ejemplo, podría ser que un valor especifico deba ser escrito en un archivo o una respuesta deba ser recibida dentro de un cierto tiempo.
Resultado actual	(Opcional) se usa más adelante para registrar el resultado de ejecutar el caso de prueba
Estado de Caso de Prueba	Indica si es caso de prueba culminó de manera exitosa o fallida.

5.6. Nomenclatura para título de casos de prueba

5.6.1. Objetivos

Es importante elegir buenos títulos para los casos de prueba por varias razones:

- Cuando usted u otros Testers del equipo estén mirando la lista de casos de prueba, debería ser fácil averiguar qué Caso de prueba es el que necesita, y si hay algunos Casos de prueba que son similares, entonces debería quedar claro cuál uno es el adecuado para la tarea en cuestión.



- Cuando utilice busque o filtre los casos de prueba, saber cómo se nombran sus casos de prueba le ayudará a encontrar el caso de prueba correcto. De lo contrario, puede terminar pensando que no existe un caso de prueba para el flujo de trabajo actual e introducir redundancias al crear uno nuevo.
- Al leer un caso de prueba más grande o un informe de resultado de prueba, debería ser fácil seguir el flujo y los pasos de la prueba sin haberlo escrito usted mismo. Cuanto más fáciles de comprender sean los informes de resultados de las pruebas, menos tiempo dedicará a analizar las fallas de las pruebas cuando ocurran.

5.6.2. Pautas básicas de nomenclatura

La regla empírica más sencilla para nombrar los casos de prueba es describir lo que hace el caso de prueba.

Algunos estilos pueden ser:

5.6.2.1. Orientada a la conformidad/verificación

Ejemplos:

SP_001_Verificar que el usuario se puede loguear
SP_002_Verificar que el usuario puede buscar cualquier ticket
SP_003_Verificar que el usuario puede comprar un ticket

Otros ejemplos:

Verificar que el nuevo Cliente fue creado
Verificar que el cliente puede ser editado
Verificar que la nueva cuenta se ha agregado
Verificar que la cuenta se ha eliminado

Usando prefijos de agrupamiento o modulo

Eliminar Orden - Eliminar una orden futura
Eliminar Orden - Eliminar una orden pasada
Editar Orden - Editar una Orden existente con todos los datos validos
Editar Orden - Editar una Orden con datos inválidos
Reporte Clientes - Guardar lista con nombre de archivo distinto
Reporte Clientes - Guardar lista con el nombre de archivo por defecto

5.6.2.2. Orientada al comportamiento

Describe lo que se espera que realice el producto o sistema. Ejemplos:

Genera confirmación - Email
Genera confirmación - archivo PDF
Tres intentos de logueo con datos incorrectos biométrico
Tres intentos de logueo con datos incorrectos PIN
Convierte una moneda en otra
Lanza el modo de accesibilidad

Otros ejemplos:

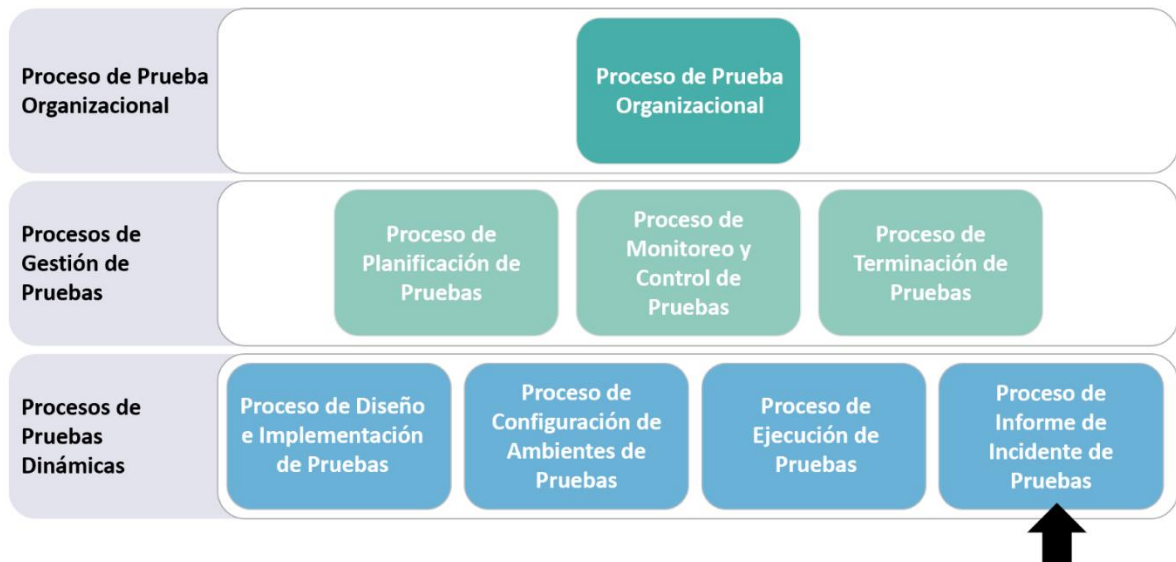
Busca por nombre
Busca por apellido
Busca por nombre parcial
Busca por código paciente
Actualiza el perfil del Paciente



6. Reportes de incidencias

6.1. Lineamientos para informe de Incidencias en las Pruebas.

6.1.1. Contexto General de los Procesos de Prueba



Informe de Incidente (de Pruebas): documentación de la ocurrencia, la naturaleza y el estado de un incidente.

Nota 1: Los informes de incidentes también se conocen como “informes de anomalías”, “informes de errores”, “informes de defectos”, “informes de fallos” e “informes de problemas”, entre otros términos.

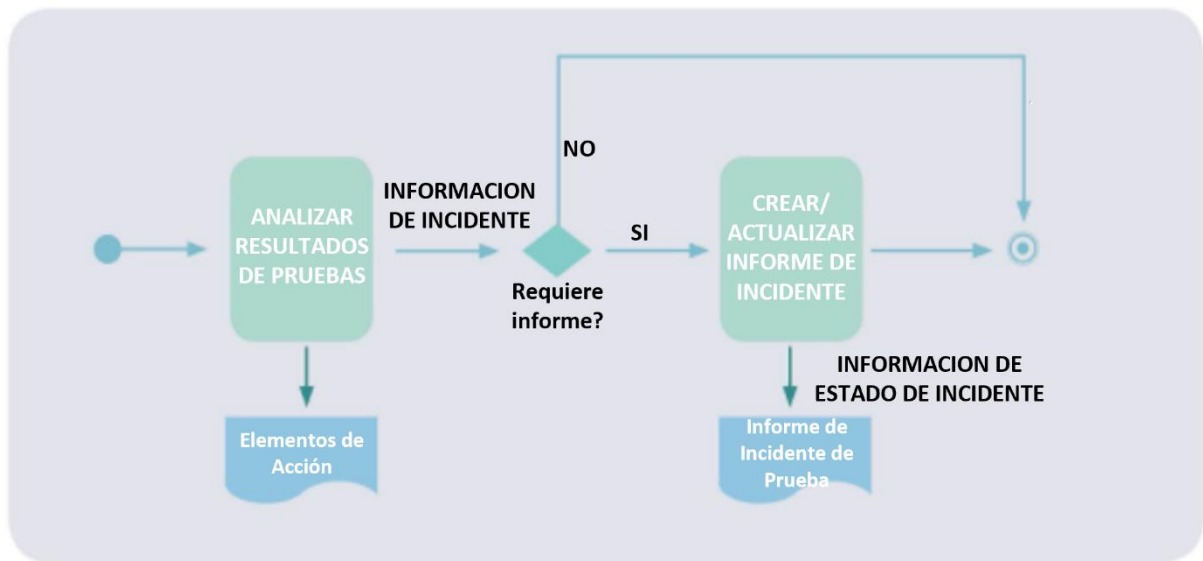
Nota 2: Un informe es un documento escrito que menciona los resultados obtenidos de algo que se hizo con comentarios de los resultados obtenidos, a diferencia de un reporte el cual es un documento escrito que menciona únicamente los resultados obtenidos de algo que se hizo sin hacer comentario alguno de estos resultados.

