



Nuestro compromiso es con el *futuro*.

**¡Les damos la
bienvenida!**

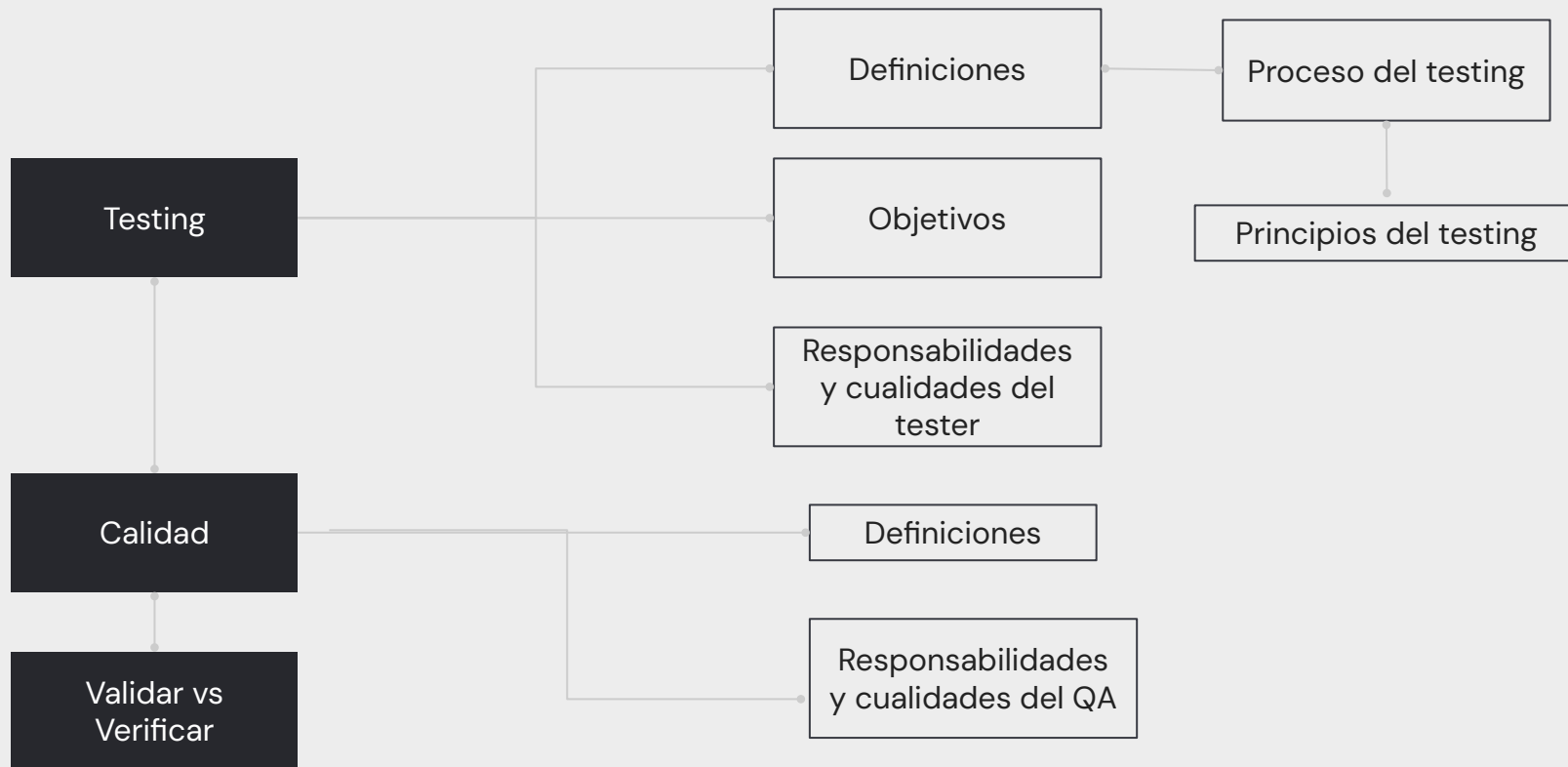
Introducción a Testing

Objetivos de la clase



- Adquirir conceptos generales y principios del testing y la calidad.
- Conocer sobre las cualidades y responsabilidades de un tester.
- Diferenciar entre Validar y Verificar.
- Proceso del testing.

Mapa de conceptos



¿Qué es el Testing?

