



Software Testing desde 0 a Experto

Para personas sin conocimientos técnicos

Software Testing desde 0 a experto

Para personas sin conocimientos técnicos

Bienvenid@ al curso

Presentación del tutor



MÓDULOS

- 01** TU CARRERA COMO TESTER EN LA INDUSTRIA DE LA TECNOLOGÍA
- 02** FUNDAMENTOS DEL PROCESO DE DESARROLLO DE SOFTWARE AGIL
- 03** PASO A PASO -FUNDAMENTOS DEL TESTING Y EJEMPLO PRÁCTICO
- 04** PASO A PASO- HERRAMIENTAS BÁSICAS PARA EL TESTING
- 05** PASO A PASO – AUTOMATIZACION DE PRUEBAS
- 06** COMO BUSCAR TRABAJO COMO TESTER



TU CARRERA COMO TESTER EN LA INDUSTRIA DE LA TECNOLOGÍA

¿Que es el Software Testing?

Y porque es tan importante

El **Testing** es una disciplina en la ingeniería de software que se realiza mediante una metodología de verificación y validación cuyo principal objetivo es localizar los errores en un **software** para ser corregidos por el equipo de desarrollo y de esta manera aumentar la confianza en el nivel de calidad del sistema



Responsabilidad del Tester / QA

MÓDULO 1
LECCIÓN # 2



Que habilidades necesita un Tester?

MÓDULO 1



Tester or Analista QA



Pensamiento Lógico

Debe saber como desglosar un Sistema en unidades más pequeñas para poder crear casos de prueba



Muy buena comunicación

Excelente comunicación verbal y escrita para comunicar los errores y documentarlos.



Atención a detalles, curiosidad y sentido común

Se requiere tener un pensamiento crítico con atención a detalles desde el punto de vista de u usuario final



Ser organizado y metódico

Esto es clave para poder ejecutar los casos de prueba en un orden y poder encontrar la mayor cantidad de errores



Apasionado por la tecnología

Todo Tester tiene que tener un gusto por la Informática, ser curioso y creativo



Paciencia y persistencia

El Desarrollo de software es un proceso que requiere flexibilidad y mucha paciencia.

¿Puedo trabajar como Tester Sin una carrera en IT o relacionada con Informática o Sistemas?

Si, tener una carrera o profesión relacionada con tecnología ayuda pero no es indispensable. Para muchas empresas hoy en día no importa que carrera tengas, importa mas tus skills (habilidades), experiencia y como puedas plasmarlos en una hoja de vida y demostrarlos en una entrevista .

FUNDAMENTOS DEL PROCESO DE DESARROLLO DE SOFTWARE

Paso a Paso

Ciclo de vida del Desarrollo de Software (SDLC) + ejemplo

Modelo de Desarrollo en Cascada



Desventajas modelo Cascada

- No permite cambios de requisitos
- Ningún producto funcional hasta casi la finalización del proyecto
- En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo..
- El usuario final no se integra en el proceso de producción hasta que no termina la programación.

<https://blog.comparasoftware.com/modelo-de-desarrollo-en-cascada/>

Diapositiva 11

E1

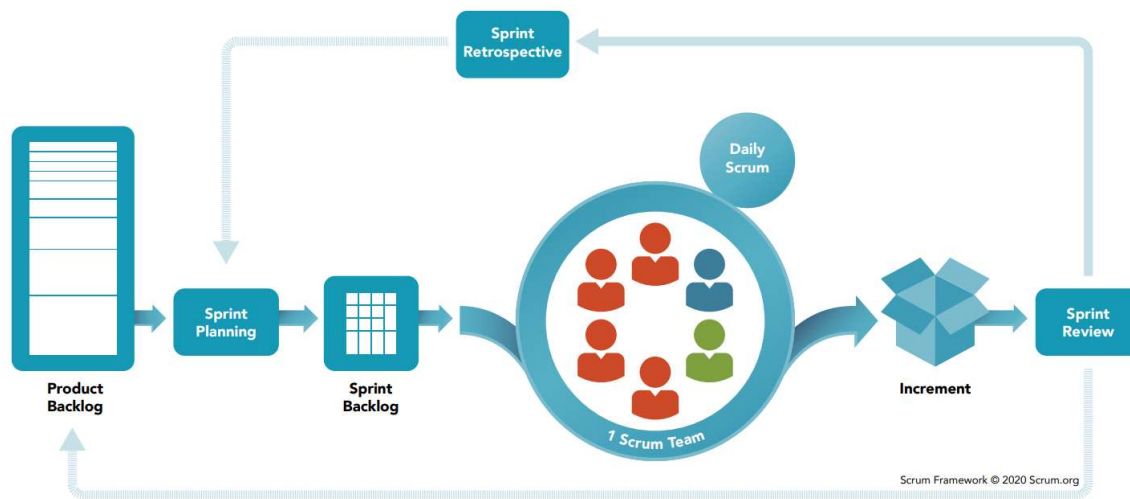
Esteban; 11/10/2021

Metodologías de Desarrollo Ágil

MÓDULO 2

Modelo Desarrollo “Scrum”

SCRUM FRAMEWORK



Ventajas de Scrum

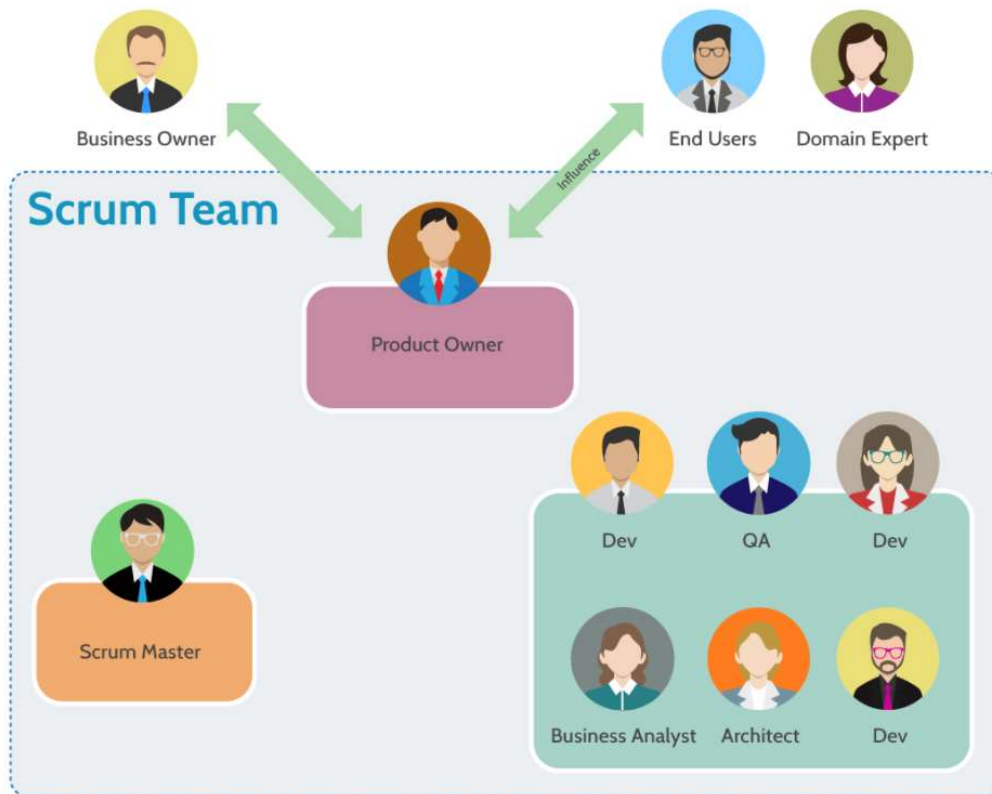
- Flexibilidad y adaptación a un mercado cambiante.
- Resultados anticipados
- Obtención de un producto mínimo viable (MVP) ...
- Feedbacks rápidos y precisos. ...
- Fecha de entrega de proyecto realista. ...
- Rápido aprendizaje del equipo. ...
- Autonomía y responsabilidad.

