Clase 06. Defectos en el Testing 

# Ambientes de desarrollo y builds

## Ambientes

### Ambientes de desarrollo

Los ambientes de desarrollo son los espacios virtuales donde se pondrá a disposición lo desarrollado para que pueda ser testeado y corregido por los distintos equipos.

Comprendemos que espacios virtuales serán computadoras, servidores, máquinas virtuales, etc...

#### Características de ambientes de desarrollo

Cada ambiente posee una característica particular relacionada con las partes que se encuentran utilizándolo.

Estos son:

* Accesibilidad
* Funcionalidad
* Salud del ambiente
* Potencia del equipo

## 5 ambientes

1. Local

2. Desarrollo

3. Staging

4. Pre-producción

5. Producción

### Ambiente local

El ambiente local es literalmente la computadora donde se puede ver el código del desarrollo.

Este ambiente puede ser llevado adelante por el desarrollador del proyecto o un QA al solicitar el repositorio del código para hacer pruebas de caja blanca.

### Ambiente desarrollo

Este ambiente implica la primera etapa de subida de código del proyecto. Es decir, el desarrollo es donde los devs unen sus proyectos y -por lo general- donde realizan las pruebas unitarias.

Suelen ser los desarrolladores quienes tienen acceso a este ambiente.

Se recomienda no realizar pruebas en este ambiente debido a que es inestable.

### Ambiente staging

Staging es el primer ambiente donde los QA realizan pruebas, debido a que posee cierta estabilidad que no se asemeja al ambiente productivo.

### Ambiente preproducción

Preproducción es el ambiente donde se realizará el UAT para la integración con lo que ya existe.

Además, en preproducción se encontrarán los desarrollos con menor cantidad de fallas porque es el paso previo a la salida a producción.

Recuerden que UAT es User Acceptance Testing, son las pruebas de aceptación mínimas para el usuario.

### Ambiente producción

Producción es el ambiente final donde los usuarios y todos tendrán acceso al contenido generado.

Los QA -al momento de realizar subidas- harán pruebas para asegurar que lo desarrollado finalmente funciona correctamente.

## Build

Build surge del término construcción en inglés y hace referencia particularmente a una construcción del código.

En definitiva, build convierte archivos de código fuente en software aislado, utilizándose mucho para aplicaciones móviles o videojuegos.

### Builds y ambientes

Producción es el ambiente final donde los usuarios y todos tendrán acceso al contenido generado.

Los QA -al momento de realizar subidas- harán pruebas para asegurar que lo desarrollado finalmente funciona correctamente.

* Los builds al ser independientes de los ambientes no se distinguen en varias partes.
* La única diferencia es que los builds consumen un ambiente u otro dependiendo el estado del desarrollo del producto.

Los builds poseen versionados, mientras los ambientes son únicos.

# ¿Cuál es la diferencia entre un error, fallo y defecto?

## Diferencias

Una persona puede cometer un error que puede llevar a la introducción de un defecto (bug) en el código software y que resulte en una falla en el sistema.

## Errores

### Error

Es una acción humana que produce un resultado incorrecto, una idea equivocada de algo. El error es una equivocación de parte del desarrollador o del analista. Un error puede llevarnos a generar uno o más defectos.

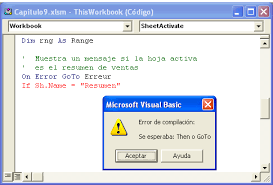
Es una desviación o algo que está mal y se encuentra en la etapa en la cual se originó.

Algunos ejemplos:

* Error en la lógica de la programación.
* Un requerimiento que esté mal especificado.

Por ejemplo el desarrollador cometió errores en el código (en una sentencia por ejemplo) y cuando realiza las pruebas unitarias encuentra este error y lo corrige.

#### Ejemplo de error



Fuente: [ediciones-eni](https://www.ediciones-eni.com/open/mediabook.aspx?idR=4af1c999e1b6d14b6284c5da3e591be2)

## Defectos

### Defecto

Es la imperfección de un componente causado por un error. El tester es quien debe reportar el defecto.

