**Manual de Buenas Prácticas para Analistas QA**

**1. Comprensión del Producto:**

* **1.1. Análisis de requisitos:** Antes de comenzar las pruebas, revisa detenidamente los requisitos del producto. Por ejemplo, si estás probando un sistema de gestión de ventas, asegúrate de entender cómo debe comportarse el sistema al procesar una venta con descuento versus una venta normal.
* **Ejemplo:** Requisito: "El sistema debe aplicar un descuento del 10% automáticamente a todas las ventas superiores a $1000". Un caso de prueba podría ser verificar que el descuento se aplique correctamente cuando el total de la venta es $1200.
* **1.2. Creación de casos de prueba:** Desarrolla casos de prueba detallados que cubran todos los escenarios posibles. Por ejemplo, en un sistema de reservas de vuelos, podrías crear casos de prueba para verificar la reserva con datos válidos e inválidos, asegurándote de probar tanto el éxito como los posibles fallos.
* **Ejemplo:** Caso de prueba: "Intentar realizar una reserva con un número de pasajeros mayor al número de asientos disponibles. Verificar que el sistema muestre un mensaje de error adecuado y no permita la reserva."

**2. Planificación de Pruebas:**

* **2.1. Elaboración de planes de prueba:** Documenta un plan de prueba detallado que incluya la estrategia de prueba. Por ejemplo, si estás trabajando en un nuevo módulo de pagos para una aplicación móvil, tu plan de prueba podría incluir pruebas manuales y también pruebas de carga para asegurar que el sistema pueda manejar múltiples transacciones simultáneas.
* **Ejemplo:** Estrategia de prueba: "Realizar pruebas manuales para validar cada paso del proceso de pago (selección de método de pago, ingreso de datos, confirmación) y ejecutar pruebas de carga simulando 100 usuarios para verificar la estabilidad del sistema bajo carga máxima."
* **2.2. Priorización de pruebas:** Usa el análisis de riesgos para priorizar las pruebas. Por ejemplo, en un sistema de gestión hospitalaria, las pruebas relacionadas con la seguridad del paciente y la integridad de los datos clínicos tendrían una prioridad más alta que las pruebas de funcionalidades menos críticas.
* **Ejemplo:** Análisis de riesgos: "Identificar que un error en la actualización de datos médicos podría tener consecuencias graves para el paciente. Priorizar las pruebas relacionadas con la integridad y seguridad de los datos clínicos para minimizar este riesgo."

**3. Ejecución de Pruebas:**

* **3.1. Ejecución manual de pruebas:** Realiza pruebas exhaustivas basadas en los casos de prueba definidos. Por ejemplo, en un sistema de gestión de inventarios, verifica la capacidad del sistema para manejar múltiples órdenes de compra simultáneas desde diferentes ubicaciones geográficas.
* **Ejemplo:** Caso de prueba: "Crear y procesar tres órdenes de compra simultáneas desde diferentes ubicaciones geográficas y verificar que el sistema actualice correctamente los niveles de inventario y genere los registros de transacción correspondientes."
* **3.2. Automatización de pruebas:** Identifica oportunidades para automatizar pruebas repetitivas y de regresión. Por ejemplo, en una aplicación de comercio electrónico, automatiza las pruebas para asegurar que la funcionalidad de agregar productos al carrito y proceder con el pago siga funcionando correctamente en cada nueva versión del software.
* **Ejemplo:** Script de prueba automatizado: "Desarrollar un script que simule el proceso de agregar varios productos al carrito, aplicar un cupón de descuento y completar el proceso de pago. Ejecutar este script después de cada actualización del software para verificar que no se haya introducido ninguna regresión en la funcionalidad."

