


# Tester QA Manual

Módulo 1



# Fundamentos del *testing*

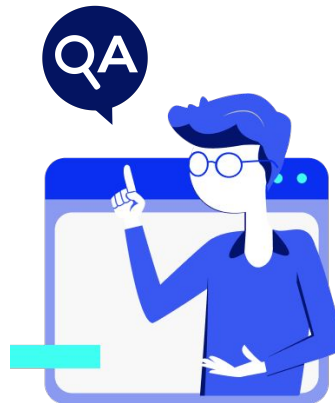
## Contenido

- Principio 1 – *La prueba muestra la presencia de defectos.*
  - Principio 2 – *La exhaustividad es imposible.*
  - Principio 3 – *Detección temprana de defectos.*
  - Principio 4 – *Agrupamiento de defectos.*
  - Principio 5 – *Paradoja del pesticida.*
  - Principio 6 – *Es dependiente del contexto.*
  - Principio 7 – *La ausencia de defectos es una falacia.*
- 




## Los 7 fundamentos o principios del testing

1. Muestra la presencia de defectos, no su ausencia.
2. La exhaustividad es imposible.
3. Detección temprana de defectos.
4. Agrupamiento de defectos.
5. Paradoja de los pesticidas.
6. Es dependiente del contexto.
7. La ausencia de defectos es una falacia.



## 1. Muestra la presencia de defectos

*“La prueba puede mostrar la presencia de defectos, pero no que no haya defectos. Se reduce la probabilidad de que queden defectos no descubiertos en el software pero, incluso si no se encuentran defectos, el proceso de prueba no es una demostración de la corrección.”* ISTQB



Como parte del proceso de pruebas, el tester interactuará con el sistema desde diferentes miradas buscando posibles desviaciones respecto a lo que el usuario solicitó. Uno de los objetivos del Testing es precisamente **evidenciar las fallas y gestionar su corrección** antes que el sistema llegue al usuario final.



## 2. La exhaustividad no es posible

El *testing* tiene diferentes miradas: un sistema puede probarse tomando en cuenta la precisión en los cálculos, el tamaño de los botones, las validaciones que hace, etc.

Aunque el sentido común lleva al *Tester* a querer considerar la prueba de todas las posibles combinaciones para evaluar la funcionalidad de un sistema, esto no es factible porque **las pruebas pueden ser infinitas** y porque en general tendrá un tiempo limitado.

Ante ello el *Tester* tendrá que **decidir qué probar**.

Veámoslo a través de un ejemplo:

