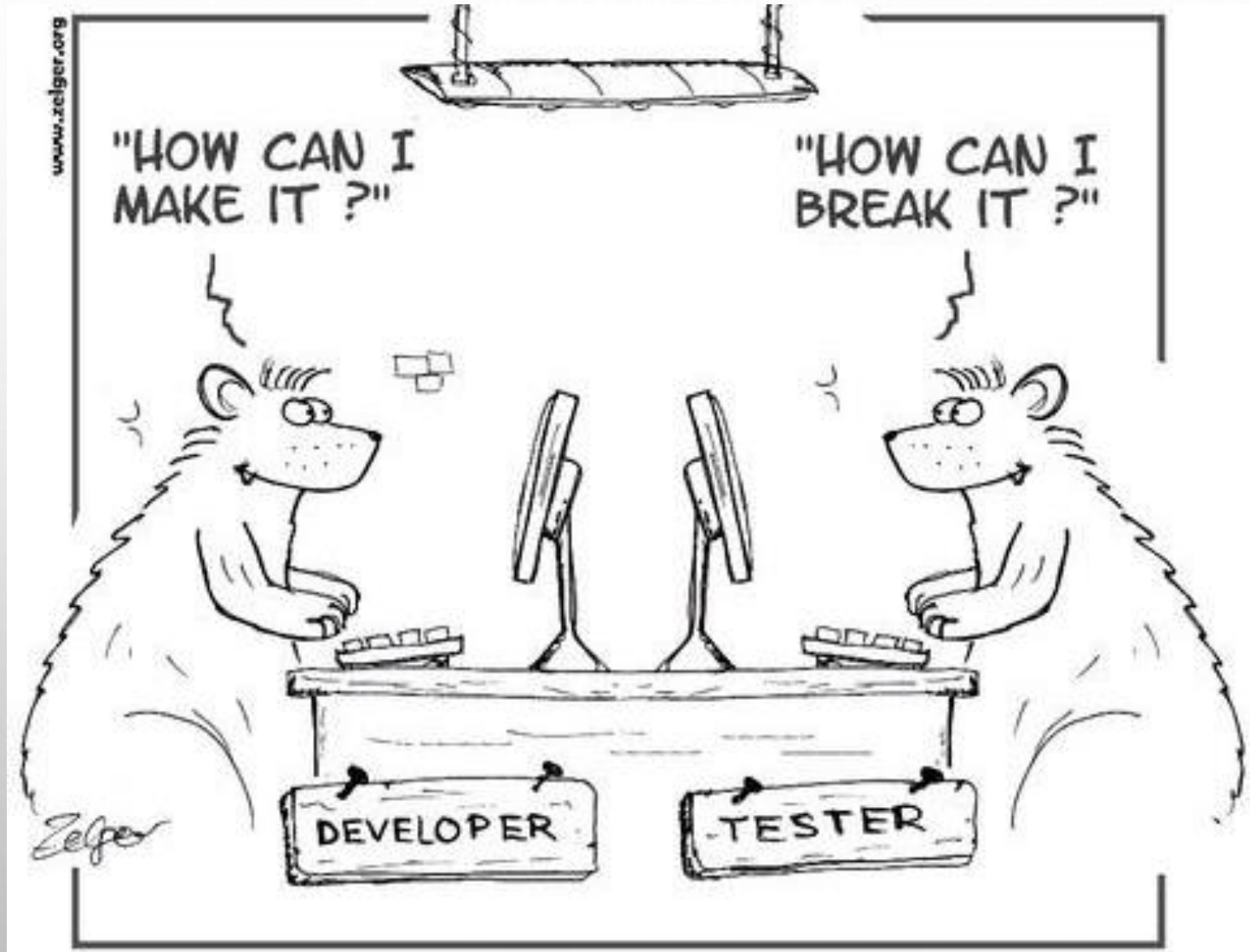


METODOLOGÍA DE PRUEBAS DE SOFTWARE

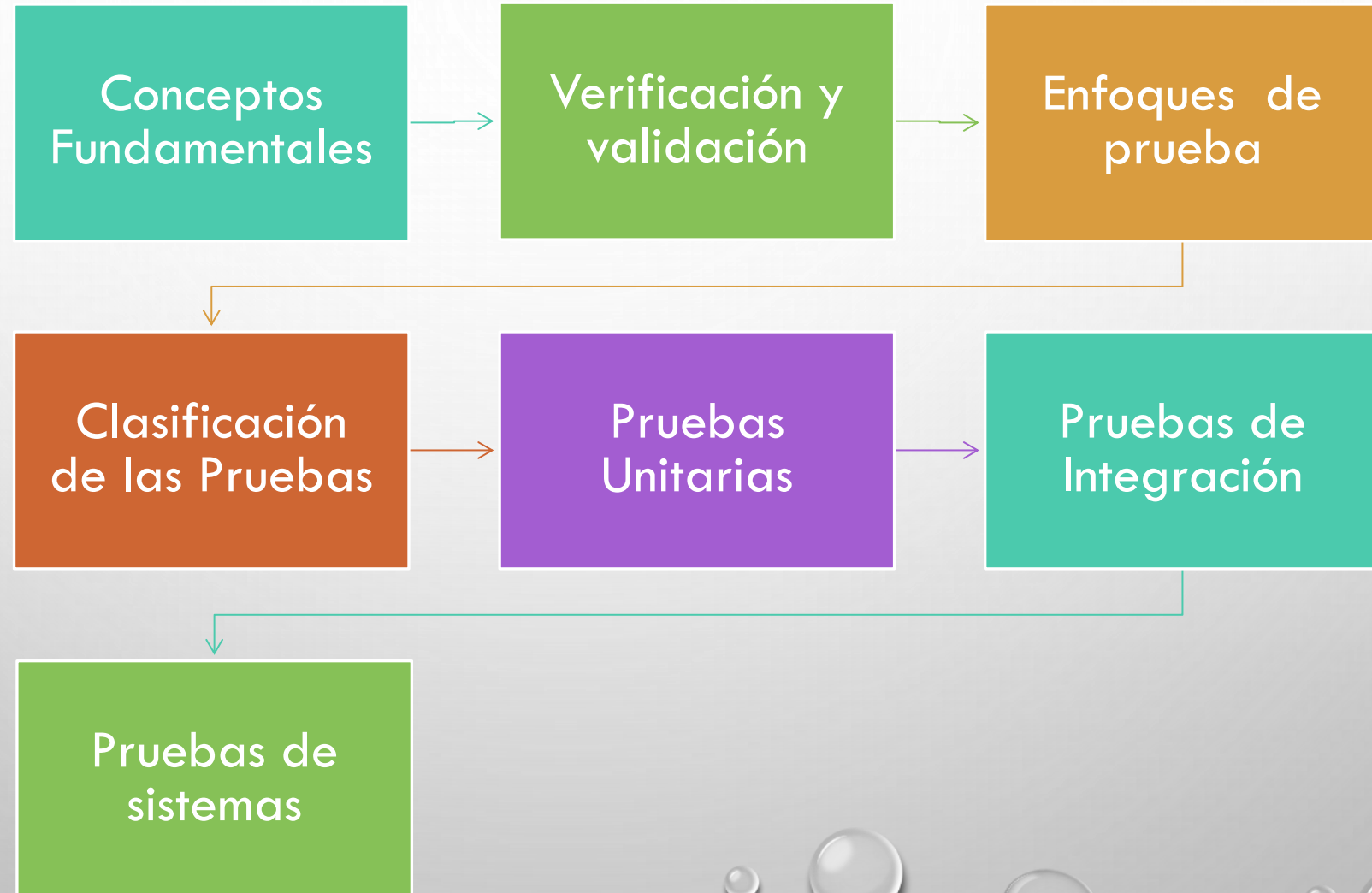


Materia: Metodología de Pruebas de Software

Profesor: Ing. Pablo Andrés Pérez



RESUMEN DE LO VISTO



PRUEBAS DE SISTEMAS

- UNA VEZ QUE SE HAN PROBADO LOS COMPONENTES INDIVIDUALES Y SE HAN INTEGRADO, SE PRUEBA EL SISTEMA DE FORMA GLOBAL. **EN ESTA ETAPA PUEDEN DISTINGUIRSE LOS SIGUIENTES TIPOS DE PRUEBAS, CADA UNO CON UN OBJETIVO CLARAMENTE DIFERENCIADO:**

- Pruebas funcionales.
- Pruebas de comunicaciones.
- Pruebas de rendimiento.
- Pruebas de volumen.
- Pruebas de sobrecarga.
- Pruebas de disponibilidad de datos.
- Pruebas de facilidad de uso.
- Pruebas de operación.
- Pruebas de entorno.
- Pruebas de seguridad.

