



Universidad
Tecmilenio®



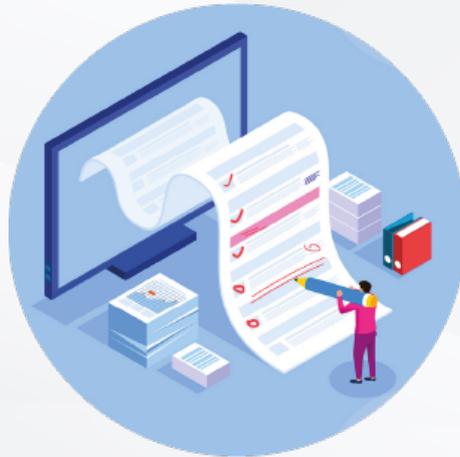


Software Testing

Introducción y procesos
de prueba

Semana 1





Te encuentras trabajando para una empresa desarrolladora de software. Tu líder te cita junto con tu equipo para platicar sobre el siguiente proyecto, el cual consiste en desarrollar un software para un nuevo hospital.

Tu equipo se encargará del desarrollo del módulo de pacientes, donde el usuario debe poder llenar todos los datos del paciente y, en cuanto haga clic en “Guardar”, estos se deben almacenar en una base.

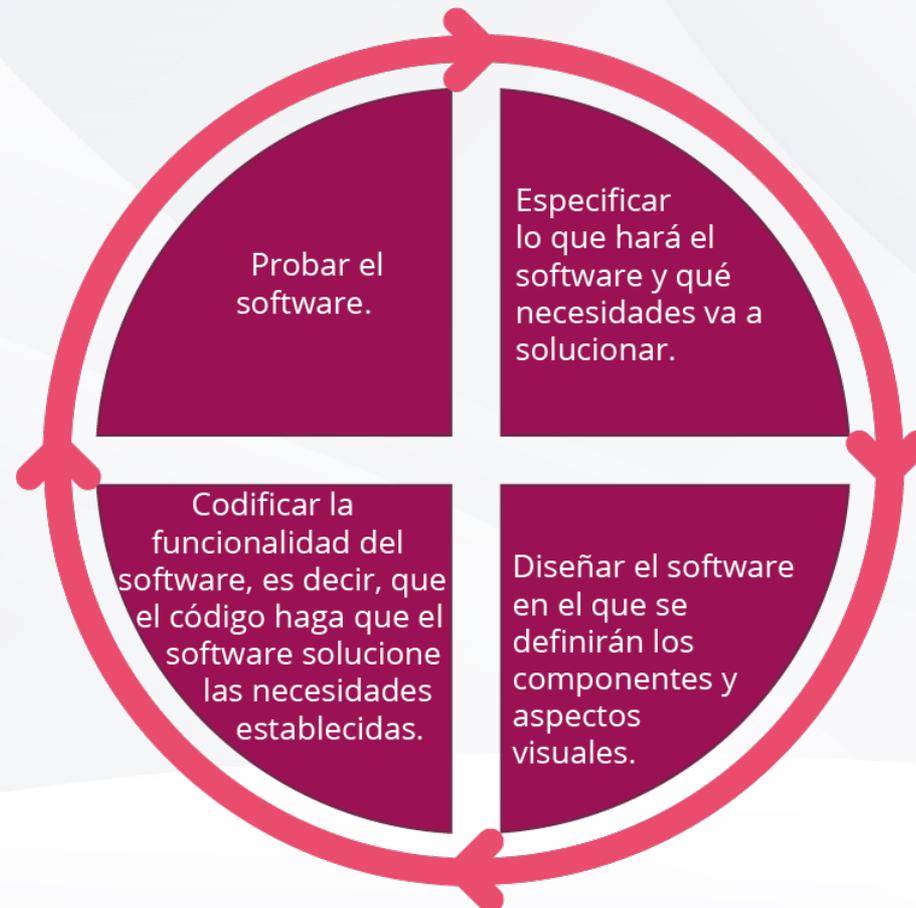
Tu jefe hace hincapié en que todo debe funcionar correctamente y tienes poco tiempo para hacerlo. Realizas el proyecto y, en cuanto terminas, informas a tu líder que está listo.

