

PRUEBAS DE SOFTWARE





Liliana Rangel Cano
lilianarc@unam.mx



MITOS



- × Los *testers* no requieren de formación
- × Las pruebas de software son actividades destructivas sin un objetivo específico
- × Al realizar *testing* se cuenta con un sistema sin bugs, robusto y cumple con las expectativas del cliente
- × El software tiene la misma calidad al realizar o no testing
- × La calidad del software es responsabilidad del *tester*



¿QUÉ ES PROBAR?

“Actividad realizada para evaluar y mejorar la calidad del producto, identificando defectos y problemas”

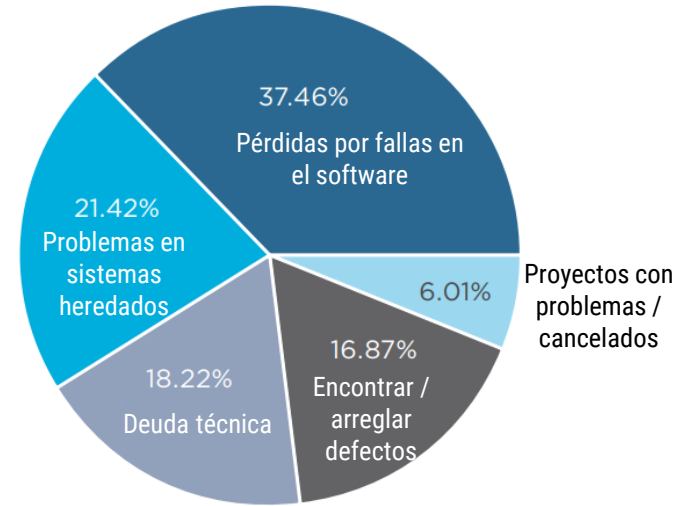
[SWEBOK]



¿PORQUÉ PROBAR?



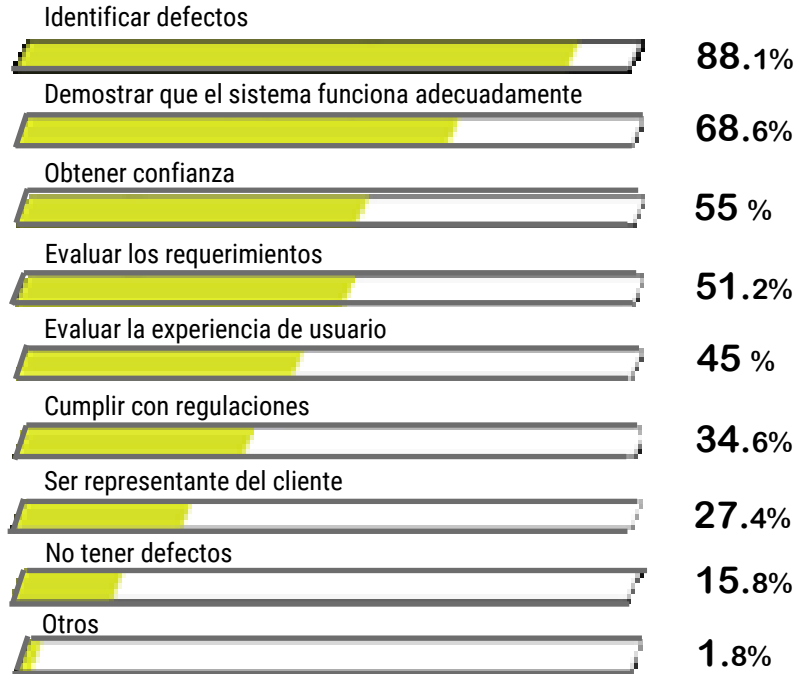
En 2017 se reportaron:
606 fallos de 314 empresas
1.7 billones en pérdidas financieras
3.6 mil millones de personas afectadas



En EUA las pérdidas del 2018 causadas por software con baja calidad ascienden es de aproximadamente \$ 2.84 billones