6.2. Proceso de Informe de Incidente de Pruebas

6.2.1. Antes de crear un Incidente de Pruebas

Este proceso implica verificar los resultados de una prueba fallida para determinar si la falla representa un problema que debe informarse formalmente.

En algunos casos, la falla puede identificarse como un incidente con la prueba, o podría decidirse que el incidente no es lo suficientemente importante como para justificar la presentación de un informe de este. En ambos casos, es posible crear y asignar un **elemento de acción** para solucionar el problema de manera informal. Ejemplos: que el caso de prueba fue incorrecto debido a un malentendido o a que los datos de prueba deben actualizarse.



6.2.2. Luego de crear un Incidente de Pruebas (Seguimiento)

Son diferentes los estados por los cuales transita un Incidente luego de ser creado.

En herramientas como Azure Devops dependerá de la plantilla de proceso utilizada. La siguiente imagen ilustra el ciclo de vida del flujo de trabajo predeterminado de un Incidente (bug) para los procesos Agile.





6.3. Informe de Incidente

Los informes de incidentes (también conocidos como informes de errores o informes de defectos) se escriben para describir los problemas encontrados en las pruebas que deben registrarse.

En la mayoría de las organizaciones, se utilizan herramientas para registrar informes de incidentes; estas herramientas están disponibles tanto comercialmente como de código abierto y algunas personas incluso usan una hoja de cálculo. Si se está utilizando una herramienta de gestión de pruebas en el proyecto, los incidentes se pueden registrar dentro de esta herramienta, si tiene esa capacidad.

Un informe de incidente en nuestro contexto documenta un incidente reconocido durante la prueba.

6.4. Descripción por cada Incidente

Elemento	Descripción
Identificador único	Correlativo generado por la herramienta
Fecha/Hora	Fecha/Hora en la que se encontró el Incidente
Originador	Quien identifico el incidente
Titulo (o Resumen de una línea)	Resumen en una línea del Incidente (ver Apéndice)
Descripción	Información las descriptiva del incidente
Pasos para reproducir	Esto incluye las acciones realizadas para encontrar o reproducir el error y el comportamiento esperado
Información del Sistema (Entorno)	Configuración del entorno (S.O. móvil, navegador, etc.)
Severidad	<p>Una calificación subjetiva del impacto de un error en el proyecto o sistema de software. Ejemplo: si hacer clic en un enlace remoto (un evento poco común) hace que una aplicación o página web se bloquee (una experiencia grave del cliente), puede especificar Severidad = 2 (Alto) y Prioridad = 3. Los valores permitidos y las pautas sugeridas son:</p> <p>1. Crítico: Debe solucionarse. Un defecto que causa la terminación de uno o más componentes del sistema o del sistema completo, o causa una gran corrupción de datos. Y no existen métodos alternativos aceptables para lograr los resultados requeridos.</p> <p>2. Alto: considere arreglarlo. Un defecto que causa la terminación de uno o más componentes del sistema o del sistema completo, o causa una gran corrupción de datos. Sin embargo, existe un método alternativo aceptable para lograr los resultados requeridos.</p> <p>3. Medio: (predeterminado) Un defecto que hace que el sistema produzca resultados incorrectos, incompletos o inconsistentes.</p>



	4. Bajo: Un defecto menor o cosmético que tiene soluciones aceptables para lograr los resultados requeridos.
Prioridad	Una calificación subjetiva del error en relación con los requisitos del cliente o de la empresa. Prioridad indica el orden en el que se deben corregir los defectos del código. Puede especificar los siguientes valores: <ul style="list-style-type: none">• El producto no puede enviarse sin la resolución exitosa del incidente y debe abordarse lo antes posible.• El producto no se puede enviar sin la resolución exitosa del incidente, pero no es necesario abordarlo de inmediato.• La resolución del incidente es opcional en función de los recursos, el tiempo y el riesgo.
Estado	Nuevo, Resuelto, Activo, Cerrado
Documento Adjunto	Observaciones y evidencias

6.5. Nomenclatura para Título (o resumen) de incidentes

6.5.1. Consideraciones

- El Título (o resumen) es un resumen en una línea del incidente (o problema)
- Las partes interesadas lo usarán al revisar la lista de incidentes que no se han corregido.
- Muchas partes interesadas dedican tiempo adicional solo a los incidentes con resúmenes "interesantes"

6.5.2. Pautas básicas de nomenclatura

El Título (o resumen) ideal brinda al lector suficiente información para ayudarlo a decidir si debe solicitar más información. Debe seguir una estructura FALLO-CUANDO:

- Breve descripción que sea lo suficientemente específica para que el lector pueda visualizar la falla.
- Debe seguir una estructura FALLO-CUANDO
- Puede contener la categoría del error o el componente de la aplicación donde ocurrió el error (por ejemplo, carrito, interfaz de usuario, etc.) y la acción o circunstancias en las que ocurrió (CUANDO).

Ejemplos correctos:

Fallo	Cuando
"El nuevo usuario no puede iniciar sesión con la contraseña predeterminada proporcionada por correo electrónico o SMS"	
"[Perfil] La imagen de perfil se oscurece cuando se ingresa al chat"	
"CARRITO - Los nuevos artículos agregados al carrito no aparecen"	
"PRECIOS - El tamaño de fuente del texto del título es incorrecto"	



Ejemplos incorrectos:

Redacción incorrecta	Razón
"No puedo ver el producto cuando lo agrego, por alguna razón lo intento y no es así. ¿POR QUÉ? Solucionarlo lo antes posible".	Este ejemplo es malo porque es: vago, agresivo, incluye demasiadas palabras y pide que se implemente una solución.
"El texto de la página de precios parece extraño"	No es un hecho reproducible, es una opinión subjetiva (percepción de la persona).
"El nuevo usuario no puede iniciar sesión"	Indica el FALLO, no el cuándo. Por ejemplo: ¿Cuándo utiliza cuales credenciales?
"El inicio de sesión del usuario no funciona como se esperaba"	Opinión subjetiva (percepción)

Errores críticos		SEVERIDAD AZURE
Cuelgue o Fallo:	Crash general de la aplicación.	Critical
Bloqueo:	Impide la continuidad de las pruebas de la aplicación.	Critical
Mayor:	Es una pérdida mayor de funcionalidad, un error de salida (los datos grabados, visualizados o impresos son incorrectos) o la aplicación no concuerda con las especificaciones.	Critical
Errores NO críticos		
Menor:	Es una pérdida menor de funcionalidad, o un problema al cual se le puede encontrar una vuelta (un error, que no es cuelgue o fallo, bloqueo, mayor ni de texto).	High
Texto:	Se usa en los casos de falta de ortografía, y rótulos o mensajes de error o advertencia, incorrectos o faltantes.	Low
Mejora:	Sugerencia que aporta una nueva característica o facilidad a la aplicación	Low
Ambientes:	Falta de configuración, parámetros y/o permisos	Medium

Severity

3 - Medium

1 - Critical

2 - High

3 - Medium

4 - Low



7. Prueba Estática

Proceso de revisión

Revisión Informal (previo al refinamiento)

Por ejemplo, comprobación entre pares (tríada).

