

Manual

Training de Analista QC

Norman Ruiz

May 19, 2019

Contents

1	Introduccion	3
2	Testing de software	4
2.1	Roles en el el desarrollo de software	4
2.2	Que es testing	5
2.3	Cual es el objetivo del testing	5
2.4	Y cual es su importancia	5
3	Fundamentos del Testing	6
3.1	Concepto: Error, Defecto y fallo	6
3.2	Mas conceptos: Requisito y Calidad	7
3.3	Atributos de la Calidad	7
3.4	¿Porque son necesarias las pruebas?	9
4	Proceso Fundamental de Testing	10
4.1	10
4.2	11
4.3	11
4.4	11
4.5	11
5	Modelos de Desarrollo de Software	12
6	Diseño de Pruebas	13
7	Técnicas Estáticas	14
8	Técnicas Dinámicas	15
9	Criterios para seleccionar el diseño apropiado de caso de prueba	16
10	Creación de Casos de Prueba	17
11	Resumen Curso	18
12	Para rendir el ISTQB Foundation Level	19

1 Introduccion

..."el testing puede probar la presencia de errores, pero no la ausencia de ellos"...

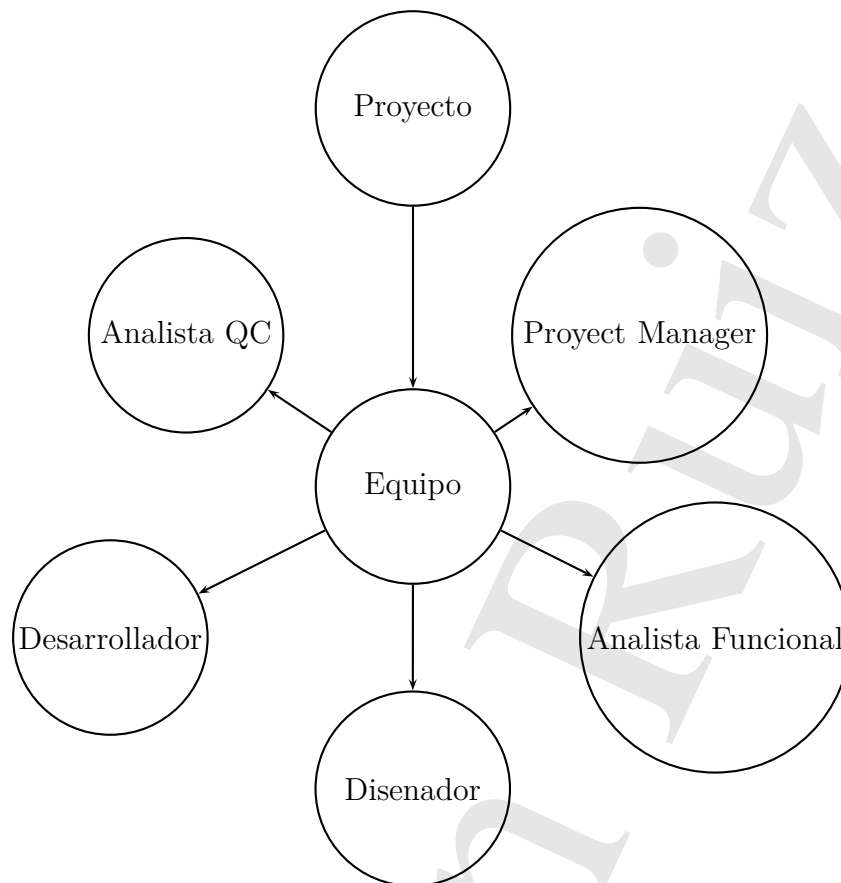
*L*a idea de escribir este pequeño manual es dar una introduccion al puesto de Analista QC a las personas que se inician en este puesto y no tienen experiencia previa. Abordaremos temas como la definicion de testing de software, hasta creacion de casos de pruebas, incluyendo fundamentos y tecnicas. Incluiremos material como ejemplos de casos de prueba para lograr una rapida insercion en sus tarea y sentar las bases para que sea capas de redactar su propia documentacion eficazmente.

2 Testing de software

2.1 Roles en el desarrollo de software

El desarrollo de software es una tarea que se lleva a cabo en equipo. Un equipo puede estar formado por dos o más personas que pueden desempeñar uno o más roles, cada rol realiza funciones específicas para lograr el objetivo común. El siguiente es un ejemplo de un equipo bastante simple:

- **Project Manager:** Coordina el equipo y asegura que todos los demás roles cumplan con su trabajo. Este debe tener una visión y misión bien clara del proyecto para lograr el éxito final.
- **Analista funcional:** Es el que se encarga de interactuar con el cliente para definir y redactar la documentación con las características del software a desarrollar.
- **Diseñador:** es quien toma la documentación del Analista funcional y genera el diseño y prototipo del software a desarrollar.
- **Desarrollador:** El desarrollador toma la documentación del Analista funcional y el diseño y prototipo del Diseñador y escribe el código correspondiente en un lenguaje definido para obtener finalmente el software solicitado.
- **Analista QC:** Es quien asegura la calidad de producto contrastando los requerimientos del cliente a través de la documentación del Analista Funcional y el diseño y prototipo del Diseñador.