Otros marcos ágiles

Kanban, Scrumban, Lean, XP, Híbridos

Fuente: <https://www.scrum.org/resources/scrum-framework-poster>

Roles en Scrum (Ejemplo Amazon)



Fuente: <https://www.tutorialscampus.com/agile/scrum-roles.htm>

- ❖ **Scrum Master:** Es la persona experta que ayuda al equipo a trabajar con el marco de scrum, es un facilitador y ayuda a resolver impedimentos, no es un PM
- ❖ **Dueño de Producto (PO):** Es una persona que representa la voz del cliente o el negocio dentro del equipo de Desarrollo, es el encargado de indicar que se debe de hacer y controla el flujo del trabajo (Backlog)
- ❖ **Equipo de Desarrollo:** Compuesto por desarrolladores (Dec), Analistas de requisitos (BA), testers (QA) , Arquitectos

Eventos de Scrum (Ejemplo Amazon)

MÓDULO 2



Fuente: <https://www.invensislearning.com/blog/what-are-scrum-ceremonies/>

- ❖ **Planeacion Sprint:** Es la primera reunión que se realiza al inicio del sprint, sirve para seleccionar los ítems en los que se va a trabajar y como se van a hacer
- ❖ **Reunion Diaria (Daily):** Es una reunión diaria de 15 minutos en la que cada miembro del equipo de Desarrollo da un update de lo que esta haciendo como Tambien que impedimentos tiene
- ❖ **Revision Sprint:** Es una reunión que ocurre al final del sprint donde el PO y el equipo presentan a los usuarios (stakeholders) el incremento terminado del product para su inspección y adaptación.
- ❖ **Retrospectiva:** Reunion que ocurre al final del sprint donde se hace una reflexión del sprint y se discuten oportunidades de mejora para el próximo sprint

FUNDAMENTOS DEL TESTING

Paso a Paso con ejemplos



7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 1.

Ejecutar pruebas nos muestra la presencia de defectos... pero no pueden probar que NO los hay.

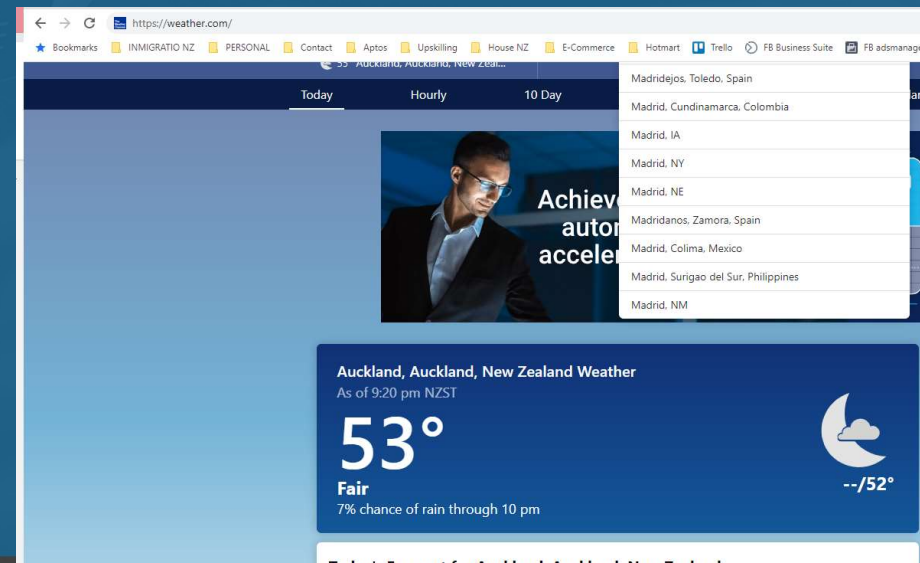


7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 2

El testing exhaustivo es imposible

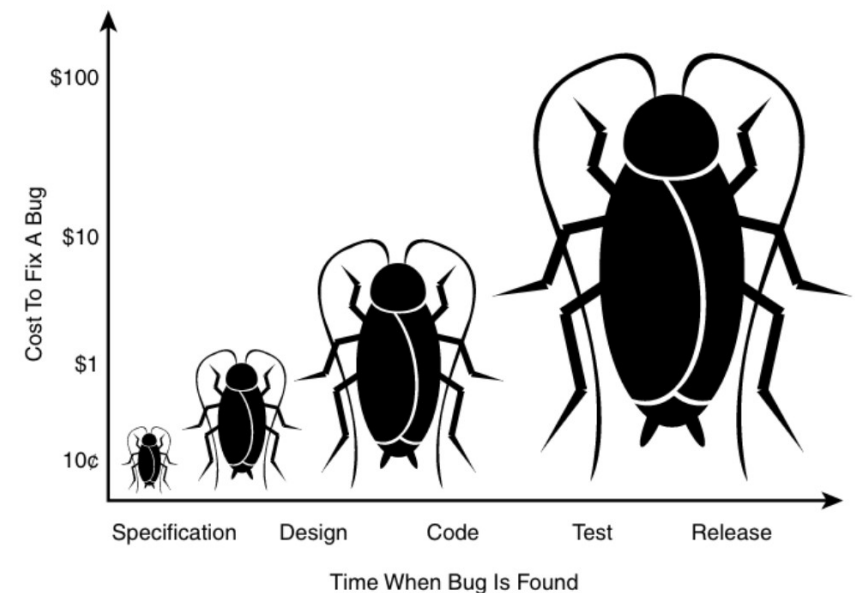


7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 3

La verificación de la calidad de un sistema debe empezarse lo antes posible



7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 4

La mayoría de defectos relevantes suelen concentrarse en un grupo muy determinado de módulos de nuestro producto.