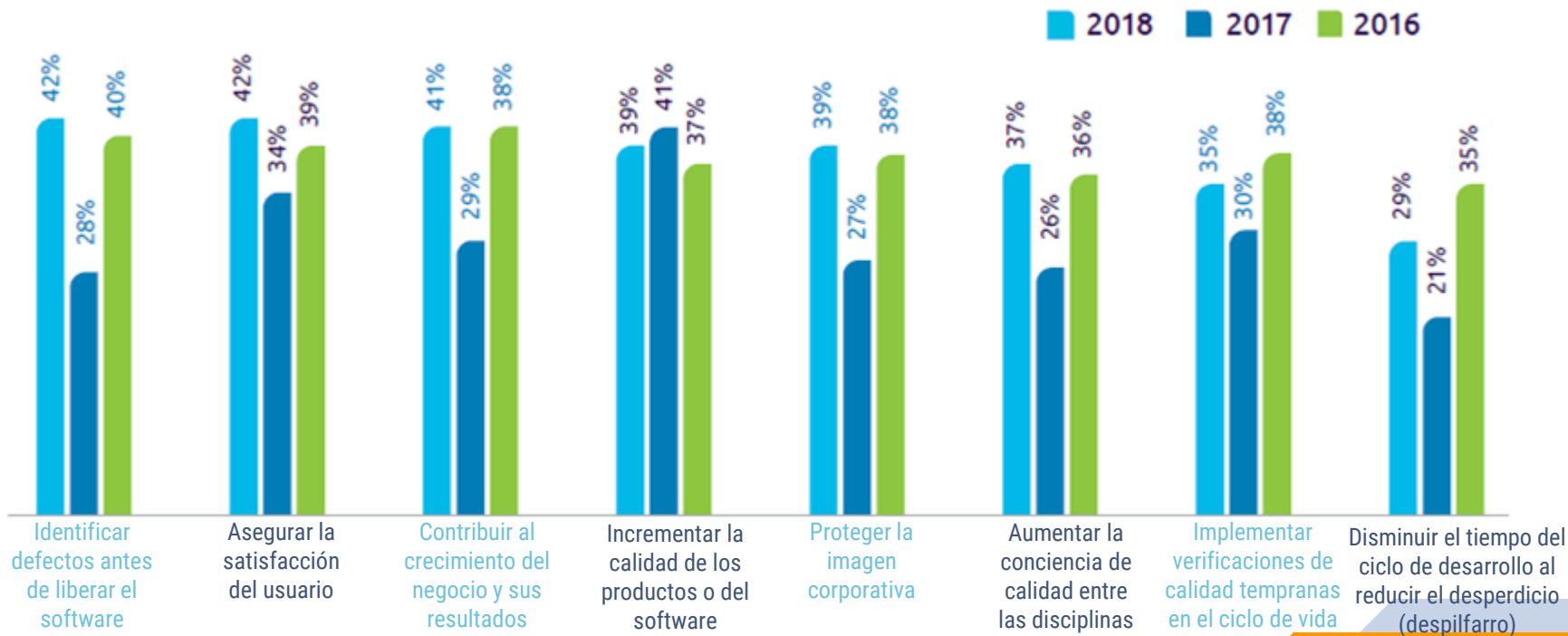


OBJETIVOS





OBJETIVOS DE QA Y PRUEBAS A NIVEL EJECUTIVO



**LAS PRUEBAS REVELAN
LA PRESENCIA DE
DEFECTOS, NO SU
AUSENCIA**



Te revisado cada metro cuadrado en esta casa, y puedo decir con confianza que no hay ratones aquí



Ausencia de pruebas no es prueba de ausencia

-William Cowper

**ES IMPOSIBLE
REALIZAR PRUEBAS
EXHAUSTIVAS**

Te realizado todas las pruebas, tenemos una cobertura del 100%



Pinocho nunca entendió por qué su nariz crecía cuando hablaba de cobertura

Andy Glover cartoonfester.blogspot.com Copyright 2011

Los defectos son criaturas sociales

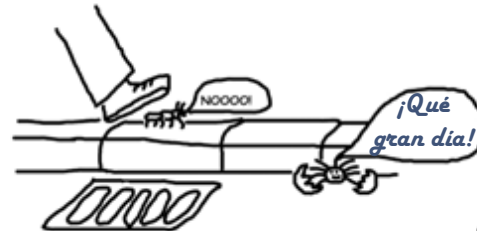
A menudo se dejan ver en grupos



**AGLOMERACIÓN DE
DEFECTOS**

Las pruebas deben concentrarse de manera proporcional en la densidad esperada

*Los defectos son criaturas religiosas
Cuando uno muere (se elimina), puede reencarnar en otro diferente*



Andy Glover cartoonfester.blogspot.com copyright 2010

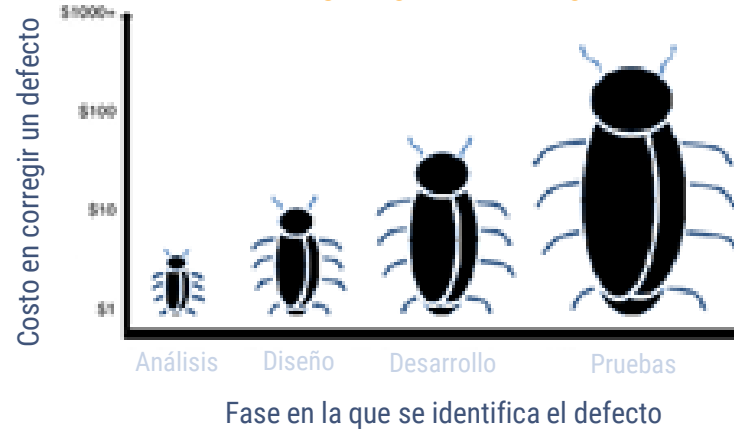
**PARADOJA DEL
PESTICIDA**

Cada ciclo de pruebas debe cambiar la estrategia, escribir nuevas y diferentes pruebas

LAS PRUEBAS DEBEN ADAPTARSE A LAS NECESIDADES DEL PROYECTO



PRUEBAS TEMPRANAS



Costo de corrección de defectos:

- 1 prevención,
- 10 corrección,
- 100 fallo



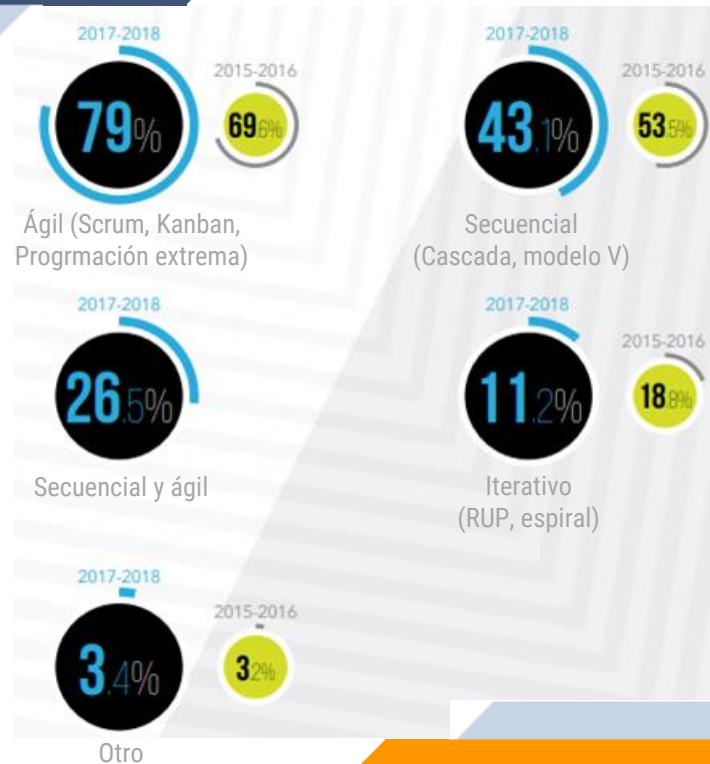
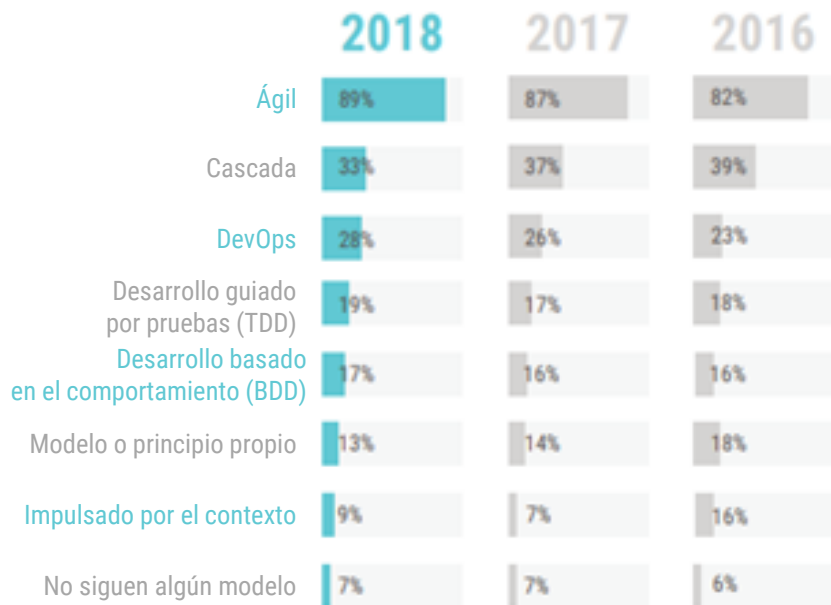
**¡CIELOS!
PESO LO MISMO
QUE UNA
PERSONA DE
DOS METROS
CUARENTA ...
¡TENGO QUE
CRECER
URGENTEMENTE!**

FALACIA DE LA AUSENCIA DE ERRORES

Identificar y corregir defectos no sirve de nada si el software no es usable, y no cumple con las expectativas y necesidades de los usuarios



MODELO DE DESARROLLO



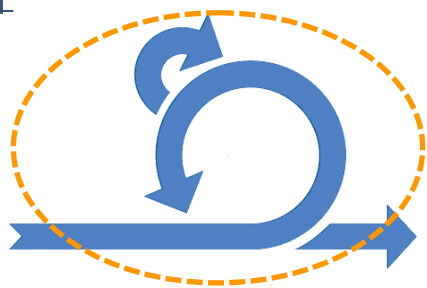
State of Testing Report © 2018 – PractiTest & Tea Time with Testers.
https://qablog.practitest.com/wp-content/uploads/2018/07/2018_state_of_testing_report_1.2.pdf

ISTQB @ Worldwide Software Testing Practices Report 2017 – 2018
<https://www.istqb.org/references/surveys/istqb@-worldwide-software-testing-practices-survey-2017-18.html>



INTERVENCIÓN DE LAS PRUEBAS

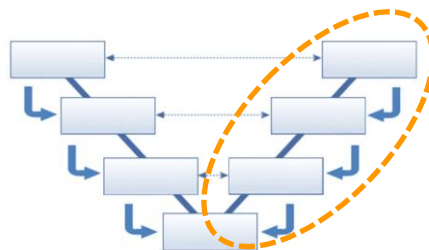
ÁGIL



Durante la realización de cada iteración

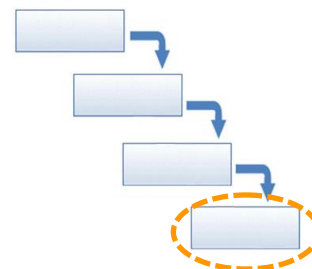
- Testers como parte del equipo de trabajo en todo el proyecto
- Inclusión del testers en fases específicas
- Aplicación de pruebas por un equipo separado e independiente

MODELO V



- Al término de cada fase del ciclo de vida

CASCADA



- Al final del ciclo de vida de desarrollo



QUIEN REALIZA LAS PRUEBAS



Nivel del objetividad

✓ Estáticas

* Preventivas

22%

✓ Análisis estáticos

✓ Revisiones
(IEEE 1028)

- ✓ Revisión informal
- ✓ Revisión técnica
- ✓ Revisión guiada
- ✓ Inspección

- ✓ Basadas en la experiencia
- ✓ Regresión
- ✓ Confirmación
- ✓ Pruebas de humo (smoke test)
- ✓ Pruebas exploratorias

53.3%

✓ Caja blanca

- ✓ Componente
- ✓ Integración 46.1%
- ✓ Unitarias

- ✓ Cumplimiento a requerimientos
- ✓ Visibilidad / Accesibilidad 28.2%
- ✓ Interoperabilidad
- ✓ Fiabilidad
- ✓ Portabilidad
- ✓ Mantenimiento 19.3%

✓ Dinámicas

* Correctivas

✓ Caja negra

✓ Funcional
83%

✓ Desempeño
60.7%

- ✓ Carga
- ✓ Volumen
- ✓ Estrés
- ✓ Concurrencia

✓ No funcional

✓ Usabilidad
44.1%

- ✓ Heurísticas
- ✓ Con usuarios

✓ Aceptación
66%

✓ Seguridad
44.6%

- ✓ Alpha
- ✓ Beta



PRUEBAS APLICADAS EN LOS MODELOS

ÁGIL



Estáticas

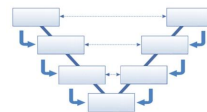
- » ~~Revisiones~~
- » Compilaciones

Dinámicas

- » Pruebas unitarias
- » Pruebas de integración
- » Pruebas funcionales
- » Pruebas de desempeño
- » Pruebas usabilidad
- » Pruebas de aceptación



MODELO V



Estáticas

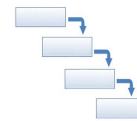
- » Revisiones
- » Compilaciones

Dinámicas

- » Pruebas unitarias
- » Pruebas de integración
- » Pruebas funcionales
- » Pruebas de desempeño
- » Pruebas usabilidad
- » Pruebas de aceptación