Roles en el el desarrollo de software

2.2 Que es testing

Es el proceso empleado para identificar la correctitud, completitud, seguridad y calidad en el desarrollo de software.

2.3 Cual es el objetivo del testing

El proceso de testeo es una investigación técnica que intenta revelar información de calidad, posibles fallos y/o usabilidad acerca del software con respecto al contexto en donde operará.

2.4 Y cual es su importancia

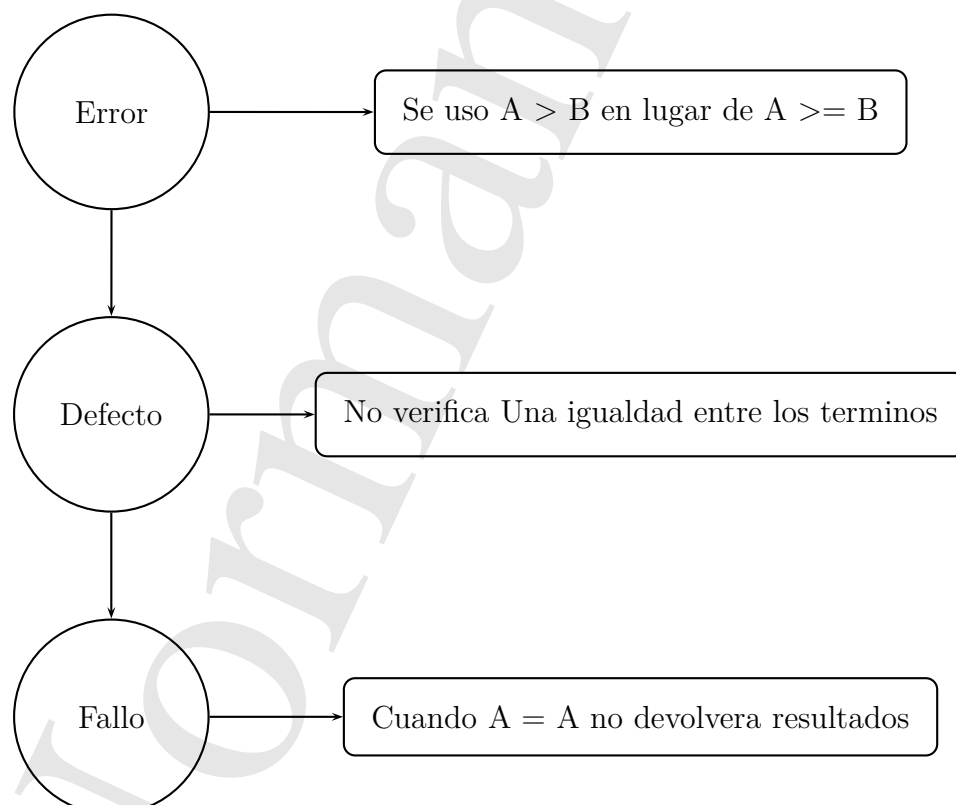
El testeo de un producto de software es uno de los pasos más complejos e importantes en el desarrollo de software. Esto incluye el proceso de encontrar errores en el software; pero el testeo no sólo se limita a eso. El testeo o prueba de un software se relaciona a atributos como la fiabilidad, eficiencia, portabilidad, escalabilidad, mantenibilidad, compatibilidad, usabilidad y capacidad del mismo.

3 Fundamentos del Testing

*E*n esta seccion introduciremos conceptos basicos que le permitiran expresarse, redactar documentacion o simplemente escribir un mail usando los terminos correctos. Esto ademas le permite situarse correctamente en el proceso y comunicar con mayor esactitud el estado del mismo.

3.1 Concepto: Error, Defecto y fallo

- **Error:** Un error es una accion humana o una idea equivocada que produce un resultado incorrecto. Los errores generan defectos.
- **Defecto:** El defecto es una imperfeccion que puede generar que uno o mas componentes no den los resultados esperados o trabajen de la manera correcta.
- **Fallo:** Un fallo es un resultado incorrecto o no esperado devuelto por un sistema defectuoso. El fallo tambien puede ser producido por una anomalia en el hambiente, como puede ser un afalla de escritura en disco.