Definición

El testing de software es una suma de metodologías, herramientas y procesos para probar y encontrar defectos en un sistema software o aplicación. El tester es el profesional que los lleva adelante.

Es un proceso destructivo para tratar de encontrar defectos (cuya presencia se asume) en el software.

Una percepción errónea común de la prueba es que sólo consiste en ejecutar pruebas. La prueba de software es un proceso que incluye muchas actividades diferentes, además de la ejecución, el proceso de prueba también incluye actividades tales como: **planificar la prueba, analizar, diseñar e implementar pruebas**, informar del avance y de los resultados de la prueba, y evaluar la calidad de un objeto de prueba

Objetivos

El objetivo principal del testing es aportar calidad del producto que se está verificando.

Otros objetivos son:

- Verificar que se cumplen los requerimientos funcionales y no funcionales.
- Validar si lo que se prueba funciona como debe, y cómo los usuarios y otros implicados esperan.
- Reducir la posibilidad de funcionamientos inadecuados del software.
- Generar confiabilidad en el nivel de calidad del programa.
- Prevenir defectos.
- Entregar información a todos los implicados.

- Validar si lo que se prueba funciona como debe, y cómo los usuarios y otros implicados esperan.



Desarrollador: Hace una UI intuitiva



Usuarios

Rompiendo mitos sobre el testing

- El Testing es probar que el software funciona.
- El Testing es una etapa que comienza al terminar el desarrollo.
- El tester es el enemigo del desarrollador.

TESTING = CALIDAD de PRODUCTO

TESTING = CALIDAD de PROCESO



Principios del testing

Existen 7 principios dentro del universo del testing, que siempre se registrarán, sin importar el proyecto en el que estés.

1. Las pruebas muestran la presencia de defectos, no su ausencia
2. Las pruebas exhaustivas son imposibles
3. Las pruebas tempranas ahorran tiempo y dinero
4. Los defectos se agrupan
5. Cuidado con la paradoja del pesticida
6. Las pruebas dependen del contexto
7. La ausencia de errores es una falacia



Las pruebas muestran la presencia de defectos, no su ausencia

Las pruebas pueden mostrar la presencia de defectos, pero no pueden probar que no existan.

Las pruebas reducen la probabilidad de que queden defectos no descubiertos en el software pero, incluso si no se encuentran, el proceso de prueba no es una demostración de la corrección.

Las pruebas exhaustivas son imposibles

Probar todo no es factible, excepto en casos triviales.

Por ejemplo, supongamos que tenemos un campo de entrada que acepta letras, caracteres especiales, y números del cero al 1000 No sería posible probar todas las combinaciones para cada tipo de entrada.

Las pruebas tempranas ahorran tiempo y dinero

Para detectar defectos de forma temprana, las pruebas tanto estáticas como dinámicas, deben iniciarse lo antes posible en el ciclo de vida de desarrollo de software.

La prueba temprana ayuda a reducir o eliminar cambios costosos.

Los defectos se agrupan

Un pequeño número de módulos, por lo general, contiene la mayoría de los defectos descubiertos durante la prueba previa a la entrega, o es responsable de la mayoría de los fallos funcionales. Las agrupaciones de defectos observadas en la prueba o producción son una aportación importante a un análisis de riesgos utilizado para centrar el esfuerzo de la prueba.

Cuidado con la paradoja del pesticida

Si las mismas pruebas se repiten una y otra vez, eventualmente estas pruebas ya no encontrarán ningún defecto nuevo.

Las pruebas ya no son efectivas para detectar defectos, de la misma manera que los pesticidas ya no son efectivos para matar insectos después de un tiempo.

Las pruebas dependen del contexto

Las pruebas se realizan de distintas maneras en diferentes contextos. Comprendiendo que el contexto va a indicar herramientas que pueden utilizarse para probar, ambiente o tipo de objeto que se prueba, metodología implementada por la empresa, o cualquier otro contexto que pueda existir.