CASCADA



Estáticas

- » Revisiones
- » Compilaciones

Dinámicas

- » ~~Pruebas unitarias~~
- » ~~Pruebas de integración~~
- » Pruebas funcionales
- » Pruebas de desempeño
- » Pruebas usabilidad
- » Pruebas de aceptación





ESTRATEGIAS DE PRUEBAS

Basada en riesgos de producto

Identificación de riesgos, clasifican y priorizan

Analítica

Pruebas basadas en requisitos y flujos completos

Basada en modelos

Metódica, basada en procesos o estándares

Reactiva

Las pruebas responden a la puntualidad en la entrega del ambiente, y el tiempo disponible para realizarse

Consultiva

Las pruebas pueden ser dirigidas por el usuario, o pueden ser por pares

Aversión a regresión

Automatización de pruebas de regresión, se aplican cuando se modifica el software

Preventiva

Pruebas estáticas



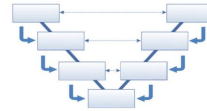
ESTRATEGIAS DE PRUEBAS

ÁGIL



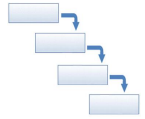
- » Analítica
- » ~~Basada en modelos~~
- » ~~Metódica/Procesos o estándares~~
- » ~~Reactiva~~
- » Consultiva
- » Aversión a regresión
- » Preventiva
- » **Basada en riesgos**

MODELO V



- » **Analítica**
- » Basada en modelos
- » Metódica/Procesos o estándares
- » ~~Reactiva~~
- » ~~Consultiva~~
- » Aversión a regresión
- » Preventiva
- » Basada en riesgos

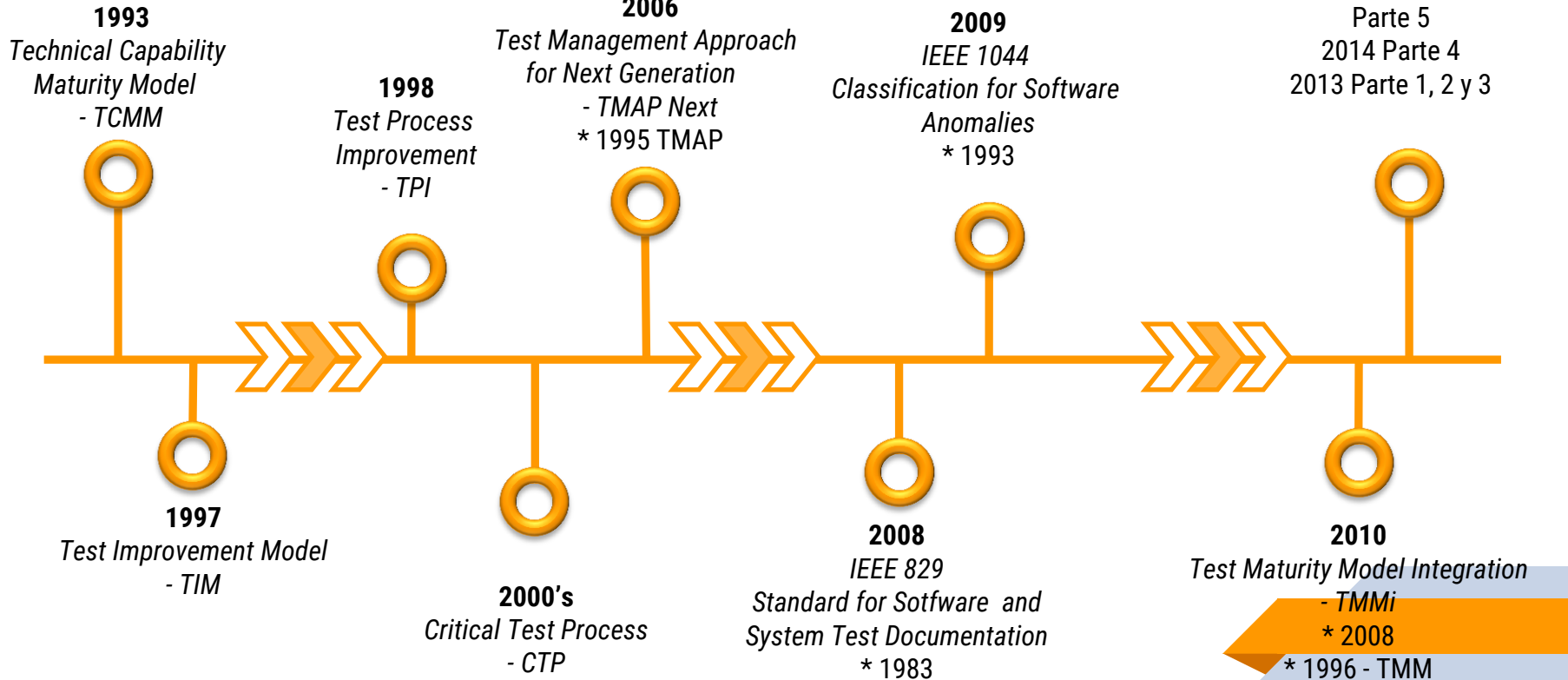
CASCADA



- » **Analítica**
- » Basada en modelos
- » Metódica/Procesos o estándares
- » ~~Reactiva~~
- » ~~Consultiva~~
- » ~~Aversión a regresión~~
- » ~~Preventiva~~
- » Basada en riesgos