- **Objetivo principal:** detectar defectos potenciales antes del inicio del desarrollo/pruebas.
- **Posibles objetivos adicionales:** generar nuevas ideas o soluciones, resolver de forma rápida problemas menores.
- Varía en utilidad dependiendo de los revisores (lo ideal sería con personas que conocen el área de negocio involucrada).
- El uso de checklist de refinamiento es opcional.

Inspección o Revisión Formal (en el refinamiento)

- **Objetivos principales:** detectar defectos potenciales, evaluar la calidad y generar confianza en el producto de trabajo, prevenir futuros defectos similares mediante el aprendizaje del autor y el análisis de la causa raíz.
- **Posibles objetivos adicionales:** motivar y capacitar a los autores para que mejoren los futuros productos de trabajo y el proceso de desarrollo de software, alcanzar un consenso
- Se elaboran registros de defectos potenciales e informes de revisión.
- Se recopilan y utilizan métricas para mejorar el proceso de desarrollo de software completo, incluido el proceso de inspección.
- El uso de Checklist de refinamiento es obligatorio.

Checklist Células

Descripción de la historia	Invalidante
Menciona el rol beneficiario de la funcionalidad solicitada (como ... rol)	Si
Se describe en forma clara y concisa la funcionalidad que se requiere (quiero/necesito)	Si
Se describe claramente el problema que el rol desea solucionar (para ...)	Si

Criterios de Aceptación	Invalidante
Están claramente definidos los criterios	Si
Los criterios son consistentes con la descripción de la historia	Si
Se encuentran especificados los criterios de aceptación críticos	Si



Los criterios no contienen ambigüedades o comparaciones con otras funcionalidades, se encuentran personalizados de acuerdo con la funcionalidad exacta que se debe construir	Si
Se encuentran los campos de entrada (obligatorios y opcionales) y/o campos de salida definidos.	Si
Se encuentran claramente definidos los procesos previos y precondiciones para las funcionalidades solicitadas.	Si
Se pueden identificar campos generadores de eventos y condiciones que diversifican las funcionalidades	Si

Sobre la historia	Invalidante
Se encuentra estimada la funcionalidad solicitada	Si
Es lo suficientemente pequeña para que pueda ser desarrollada como unidad (o debería separarse en otras historias)	Si
Puede ser probada esta funcionalidad	Si
Se han identificado restricciones, supuestos, requerimientos no funcionales particulares.	Si
Se han identificado reglas de los negocios particulares, consideraciones sobre el alcance.	Si
Los prototipos de interfaz gráfica, si existen, son consistentes con la funcionalidad definida en la HU.	Si

Fábrica

Descripción del relevamiento	Invalidante
La descripción de la solicitud es clara e indica el problema que se desea resolver o la justificación de la funcionalidad requerida por el usuario	Si
Se encuentran especificadas las funcionalidades críticas.	Si
Las descripciones de las funcionalidades no contienen ambigüedades o comparaciones con otras funcionalidades, se encuentran de acuerdo con la solicitud exacta que se debe construir.	Si
Se encuentran claramente definidos los procesos previos y precondiciones para las funcionalidades solicitadas.	Si
Se encuentra definida la secuencia/orden de los pasos para completar la funcionalidad.	Si
Se especifica que tipo de usuario o rol debe acceder o utilizar la funcionalidad.	Si
Se pueden identificar campos generadores de eventos y condiciones que diversifican las funcionalidades.	Si
Se encuentran los campos de entrada (obligatorios y opcionales) y/o campos de salida definidos.	Si
Se encuentran evidenciadas las pruebas unitarias que abarca el flujo básico.	Si
Componentes involucrados	
Se han especificado las tablas y campos involucrados dentro de la funcionalidad y/o el proceso	Si



Se han adjuntado las pantallas modificadas (antes y después del cambio) en los casos requeridos	Si
---	----

8. Checklist General de Proyectos

Los siguientes ítems son para tener en cuenta para el Control de Calidad de los proyectos que entran tanto por Solicitud de Programas, Solicitud de Pedidos Internos, Pedidos Células e Historias.

Proyectos C# y Fox	
1.1	Verificar que no existan sentencias SQL ejecutadas directamente dentro del formulario FOX.
1.2	Verificar que las configuraciones de proyectos estén de la siguiente forma: Biblioteca de clases, Any CPU, Release, Namespace f2.
1.3	Verificar que la conexión de formularios C#: Es producción: true. TNS: Datosita SRVORACLE dbitacua.
1.4	Verificar proyectos en donde se utiliza la webservice de imágenes que este apuntado a http://servicebus.bancontinental.com.py:10100/extractos.asmx .
1.5	Verificar que los nuevos proyectos C# estén estructurados en base al Project base establecido Form1.
1.6	Verificar que el botón de ayuda que llama al manual correspondiente posea el código correcto que es proporcionado por el dpto. De Pase Producción. El mismo debe llamar a SRVORACLE Dbitacua y F2lanza3.
1.7	Verificar la tabla programa_objeto , el código utilizado debe ser el mismo que el código correspondiente al menú del proyecto en cuestión.
1.8	Verificar que los proyectos C# y Fox de solo reportes o que tengan la funcionalidad de generar reportes apunten a SRVACTIVE dbticua y DATOSITA1 respectivamente.
1.9	Verificar la documentación en proyectos C# y Fox. Deben nro. pedido, programador y descripción (los métodos y funciones deben estar comentados) .
1.10	Verificar que el nombre las DLL generadas para Proyectos C# 2005 terminen con “_dx” y proyectos C# 2010 en adelante terminen con “_4”
1.11	Verificar que todos los proyectos C# se deben migrar a visual studio 2019. En caso de que este no pueda ser migrado se requiere una justificación escrita (comentario del tech lead en las HU o por correo).
1.12	Verificar que todos los proyectos C# deben utilizar el .net framework con versión 4.5 o superior.
1.13	La Plataforma debe de ser x64 y la versión de la Oracle.DataAccess.dll debe de ser 4.122.19.1 o superior.
1.14	Se debe de cargar un soporte para cargar las pantallas C# a migrarse en la tabla wilson1.dt_parametros tipo_param 19191 (no aplica para Fox o Daemon)
1.15	Se debe de actualizar la librería ChilkatDotNet45.dll con la versión 9.5.0.93 o superior y la librería F2_Lanza3 con la versión 2023.9.3.1 o superior debido a la actualización del framework del proyecto y la arquitectura estas librerías dejan de funcionar. La librería del F2_Lanza3 se podrá obtener descargando del proyecto del ecosistema de