Error, defecto y Fallo

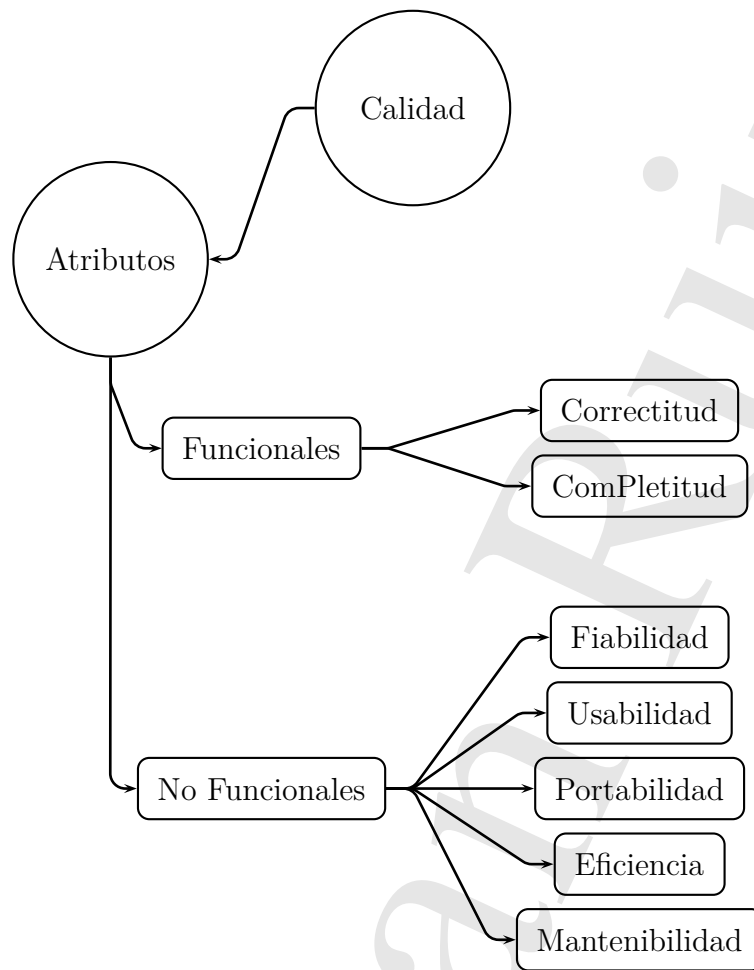
3.2 Mas conceptos: Requisito y Calidad

- **Requisito:** Es un atributo funcional definido por el cliente para el desarrollo solicitado. El conjunto de requisitos conforman la ERS o Especificacion de requisitos de Software.
- **Calidad:** El el nivel de satisfaccion del desarrollo con respecto al requerimiento del cliente. La calidad del software hace mencion al nivel de satisfaccion de la suma de todos los requerimientos respecto a las expectativas del cliente del software solicitado.

3.3 Atributos de la Calidad

La calidad esta determina por el cumplimiento de atributos los cuales podemoc clasificar en Funcionales y No Funcionales.

- **Atributos funcionales:** Son aquellos atributos que podria decirse nacen de los requerimientos y son determinantes en la calidad del software .
 - **Correctitud:** Este atributo determina que la funcionalidad satisface el requisitos solicitado.
 - **Compleitud:** Este atributo determina que el software satisface todos los requisitos solicitados por el cliente.
- **Atributos no funcionales:** Son atributos de linemientos generales y su cumplimiento en mayor o menor medida dependen de ciertos factores que conforman el software, como por ejemplo el tipo de proyecto, el entoro donde va a correr, etc. Para los inquietos, este punto es el de partida para investigar tecnologias como UX.
 - **Fiabilidad:** La fiabilidad es determinada por la estabilidad y operatividad del software e un lapso de tiempo determinado.
 - **Usabilidad:** La usabilidad es determinada por la practicidad de opercion y lo intuitivo y facil de aprender.
 - **Portabilidad:** La portabilidad refiere a la facilidad de instalacion y desisntalacion, configuracion y/o parametrizacion y a la simplisidad de poder reubicar el software por ejemplo entre servidores.
 - **Eficiencia:** La eficiencia es determinada por el rendimiento y los recurso utilizados en la opercion.
 - **Mantenibilidad:** Es determinada por el mayor o menor grado de esfuerzo para realizar cambios, actualizaciones, mejoras y/o correcciones en el software.