CLASIFICACIÓN DE LAS PRUEBAS

Según el conocimiento del código

Caja Negra

Caja Blanca

Según la etapa de desarrollo

Unitario

Integración

Sistema

Aceptación

Regresión

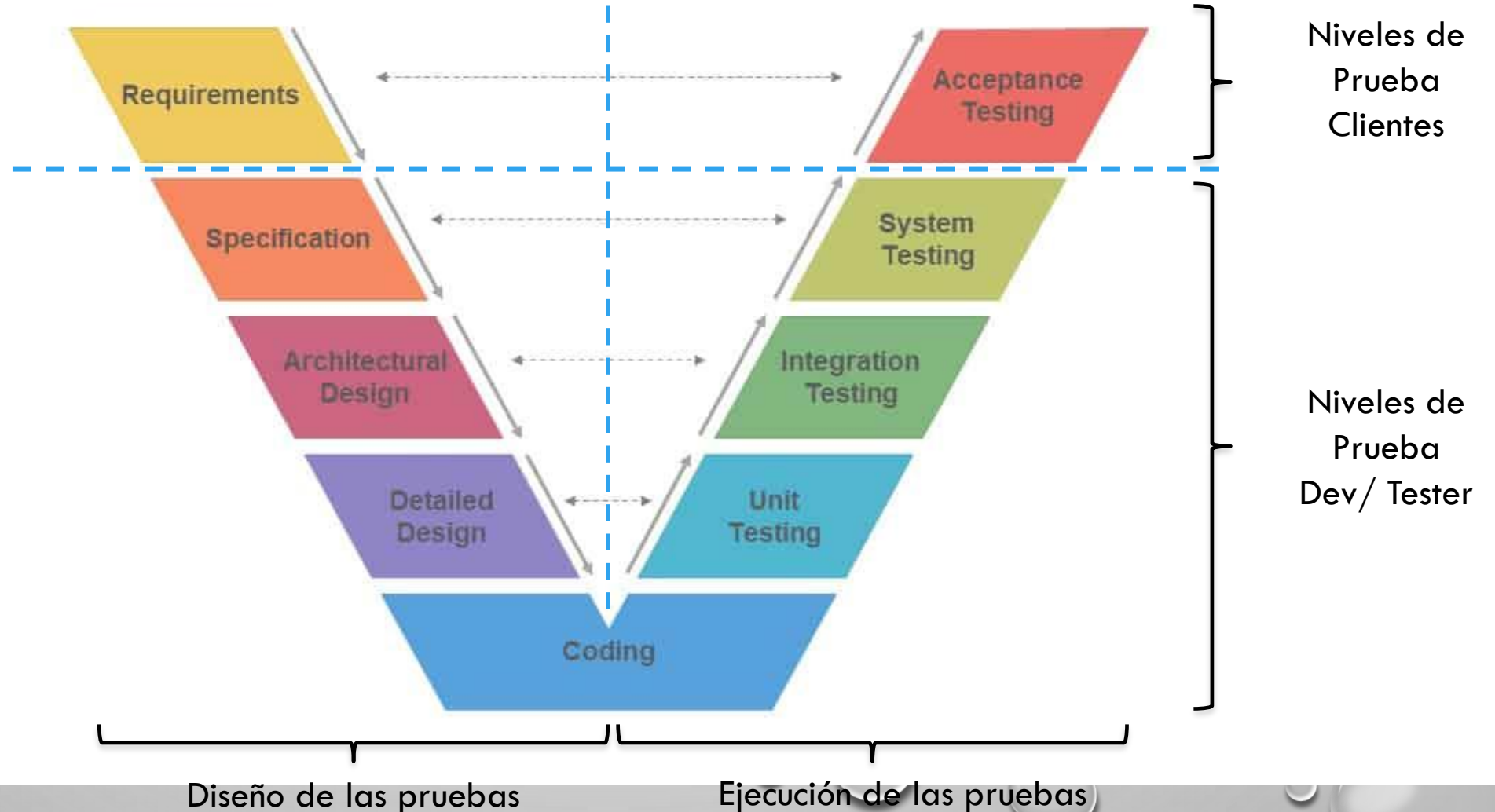
Según el aspecto a evaluar

Funcional

Para-funcional

Diagram

V-MODEL SOFTWARE DEVELOPMENT



FUNCIONALES Y NO FUNCIONALES

Pruebas funcionales

Pruebas unitarias
Pruebas de integración
Pruebas de sistema
Pruebas de sanidad
Pruebas de humo
Pruebas de interfaz
Pruebas de regresión
Pruebas de aceptación