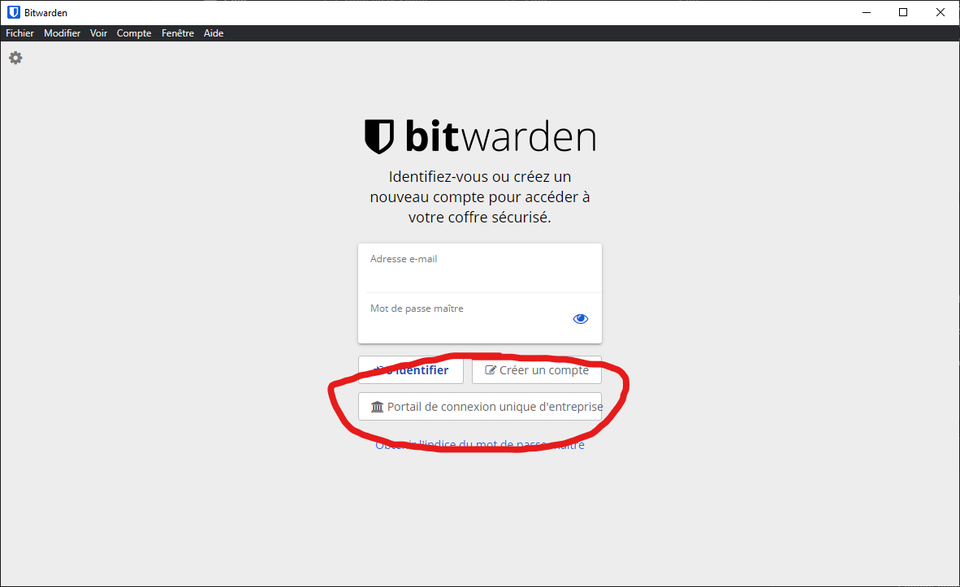
Bug encontrado en la etapa de testing

Es un desvío o algo que está mal y se encuentra en la etapa posterior en la que se originó.

Algunos ejemplos:

* Un módulo de registro de usuarios tiene mala configuración en la función de conexión a base de datos.
* Una función de login cuenta con las variables de usuario y contraseña declaradas incorrectamente.

#### Ejemplo de defecto

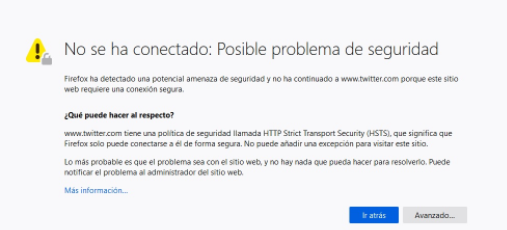


Fuente: [reddit](https://www.reddit.com/r/Bitwarden/comments/m6ho89/bug_report_visual_bug_french_version_windows/)

## Fallas

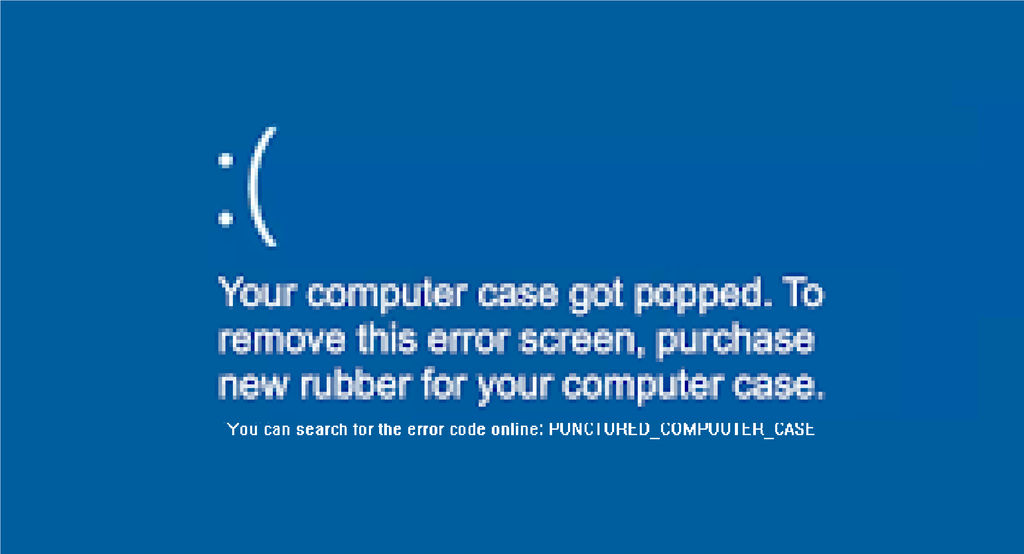
### Falla

Es la manifestación visible de un defecto. Es decir que si un defecto es encontrado durante la ejecución de una aplicación entonces va a producir un fallo.