7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 5

La paradoja del pesticida



Software Tester

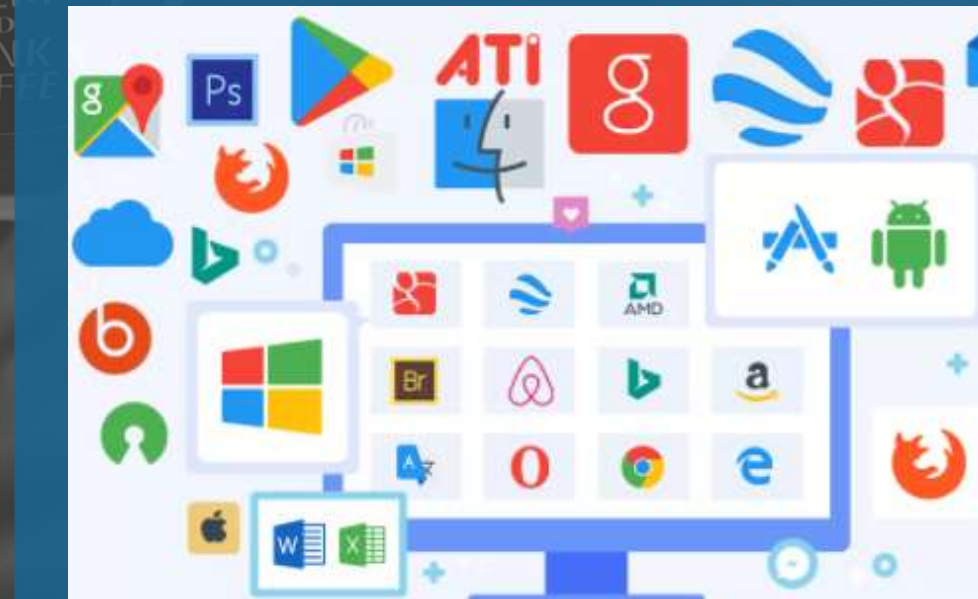
Software Bug

7 principios del Software Testing

MÓDULO 3

PRINCIPIO # 6

El testing es totalmente dependiente del contexto

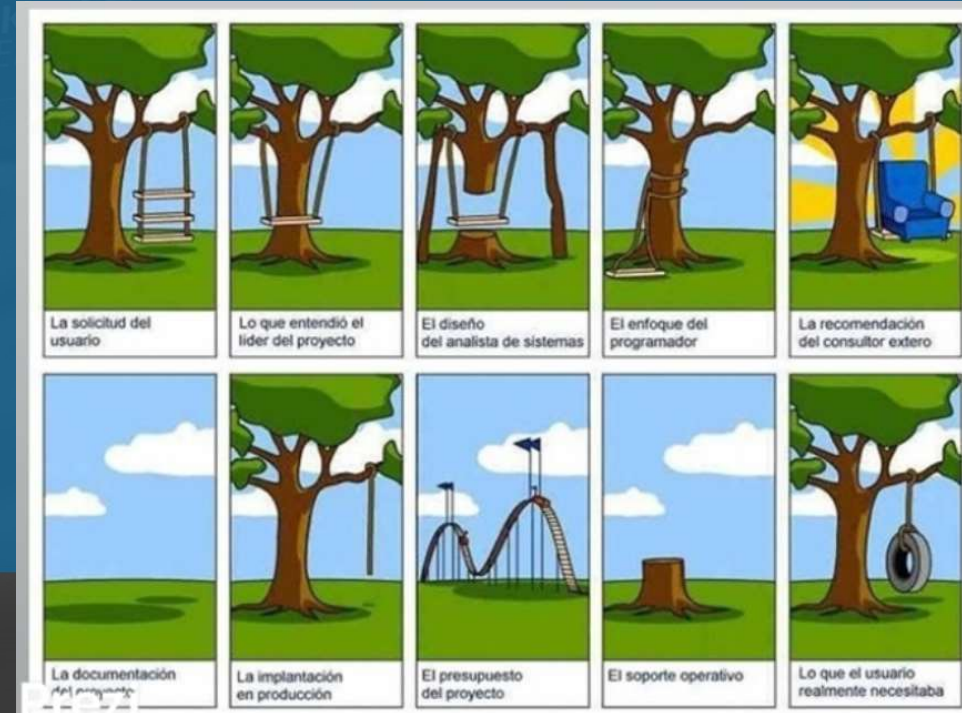


7 principios del Software Testing

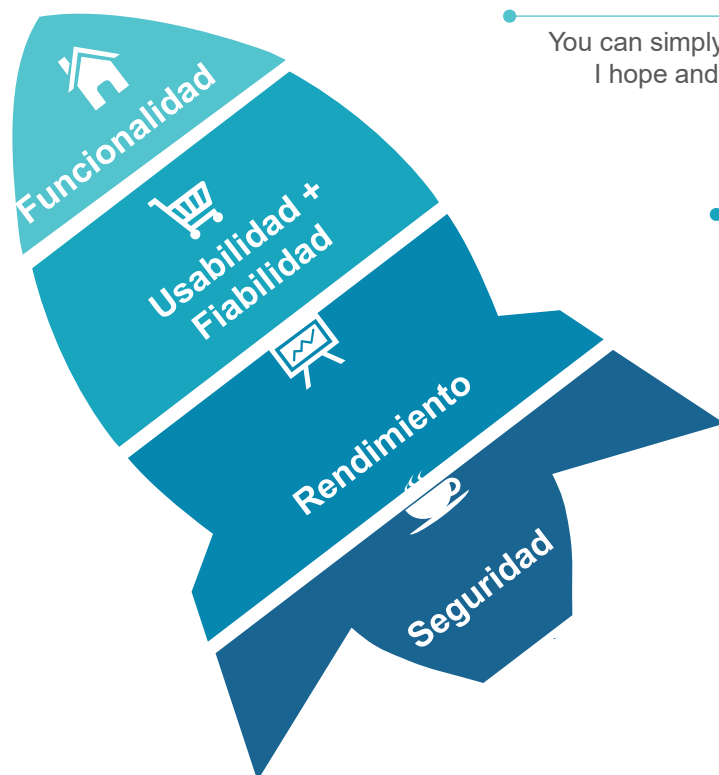
MÓDULO 3

PRINCIPIO # 7

La falacia de ausencia de errores



Atributos de calidad



FUNCIONALIDAD



You can simply impress your audience and add a unique zing and appeal to your Presentations. I hope and I believe that this Template will your Time, Money and Reputation. Get a modern PowerPoint Presentation that is beautifully designed.