Pruebas no funcionales

Pruebas de rendimiento
Prueba de carga
Pruebas de estrés
Pruebas de volumen
Pruebas de seguridad
Pruebas de compatibilidad
Pruebas de instalación
Pruebas de recuperación
Pruebas de confiabilidad
Pruebas de usabilidad
Pruebas de conformidad
Pruebas de localización

*“Las pruebas funcionales se centran en verificar que el software **funcione de acuerdo con los requisitos especificados.**”*

*“Las pruebas no funcionales evalúan aspectos no relacionados directamente con funciones o características específicas del software, **sino con atributos de calidad**”*

Pruebas funcionales

Se definen como un tipo de prueba cuya finalidad es comprobar que el *sistema bajo prueba se comporte de acuerdo con las especificaciones o requisitos funcionales*. Algunos pasos generales para realizar pruebas funcionales son:

Identificar los datos de entrada

Determinar los resultados esperados de acuerdo a los datos de entrada

Ejecutar los casos de prueba

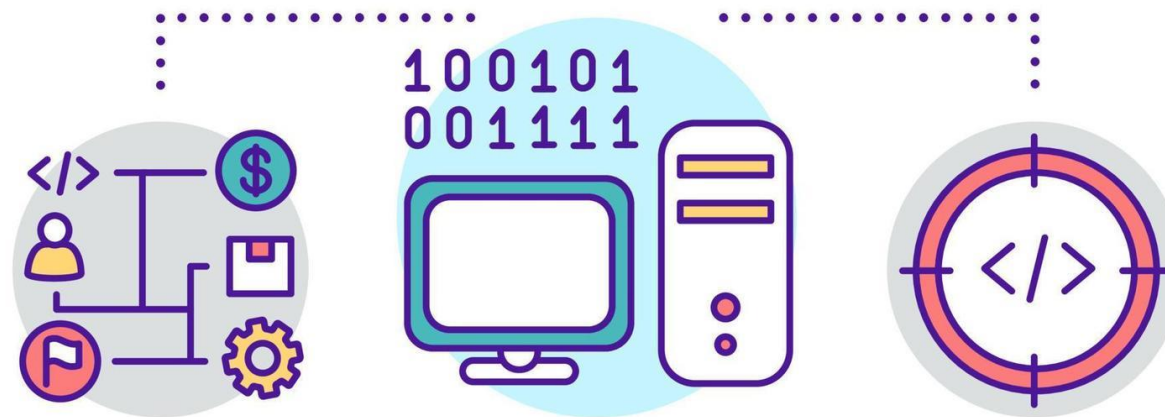
Comparar los resultados actuales con los resultados esperados

PRUEBAS UNITARIAS

- LAS PRUEBAS UNITARIAS SON LAS QUE ASEGURAN QUE CADA CÉLULA DEL CÓDIGO DESARROLLADO EN UN COMPONENTE BRINDE LOS **RESULTADOS ADECUADOS**. EN ESTAS PRUEBAS LOS DESARROLLADORES OBSERVAN LA INTERFAZ Y LA ESPECIFICACIÓN DE UN COMPONENTE, PROPORCIONANDO LA DOCUMENTACIÓN DEL DESARROLLO DEL CÓDIGO SE PRUEBA EXHAUSTIVAMENTE, CLARO QUE DE FORMA INDEPENDIENTE ANTES DE PASAR A OTRA UNIDAD.
- LAS **PRUEBAS UNITARIAS** ADMITEN PRUEBAS FUNCIONALES AL EJERCER EL CÓDIGO QUE ES MÁS PROBABLE QUE SE *ROMPA*. POR ELLO, SI SE USA PRUEBAS FUNCIONALES SIN PRUEBAS UNITARIAS, PUEDES EXPERIMENTAR ALGUNAS DIFICULTADES PARA DIAGNOSTICAR PRUEBAS FALLIDAS.

PRUEBAS DE HUMO

Las **pruebas de humo** se realizan para verificar si las funcionalidades más significativas de la aplicación funcionan o no. De forma que lo más básico del software se ejecute de forma correcta con **pruebas sencillas y rápidas**.