#### Ejemplos de fallas

* Visualización de un mensaje de alerta que no fue definido previamente por el desarrollador.
* Un formulario de login que contenga los datos de acceso no te permita ingresar a la aplicación al hacer clic en el botón de ingresar.



## Defectos, causas raíz y efectos

### Definiciones

Se pueden analizar los defectos para identificar sus causas raíz, con el propósito de reducir la ocurrencia de defectos similares.

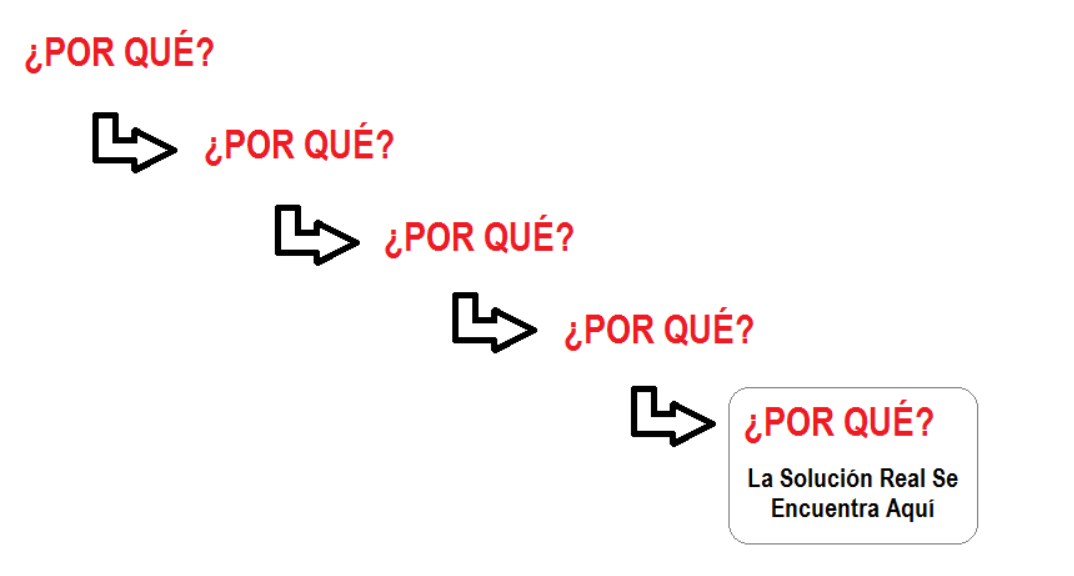
Al centrarse en las causas raíz más significativas, el análisis puede conducir a mejoras en el proceso que previenen la introducción de un número significativo de futuros defectos.

Como consecuencias de estos defectos, tendremos acciones que surgen y serán los efectos.

### Los 5 ¿por qué?

Es una técnica para realizar preguntas iterativas, usadas para explorar las relaciones de causa y efecto subyacentes a un problema particular.​ El objetivo principal de la técnica es determinar la causa raíz de un defecto o problema repitiendo la pregunta "¿por qué?"

#### ¿Cómo se implementa?



Cada respuesta forma la base de la siguiente pregunta.

El "5" en el nombre se deriva de la observación empírica en el número de iteraciones típicamente requeridas para resolver el problema.

No todos los problemas tienen una sola causa raíz.

Si se quieren descubrir múltiples causas raíz, el método debe ser repetido; haciendo una secuencia de preguntas por cada causa raíz.

#### Ejemplo de implementación

El vehículo no arranca. (El problema)

* ¿Por qué? - La batería está muerta. (Primer ¿por qué?)
* ¿Por qué? - El alternador no está funcionando. (Segundo ¿por qué?)
* ¿Por qué? - La correa del alternador se ha roto. (Tercer ¿por qué?)
* ¿Por qué? - La correa del alternador fue mucho más allá de su vida de servicio útil y no se ha sustituido. (Cuarto ¿por qué?)
* ¿Por qué? - El vehículo no se mantiene de acuerdo al recomendado programa de servicio. (Quinto ¿Por qué?, una de las causas)

# Clasificación de defectos

### Registrar defectos

Dado que uno de los objetivos de la prueba es detectar defectos, los defectos encontrados durante la prueba deben registrarse.