ESTÁNDARES Y MARCOS DE REFERENCIA





ESTÁNDARES Y MARCOS DE REFERENCIA

IEEE 1008

*Standard for Software
Unit Testing*

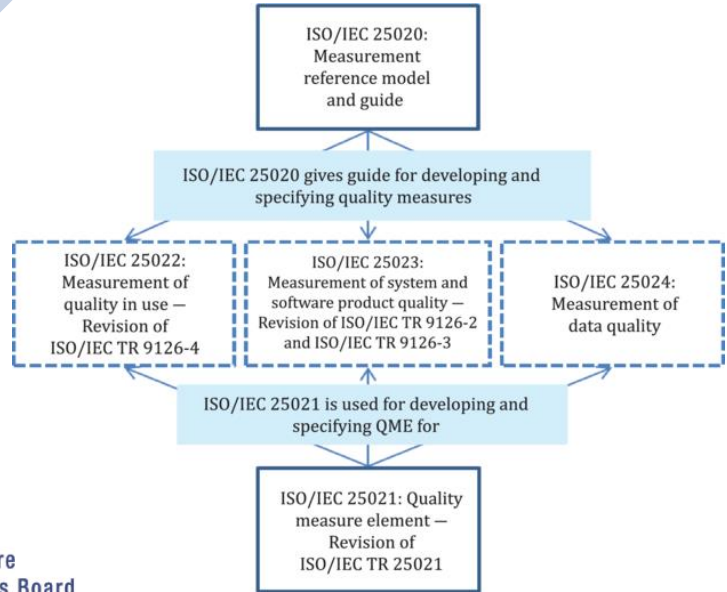
BS 7925

*Standard for Software
Component Testing*



Syllabus

MAAGTIC-SI





TÉCNICAS DE DISEÑO DE PRUEBAS



Seleccionar los casos de prueba, que permitan reportar el mayor número de defectos, mediante la identificación de las condiciones de pruebas, casos y datos de prueba

Se clasifican de acuerdo con la fuente de información que se usa para el diseño de los casos de prueba:

Basadas en la especificación / comportamiento (caja negra)

- » Partición de equivalencias
 - » Valores límite
 - » Tablas de decisión
 - » Transición de estados
 - » Casos de uso
- Aplicables a las pruebas dinámicas (implican la ejecución del código)
- Basadas en un análisis de las bases de pruebas

Basadas en la estructura o en el código (caja blanca)

- » Cobertura de sentencia
- » Cobertura de rama
- » Cobertura de camino
- » Cobertura de condición

Basadas en la experiencia

- » Pruebas exploratorias
 - » Predicción de errores
- Trabajan en conjunto con las anteriores
- Basadas en la experiencia, derivan y seleccionan las pruebas basadas en el conocimiento y experiencia del probador



TÉCNICAS BASADAS EN LA ESTRUCTURA O EN EL CÓDIGO – CAJA BLANCA

Cobertura de sentencias

Su objetivo es ejecutar al menos una vez cada sentencia de código

¿Qué valores de prueba necesitamos para cubrir todas las instrucciones?

21.6%

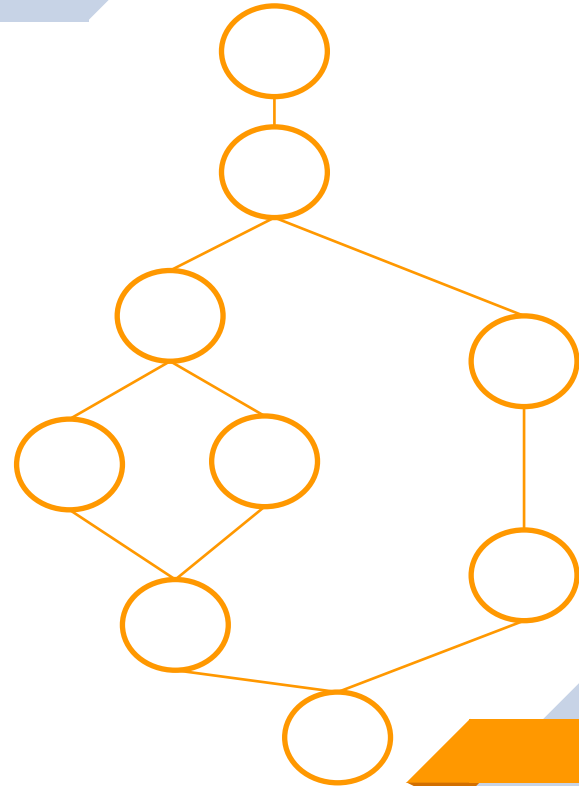
```
1 #include <stdio.h>
2 main()
3 {
4     int i, n, f;
5     printf («n = »);
6     scanf («%d», &n);
7     if (n < 0) {
8         printf («Invalid: %d\n», n);
9         n = - 1;
10    } else {
11        f = 1;
12        for (i = 1; i <= n; i++) {
13            f * = i;
14        }
15        printf («%d! = %d \n», n, f);
16    }
17    return n;
18 }
```



TÉCNICAS BASADAS EN LA ESTRUCTURA O EN EL CÓDIGO – CAJA BLANCA

Cobertura de ramas / decisión

El objetivo es ejecutar al menos una vez cada rama (decisión) tomada en cada sentido, verdadero y falso





TÉCNICAS BASADAS EN LA ESTRUCTURA O EN EL CÓDIGO – CAJA BLANCA

Cobertura de condición

(simple, múltiple, múltiple simplificada)

Simple:

Cada condición será evaluada para verdadero como para falso.

Múltiple:

Cada combinación será evaluada de las condiciones verdaderas y falsas (tabla de verdad).

Múltiple simplificada:

Solamente las combinaciones de condiciones que pueden influir en la decisión.