	Procesos Automáticos Daemon y la librería del ChilkatDotNet y su respectiva contraseña podrán obtenerla con el equipo de migración.
1.16	Los responsables de Aprobación de Pull Request realizarán rechazados en casos de pantallas C# con referencias apuntando a desarrollo, responsabilidad compartida con los líderes técnicos.
Base de Datos	
2.1	Verificar los comentarios en la creación y alteración de tablas y sus respectivas columnas.
2.2	Verificar la documentación en los paquetes o programas de la base de datos (los procedimientos y funciones deben estar comentados).
2.3	Verificar que dentro de los paquetes o tablas no exista "Ñ" como variables o campos de tablas. Tener en cuenta el mismo criterio para los comentarios y las estructuras de las tablas.
2.4	Verificar que, en la creación de tablas, las mismas cuenten con el tablespace para tablas e índices correspondientes según los datos provistos por DBA.
2.5	Verificar los objetos inválidos en la base de datos antes y después de la compilación de fuentes.
2.6	Verificar la creación de tablas, el tamaño que le asignen a la tabla nueva en el caso que lo indiquen o de lo contrario puede omitirse dentro del script (initial 64K next 1M).
2.7	Verificar los estándares de programación, las nomenclaturas de los objetos creados en la base de datos.
2.8	Verificar que en la creación/alter de tablas no se incluyan permisos de ningún tipo.
2.9	Verificar que, si el pedido involucra creación de tablas en el servidor de Producción, se debe de adjuntar un documento con el ok de DBA y del sr Wilson Medina.
2.10	Verificar el costo de sentencias de fuentes de base de datos modificadas y/o nuevas no supere los 1000, en caso hipotético de superar este número se debe pasar a producción con previa autorización de DBA vía correo.
2.11	Verificar que la tabla no cuente con TABLE ACCESS FULL sin índices creados, en caso de no contar con índices creados y tener TABLE ACCESS FULL será un punto de rechazo hasta la creación de los índices con OK de DBA, en caso de contar con TABLE ACCESS FULL con índices creados se pasará por excepción con OK de DBA.
2.12	Verificar que en caso de contar con Alters de una tabla dentro del Listado de tablas que Migra el Daemon Tablero , se controlara que se realice también los alters en el BI en los esquemas stage, stage_fm, en caso de que este dentro de la lista, será punto de rechazo no contar con alters en el BI en caso de modificar tablas que se migran al BI.
Archivos Fuentes	
3.1	Verificar que las fuentes levantadas existentes para su paso a producción hayan sido solicitadas previamente al dpto. de Pase a Producción.
3.2	Verificar que los archivos de objetos de la base de datos que se levanten posean siempre el nombre del objeto que se está creando o afectando.
3.3	Verificar que los nombres de las fuentes no se cambien , a no ser que sea un nuevo proyecto.



3.4	Verificar que los scripts de creación/alteración de Objetos de la Base de Datos se presenten de acuerdo los estándares establecidos.
3.5	Normar las extensiones de los scripts de creación/alteración de Objetos de la Base de Datos. Por ejemplo: trigger. trg, paquete. pck, funcion .fnc , types .typ , procedures .prc ., cuando sean sentencias ddl, alteraciones de tables, creaciones de índices, secuencias, debe tener la extensión. pdc
3.6	<p>Cuando un pedido de desarrollo, sea un bug o una Historia de Usuario que cuente con fuentes nuevas y/o modificadas de Oracle debe de crearse un archivo de “Pasos de Compilación {HU XXXXX/Bug XXXXX}” con formato .txt adjunto en la HU/Bug dentro del Azure, debe de incluir en que base/s (Punto 9 del Documento) va/n a compilarse el/los objeto/s, debe de especificarse el orden en caso de que exista de los archivos y los permisos o “granteos” de compilación de fuentes, adicionalmente se puede agregar los permisos a usuarios de APIs, Servicios, Daemons, pero NO se debe adjuntar grants a Departamentos o a Usuarios de tres letras, permisos para Departamentos específicos deben de direccionarse por Service Desk a Seguridad cargado por el Usuario, PO o el TL a demanda.</p> <p>No aplica sentencias DML (Update/Inserts) en pasos de compilación, se debe redireccionar por Soporte cargado por el Usuario, PO o el TL a demanda.</p>
3.7	En caso de contar con archivos .pdc , verificar que no cuente con una sentencia drop al inicio del archivo, en caso de contar con esta sentencia será punto de rechazo
3.8	Los archivos .pdc de creación de tablas deben de nombrarse por tabla y deben de ser un solo archivo .pdc por creación de tabla , en caso de crear mas de una tabla por archivo es punto de rechazo
Documentación	
4.1	Verificar la descripción de las tareas realizadas por el programador y pruebas técnicas realizadas en ambiente de Desarrollo, así como también los comentarios del Líder de Proyecto, los mismos deben ser bien detallados y específicos.
4.2	Verificar que el Manual de Usuario se presente en el formato correcto y que explique detalladamente la funcionalidad de la pantalla modificada, siempre que se modifique una pantalla existente, o el proyecto sea nuevo. El archivo no debe superar los 3MB .
4.3	Verificar el Diagrama entidad relación (DER), deben estar las tablas involucradas en el proceso y la relación entre las mismas. El formato debe estar en. PDM y “DER_NOMBREPROGRAMA_NROPEDIDO”.
4.4	Verificar que el relevamiento cuente con las especificaciones técnicas y pruebas unitarias correspondientes.
4.6	Verificar que, en proyectos relacionados al área de Prevención de Lavado de Dinero , el relevamiento se debe especificar a qué ítem del Manual de PLD hace referencia.
4.7	Verificar que en el relevamiento se encuentre el flujograma del pedido.
4.8	Verificar en caso de que se modifique el proyecto general_4 u objetos de la base de datos relacionados al proyecto, siempre este presente la aprobación de modificación por parte de un director y de los Líderes de Proyectos designados.
Proyectos Daemon	
5.1	Verificar los proyectos Daemon, deben tener incluidos el timer de conexión a la base de datos Marca presencia , que hace open close en un tiempo no mayor a 5min.
5.2	Verificar que no posea conexión con ServiceBus, debe ser conexión directa.