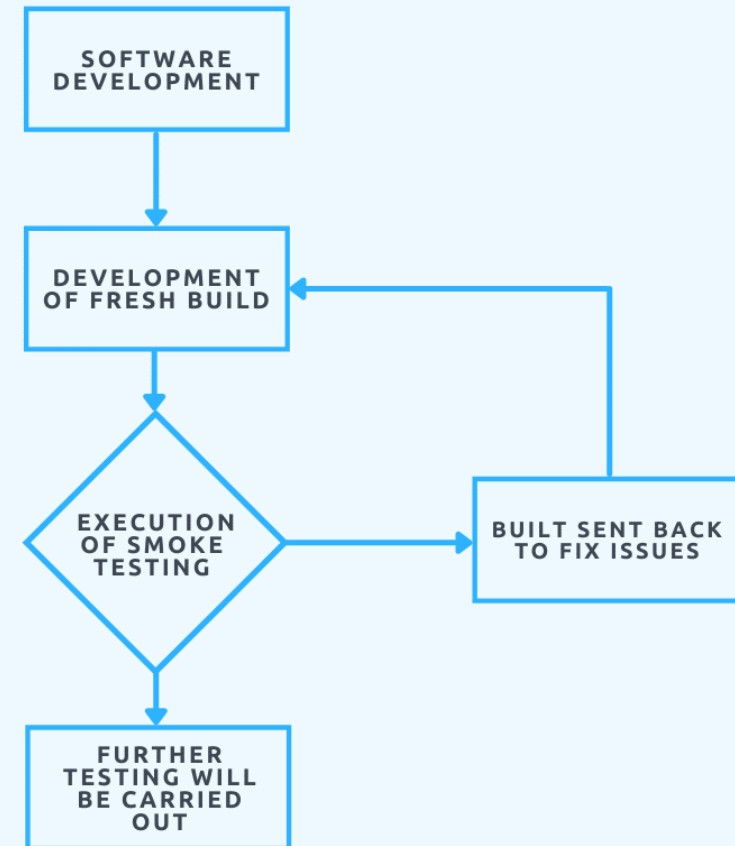
PRUEBAS DE HUMO

Es una de las **pruebas funcionales más importantes** y debería ser la primera que se realice en **una nueva compilación**.

La prueba de humo es común y a veces no se tiene claro su concepto.

No se trata de realizar pruebas exhaustivas sino de **verificar que la funcionalidad crítica del sistema realmente funciona bien**.

Smoke Testing Representation



PRUEBAS DE HUMO

Si la prueba es exitosa será entonces una compilación estable. El equipo QA realizará **pruebas funcionales** para las características o funcionalidades recién agregadas posteriormente o **pruebas de regresión** según la situación. Por otro lado, si esta no es estable y falla la compilación lo común es que se devuelva al equipo de desarrollo para solucionar los problemas de compilación y crear una nueva.



PRUEBAS DE CORDURA O SANIDAD



- CUANDO SE TIENE UNA COMPILACIÓN CON MODIFICACIONES MENORES, EN VEZ DE EJECUTAR LAS PRUEBAS DE REGRESIÓN, REALIZAMOS UNA PRUEBA DE CORDURA.
- CON ELLA PODEMOS DETERMINAR QUE LAS MODIFICACIONES REALMENTE HAYAN SOLUCIONADO LOS PROBLEMAS. Y QUE DICHAS CORRECCIONES NO HAYAN GENERADO NINGÚN PROBLEMA.
- USUALMENTE ESTAS PRUEBAS SON SUBPRUEBAS DE LA DE '*REGRESIÓN*' YA QUE ESTÁN RELACIONADAS CON LOS CAMBIOS REALIZADOS EN EL PRODUCTO.

“SMOKE TEST VS SANITY TEST”

SIMILITUDES:

- **RÁPIDAS Y ENFOCADAS:** AMBAS SON PRUEBAS RÁPIDAS QUE SE EJECUTAN PARA DETERMINAR SI EL SISTEMA ES LO SUFICIENTEMENTE ESTABLE PARA PROCEDER CON PRUEBAS MÁS EXHAUSTIVAS.
- **SELECCIÓN DE FUNCIONALIDADES:** TANTO LAS PRUEBAS DE HUMO COMO LAS DE SANIDAD SE CENTRAN EN UN CONJUNTO LIMITADO DE FUNCIONALIDADES CLAVE O CRÍTICAS DEL SISTEMA.
- **FRECUENCIA DE USO:** AMBAS SE UTILIZAN REGULARMENTE DURANTE EL CICLO DE DESARROLLO DE SOFTWARE, ESPECIALMENTE DESPUÉS DE CAMBIOS EN EL CÓDIGO.