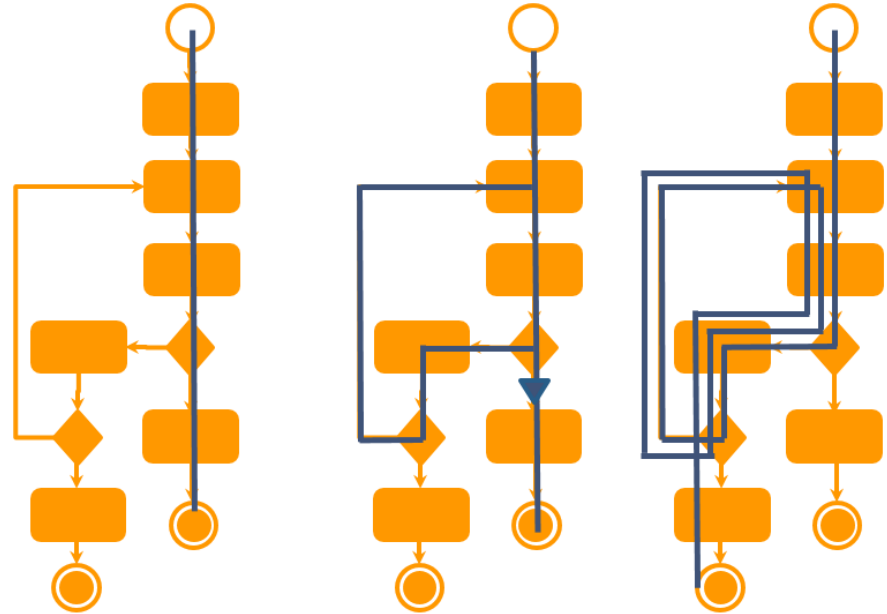


TÉCNICAS BASADAS EN LA ESTRUCTURA O EN EL CÓDIGO – CAJA BLANCA

Cobertura de caminos y cobertura de bucle

Todos los caminos del bucle tomados, cero, una vez, múltiples (idealmente el máximo) veces

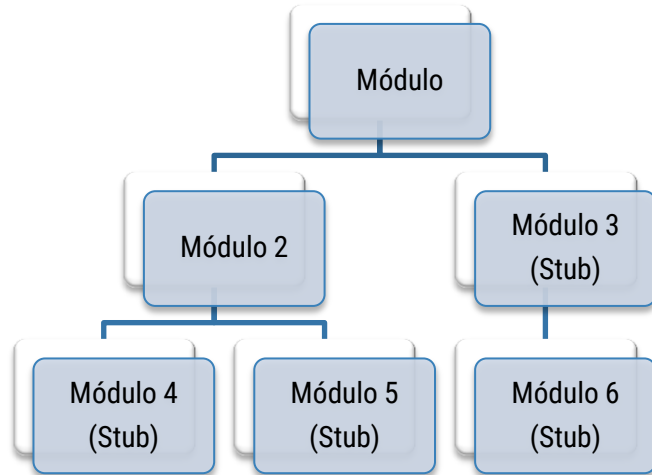
- Máximo -1
- Máximo +1
- Valor típico del bucle





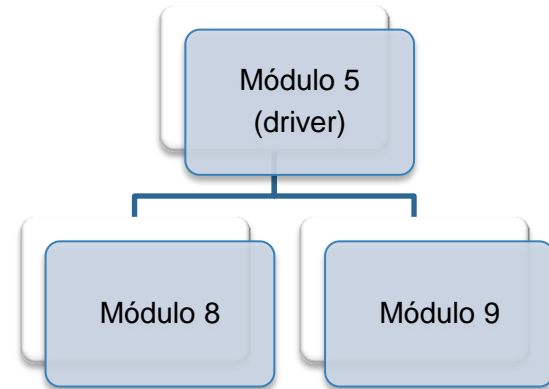
TÉCNICAS BASADAS EN LA INTEGRACIÓN

Top down



Prueba primero las rutinas de control de menor profundidad

Bottom up



Prueba las rutinas de control de mayor profundidad



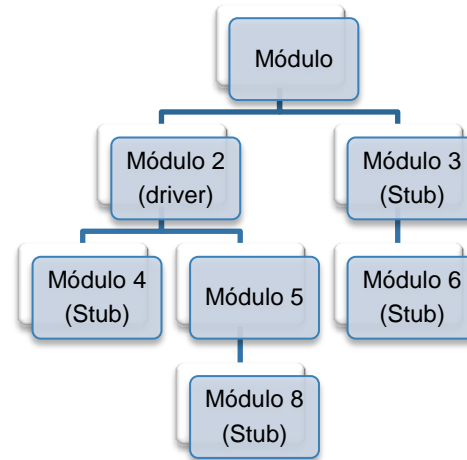
TÉCNICAS BASADAS EN LA INTEGRACIÓN

Big bang



Ensambla todos los módulos y corre las pruebas al sistema completo

Middle-out / Sandwich



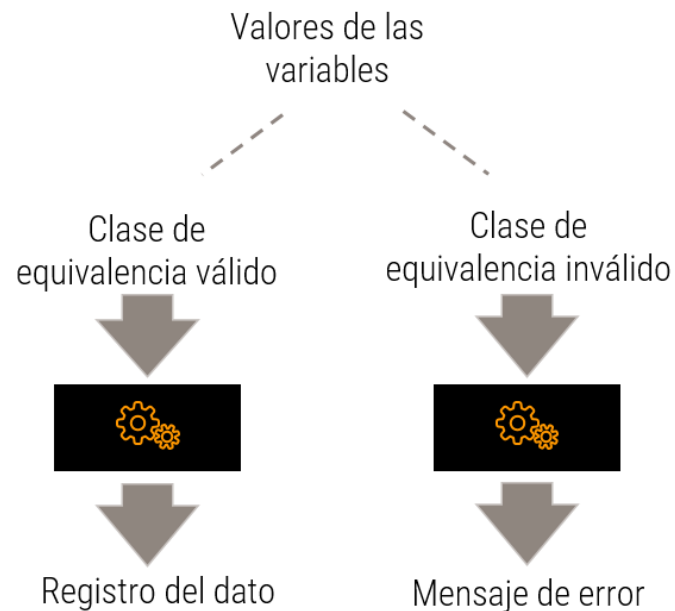
Navega a través de la funcionalidad, haciendo énfasis en los módulos relevantes



TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Partición de equivalencias

Una clase de equivalencia representa al conjunto de valores válidos o inválidos para todas las condiciones de entrada que serán tratadas de manera similar