5.3	Verificar en el caso nuevo Daemon, que se definan la contraseña y el usuario a ser utilizados en el Daemon, así como la TNS dentro del App_config .
5.4	Los daemons que no cuenten con usuarios propios se exigirá la creación de un nuevo Usuario con la nomenclatura inicial DMN_ (nombre del Proceso) siguiendo el nombre del proceso, en el relevamiento de desarrollo deberá incluirse los permisos correspondientes al nuevo usuario, se controlara el alta del usuario y excepción de IP en los apartados de Análisis de Seguridad en la HU, este punto será obligatorio y en caso de que no se cumpla será punto de rechazo para el pase a producción del pedido.
5.5	Se solicitará que en el relevamiento de Desarrollo se incluya los timers existentes en los Daemons modificados indicando el motivo de la configuración realizada en función a su contexto según las reglas de negocio.
5.6	La Plataforma debe de ser x64 y la versión de la Oracle.DataAccess.dll debe de ser 4.122.19.1 o superior.
5.7	Verificar que todos los proyectos C# deben utilizar el .net framework con versión 4.5 o superior.
5.8	Se debe de actualizar la librería ChilkatDotNet45.dll con la versión 9.5.0.93 o superior, debido a la actualización del framework del proyecto y la arquitectura estas librerías dejan de funcionar, la librería del ChilkatDotNet y su respectiva contraseña podrán obtenerla con el equipo de migración
Proyectos WEB	
6.1	Verificar que los proyectos web que impliquen nuevos Servicios/Productos, tengan una forma de parametrizar la habilitación para clientes específicos, tipos de clientes o a todo el banco.
6.2	Verificar las pruebas hechas con los proyectos web en los principales navegadores (Chrome, Internet Explorer, Firefox, Edge, Safari).
Otros Controles	
7.1	Verificar que los archivos adjuntos no se guarden en tablas, sino en el Webservice, y en la base debe guardarse la dirección del archivo.
7.2	Verificar los usuarios, departamentos o correos específicos de proyectos o procedimientos de la base. Deben ser paramétricos . En lo posible los registros ya deben estar insertados en producción al pasar a Control de Calidad.
APIS	
8.1	Verificar que los despliegues cuenten con una HU o Bug de Tipo Producción vinculado al commit que se desea desplegar para no perder trazabilidad en pedidos.
8.2	Verificar que si es una API solo de Consulta que exista una cadena de conexión en la variable del Pipeline y que este apunte correctamente al Active de QA y la misma variable del grupo de producción este apuntando al Active.
8.3	Para solicitar pruebas de rendimiento la API aplicara para casos donde esta es una de uso masivo o transaccional previamente definida en conjunto con el líder técnico.
8.4	Se debe contemplar en el análisis, el diseño y la ejecución el realizar pruebas con el cierre diario y con el cierre mensual cuando se trate de APIs transaccionales (realizan movimiento de dinero) que estén disponible de manera externa para clientes.
8.5	Las APIs que no cuenten con usuarios propios y utilicen el usuario de API_CONSULTA se exigirá la creación de un nuevo Usuario en conjunto con la definición de



	Integraciones, Seguridad e Infraestructura, se controlara el alta del usuario y excepción de IP en los apartados de Análisis de Seguridad en la HU, y la creación de las Variables de Grupo en el apartado de Análisis de Infraestructura en la HU, ambos serán obligatorios y en caso de que no se cumpla serán puntos de rechazo para el pase a producción del pedido.
8.6	Se aplicara pruebas de rendimiento a todas las APIs nuevas o modificadas, los controles se aplicaran en el rendimiento de la base de datos Oracle, visualizando el monitor de la base de datos, serán punto de rechazo: un consumo mayor al 50% de capacidad de la base de datos, casos de bloqueo de tablas o alta concurrencia, el resultado de las mismas se autorizara con el OK de DBA , el Encargado de QA podrá aplicar excepciones de este control para los Bugs de Tipo Producción o HU dependiendo del caso.
8.7	<p>Se aplicarán controles de verificaciones de Códigos de Respuesta, basado en el estándar actual que rige el significado de los códigos HTTP de las APIs es el RFC 7231 El RFC 7231 define una serie de códigos de estado de respuesta HTTP que se dividen en cinco clases, cada una con un rango de códigos específico:</p> <p>Clase 1xx: Respuestas informativas Clase 2xx: Respuestas satisfactorias Clase 3xx: Redirecciones Clase 4xx: Errores del cliente Clase 5xx: Errores del servidor</p> <p>Estos controles aplicarán tanto para APIs nuevas o Modificadas, en caso de no controlar correctamente las respuestas o validaciones obligatorias según el estándar serán punto de rechazo.</p>
8.8	Al finalizar las pruebas funcionales de una API nueva y/o modificada subir la colección de Postman utilizada en el repositorio del equipo correspondiente de la API dentro del proyecto de Quality Assurance , en caso de ser una API nueva actualizar el inventario de pruebas de APIs , flujo de pruebas de APIs definido en Anexo - Punto 6.
8.9	Se crea el Rol de Automatizador de Pruebas, el mismo es responsable de automatizar las pruebas de las colecciones creadas por su equipo antes del pase a produccion, el mismo es el responsable de seguir las buenas prácticas definidas en nuestra Wiki Colecciones de Postman - Buenas Prácticas , Archivo YAML , Automatizacion con Postman y de realizar el Pull Request segun lo definido en la Wiki Creacion de Ramas y Pull Request - GitHub Desktop
8.10	En caso de existir pruebas de APIs nuevas y/o modificadas deben de contar con la tarea " sonar-buildbreaker " dentro del pipeline de la API basándonos en la wiki referente al Control de Sonar Qube verificando el archivo .yaml, esta responsabilidad es compartida con los lideres técnicos aprobadores de Pull Request y será punto de rechazo en caso de no cumplirse.



9. Ambientes de Control de Calidad

9.1. Definición

Conjunto de datos representativos el entorno de producción replicados a un ambiente segregado e independiente para realizar las pruebas sin alterar las operaciones o procesos rutinarios de la Entidad. El entorno de prueba consta de elementos que respaldan la ejecución de la prueba con software, hardware y red configurados.

Los entornos de prueba o de homologación deben ser definidos y diseñados precautelando los principios y buenas prácticas de seguridad, control interno y criterios de desempeño requeridos.

Los entornos definidos por Continental son los siguientes:

AMBIENTES DE CONTROL DE CALIDAD		
Servidores	IP	Descripción
PRUEBAS_212	10.6.2.212	Servidor Principal
CIERRE_47	10.6.2.47	Pruebas de cierres contables
BI_QA	10.6.2.53	Pruebas BI y T5
ACTIVE_QA	10.6.2.96	Pruebas de Reportes y Consulta
PRUEBAS_211	10.6.2.211	Pruebas de Oracle 19
SRVHISTORICOS	10.6.2.19	Pruebas con la Base de Históricos compartido con Desarrollo

WEB Y APP		
Servidores	IP	Descripción
DESAWEBAPP-QA	10.6.2.93	Publicaciones de pruebas
DEV-API-QA	192.168.48.15	Backend de la APP

HANGFIRE		
Servidores	IP	Descripción
DESAHF-QA	10.6.2.94	Servidor de hangfire, procesos internos

CRM		
Servidores	IP	Descripción
DESAZOHO	10.6.2.91	Servidor CRM ZOHO

NINTEX	
SERVIDORES	Descripción
PROCESO DE CREDITO	Célula de Proceso de Crédito
https://dev-intranet/sitios/Procesos/SitePages/Inicio.aspx	

DESARROLLOS VARIOS	Desarrollo en fábrica y otras células
https://dev-intranet/sitios/Sistemas/SitePages/Home.aspx	



Monitoreo de Base de Datos	
SERVIDORES	Descripción
Oracle Cloud Control	Monitoreo de Base de Datos
https://10.6.2.67:7803/em/faces/logon/core-uifwk-console-login	

Anexo

1 - Plantilla Caso de Prueba

TEST CASE (Nombre)										
Nro. de TC:			Fecha: / /							
Objetivo(s) de la prueba: •										
Responsable(s): •										
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Prioridad: Alta Media Baja										
Trazabilidad:			Dependencia:							
Precondiciones: •										
Escenario										
Item	Descripción	Entradas	Resultado esperado	Resultado actual (si es diferente a lo esperado)	Estado (Successful/Failed)					
				Total Estado						

1.1 - Entorno Azure



Banco Continental S.A.E.C.A
Departamento de Control de Calidad -Tecnología
Documentación Interna
Versión: <2.5> 17/10/2023

Identificador único

Título

Responsable

Entradas

- Objetivo
- Pre-condiciones

Trazabilidad

Prioridad

NEW TEST CASE * Field 'Title' cannot be empty.