La ausencia de errores es una falacia

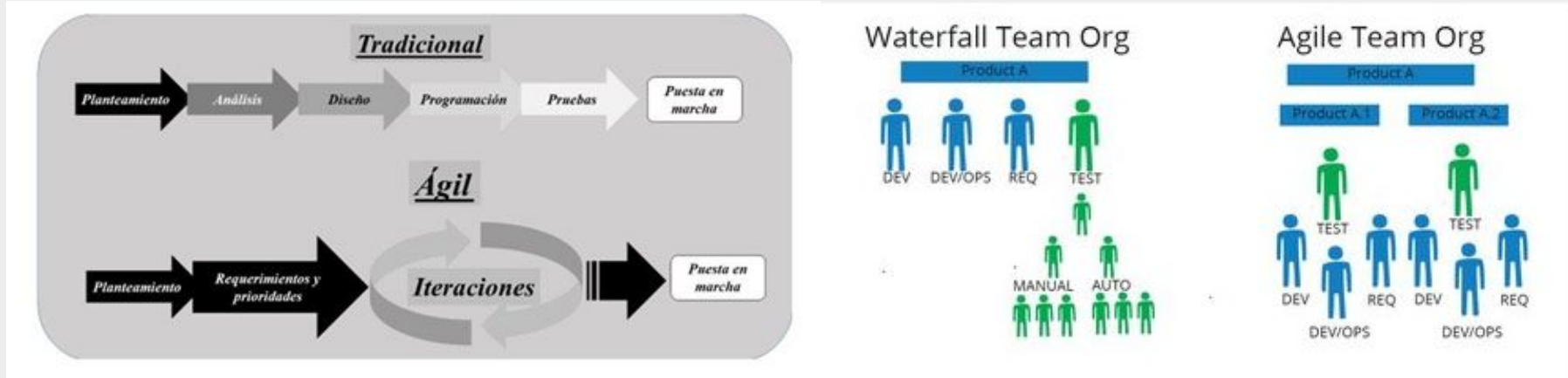
Algunas organizaciones esperan que los tester puedan realizar todas las pruebas posibles y encontrar todos los defectos posibles, pero esto no es posible. Además, es una creencia equivocada esperar que sólo con encontrar y corregir un gran número de defectos se asegure el éxito de un sistema.

Importancia del testing en el ciclo de vida del desarrollo de software

Un modelo de ciclo de vida de desarrollo de software describe los tipos de actividad que se realizan en cada etapa de un proyecto de desarrollo de software, y cómo las actividades se relacionan entre sí de forma lógica y cronológica. Hay diferentes modelos de ciclo de vida de desarrollo de software, cada uno de los cuales requiere **diferentes enfoques de prueba**



¿Cuándo probamos?



El alcance del trabajo del tester, recorre todas las etapas del desarrollo de software, ya que el testing está desde la creación de la documentación, hasta la implementación del producto. El tester siempre está realizando pruebas, y su alcance es completo al producto.

Comienzo del desarrollo

En etapas tempranas del desarrollo, nuestro trabajo va a ser el de leer documentación, participando de reuniones que decidan el funcionamiento para dar nuestro punto de vista experto, y también para hacer los pedidos de herramientas necesarias para hacer las pruebas.

Cuando los testers se ponen a hacer el trabajo de los analistas



Cierre del desarrollo

Una vez que los desarrolladores nos entregan el producto, y comienzan nuestras pruebas, estas se extienden hasta la salida del producto

Si algo funciona mal luego de la implementación, somos nosotros los testers, quienes debemos informar.



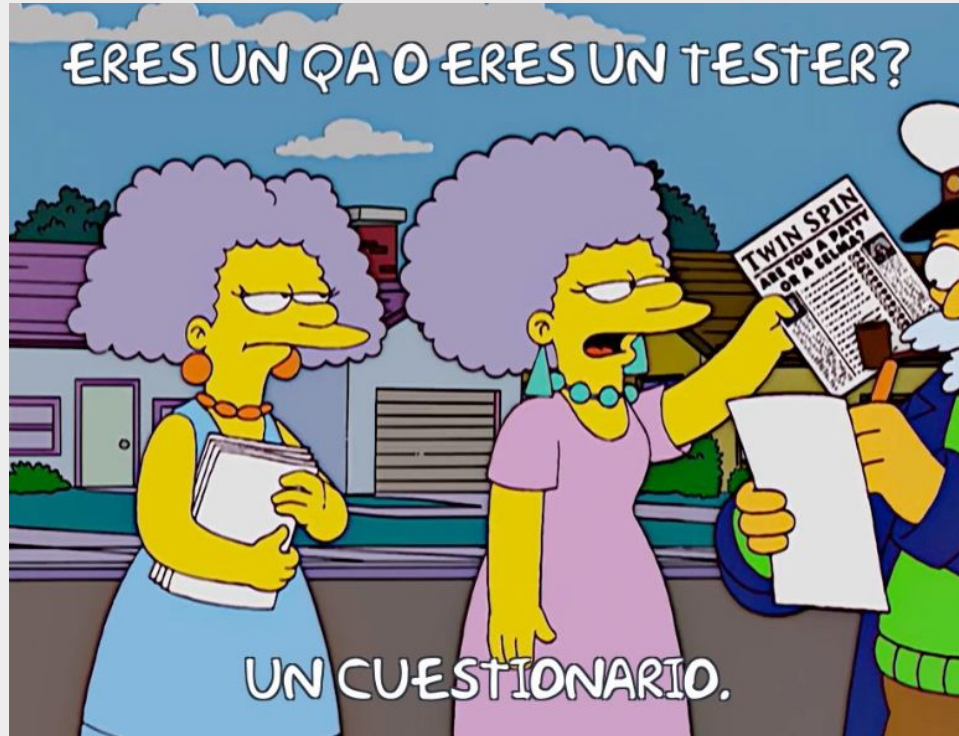
¿Qué es la Calidad?