“SMOKE TEST VS SANITY TEST”

- **PRUEBAS DE HUMO:** TIENEN UN ALCANCE MÁS AMPLIO PERO SUPERFICIAL, CUBRIENDO LAS FUNCIONALIDADES BÁSICAS DEL SISTEMA PARA CONFIRMAR LA ESTABILIDAD INICIAL.
- **PRUEBAS DE SANIDAD:** TIENEN UN ALCANCE MÁS LIMITADO Y PROFUNDO, ENFOCADAS EN UN ÁREA ESPECÍFICA DEL SISTEMA DIRECTAMENTE RELACIONADA CON LOS CAMBIOS REALIZADOS.



“SMOKE TEST VS SANITY TEST”

- **PRUEBAS DE HUMO:** SE SUELEN EJECUTAR DESPUÉS DE CADA NUEVA COMPILACIÓN DEL SOFTWARE.
- **PRUEBAS DE SANIDAD:** SE EJECUTAN DESPUÉS DE APLICAR PEQUEÑOS CAMBIOS EN EL CÓDIGO O REALIZAR CORRECCIONES DE ERRORES EN UNA COMPILACIÓN EXISTENTE.
- LAS PRUEBAS DE HUMO PRUEBAN SI UN NUEVO BUILD DEL SOFTWARE ES MÍNIMAMENTE FUNCIONAL
- LAS PRUEBAS DE SANIDAD CONFIRMAN SI CAMBIOS RECIENTES HAN SIDO EXITOSOS SIN INTRODUCIR ERRORES ADICIONALES.



PRUEBAS INTEGRACION

- LA PRUEBA DE INTEGRACIÓN ES UNO DE LOS TIPOS DE PRUEBA FUNCIONAL MÁS COMÚN Y SE REALIZA **DE FORMA AUTOMATIZADA**. SE REALIZAN PARA **PROBAR COMPONENTES INDIVIDUALES** CON EL OBJETIVO DE VERIFICAR CÓMO LOS MÓDULOS, QUE TRABAJAN DE FORMA INDIVIDUAL, FUNCIONAN CUANDO ESTÉN INTEGRADOS.
- EL OBJETIVO DE REALIZAR ESTAS PRUEBAS **ES PORQUE COMÚNMENTE LOS DESARROLLADORES SE ENFOCAN EN CONSTRUIR DIFERENTES MÓDULOS DEL SISTEMA SIMULTÁNEAMENTE Y NO SE CENTRAN EN OTROS.**

Usualmente nos ayuda a identificar problemas en las operaciones de la interfaz de usuario, formatos de datos, invocar API, acceso a bases datos, entre otras.