La forma en que se registran los defectos pueden variar, según el contexto del componente o sistema que se está probando, el nivel de prueba, etc.

Cualquier defecto identificado debe ser investigado y debe ser objeto de seguimiento desde su descubrimiento y clasificación hasta su resolución.

### Proceso de gestión

Para gestionar todos los defectos hasta su resolución, la organización debe establecer un proceso de gestión de defectos que incluya un flujo de trabajo y reglas de clasificación.

Este proceso debe ser acordado con todos aquellos que participan en la gestión de defectos.

En algunas organizaciones, el registro y seguimiento de defectos puede ser muy informal.

El equipo que participa incluye diseñadores, desarrolladores, testers y propietarios de producto

## Criticidad y urgencia

### Diferencia

La criticidad es la medida del alcance del daño potencial que el incidente puede causar.

La urgencia es la rapidez con que se requiere una resolución.

Los incidentes graves, por ejemplo, tienen un tiempo de resolución menor, puesto que su impacto en el servicio es mayor.

### Criticidad

La criticidad es la condición o el estado de crítico. Esto mide que tan grave o cómo impacta, un defecto sobre el sistema. Depende del acuerdo de equipos, pero en general los estados de criticidad pueden ser bajo, medio, alto, crítico.

### Urgencia

La urgencia mide que tan inmediata es la necesidad de corregir un defecto en el sistema.

Depende del acuerdo de equipos, pero en general los estados de criticidad pueden ser bajo, medio, alto, urgente.

### Niveles de bugs

Para medir los niveles del bug, usamos una mezcla de criticidad y urgencia.

Podemos tener defectos de criticidad baja y urgencia alta, y viceversa. No siempre lo critico es urgente.

Ejemplos:

* El login tiene una imagen que rompe y no permite ver algún dato poco importante: Baja criticidad y urgencia - Nivel bajo.
* El login no permite ingresar caracteres para loguearse: crítico y urgente. - Nivel alto.

## Tipos de defectos

No existe una clasificación de tipos de errores a nivel documentación, pero es importante conocer con qué tipo de defectos nos podemos encontrar y cómo trabajar sobre ellos para reportarlos.

* Defectos UX/UI
* Defectos visuales
* Defectos componente
* Defectos contenido
* Defectos disruptivos