USABILIDAD



La facilidad con la que el sistema o componente se puede utilizar o bien aprender a utilizar.

RENDIMIENTO + SEGURIDAD



La performance es una medida de la eficiencia en el uso de recursos del sistema ejecutándose, por ejemplo: Uso de procesador, Memoria , Almacenamiento permanente (discos rígidos, etc).
Uso de redes , o cualquier otro recurso físico.

SEGURIDAD



Algunas visiones de la seguridad son: Comprobar la identidad de las personas que intentan acceder al sistema., Garantizar que sólo las personas específicamente autorizadas pueden ver determinada porción de la información del sistema . Garantizar que sólo las personas específicamente autorizadas pueden modificar determinada porción de la información del sistema o bien realizar determinadas acciones

Pruebas Ágiles (Agile testing)

MÓDULO 3

Agile Testing es una práctica de pruebas de software que sigue los principios del desarrollo ágil de software.



Agile tester durante el sprint

- ✓ Ayudar al equipo a la revisión de común entendimiento de requisitos.
- ✓ Revisión que los programadores incluyan pruebas unitarias (pruebas en el código), creación de datos de prueba y ambientes de prueba
- ✓ Verificación y pruebas de los criterios de aceptación + pruebas exploratorias de la página web. Guiar al resto del equipo a ayudar a hacer pruebas
- ✓ Verificación de la liberación de la página y sanity tests.

Plan de pruebas de un Proyecto de Software

MÓDULO 3

- ❖ Sirve para definir la estrategia de las pruebas y guía al equipo como probar la aplicación.
- ❖ Define el QUÉ?, COMO?, DONDE? QUIEN?
- ❖ **Depende mucho de la complejidad del Proyecto, por lo general para sprints cortos y con alcance muy definido un plan de pruebas lite o ágil es más que suficiente.**
- ❖ Lo realiza el tester al principio del sprint en conjunto con el equipo de Desarrollo.
- ❖ Debe estar al alcance de todos



PLAN DE PRUEBAS ÁGIL

ALCANCE DE LAS PRUEBAS

TIPOS DE PRUEBA / ESTRATEGIA

AMBIENTE DE LA PRUEBA

DATOS DE LA PRUEBA

PLAN DE RIESGOS

Casos de prueba

MÓDULO 3

- Un **caso de prueba** es un conjunto de condiciones o variables bajo las cuales un tester determinará si una [aplicación](#), un sistema software o una característica de éstos es parcial o completamente satisfactoria.
- Generalmente los casos de prueba se diseñan a partir de los criterios de aceptación definidos en las historias de usuario, sin embargo si no se tienen dichos requisitos entonces los casos se escriben basados en la operación normal de un programa similar.



ESTRUCTURA CASO DE PRUEBA

IDENTIFICADOR DEL CASO

DESCRIPCION DEL CASO DE PRUEBA

PRECONDICIONES

DATOS DE ENTRADA

PASOS

RESULTADOS ESPERADOS

RESULTADOS OBTENIDOS

PRIORIDAD

ESTADO

Combinaciones de casos de prueba

MÓDULO 3

PRUEBA POSITIVA

La prueba del camino feliz (HAPPY PATH), o prueba positiva, es un tipo de prueba que se diseña para probar el resultado satisfactorio de un requisito tal cual fue especificado en los criterios de aceptación.



PRUEBA NEGATIVA

Es una prueba diseñada para romper el flujo normal o esperado de un requerimiento, se utilizan distintos datos de prueba o cambios en alguna variable para probar un camino alternativo del sistema.



Que es un bug?

MÓDULO 3

- ❖ En términos informáticos se refiere a un **error o defecto en el software** que hace que un programa no funcione correctamente.
- ❖ La mayoría de los bugs se originan por **errores de quienes desarrollan el código de un software o sistema**, al diseñar éstos o al no ser compatibles con otros programas o hardware.



ESTRUCTURA BÁSICA DE UN BUG

IDENTIFICADOR DEL BUG (#)