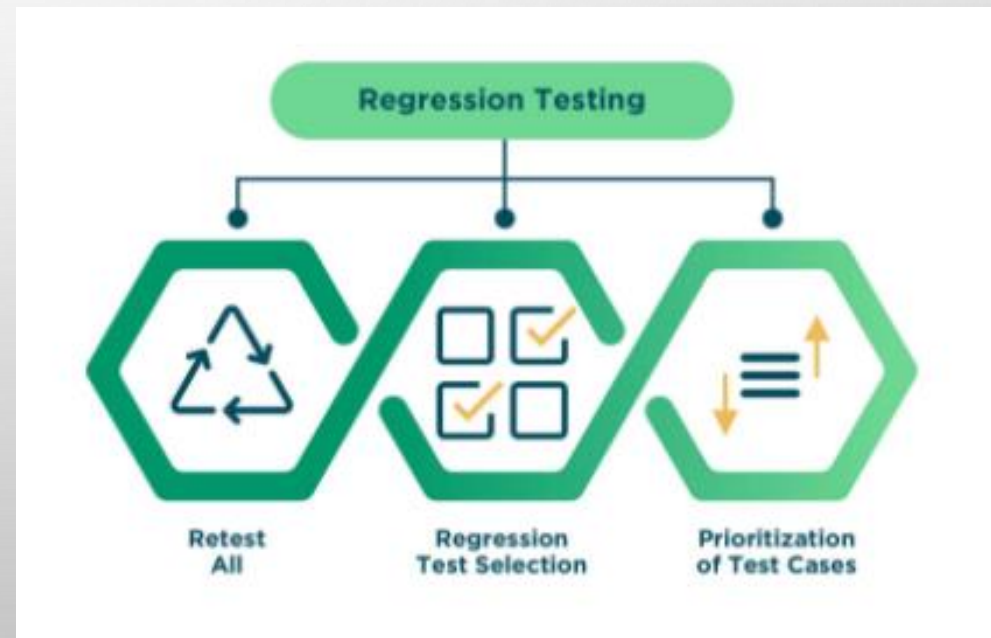
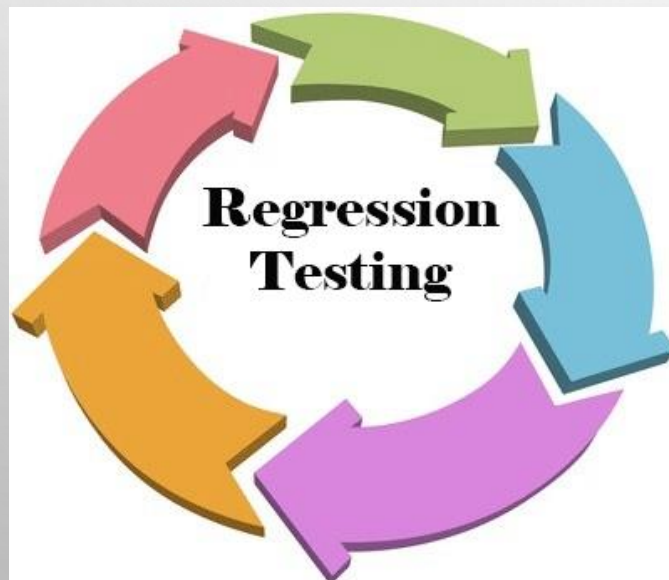


PRUEBAS INTEGRACION

- ALGUNAS DE LAS VERIFICACIONES QUE SE REALIZAN EN LAS PRUEBAS DE INTEGRACIÓN SON:
- **PRUEBA DE INTERFAZ:** EN LA COMPROBACIÓN DE LAS TRANSFERENCIAS DE DATOS ENTRE DOS COMPONENTES. PRUEBA DE INTERFACES COMO SERVICIOS WEB, API, ENTRE OTROS.
- SE REALIZA PARA VERIFICAR QUE LOS COMPONENTES ESTÉN SINCRONIZADOS ENTRE SÍ. AYUDAN A DETERMINAR QUE DIFERENTES FUNCIONES, COMO LA TRANSFERENCIA DE DATOS ENTRE LOS DIFERENTES ELEMENTOS DEL SISTEMA, SE REALIZAN DE ACUERDO CON LA FORMA EN QUE FUERON DISEÑADAS.