### Defectos UX/UI

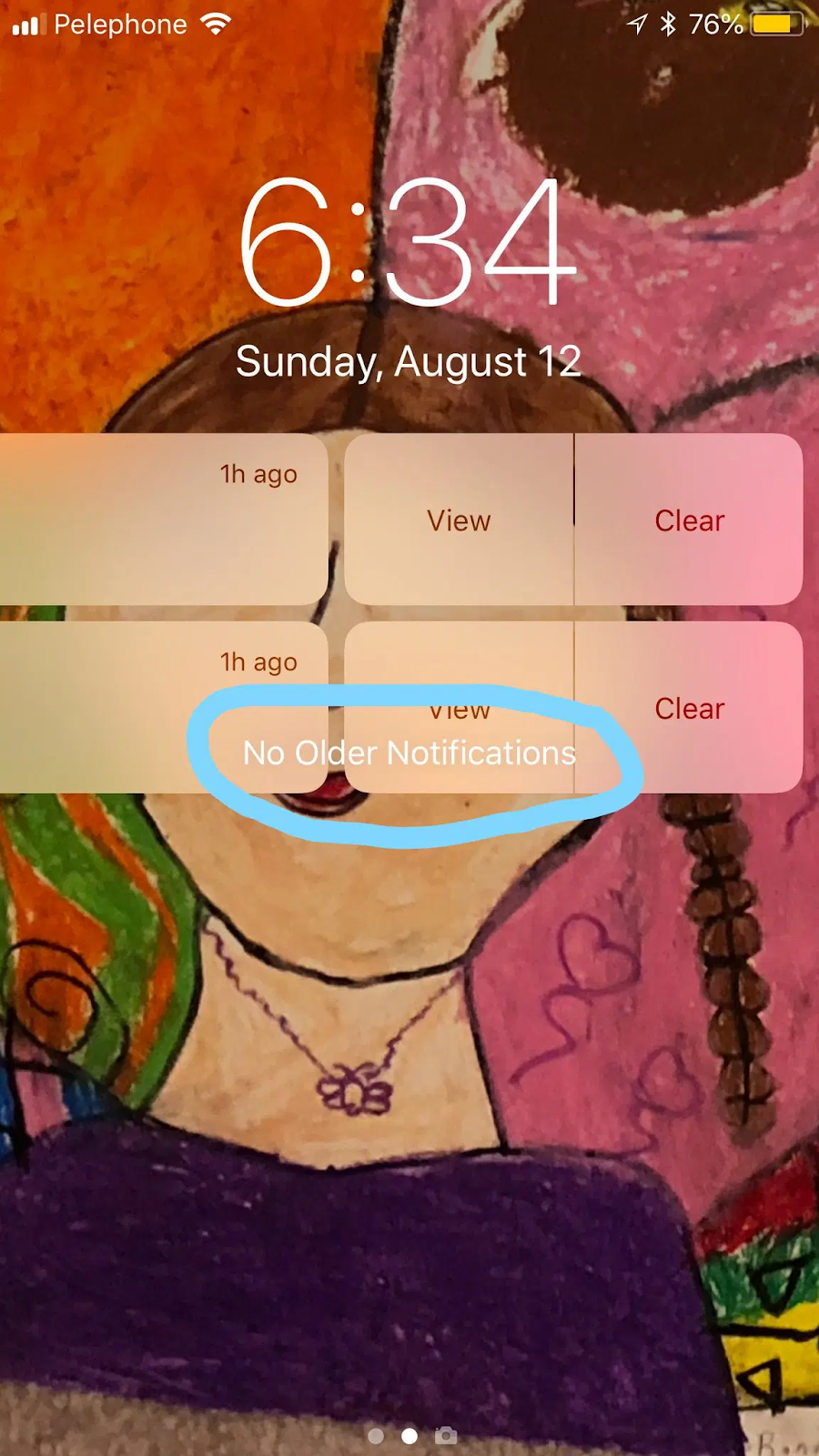
Los defectos UX/UI(también conocidos como visuales y de usabilidad) se pueden dar por una diferencia entre las maquetas entregadas a los desarrolladores y lo programado.

### Defectos visuales

Los defectos visuales (también conocidos como UX/UI) se pueden dar por una diferencia entre las maquetas entregadas a los desarrolladores y lo programado.

También puede haber defectos visuales relacionados con la usabilidad de la plataforma.

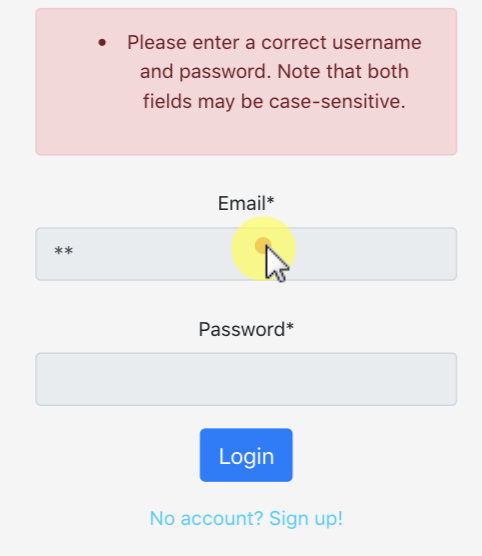
#### Ejemplos de defectos visuales



### Defectos componentes

Los defectos de los componentes son específicamente cuando algo particular de la plataforma no está funcionando como corresponde.

#### Ejemplo de defectos componentes

En la página del login vemos que al ingresar, no muestra nuestro email de forma correcta y tampoco alguna acción que nos permita visualizarlo.

Ese componente de forma aislada no funciona correctamente.

### Defectos contenido

Los defectos de contenido están medianamente relacionados con los defectos de UX/UI, pero se desprenden de ellos porque comprenden contenido específicamente en la plataforma, y no específicamente algo visual o de usabilidad.

Ejemplo: el contenido de legales está desactualizado con fechas del año anterior.

### Defectos funcionales

Este caso aplica específicamente cuando algo no funciona de acuerdo a lo que la documentación indica.