Definición

La calidad se define como un grado de excelencia.

La idea de calidad de un usuario puede ser definida por la cobertura de ciertas funciones en el sistema como un buen desempeño del producto, su compatibilidad y adaptabilidad entre otras cosas. Muchas veces esos criterios no son señalados en ningún sitio, pero aun así se esperan en el sistema.

Tester (QC) y Tester (QA)



Aseguramiento de Calidad (QA)

QA (Quality Assurance) o Aseguramiento de Calidad, hace referencia a la forma de medir la calidad, no solo del producto, sino también del proceso de desarrollo. Las personas utilizan el término aseguramiento de la calidad (o simplemente QA por sus siglas en inglés) para referirse a la prueba, pero el aseguramiento de la calidad y la prueba no son lo mismo, sí están relacionados. Un concepto más amplio, la gestión de la calidad, los une. La gestión de la calidad incluye todas las actividades que dirigen y controlan una organización con respecto a la calidad. Entre otras actividades, la gestión de la calidad incluye tanto el aseguramiento de la calidad como el control de la calidad. El aseguramiento de la calidad se centra, por lo general, en el cumplimiento de los procesos adecuados, a fin de proporcionar la confianza de que se alcanzarán los niveles de calidad adecuados. Cuando los procesos se llevan a cabo de forma correcta, los productos de trabajo creados por esos procesos son generalmente de mayor calidad, lo que contribuye a la prevención de defectos. Además, el uso del análisis de la causa raíz para detectar y eliminar las causas de los defectos, junto con la aplicación adecuada de los hallazgos de las reuniones retrospectivas para mejorar los procesos, son importantes para un aseguramiento de la calidad eficaz.

Existen dos perfiles muy marcados en el mundo del software, los cuales son el Tester y el QA...

Tester QC

Es la persona que prueba software o proyectos similares en busca de errores, defectos o cualquier problema que pueda encontrar el usuario final. El papel de un tester es probar productos y proporcionar informes al equipo del proyecto sobre cualquier problema o mejora que el producto pueda requerir.

Tareas del tester

- Diseñar el plan de pruebas.
- Definir los casos de prueba en base a los requerimientos funcionales, no funcionales y técnicos.
- Test de integración: Definir las pruebas de Integración que se realizarán.
- Generar datos o lotes de prueba.
- Ejecutar los casos de prueba.
- Realizar la documentación de las pruebas (evidencia).
- Registrar los incidentes encontrados durante la fase de pruebas, y realizar su seguimiento para asegurar su adecuada corrección.
- Pruebas no funcionales (carga, estrés, performance, seguridad, etc).

Tester QA

Un QA (Quality Assurance) o analista QA es el profesional responsable de asegurar la calidad del software y de prevenir fallos en él. Es la persona encargada de garantizar el correcto funcionamiento del producto desde el primer momento, pero también de confirmar que satisface las expectativas de los usuarios que lo utilizarán.

Como su propio nombre indica, es un rol vinculado a la calidad. Un QA trabaja en conseguir tanto la calidad del producto como del proceso, esforzándose para que todas las fases del desarrollo de la aplicación se lleven a cabo eficientemente.