*¿Cuántos números tendría para probar un DNI?*

*Tal vez: 1,2,3, 890.999,... 27.545.455,...*

**Respuesta:** Para asegurar que cumpla estaría realizando **millones de pruebas**.

**Nota:** De aquí el empleo de técnicas para minimizar drásticamente la cantidad.

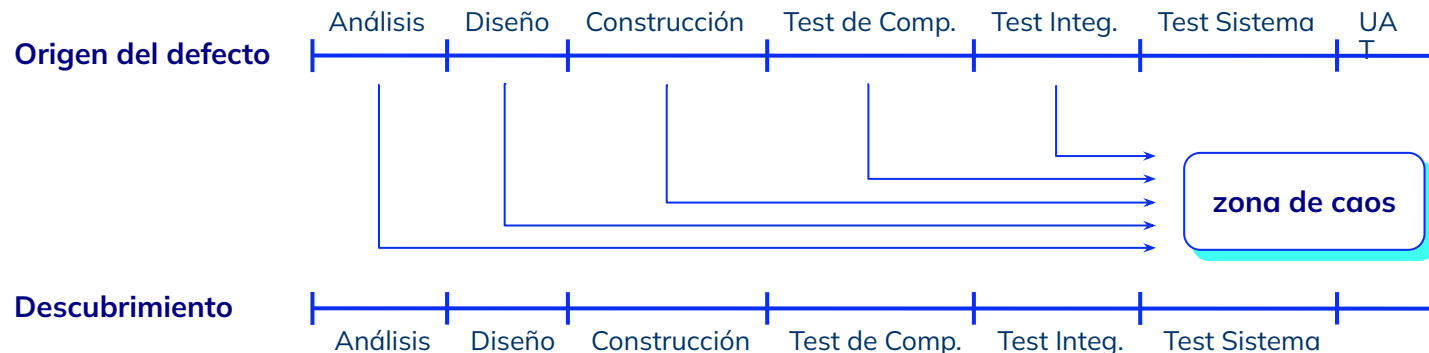
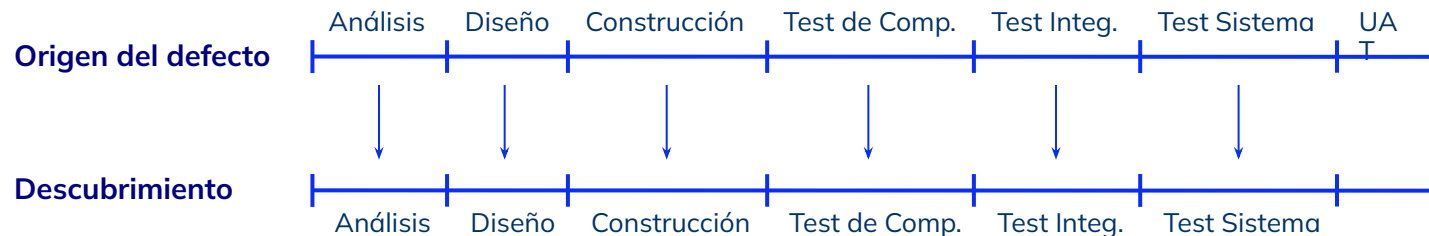


### 3. Detección temprana

El proceso de *testing* más efectivo no es aquel que logra identificar más fallas, sino el que logra que el sistema que llega a manos del usuario tenga la **mejor calidad posible**.

Para ello, el *Tester* tendrá que comenzar sus actividades de prueba lo más pronto posible y reportar oportunamente cualquier desviación respecto a lo que el usuario espera, para que haya mayor probabilidad que el **sistema sea corregido antes de finalizar las pruebas**.

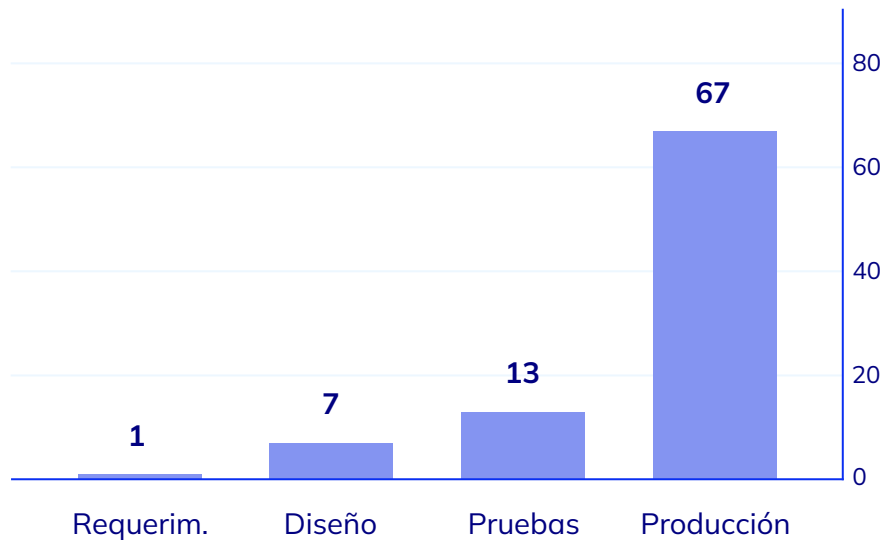


**Sin control  
en etapas****Con control  
en etapas**



### Ejercicio: *¿Qué es más costoso?*

1. El *Tester* encontrando los defectos en las pruebas de unitarias.
2. El *Tester* encontrando los defectos cuando prueba a nivel sistema.
3. El *usuario* encontrando los defectos en producción.
4. El *usuario* encontrando los defectos en pruebas de aceptación.