### Defectos disruptivos

El más grave de todos, que se da a causa de una nueva implementación que hace que toda la plataforma se rompa y no ande nada o nos envíe a una sección vacía.

Cualquier defecto que rompa toda la pantalla, es disruptivo.

## Reporte de bugs

### Función de reporte de defectos

El reporte de defectos es nuestra herramienta principal para demostrar el trabajo realizado.

### ¿Quién define el reporte del bug?

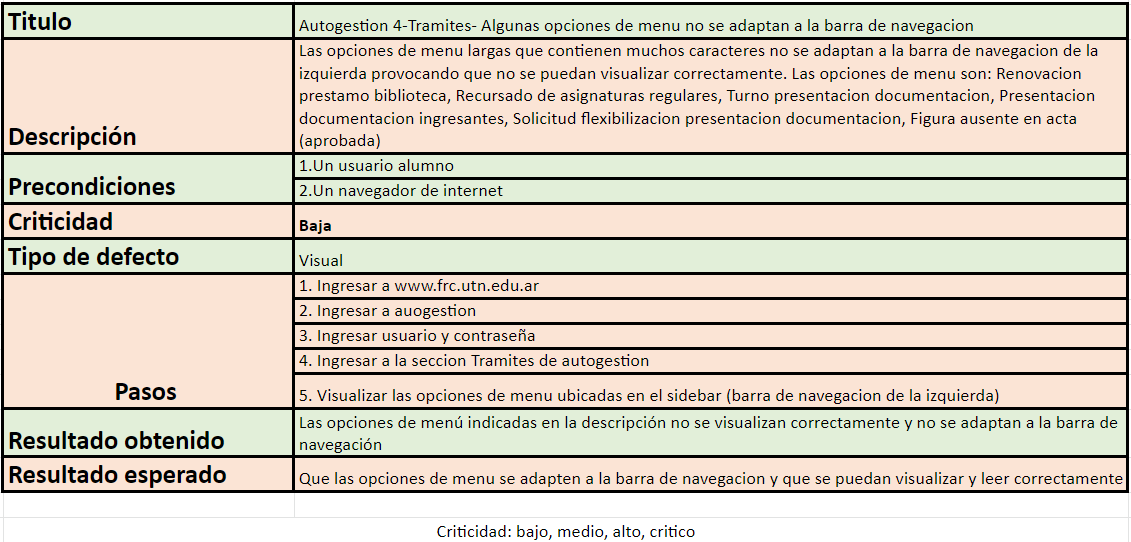
La definición de la información agregada en los reportes es un trabajo de equipo que se realiza previo al comienzo del ciclo del desarrollo. El equipo comprende a todas las entidades que pertenezcan, como desarrolladores, tester, product owners, sysadmins, diseñadores, etc.

### Características destacables

Algunos de los datos que NO pueden faltar en los reportes son:

* Resultado esperado
* Resultado obtenido
* Captura de pantalla del error
* Sistema utilizado (sistema operativo o equipo físico).
* Nivel del bug

#### Ejemplos de reporte de defecto



## Herramientas de reporte

Algunas de las herramientas de reportes de bugs son:

* Jira
* Bugzilla
* Mantis
* Gitlab

## Glosario

**Build:** Versión operativa de un producto de software que incorpora un subconjunto de las funciones que se incluirán en el producto final. Generalmente se identifica por un número de compilación, en lugar de por un número de versión.

**Error:** Es una desviación o algo que está mal y se encuentra en la etapa en la cual se originó.

Defecto: Es un desvío o algo que está mal y se encuentra en la etapa posterior en la que se originó.

**Falla:** Error detectado en un producto de trabajo.

**Criticidad:** es la medida del alcance del daño potencial que el incidente puede causar

**Urgencia:** la urgencia mide que tan inmediata es la necesidad de corregir un defecto en el sistema.

#### ¿QUIERES SABER MÁS? TE DEJAMOS MATERIAL AMPLIADO DE LA CLASE

* [Documento ISTQB en Español - Página 23 a 24](https://141647fc-7a5b-4407-a84a-78567bb3f5b0.filesusr.com/ugd/acfdb9_743bf3d31dde49578c94d97e5b96b9da.pdf)

#DEMOCRATIZANDOLAEDUCACIÓN