Tareas del QA

Las tareas del QA son las mismas que las tareas que realiza un tester más las siguientes:

- Diseño y preparación de la estrategia general de Calidad.
- Revisión de historias de usuario con el Product Owner.
- Revisión de criterios de aceptación (a partir de los cuales se generan escenarios de prueba).
- Identificación de dependencias entre las historias de usuario.
- Sugiere mejoras funcionales y de usabilidad.
- Identifica los escenarios y flujos más vitales del negocio.
- Colaboración junto con los stakeholders en las pruebas UAT.
- Preparación de ambientes de prueba.
- Seguimiento de releases.
- Automatización de pruebas funcionales.
- Creación y aplicación de métricas de calidad de Software.

Quality Assurance	Quality Control
QA significa Planificación para realizar un proceso.	QC Significa Acción para ejecutar el proceso planificado.
QA es un proceso que delibera en brindar garantía de que se logrará la solicitud de calidad.	QC es un proceso que delibera sobre el cumplimiento de la solicitud de calidad.
Un objetivo de QA es prevenir el defecto.	Un objetivo de QC es identificar y mejorar los defectos.
QA es la técnica de gestión de la calidad.	QC es un método para verificar la calidad.
El control de calidad no implica la ejecución del programa.	El control de calidad siempre implica la ejecución del programa.
Todos los miembros del equipo son responsables del control de calidad.	El equipo de prueba es responsable del control de calidad.
Ejemplo de asegurar la calidad: verificación y validación (no siempre)	Ejemplo de control de calidad: validación.
QA Define estándares y metodologías a seguir para cumplir con los requisitos del cliente.	QC asegura que se sigan los estándares mientras se trabaja en el producto.
QA es responsable del ciclo de vida completo del desarrollo de software.	QC es responsable del ciclo de vida completo de las pruebas de software





Nuestro compromiso es con el *futuro*.

BREAK - ¡10 MINUTOS Y VOLVEMOS!