Enter title

Guino Henostroza

0 comments Add tag

Save

Steps Summary Associated Automation

Deployment

To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

Development

+ Add link

Related Work

+ Add link ~

Add an existing work item as a parent

Parameter values

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Automation status

Not Automated

Steps

Click or type here to add a step

Steps Action Expected result

Priority

2



2 - Plantilla Informe de Incidente de Pruebas.

INFORME DE INCIDENTE DE PRUEBAS	
Nro de Incidente:	Fecha: / /
Originador: (persona que identifica el incidente)	Asignado: (persona asignada para resolver el incidente)
Título:	
Descripción del Incidente:	
Pasos para reproducir:	
<ul style="list-style-type: none">	
<div><input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Severidad:</div> <div>Alta Media Baja</div>	<div><input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Prioridad:</div> <div>Crítico Alta Media Baja</div>
Información adicional:	
Estado	

2.1 - Entorno Azure

Nro. de Incidente: 27510

Estado: Active

Fecha: 23/11/2020 13:18

Originador: Carlos Micael Uriarte Lopez

Título: No se genera ninguna alerta para las cuentas corrientes ya que el procedimiento sp_situaCtaCC_alerta se queda congelado, sin hacer nada

Descripción: Bug filed on "Caso de prueba N°-005. Verificar generación de alerta a ser enviada por correo para Gerentes."

Repro Steps:

Step no.	Result	Title
1.	Passed	Ejecución del procedimiento wison1.pkg_notificaciones_cuentas.sp_situaCtaAHO_alerta o wison1.pkg_notificaciones_cuentas.sp_situaCtaCC_alerta (dependiendo del tipo de cuenta).
2.	Failed	Ver tabla userweb.emailservir

Expected Result: Se debe visualizar el informe consolidado con los datos de las cuentas correctamente según la fecha de generación de la alerta y la cantidad de días sin movimiento.

Comments: El procedimiento sp_situaCtaCC_alerta tarda más de una hora y no envía ninguna alerta

Attachments: [Attachment]

Planning:

- Asignado: Carlos Micael Uriarte Lopez
- Severidad: 1 - Critical

Deployment:

- Ready

Related Work:

- 27510: Caso de prueba N°-005. Verificar generación de alerta a ser enviada por correo para Gerentes.

System Info:

Documents adjuntos: [Attachment]

Pasos para reproducir: [Attachment]

Prioridad / Severidad: 1 - Critical

Asignado: Carlos Micael Uriarte Lopez



3 - Política especial para Azure DevOps.

1. El enlace del test suite debe estar anexo a la HU que va a ser probada. Con el fin de tener trazabilidad desde el Azure Test Plan al Azure Backlog y viceversa.
2. El test plan creado (Test Suite) debe de ser cerrado posterior a la finalización completa del test suite en el Azure a modo de poder dar por finalizada las pruebas del pedido o HU y dicha trazabilidad sea reflejada en el work item.

Paso 1: Click Derecho sobre la carpeta del Plan de Pruebas creado y darle click en Open

The screenshot shows the Azure DevOps Test Plan interface. On the left, a sidebar lists test suites under the heading 'Migración de IIS a K8 - NET 6 - Finasys Utilitario'. One suite, '110447 : Actualizar a net 6 APIs de openbanking - Transferencias Interbancarias', is selected. A right-click context menu is open over this suite, with the 'Open' option highlighted in red. The main area displays a table of test points for this suite, all of which are marked as 'Passed'.

Title	Outcome	Order	Test Case Id	Assigned To
CP_001_Aprobar Transferencia Interbancaria con cuenta origen Habilitado.	Passed	1	31140	Steven Gilbert Gra...
CP_002_Aprobar Transferencia Interbancaria con cuenta origen Inactiva.	Passed	2	31141	Steven Gilbert Gra...
CP_003_Aprobar Transferencia Interbancaria con cuenta origen Bloqueado.	Passed	3	31142	Steven Gilbert Gra...
CP_004_Aprobar Transferencia Interbancaria con Saldo disponible mayor al monto a transferir.	Passed	4	31143	Steven Gilbert Gra...
CP_005_Aprobar Transferencia Interbancaria con Saldo disponible igual al monto a transferir.	Passed	5	31144	Steven Gilbert Gra...
CP_006_Aprobar Transferencia Interbancaria con Saldo disponible menor al monto a transferir.	Passed	6	31145	Steven Gilbert Gra...
CP_007_Aprobar Transferencia Interbancaria con Saldo disponible mayor al monto a transferir (...)	Passed	7	31146	Steven Gilbert Gra...
CP_008_Aprobar Transferencia Interbancaria con Saldo disponible igual al monto a transferir (...)	Passed	8	31147	Steven Gilbert Gra...
CP_009_Aprobar Transferencia Interbancaria con Saldo disponible menor al monto a transferir (...)	Passed	9	31148	Steven Gilbert Gra...
CP_010_Aprobar Transferencia Interbancaria con Fecha de Agendamiento menor a la Fecha Ac...	Passed	10	31149	Steven Gilbert Gra...
CP_011_Aprobar Transferencia Interbancaria con Fecha de Agendamiento menor a la Fecha Ac...	Passed	11	31150	Steven Gilbert Gra...
CP_012_Aprobar Transferencia Interbancaria con Fecha de Agendamiento menor a la Fecha Ac...	Passed	12	31151	Liz Paola Ferreira ...
CP_013_Aprobar Transferencia Interbancaria con Fecha de Agendamiento menor a la Fecha Ac...	Passed	13	31152	Liz Paola Ferreira ...
CP_014_Aprobar Transferencia Interbancaria con Fecha de Agendamiento menor a la Fecha Ac...	Not Applicable	14	31153	Liz Paola Ferreira ...

Paso 2: Cambiar estado de In Progress a Completed

The screenshot shows the Azure DevOps Work Item interface for a work item titled '110447 : Actualizar a net 6 APIs de openbanking - Transferencias Interbancarias'. The 'State' dropdown menu is open, and the 'Completed' option is highlighted in red. The 'Reason' field is set to 'Completed testing for th...'. The 'Classification' section shows 'Test Suite Type' as 'Requirement Based' and '7pace Timetracker' as 'Please navigate to the "TIME" hub first, to see info'.