TITULO DEL ERROR

REPORTADO POR

DESCRIPCIÓN DEL ERROR

PASOS DE REPRODUCCIÓN

RESULTADO ESPERADO

RESULTADO OBTENIDO

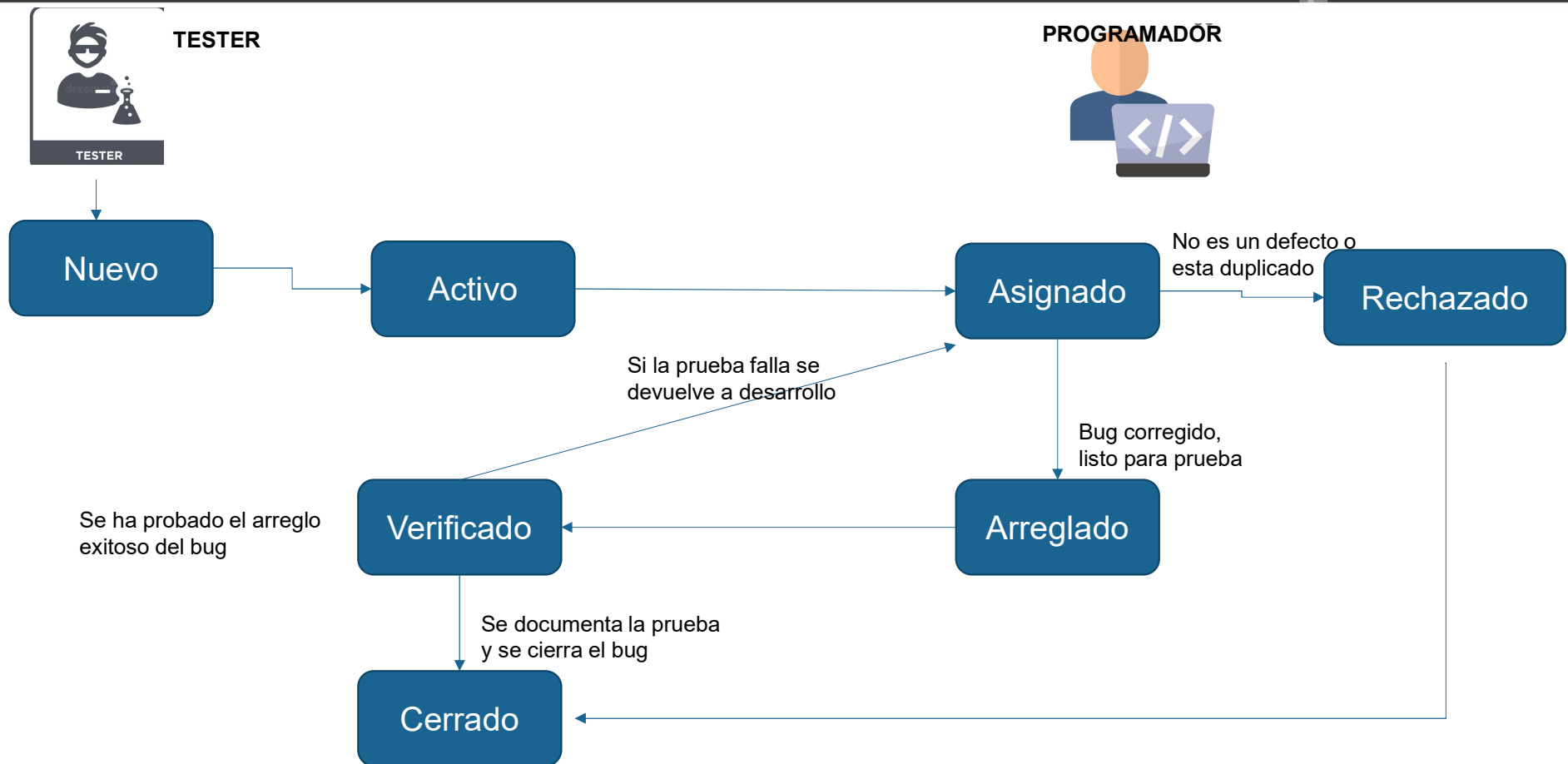
IMPRESIÓN DE PANTALLA/VIDEO ETC

SISTEMA OPERATIVO/BROWSER

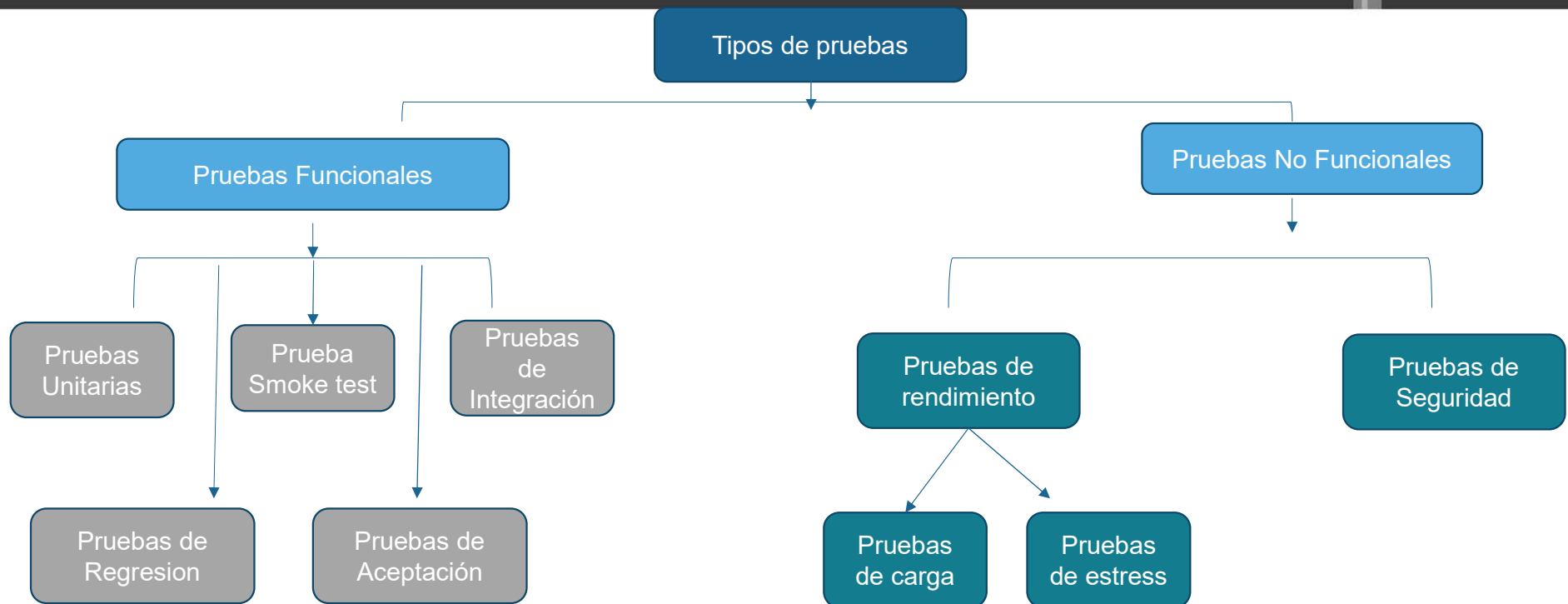
SEVERIDAD / PRIORIDAD

Ciclo de vida de un bug

MÓDULO 3



Tipos de pruebas

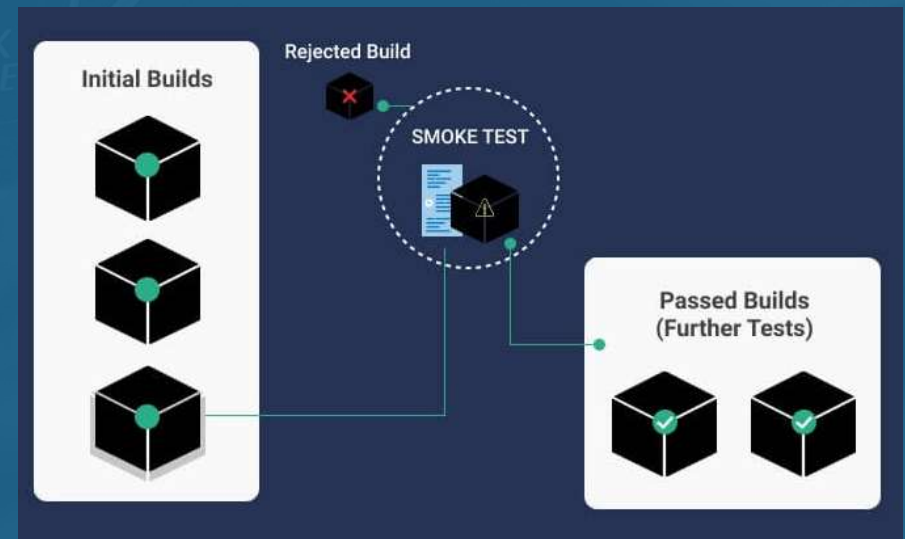


Pruebas Funcionales

MÓDULO 4

Smoke Tests

El smoke test se realiza después de ejecutar una build del software para asegurarnos de que ciertas funcionalidades básicas y críticas del programa funcionan correctamente.



Pruebas Funcionales

MÓDULO 4

Pruebas de Regresión

La prueba de regresión consiste en probar un sistema que ha sido analizado previamente para asegurar que no se haya introducido algún tipo de defecto como resultado de cambios realizados.