Él te pregunta: ¿realizaron las pruebas necesarias para asegurarse de que todo esté bien?

Tú respondes: todas las pruebas fueron aplicadas de acuerdo con el paso del ciclo de vida de desarrollo de software, conocido como *software testing*.

El software testing es un proceso en el cual se verifica si el producto tiene un correcto funcionamiento y si lleva a cabo el propósito por el cual fue definido desde un inicio.

El ciclo de desarrollo de vida de una prueba de software es el siguiente:



Conceptos básicos de software testing

Concepto	Definición
Accesibilidad	Grado de un sistema o componente que puede ser usado por el usuario para cumplir cierto propósito.
Anomalía	Condición que se desvía del resultado esperado.
Compatibilidad	Grado de un componente o sistema que puede intercambiar información con otros componentes o sistemas.
Componente	Parte de un sistema que puede probarse o revisarse por separado.
Defecto	Imperfección o deficiencia en un producto.
Disponibilidad	Grado de un componente o sistema que está en funcionamiento y es accesible en el momento que se necesite.



Conceptos básicos de software testing

Concepto	Definición
Failed	Es el resultado de una prueba que no hace match con el resultado esperado.
Emulador	Software utilizado en las pruebas que imita el comportamiento del hardware.
Failure	Es un evento donde un componente o sistema no realiza una función dentro de los límites especificados.
Terminación anormal	Es la terminación sin intención de un componente o sistema al momento de ejecutarse.

Principios de las pruebas de software

- Las pruebas muestran la presencia de defectos, no su ausencia.
- Las pruebas exhaustivas son imposibles.
- Las pruebas tempranas ahorran tiempo y dinero.
- Agrupación de defectos.
- La paradoja del pesticida.
- Las pruebas dependen del contexto.
- La falacia de ausencia de errores.

¿Cuáles son las consecuencias de no realizar pruebas de software?

¿Consideras que las pruebas aportan beneficios o inconvenientes?



Las pruebas de software son un paso básico dentro del ciclo de vida para el desarrollo de un software, ya que permitirán minimizar el riesgo de un mal funcionamiento, ahorrando tiempo y dinero.

Recuerda, las pruebas son esenciales para que el software tenga éxito y cumpla con todos los requisitos.





Software Testing

Pruebas a través del ciclo
de vida de desarrollo de
software



Semana 1



El desarrollo de un producto se debe realizar mediante alguna metodología o ciertos pasos.

El software no es la excepción a la regla, por lo cual, se definió el ciclo de vida de desarrollo de software, que contiene los pasos a seguir para asegurar que este tenga el resultado esperado por el cliente.

El ciclo de vida de desarrollo de software brindará la oportunidad de realizar un proceso dividido en diferentes etapas, asegurando el correcto desarrollo del software, ya que en cada etapa podrás verificar y tener la certeza de realizar lo solicitado por el cliente.



Ciclo de vida de desarrollo de software

Planificación

- Antes de comenzar a diseñar un software, necesitas entender las necesidades del cliente y cuál es el problema que el software debe solucionar.