Paso 3: Al finalizar la HU debe de estar Test Suite con estado Completed

The screenshot shows the Azure DevOps Work Item interface for the same work item. The 'State' dropdown menu is now set to 'Completed'. The 'Reason' field is 'Completed testing for th...'. The 'Classification' section shows 'Test Suite Type' as 'Requirement Based' and '7pace Timetracker' as 'Please navigate to the "TIME" hub first, to see info'.



4 - Política de Creación de Usuarios de APIs:

PROPUESTA PARA MANTENER TRAZABILIDAD Y CONTROL EN EL FLUJO DE PASE A PRODUCCION DE APIs CON NUEVOS USUARIOS

USER STORY DESARROLLO 69479

69479 Registro de versiones y modificaciones backend

Nestor Ariel Duarte Cardozo 0 comments Informe de Riesgos X +

State Test Area Analisis de Riesgo
Reason Moved to state Test Iteration Analisis de Riesgo\Sprint 56

Detalles Relevamiento Anál

Análisis DBA

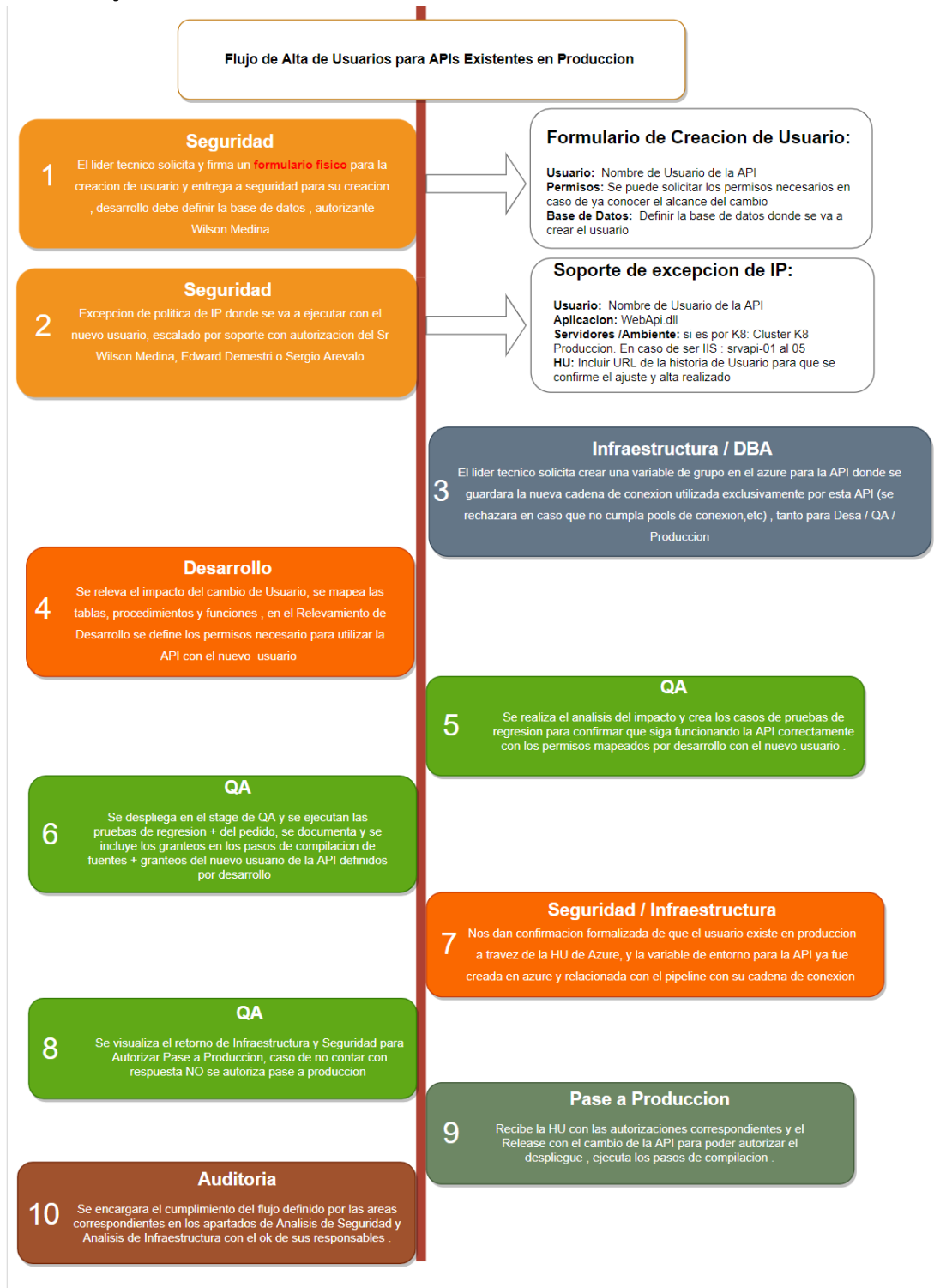
@Oscar Gabriel Sanchez Vazquez Favor autorizar creación de tabla y secuencia.

Ok de mi parte @Hugo Ramon Benitez Cañete

Análisis Infraestructura		En este apartado se solicita / confirma que se procedió a crear las nuevas variables de grupo para DESA/QA / PROD por el Responsable de Infraestructura
n/a		
Análisis Seguridad		En este apartado se confirma que el usuario fue creado correctamente en produccion por el Responsable de Seguridad
n/a		



4.1 – Flujo de Creación de Usuarios de APIs:

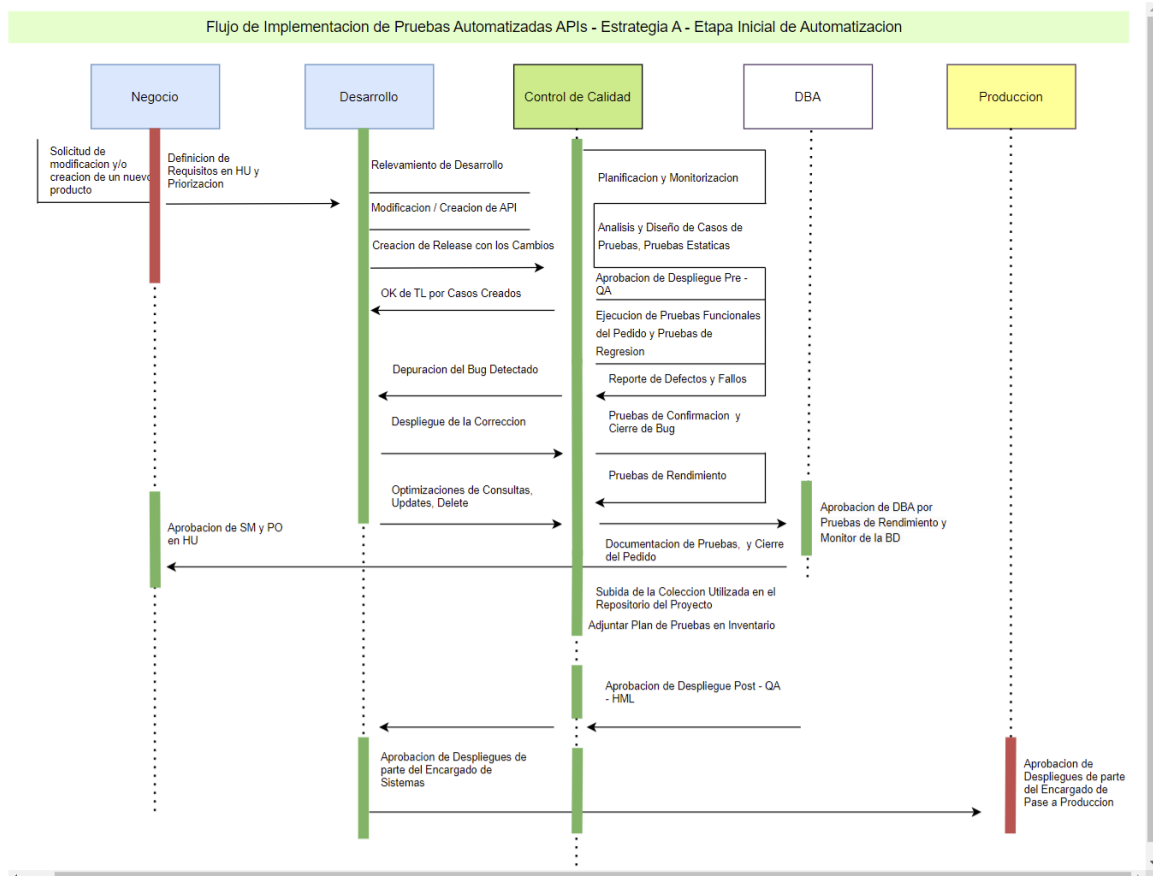




5 – Flujo de Control de Alters:

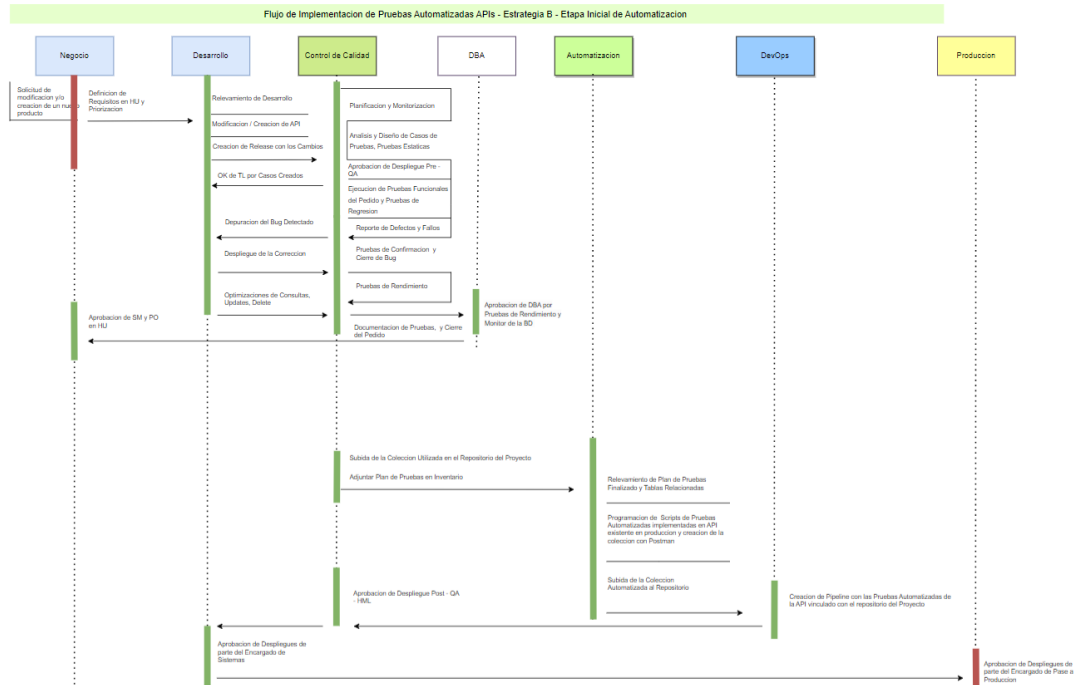


6 – Actualización de Flujo de Pruebas de APIs: Flujo Inicial (Subida de Colecciones al Repositorio)





6.1 – Actualización de Flujo de Pruebas de APIs: Automatización, creación de Pull Request e implementación de Pipelines



Responsable/s	Versión	Fecha actualización	Obs
Diego Martinez, IT Alianza	V1	12/02/2021	Política General del Dpto. de Control de Calidad
Eric Aranda, Steven Ayala	V1.1	04/07/2022	C# - Fox: Cadena de Conexión al Active Base de Datos: Ok de Creación de Tablas APIs: Conexión a Active
Steven Ayala	V1.2	02/08/2022	APIs: 8.4 (Cierre) Base de Datos: 2.10 (Costo 1000)
Steven Ayala	V1.3	05/09/2022	APIs: 8.5 (Cambio de Usuario) Base de Datos: 2.11 (Table Access Full)
Steven Ayala	V1.4	13/09/2022	APIs: 8.5 (Actualización)
Steven Ayala	V1.5	03/11/2022	APIs: 8.6 (Pruebas de Rendimiento)
Steven Ayala	V1.6	13/12/2022	Daemon: 5.4 (Cambio de Usuario) y 5.5 (Timer)
Steven Ayala	V1.7	06/02/2022	Base de Datos: 2.12 (Alters con migración a BI)



Banco Continental S.A.E.C.A
Departamento de Control de Calidad -Tecnología
Documentación Interna
Versión: <2.5> 17/10/2023

Steven Ayala	V1.8	13/04/2023	Controles Daemon: 5.6 (Versión de Oracle) y 5.7 (Framework) Pantalla C#: 1.12 (Framework)
Steven Ayala	V1.9	03/05/2023	Controles Pantalla C#: 1.13 (Versión de Oracle)
Steven Ayala	V2.0	09/05/2023	Controles de Migración Oracle 19 Ambiente de Pruebas: 10.6.2.211 Daemon: 5.8 Pantalla C#: 1.14 - 1.15
Steven Ayala	V2.1	24/05/2023	Política especial para Azure DevOps: 1 – Cierre de Test Suite al finalizar las pruebas de un pedido
Steven Ayala	V2.2	20/07/2023	APIs: 8.7 – Manejo de Códigos de Respuesta y Validaciones de Contrato – Estándar RFC 7231 Pantallas C#: 1.16 – Rechazo de Pull Request por referencias a desarrollo
Steven Ayala	V2.3	18/08/2023	APIs: 8.8 – Subida de Colección en Repositorio y Actualización de Inventario de Pruebas de APIs Anexo – Enumeración de Cada Anexo y se define el flujo de Pruebas de APIs con el punto 8.8.
Steven Ayala	V2.4	10/10/2023	Archivos Fuentes: 3.6 – Definiciones de Pasos de Compilación 3.7 – Control de Archivos .pdc APIs: 8.9 – Se definen las prácticas para automatización, creación de Pull Request y Archivo YAML. Anexo: 6.1 – Se Agrega Flujo de Pruebas de APIs con Automatización
Steven Ayala	V2.5	17/11/2023	APIs: 8.10 – Se crean controles para utilización de tareas con Sonar Qube en Pipelines