Regression:
“when you fix one bug, you introduce several newer bugs.”



Pruebas Funcionales

MÓDULO 4

Pruebas de Aceptación (UAT)

También conocida como **prueba** beta o de **usuario** final, se define como la **prueba** del software por parte del **usuario** o cliente para determinar si puede ser aceptado o no. ... El objetivo principal de esta **prueba** es validar el software frente a los requisitos comerciales.

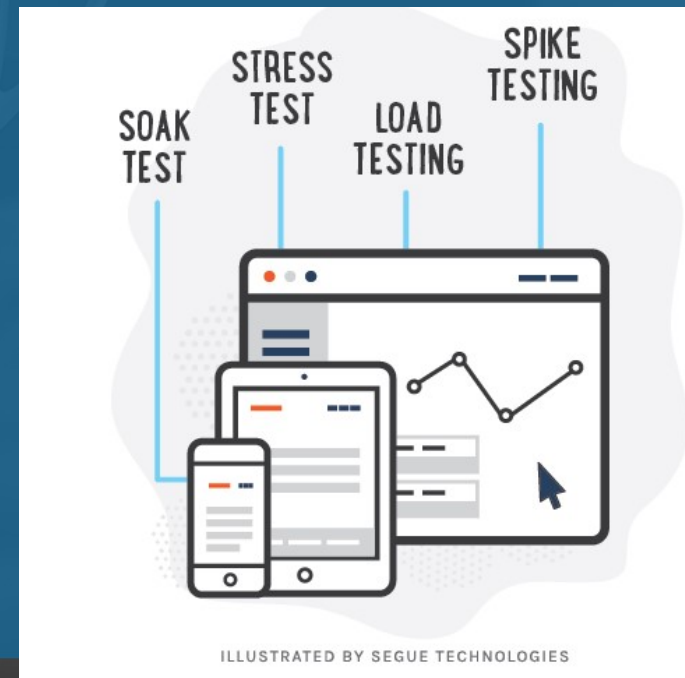


Pruebas No Funcionales

MÓDULO 4

Pruebas de Rendimiento (Performance)

Es una técnica de **prueba** de software no funcional **que** determina cómo la estabilidad, la velocidad, la escalabilidad y la capacidad de respuesta de una aplicación se mantiene bajo una determinada carga de trabajo.

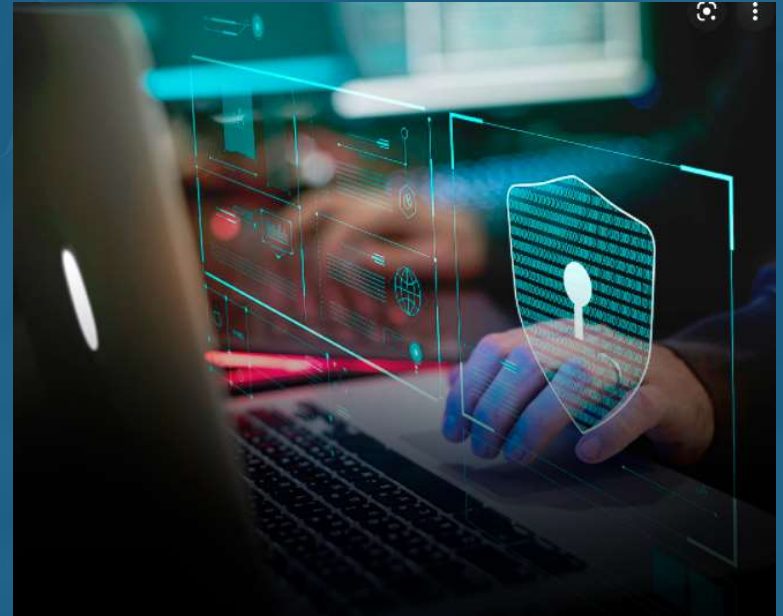


Pruebas No Funcionales

MÓDULO 4

Pruebas de Seguridad

Las **pruebas de seguridad** se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web, buscando disminuir el impacto de ataques a ellas y pérdida de información importante



HERRAMIENTAS BÁSICAS PARA TESTING

Instalación y ejemplos

Herramientas básicas en Testing



Capturas de pantalla

Lightshot (Fotos) , Screencastify (Videos)



Gestor de incidencias / Scrum

JIRA , Azure DevOps, Trello



Gestor de casos de prueba

TestRail, Zephyr, TestLodge, TestLink,
SpiraTest



Automatización de pruebas / DevOps

Selenium IDE, Visual Studio code(editor),
Node JS, Cypress (Automatización),
Selenium, TestCafe, SOAPUI, Postman (API
testing) , Jenkins, Bamboo,

AUTOMATIZACIÓN DE PRUEBAS

Paso a Paso con ejemplos

Pruebas manuales vs Automatizadas



MÓDULO 5

Las pruebas manuales, son ejecutadas directamente por uno o más testers, simulando las acciones del usuario final, apoyándose de las herramientas necesarias.

Las pruebas automatizadas son pruebas que son programadas por desarrolladores o testers por medio de scripts y programas, simulando lo que el usuario hace en el sistema pero ejecutado automáticamente en un laptop o un servidor de pruebas.

Pruebas Manuales



MÓDULO 5

Ventajas

- ✓ Para proyectos cortos son más rápidas.
- ✓ Son más flexibles
- ✓ Al ser ejecutadas por un humano pueden detectar problemas o bugs no funcionales que no son detectados por un programa como lo es la usabilidad o la navegabilidad o experiencia de usuario.
- ✓ No requieren de conocimiento técnicos profundos o programación aunque sería idea.
- ✓ Permiten probar escenarios complejos con secuencias largas de pasos en sistemas antiguos difíciles de automatizar.

Desventajas

- Se pueden volver una actividad repetitiva y tediosa cuando se trata de proyectos grandes.
- Cuando el proyecto tiene muchos bugs, las repruebas y pruebas de regresión se vuelven muy costosas de ejecutar.
- En proyectos grandes, la repetición de pruebas manuales puede hacer que el QA omita defectos debido al cansancio.
- Poca reusabilidad, cada que se hace un cambio se debe volver a probar todo desde cero.

Pruebas Automatizadas



MÓDULO 5

Ventajas

- ✓ La ejecución de casos es rápido y eficiente.
- ✓ Brindan un mayor grado de confiabilidad y precisión.
- ✓ Tienen un mayor ROI, en proyectos grandes ayudan a reducir costos en pruebas y detección de bugs.
- ✓ Son reusables, se pueden correr una y otra vez en minutos con alto grado de cobertura.
- ✓ Las pruebas se pueden programar para que corran a cualquier momento sin intervención humana. CI/CD

Desventajas

- Automatizar las pruebas requiere una inversión de tiempo.
- Automatizar pruebas requiere de un conocimiento técnico que algunos QAs principiantes aún no tienen.
- Algunas herramientas de automatización son costosas.
- Las pruebas automáticas no proveen un feedback de aspectos del sistema como la usabilidad.