Análisis

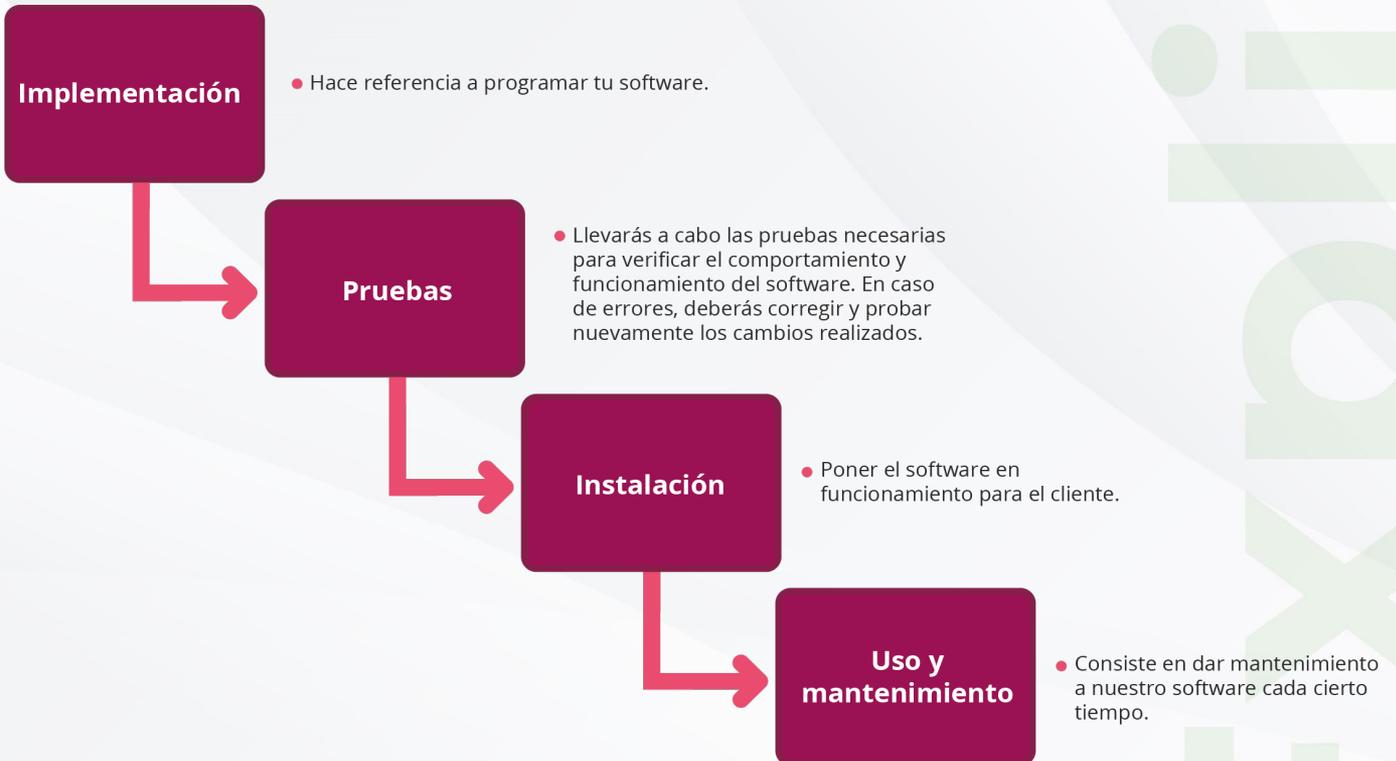
- Te ayudará a saber exactamente la función que debe hacer el software que vas a desarrollar.

Diseño

- Para el diseño del software, debes tomar en cuenta su estructura general, los módulos, el acceso a la información, etc.



Ciclo de vida de desarrollo de software



Modelos del ciclo de vida de desarrollo de software



Cascada.

Con componentes.

En V.

Sashimi.