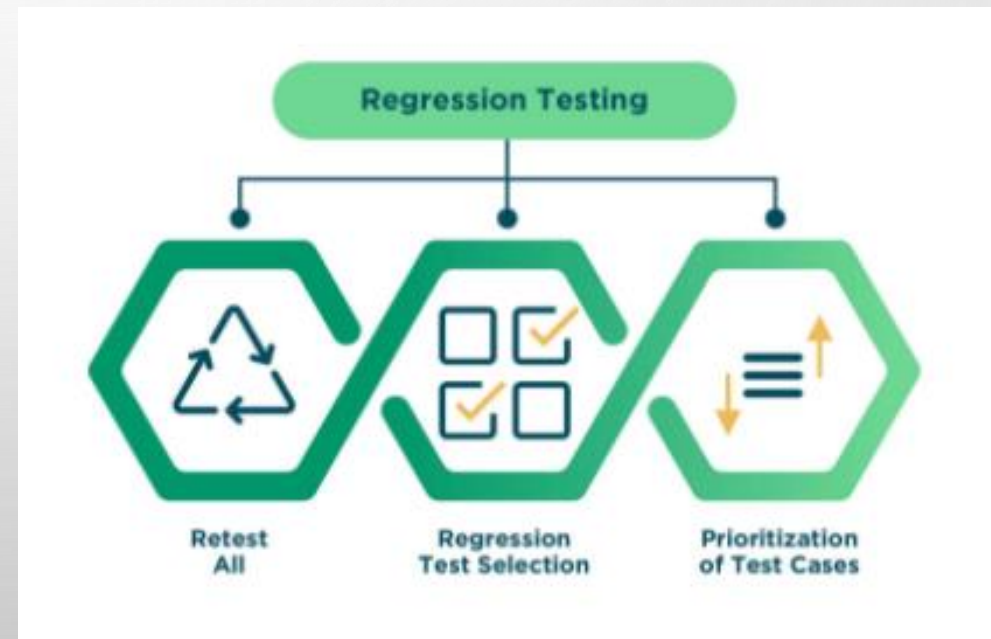
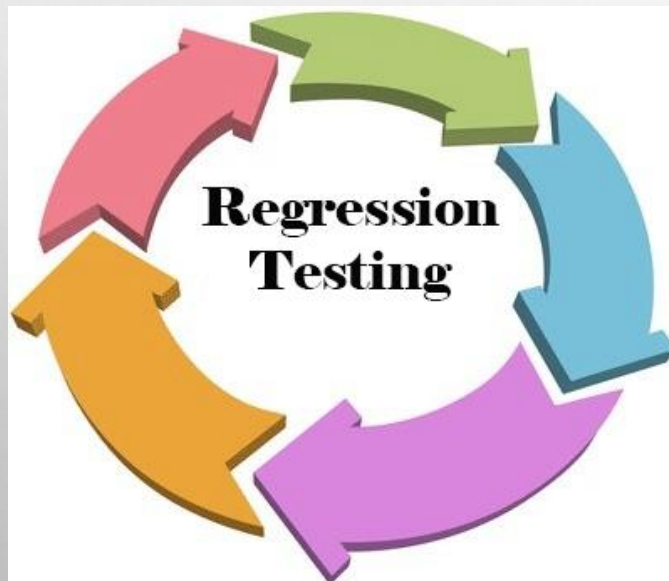
PRUEBAS DE REGRESION

ES NORMAL QUE LOS DESARROLLADORES **MODIFIQUEN Y MEJOREN** LAS FUNCIONALIDADES DE SU DESARROLLO. POR ELLO EXISTE UNA GRAN POSIBILIDAD DE QUE PUEDAN CAUSAR 'EFECTOS' INESPERADOS EN SU COMPORTAMIENTO. ESTAS PRUEBAS DE REGRESIÓN SE REALIZAN PARA ASEGURAR QUE LOS CAMBIOS O ADICIONES NO HAYAN ALTERADO NI ELIMINADO LAS FUNCIONALIDADES EXISTENTES.



PRUEBAS DE REGRESION

EL OBJETIVO DE LAS **PRUEBAS DE REGRESIÓN** ES ENCONTRAR ERRORES QUE PUEDAN HABER SIDO INTRODUCIDOS ACCIDENTALMENTE EN LA COMPILACIÓN EXISTENTE Y ASÍ GARANTIZAR QUE LOS ERRORES ELIMINADOS CONTINÚEN ASÍ.



PRUEBAS DE ACEPTACION

- LAS PRUEBAS DE ACEPTACIÓN CONSTITUYEN LA ÚLTIMA FASE DEL PROCESO DE TESTING. AQUÍ LOS USUARIOS REALES DEL SOFTWARE *LO USAN PARA VERIFICAR QUE CUMPLA CON LAS TAREAS REQUERIDAS EN UN AMBIENTE 'REAL'*. EN OCASIONES SE REALIZA CUANDO SE HACE LA ENTREGA DEL PRODUCTO *“COMO PUNTO DE CONTROL FINAL ENTRE TODOS LOS TIPOS DE PRUEBAS FUNCIONALES”*.



PRUEBAS DE ACEPTACIÓN

- SON LAS PRUEBAS DE ASEGURAMIENTO DE CALIDAD MÁS IMPORTANTES QUE SE PUEDEN IMPLEMENTAR PARA ENTREGAR DESARROLLOS PRODUCTOS Y/O APLICACIONES DE OTRO NIVEL. ASÍ SE PODRÁ CUMPLIR CON LOS REQUERIMIENTOS DEL CLIENTE Y ENTREGAR SOLUCIONES FUNCIONALES Y DE CALIDAD.