Validar vs Verificar



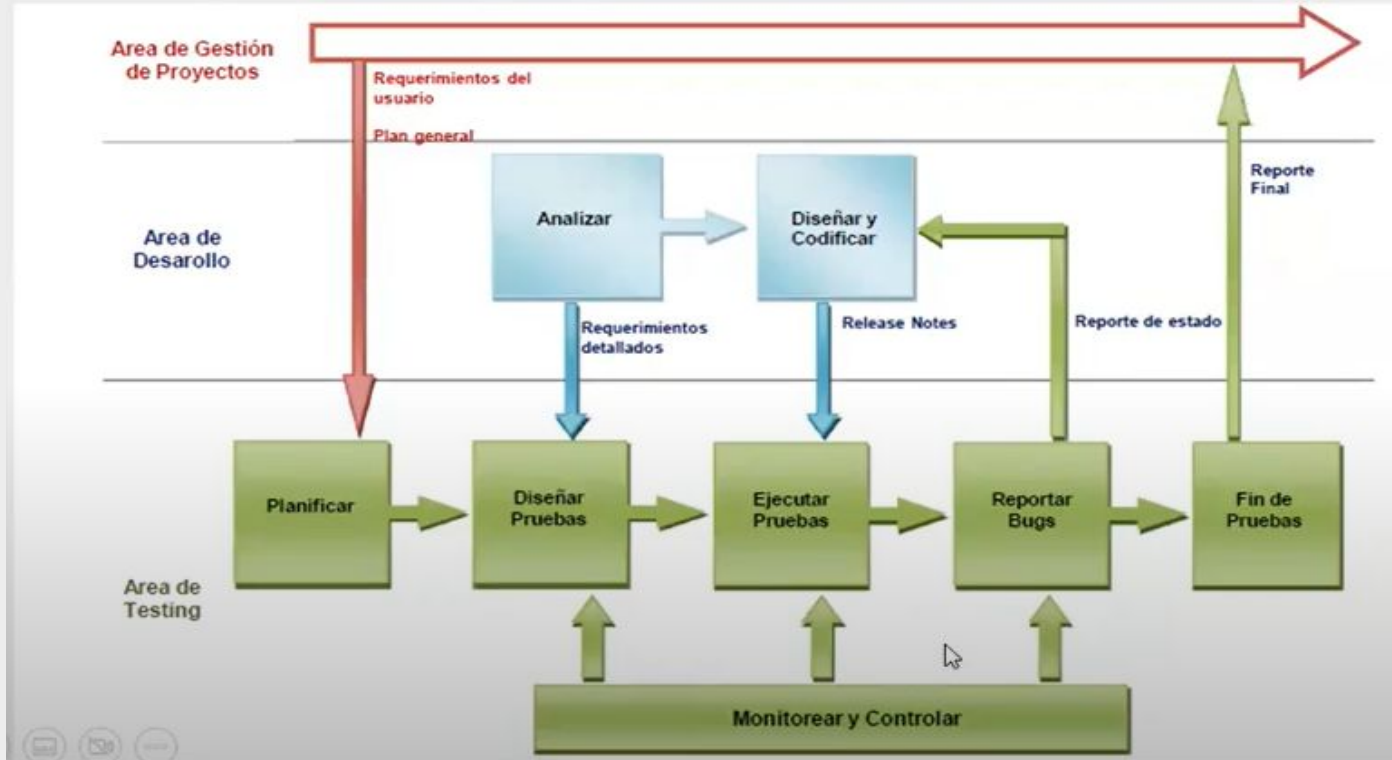
Validar

- ★ El software debería ajustarse a lo que el cliente realmente pidió.
- ★ Responde a la pregunta: ¿estamos construyendo el sistema correcto?

Verificar

- ★ El software debería ajustarse a su especificación.
- ★ Responde a la pregunta: ¿estamos construyendo el sistema correctamente?

Proceso de testing



Planificación y Control

El **Control** de la prueba implica tomar las medidas necesarias para cumplir los objetivos del plan de prueba (que puede actualizarse con el tiempo). Además en esta actividad se revisan los resultados de las pruebas y se toman decisiones según estos resultados.

La **Planificación** de las pruebas es la actividad de verificar que se entienden las metas y los objetivos del cliente, las partes interesadas (stakeholders), el proyecto, y los riesgos de las pruebas que se pretende abordar.

Identificación y Especificación

Se basa en las siguientes tareas:

- Revisión de la base de pruebas
- Verificación de las especificaciones para el software bajo pruebas
- Evaluar la testeabilidad de los requerimientos y el sistema
- Identificar los datos necesarios
- Diseño y priorización de los casos de las pruebas
- Diseño del entorno de prueba

Ejecución

La ejecución de la prueba incluye las siguientes actividades principales:

- Desarrollar y dar prioridad a nuestros casos de prueba
- Crear los datos de prueba
- Automatizar lo que sea necesario
- Creación de conjuntos de pruebas de los casos de prueba para la ejecución de la prueba eficientemente
- Implementar y verificar el ambiente
- Ejecutar los casos de prueba
- Registrar el resultado de la ejecución de pruebas y registrar la identidad y las versiones del software en las herramientas de pruebas
- Comparar los resultados reales con los resultados esperados

Evaluación y Reporte

Comprende las siguientes actividades:

- Evaluar los criterios de aceptación
- Reporte de los resultados de las pruebas para los stakeholders
- Recolección de la información de las actividades de prueba completadas para consolidar
- Verificación de los entregables y que los defectos hayan sido corregidos
- Evaluación de cómo resultaron las actividades de testing y se analizan las lecciones aprendidas

Soft Skills del Tester



Curiosidad

Más allá del conocimiento a la hora de probar, un tester debe sentir curiosidad por el producto para poder imaginar mejores planes de prueba.

Observación

El tester está siempre en los detalles que el resto del equipo no observa (**sesgo de confirmación**)

Comunicación

Dado que el tester es el informador de reportes o defectos, es importante tener un buen nivel de comunicación tanto oral como escrito para poder transmitir la información.

Empatía

Detrás de cada desarrollo, hay , al menos, una persona que trabajó en eso. Es importante tener empatía al momento de entregar información para cuidar las buenas relaciones con los pares.

Pensamiento lateral o razonamiento creativo

Viene junto con la creatividad para intentar resolver problemas buscando vías alternativas, siempre siendo imaginativo en los resultados.



¿Preguntas?

¡Muchas gracias!



Actividad

Pensar y comentar ejemplos de casos en los que la falta de testing ocasiono perdida de clientes y/o dinero.

Tiempo estimado: 10 minutos

Actividad

Enumerar todos los defectos y oportunidades de mejoras encontradas en el siguiente sitio web:
Demoblaze

<https://www.demoblaze.com/>

Tiempo estimado: 15 minutos



Nuestro compromiso es con el *futuro*.