Como saber cual tipo prueba usar?

Pruebas manuales:

- ✓ Siempre para cada proyecto hay que hacer pruebas exploratorias manuales para conocer el sistema y poder empezar a diseñar casos de prueba.
- ✓ Cuando se tienen escenarios complejos en sistemas que requieren pre-condiciones y un grado de análisis complejo.
- ✓ Se recomiendan correr cuando se prueban sistemas que no tienen un resultado específico o es muy dinámico o variable

Pruebas Automatizadas

- ✓ Todas las pruebas de regresión son candidatas a automatización.
- ✓ Toda prueba que se corre constantemente en alguna aplicación y que tome mucho tiempo.
- ✓ Las pruebas de rendimiento o carga.
- ✓ Pruebas con flujos largos de muchos pasos en sitios web o móvil.
- ✓ Pruebas de APIs
- ✓ Pruebas de varios navegadores (Cross Browser Testing)

Automatización de pruebas con Cypress y Javascript

MÓDULO 5

Cypress es un framework de testing automatizado moderno y todo en uno. Es rápido, fácil de usar y permite ejecutar pruebas sobre cualquier aplicación web.

Cypress te permite crear pruebas automatizadas.

- End to End
- Pruebas de Integracion
- Pruebas Unitarias



<https://docs.cypress.io/>

Como funciona Cypress

MÓDULO 5

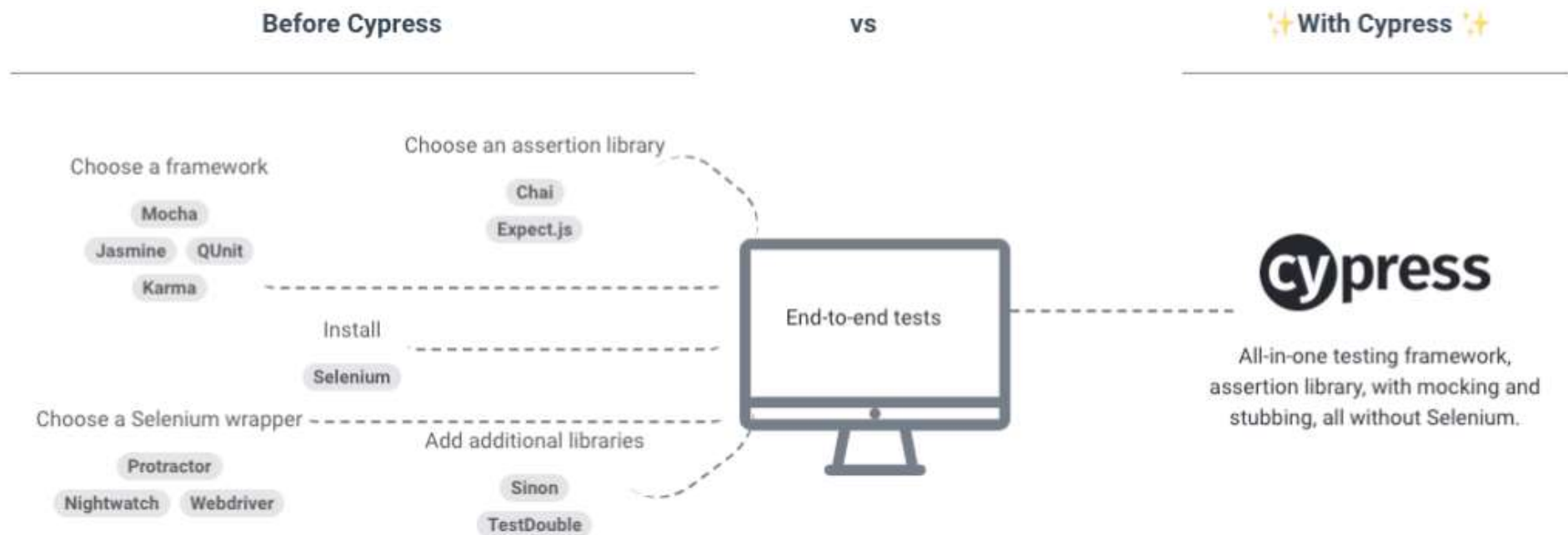
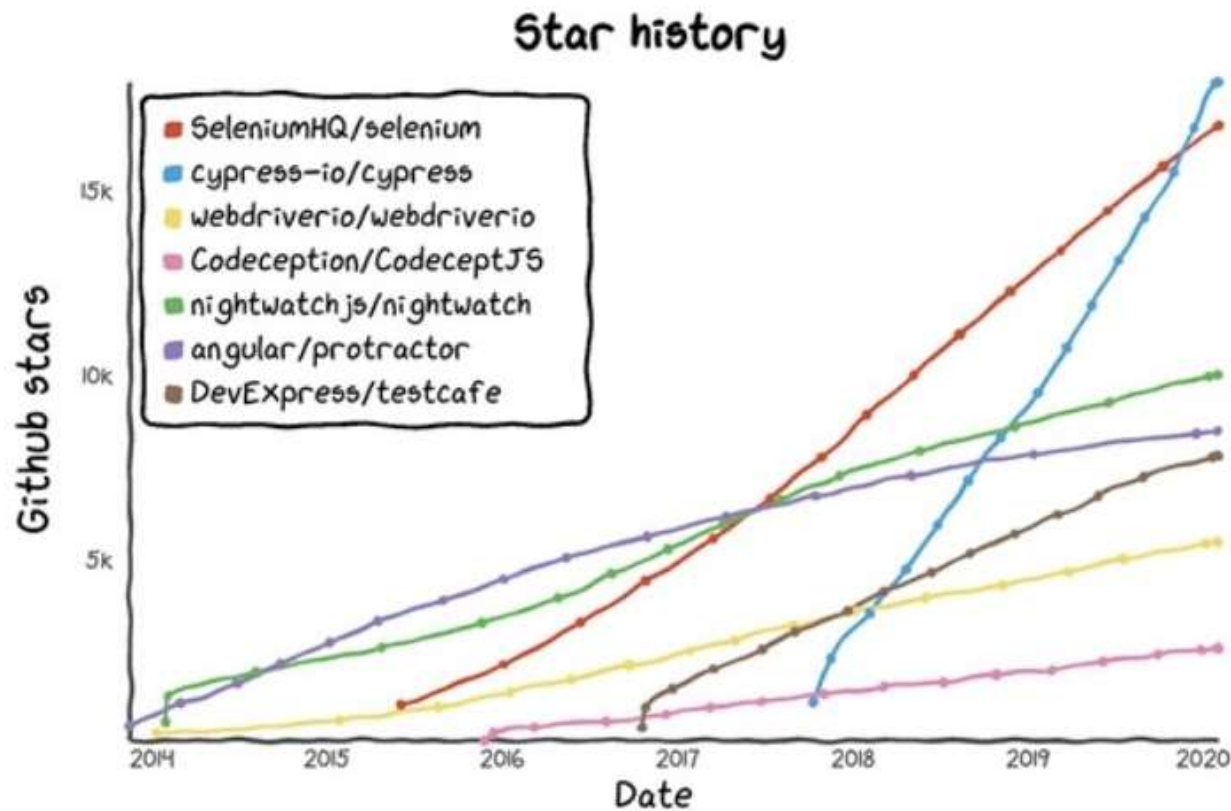


Fig:- How Cypress works

Uso de Cypress a nivel mundial..