**4. Monitoreo y Seguimiento:**

* **4.1. Gestión de defectos:** Utiliza herramientas de gestión de defectos para registrar y seguir el estado de todos los problemas identificados durante las pruebas. Por ejemplo, en un sistema de gestión de proyectos, registra cualquier defecto que impida la asignación correcta de recursos a una tarea específica.
* **Ejemplo:** Defecto registrado: "Identificar que la función de asignación automática de recursos asigna incorrectamente un recurso a una tarea debido a un error en el algoritmo de asignación. Registrar el defecto y asignarlo a un desarrollador para su corrección."
* **4.2. Reporte de resultados:** Genera informes periódicos que resuman el estado de las pruebas. Por ejemplo, en un sistema de gestión financiera, presenta un informe que incluya métricas de cobertura de pruebas, número de defectos encontrados y el porcentaje de pruebas automatizadas versus manuales.
* **Ejemplo:** Informe de pruebas: "Informe semanal que muestra que el 95% de las pruebas han sido completadas, con 15 defectos identificados y un aumento del 20% en la cobertura de pruebas automatizadas desde la última semana."

**5. Mejora Continua:**

* **5.1. Retroalimentación y aprendizaje:** Participa en sesiones de retroalimentación para discutir problemas encontrados y oportunidades de mejora. Por ejemplo, después de una versión nueva del software, discute con el equipo de desarrollo sobre cómo mejorar la claridad de los mensajes de error para facilitar la resolución de problemas.
* **Ejemplo:** Retroalimentación del equipo: "Reunión post-implementación para discutir la falta de claridad en los mensajes de error del nuevo módulo. Proporcionar sugerencias al equipo de desarrollo para mejorar la redacción de mensajes que ayuden a los usuarios a entender y resolver los problemas más fácilmente."
* **5.2. Actualización de casos de prueba:** Revisa y actualiza regularmente los casos de prueba para reflejar cambios en los requisitos del producto. Por ejemplo, si se añade una nueva funcionalidad a un sistema de gestión de contenidos, actualiza los casos de prueba para asegurarte de que cubren todas las nuevas características y escenarios.
* **Ejemplo:** Caso de prueba actualizado: "Actualizar los casos de prueba existentes para incluir pruebas de la nueva funcionalidad de programación de publicaciones en redes sociales añadida al sistema de gestión de contenidos. Asegurar que se pruebe la correcta publicación automática en cada red social configurada."

**6. Colaboración y Comunicación:**

* **6.1. Comunicación efectiva:** Mantén una comunicación abierta y clara con los desarrolladores y otros stakeholders. Por ejemplo, organiza reuniones regulares con el equipo de desarrollo para discutir el progreso de las pruebas y los problemas encontrados.
* **Ejemplo:** Reunión de progreso: "Reunión semanal con el equipo de desarrollo para revisar el progreso de las pruebas del nuevo módulo de facturación y discutir cualquier obstáculo encontrado que pueda afectar la fecha de entrega prevista."
* **6.2. Colaboración interfuncional:** Trabaja estrechamente con el equipo de desarrollo desde las fases iniciales del ciclo de vida del software. Por ejemplo, participa en reuniones de planificación de sprint para entender los objetivos del desarrollo y asegurarte de que las pruebas se alineen adecuadamente con estos objetivos.
* **Ejemplo:** Reunión de planificación de sprint: "Participar en la reunión de planificación del sprint para el desarrollo del nuevo módulo de análisis de datos. Discutir los requisitos y las funcionalidades clave para asegurar que los casos de prueba se desarrollen de acuerdo con las nuevas funcionalidades planificadas."

**7. Cumplimiento y Ética:**

* **7.1. Cumplimiento de estándares:** Sigue las políticas y estándares de QA establecidos por la organización. Por ejemplo, asegúrate de adherirte a los estándares de seguridad de datos y privacidad en todos los aspectos de las pruebas.
* **Ejemplo:** Seguridad de datos: "Asegurarse de que todas las pruebas se realicen en un entorno seguro y protegido para cumplir con los estándares de seguridad de datos de la organización, evitando el uso de datos de prueba sensibles."
* **7.2. Integridad:** Mantén la imparcialidad y objetividad en la evaluación de la calidad del producto. Por ejemplo, informa de manera justa sobre todos los defectos encontrados, sin omitir ni exagerar ningún problema para influir en la percepción del producto.
* **Ejemplo:** Evaluación imparcial: "Reportar todos los defectos identificados durante las pruebas de regresión, independientemente de su impacto potencial en la experiencia del usuario, para asegurar que el equipo de desarrollo tenga una visión completa de la calidad del producto."