#### 4. Agrupamiento de defectos

Los defectos que encuentra el *Tester* en un sistema no estarán distribuidos de forma equitativa en las diferentes funcionalidades.

Usualmente estarán concentradas en ciertas funcionalidades o aspectos del sistema: aquellos que fueron modificados por el desarrollador, aquellos que tuvieron mayor nivel de incertidumbre entre otros factores.

Por ello se recomienda que, **al encontrar una falla en una funcionalidad, se siga evaluando porque probablemente tenga alguna más.**

**Nota:** Es común ver que a mayor complejidad, mayor nivel de defectos se encuentran.



## 5. Paradoja del pesticida

*“Si las pruebas se repiten una y otra vez, eventualmente estas pruebas no encontrarán nuevos defectos. Para detectar nuevos defectos es posible que sea necesario cambiar a nuevas pruebas y a nuevos datos de pruebas.*

*Las pruebas ya no son efectivas del mismo modo que los pesticidas ya no son efectivos para matar insectos después de un tiempo.”*

ISTQB

Si bien se recomienda reutilizar las pruebas que se diseñen para evaluar cierta funcionalidad del sistema, utilizar exactamente las mismas disminuye su efectividad. Esto ocurre porque los sistemas cambian y se adaptan al negocio y así, **las pruebas deberían mantenerse actualizadas para seguir siendo efectivas.**



## 6. Depende del contexto

*“Las pruebas se realizan de manera diferente según el contexto. Por ejemplo, la prueba se desarrolla de manera diferente en marco predecible que en framework Agile.”*



ISTQB

Algunos de los factores que afectan las pruebas serían:

- **Metodología de desarrollo.**
- **Entorno** (sistema operativo, navegador, etc.)

Por ello y tomando en cuenta otro de los principios, el *Tester* tendrá que definir qué probar según el caso.

**Por ejemplo:** en una metodología ágil, seguramente sólo habrá tiempo para probar un sistema en los dos navegadores más utilizados.



## 7. La ausencia de defectos es una falacia

*“La ausencia de errores es una falacia (es decir, una creencia equivocada) conforme a que las pruebas contribuyen a minimizar los defectos en un gran número, pero eso no indica que se realice al 100%.”*

ISTQB



El proceso de pruebas transmite confianza a los usuarios e involucrados con el sistema, no obstante **en ningún momento se puede afirmar que un sistema está libre de fallas.**

En general, el *testing* evaluará el sistema y dará visibilidad sobre su calidad. A lo largo del proceso se identificarán y corregirán las fallas que se detecten pero, como no se puede probar todo, no se puede dar una garantía de 0 defectos.

## ¿Cómo utilizar los principios del testing?

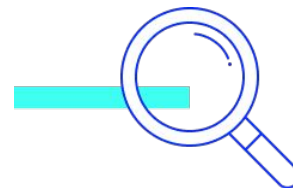
No es necesario memorizar los principios de las pruebas, estos seguramente serán utilizados como parte del sentido común del *Tester*:

- Al orientar sus actividades hacia la identificación de fallas tan temprano como sea posible.
- Al dar visibilidad sobre el estado de las pruebas.
- Al evaluar un sistema en función al contexto.
- Al indagar en aquellas funcionalidades donde se detecta una falla.
- Al decidir cuáles son las pruebas más importantes.



# Revisión

- Repase los puntos vistos en la clase.
- Realice las preguntas necesarias al o la docente antes de continuar.
- Realice los ejercicios de la práctica.



**¡Sigamos  
trabajando!**