Tipos de pruebas

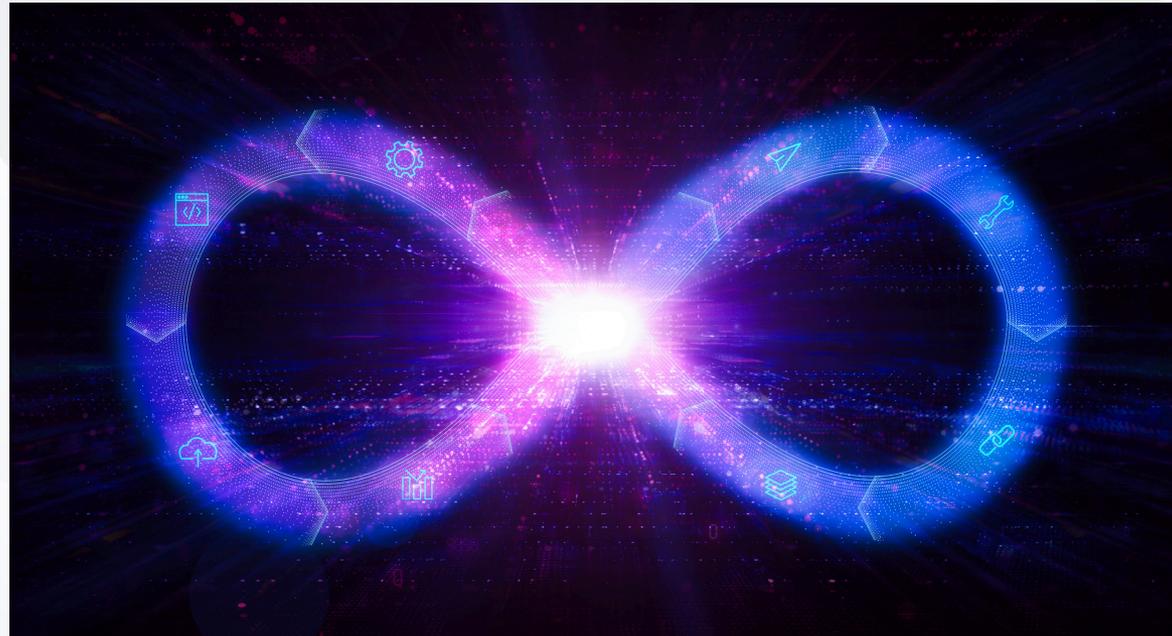
Estáticas

- Análisis estático.
- Revisiones.

Dinámicas

- Es necesario llevar a cabo la ejecución del software para determinar si todo funciona correctamente.





¿Qué paso consideras más importante dentro del ciclo de vida de desarrollo de software?

¿Puedes obviar alguno?

Actividad



Todo software debe seguir un proceso. Cada uno de los pasos del ciclo de vida de desarrollo de software es muy importante, ya que uno complementa al otro, y si uno no se encuentra definido, el software tendrá inconsistencias y su funcionamiento no será el esperado.

Una vez que llegues al paso de las pruebas, es necesario que sepas cuál es el proceso y qué técnicas debes utilizar.





Software Testing

Niveles y tipos de pruebas



Semana 1



El software que desarrollarás tendrá sus propias particularidades. Si realizas la prueba de otro software que hayas desarrollado, es muy probable que no obtengas los resultados esperados.

Es importante que conozcas los diferentes tipos y niveles de pruebas que se pueden aplicar al desarrollo y lo que evalúan.

Sin este conocimiento, las pruebas aplicadas pueden resultar no eficientes y no estarías asegurando un funcionamiento correcto del software.



Niveles de pruebas

Pruebas de componentes o unitarias

- Comprueban el correcto funcionamiento de los componentes, unidades o módulos.

Integración

- Corroboran que la interacción de los diferentes componentes, módulos o sistemas se realice de forma correcta.



Niveles de pruebas

Sistemas

- Se centran en el comportamiento del sistema y sus capacidades. Se observan desde la perspectiva del usuario.

Aceptación

- Se realiza por el cliente.
- Aceptación de usuario.
- Aceptación operativa.
- Aceptaciones contractuales y normativas.
- Alfa y Beta.



Tipos de pruebas

Pruebas funcionales

- Verifican que las necesidades del cliente se implementen en el software.
- Verifican la funcionalidad del software y su usabilidad.
- Garantizan que el software se comporte de acuerdo con lo definido.

Pruebas no funcionales

- Validan la operación del software.
- No se centran en la funcionalidad, sino en obtener información del software, como su rendimiento, estabilidad y qué tanto se puede explotar.

Pruebas estructurales

- Verifican la estructura del código y que este no tenga defectos.
- Se puede aplicar la prueba de caja blanca.

Pruebas asociadas al cambio

- Se utilizarán para verificar que, si haces algún cambio al código, compruebes que la funcionalidad del software sigue al 100%.





Actividad



¿Consideras que los niveles de pruebas son suficientes?

¿Qué es más fácil: probar todo de un software o ir poco a poco?

Gracias a los niveles de pruebas es más fácil asegurar que el software funcione de forma correcta, esto debido a que comienzas con pruebas individuales (módulos o componentes) y terminas con las pruebas de todo el software.

Si divides un problema en diferentes partes, es más fácil y rápido encontrar una solución.

Lo mismo pasa con las pruebas, pues si aseguras que las partes más pequeñas funcionan bien, conforme vayas agregando más elementos a probar, será más fácil, ya que vas de poco a poco, es decir, de lo particular a lo general.