Atributos de la Calidad

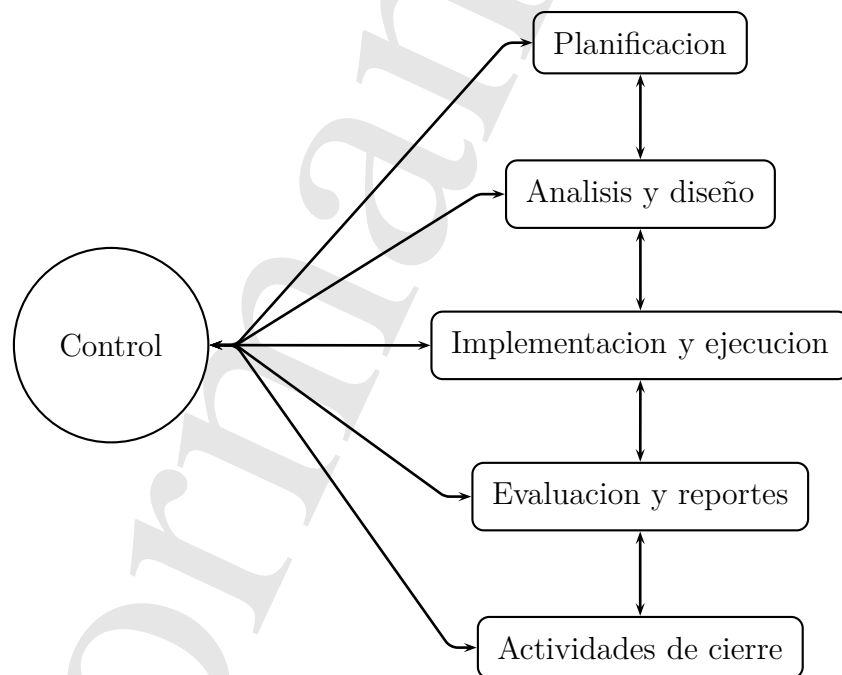
3.4 ¿Porque son necesarias las pruebas?

- **Por Que?:**
 - **Realizado por seres humanos:** Porque el desarrollo fue realizado por seres humanos, y los seres humanos se pueden cometer errores.
 - **Presion en las entregas:** Porque a medida que aumenta la presion en las entregas por que no alcanza el tiempo se empieza a asumir que todo estara bien, y se se pueden cometer errores.
 - **Medicion de la estabilidad:** Porque de las pruebas se puede medir la estabilidad o inestabilidad del software.
 - **Hace lo que debe Hacer:** porque principalmente debemos demostrar que hace lo que fue diseñado para hacer .
 - **Costos:** Porque es menos costoso encontrar fallas en la etapa de desarrollo que estando en productivo cuando el cliente esta operando el sotware.
- **Que obtenemos?:**
 - **Adquirir conocimiento:** Las pruebas nos permiten conocer el software y asi describir mejor el o los defectos y facilitar su correccion.
 - **Confirmacion de la Funcionalidad:** Las pruebas nos permiten confirmar la funcionalidad en base a los requisitos esesificados por el cliente.
 - **Generacion de informacion:** Las pruebas nos proporsionan la evidencia de posibles riesgos antes de que el software sea puesto en produccion.
 - **Confianza:** Las pruebas nos brindan la confianza de que el software cumple con la funcionalidad esperada por el cliente.
- **Cuando terminamos las pruebas?:** Para finalizar las pruebas debemo basaros en riesgos y prioridades lo cual depende de cada proyecto.
 - **Criterios de salida:** Un ejemplo de esto puede ser que despues de una determinada cantidad de pruebas no se evidencian mas defectos. Esto podria definirce como una condicion, entonces un conjunto de condiciones previamente definidas podria determinar un criterio de salida.
 - **Basado en riesgos:** El nivel de riesgo determina el grado en el cual se a probado, de quien es la responsabilidad en caso de fallos, la odcurrencia de fallos y aspectos relatios a factores economicos.
 - **Basado en plazo y Presupuesto:** Los recursos, tiempos y presupuesto son determinantes ya que de estas se asume que testear primero, que testeas mas y que no testear.

4 Proceso Fundamental de Testing

*E*l proceso de testing esta compuesto por varias etapas que veremos a continuacion, antes de empezar debemos resaltar que la etapa "Planificacion y control" tiene su propio nivel y es la primera, por esta debemos empzar, pero tambien se encarga de controlar y regular las demas etapas.

- Planificacion y control
- Analisis y diseño
- Implementacion y ejecucion
- Evaluacion de criterios de salida y reportes
- Actividades de cierre



Proceso del testing

- 4.1 Planificacion y control
- 4.2 Analisis y diseño
- 4.3 Implementacion y ejecucion
- 4.4 Evaluacion de criterios de salida y reportes
- 4.5 Actividades de cierre

5 Modelos de Desarrollo de Software

Norman Ruiz

6 Diseño de Pruebas

Norman Ruiz

7 Técnicas Estáticas

Norman Ruiz

8 Técnicas Dinámicas

Norman Ruiz

9 Criterios para seleccionar el diseño apropiado de caso de prueba

Norman Ruiz

10 Creación de Casos de Prueba

Norman Ruiz

11 Resumen Curso

Norman Ruiz

12 Para rendir el ISTQB Foundation Level

Norman Ruiz