MÓDULO 5



Que vamos a necesitar para empezar?

MÓDULO 5

Instalar Cypress es muy fácil.

1. Instalar el editor Visual studio code.
2. Instalar Node JS
3. Luego instalamos el manejador de paquetes NPM
4. Instalar Cypress como dependencia en el proyecto a probar.



COMO BUSCAR TRABAJO EN TESTING

Como encontrar trabajo en Testing?

MÓDULO 6

1. **Crea una CV resumida que impacte al empleador!** Si tienes repositorios en GitHub con proyectos trabajados seria ideal
2. **Crea tu perfil de LinkedIn**
 1. *Incluye palabras claves y herramientas de testing en tu perfil.*
 2. *En cada experiencia que tengas, debes poner tus funciones básicas, tecnologías trabajadas y si tienes algún reconocimiento.*
 3. *Incluye todos los cursos relevantes a software Testing*
 4. *Pídele a tus contactos que te validen conocimientos*
 5. *Pídele a tus contactos que te den recomendaciones y que describan proyectos de IT en los que hayas participado.*
 6. *Genera valor en LinkedIn, escribe artículos sobre Testing para ganar visibilidad y reputación en la materia.*
 7. *Sigue las empresas de Software y Testing de tu ciudad en LinkedIn.*
 8. *Adiciona otros referentes de Testing en tu área como también reclutadores.*
 9. *Escribe un post en tu perfil diciéndole a tu red que estas buscando trabajo como Tester.*

Como encontrar trabajo en Testing?

MÓDULO 6

3. Crea una red de contacto, haz networking!

1. Únete a grupos de Testing en Facebook de tu país
2. Únete a grupos de Meetup de testing en tu ciudad y atiende a los eventos
 1. <https://www.meetup.com/>
3. Únete a grupos de LinkedIn sobre testing.
4. Puedes hacer voluntariados en startups, o proyectos que apenas arrancan en internet.

4. Gana experiencia como Tester freelance

<https://www.utest.com/>
<https://www.upwork.com/>
<https://www.fiverr.com/>
<https://www.freelancer.com/>

Puedes probar proyectos o paginas de empresas existentes, crea casos de prueba, ejecuta los casos, automatiza algunos y súbelos a algún repositorio online. Si encuentras bugs repórtalos a la empresa, esto te sirve para demostrar experiencia durante las entrevistas

Recomendaciones para la entrevista

MÓDULO 6

1. Prepárate antes de la entrevista

1. Averigua sobre la empresa, sus servicios, productos, mercado y todo lo que puedas
2. Lee muy bien la descripción del trabajo publicada, sobre el rol y las responsabilidades
3. Revisa que están tech-stack están buscando, que herramientas están solicitando y refresca tus conocimientos sobre estos.
4. Averigua donde va ser la entrevista, si es videollamada, prepara tu laptop con el software y haz una prueba con anterioridad.

2. Durante la entrevista

1. Relájate, aprovecha la primera pregunta “rompe hielo” para tranquilizarte. Sonríe y te una buena postura.
2. Cuéntanos acerca de tí, responde de manera breve (los expertos recomiendan max 2-3 minutos) tu profesion, tus años de experiencia, roles destacados y relacionados al cargo, responsabilidades relevantes al cargo, tus habilidades y lo que puedes aportar al cargo en cuestión.
3. Cuando te pregunten por responsabilidades especificas, trata de resaltar las que estén relacionadas con el cargo al que estas aplicando y destaca las herramientas utilizadas con ejemplos reales donde se vea el problema presentado y como lo solucionaste.
4. Puede que te pregunten por debilidades y fortalezas, ten un ejemplo claro en mente para cada una, en las debilidades debes explicar que has hecho para mejorarlo.
5. Al final de las entrevistas generalmente preguntan si tienes alguna duda, ten en mente un par de preguntas preparadas por ejemplo : Cuales son las prioridades del cargo al que aplicas, que tecnologias estan trabando, que tipo de metodologias estan usando para el Desarrollo de software..etc.

6. Después de la la entrevista

Luego de un par de días, esta atento a cualquier email o llamada del empleador. Si te solicitan 2nda entrevista o prueba técnica prepárate y cumple con los tiempos de entrega. EXITOS en la búsqueda!!

Recomendaciones iniciales

MÓDULO 6

1. Sugiero realiza el curso en el orden establecido
2. Completa los quices y actividades de cada sección.
3. Puedes acelerar la velocidad de las clases y dejar notas.
4. Si la clase se ve borrosa , puedes cambiar la resolución.
5. No olvides descargar todo el material de cada sección, código, plantillas y los bonos extras que están al final.
6. Recuerda que el curso está en constante actualización con nuevo contenido para maximizar el valor invertido.
7. Sugiero dejar la calificación del curso para el final.

Happy Testing

¿Cómo calificarías tu experiencia con el curso hasta ahora?

Excelente, por encima de mis expectativas



Preguntarme más tarde

Preguntarme al final del curso

Tu calificación es muy importante!

MÓDULO 6

¿Por qué diste esta calificación?

Excelente, por encima de mis expectativas



Un curso muy completo, y didactico,
me ha encantado!

Guarda y Continúa

Danos más información (opcional).

¿Estás adquiriendo información que te resulte útil?

☒ Sí

☐ No

☐ No lo sé con seguridad

¿Están los conceptos bien explicados?

☒ Sí

☐ No

☐ No lo sé con seguridad

¿Crees que la forma en la que realiza las clases es un incentivo para realizarlas?

☒ Sí

☐ No

☐ No lo sé con seguridad

¿Tienes oportunidades suficientes para poner en práctica lo que has aprendido?

☒ Sí

☐ No

☐ No lo sé con seguridad

¿Cumple el curso tus expectativas?

☒ Sí

☐ No

☐ No lo sé con seguridad

¿Es el instructor un experto en la materia?

☒ Sí

☐ No

☐ No lo sé con seguridad

Guarda y Continúa

Saltar