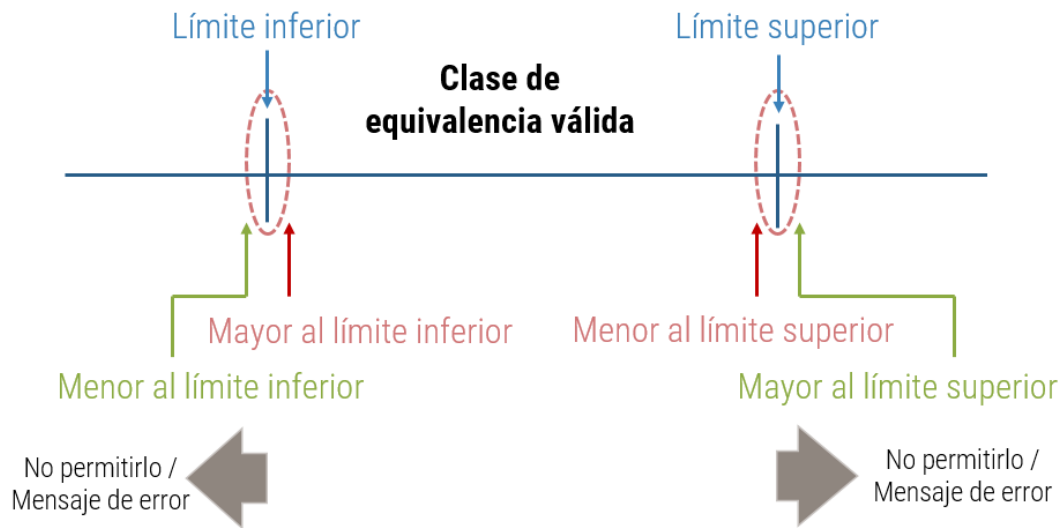
36%



TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Análisis de valores límite

Identifica defectos funcionales en condiciones lógicas, en la definición de los límites / fronteras / extremos de la clase



52.3%



TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

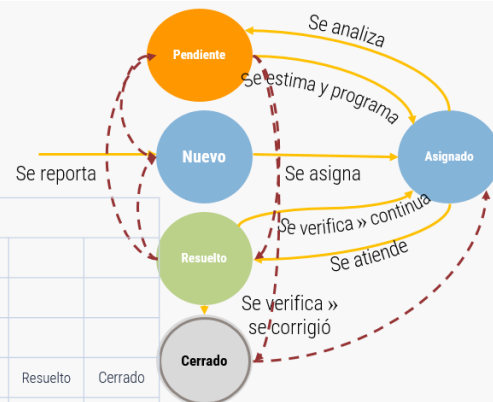
Transición de estados

El objeto tiene un ciclo de vida y queremos probar la capacidad del software bajo los diversos estados definidos a través de transiciones válidas y no válidas

EJEMPLOS

Ciclo de vida de un defecto

| CASO | ESTADOS | | | | | | | |
|------|---------|----------|----------------------|-----------|----------|----------|----------|---------|
| 1 | Nuevo | Asignado | Resuelto | Cerrado | | | | |
| 2 | Nuevo | Asignado | Resuelto | Asignado | Resuelto | Cerrado | | |
| 3 | Nuevo | Asignado | Pendiente | Asignado | Resuelto | Cerrado | | |
| 4 | Nuevo | Asignado | Pendiente | Asignado | Resuelto | Asignado | Resuelto | Cerrado |
| 5 | Nuevo | Asignado | Pendiente / Resuelto | Nuevo | Asignado | Resuelto | Cerrado | |
| 6 | Nuevo | Asignado | Resuelto | Pendiente | Asignado | Resuelto | Cerrado | |
| 7 | Nuevo | Asignado | Pendiente | Resuelto | Cerrado | | | |
| 8 | Nuevo | Asignado | Resuelto | Cerrado | Asignado | | | |





TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Gráficas de causa - efecto

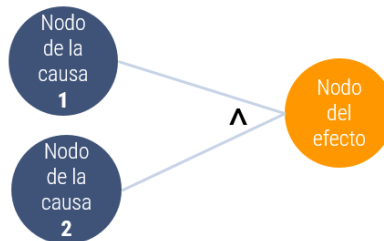
Representa la relación lógica entre un efecto y sus posibles causas de forma ordenada, clara y precisa



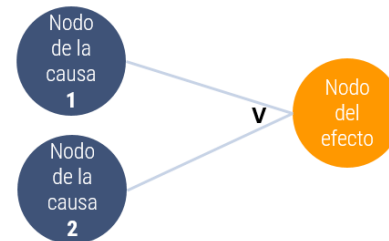
Afirmación » Si causa – entonces efecto



Negación » No causa – entonces efecto



Y » Si causa 1 y 2 – entonces efecto



O » Si causa 1 ó 2 – entonces efecto



TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Tablas de decisiones

Identifica las posibles condiciones, valores o variables de entrada y las posibles acciones de salida

Las reglas de negocio pueden especificarse en decisiones

Reglas [Definen una única combinación de condiciones]

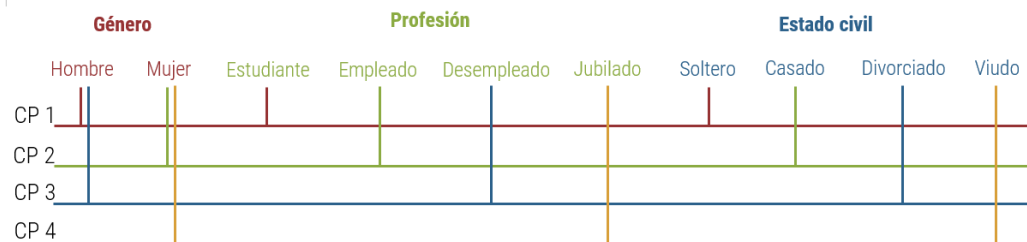
| | Regla A | Regla B | Regla N |
|--|----------------|---------|---------|
| Condiciones [Variables o datos de entrada] | | | |
| | | | |
| | | | |
| | Caso de prueba | | |
| Acciones [Resultados esperados – dependen de las condiciones de entrada] | | | |
| | | | |
| | | | |



TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Método de clasificación de árbol

Permite definir la combinación
de pruebas dependiendo del
máximo de parámetros
relacionando sus valores





TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Combinacional & Pairwise

Describe casos de prueba para ejecutar combinaciones de representantes de los dominios de entrada y/o salida

| | Parámetros | | |
|---------|------------|-----|-----|
| | C1 | C2 | C3 |
| Valores | 1 | 1 | 1 |
| | 1 | 1 | 2 |
| | 1 | 1 | 3 |
| | 1 | 1 | 4 |
| | 1 | 1 | 5 |
| | 1 | 1 | 6 |
| | 2 | 1 | 1 |
| | 2 | 1 | 2 |
| | ... | ... | ... |

| C1 | C2 | C1 | C3 | C2 | C3 |
|-----|-----|-----|-----|-----|-----|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 2 | 1 | 2 |
| 1 | 1 | 1 | 3 | 1 | 3 |
| 1 | 1 | 1 | 4 | 1 | 4 |
| 1 | 1 | 1 | 5 | 1 | 5 |
| 1 | 1 | 1 | 6 | 1 | 6 |
| 2 | 1 | 2 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 | 1 | 2 |
| ... | ... | ... | ... | ... | ... |





TÉCNICAS BASADAS EN LA ESPECIFICACIÓN / COMPORTAMIENTO (CAJA NEGRA)

Pruebas de casos de uso

Los casos de uso describen el conocimiento del negocio y la interacción entre el usuario y el sistema

Identificar defectos en los flujos de información del sistema en su uso real, en la integración de diferentes componentes

| | |
|------------------|---|
| NOMBRE | Acceso al sistema |
| Descripción | Un usuario debe registrarse para hacer uso del sistema, y para ello debe hacer "login" con su usuario y contraseña. Si no cuenta con tal, debe registrarse en el sistema creando su cuenta |
| Actores | Usuario mediante la interfaz web |
| Precondiciones | El usuario debe estar registrado en el sistema |
| Flujo principal | <ol style="list-style-type: none">1. El usuario accede al sistema en la URL principal2. El sistema solicita credenciales3. El usuario ingresa proporcionando usuario y contraseña4. El sistema valida las credenciales del usuario y le da la bienvenida |
| Flujos alternos | <p>Flujo alternativo 1.</p> <ol style="list-style-type: none">1. El usuario no recuerda su contraseña por lo que solicita que se le envíen por correo electrónico2. El sistema solicita el correo electrónico y envía una nueva clave temporal al correo <p>Flujo alternativo 2.</p> <ol style="list-style-type: none">1. El usuario no está registrado en el sistema por lo que solicita crear una cuenta2. El sistema solicita los datos necesarios para crear la cuenta3. El usuario ingresa los datos y confirma4. El sistema crea la cuenta de usuario |
| Excepciones | <p>E1. Usuario y contraseña incorrectos. Si sucede en 3 ocasiones consecutivas la cuenta del usuario se bloquea por seguridad</p> <p>E2. (A1) El correo electrónico proporcionado no está registrado en el sistema. El sistema notifica el error</p> |
| Post-condiciones | El usuario accede al sistema y se registra su acceso en la tabla de registro de actividad |

73%



TÉCNICAS BASADAS EN LA EXPERIENCIA

Predicción de errores

Basados en errores encontrados en el pasado o la suposición de errores futuros



36%

Exploratorias

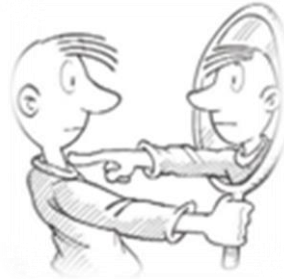
El primer contacto con el sistema es para conocer cómo funciona, saber de qué trata



67.2%

* Benchmarking

Comparación referencial, es un **análisis comparativo** planeado y sistemático de los procesos o metodologías que llevan a cabo las otras empresas





TÉCNICAS DE DISEÑO DE PRUEBAS

ÁGIL



~~Basadas en la especificación (caja negra)~~

- ~~» Partición de equivalencias~~
- ~~» Valores límite~~
- ~~» Tablas de decisión~~
- ~~» Transición de estados~~
- ~~» Casos de uso~~

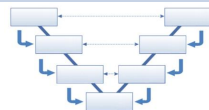
~~Basadas en la estructura (caja blanca)~~

- ~~» Cobertura de sentencia~~
- ~~» Cobertura de rama~~
- ~~» Cobertura de camino~~
- ~~» Cobertura de condición~~

Basadas en la experiencia

- » Pruebas exploratorias
- » Predicción de errores
- » Basadas en la experiencia

MODELO V



Basadas en la especificación (caja negra)

- » Partición de equivalencias
- » Valores límite
- » Tablas de decisión
- » Transición de estados
- » Casos de uso

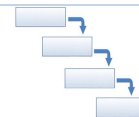
Basadas en la estructura (caja blanca)

- » Cobertura de sentencia
- » Cobertura de rama
- » Cobertura de camino
- » Cobertura de condición

Basadas en la experiencia

- » Pruebas exploratorias
- » Predicción de errores
- » Basadas en la experiencia

CASCADA



Basadas en la especificación (caja negra)

- » Partición de equivalencias
- » Valores límite
- » Tablas de decisión
- » Transición de estados
- » Casos de uso

~~Basadas en la estructura (caja blanca)~~

- ~~» Cobertura de sentencia~~
- ~~» Cobertura de rama~~
- ~~» Cobertura de camino~~
- ~~» Cobertura de condición~~

Basadas en la experiencia

- » Pruebas exploratorias
- » Predicción de errores
- » Basadas en la experiencia



TÉCNICAS API TESTING

Prueba funcional básica

Evalúa el flujo básico, considerando funciones cuyo resultado genere los resultados esperados

Condición negativa

Valida mecanismos para el manejo de errores y excepciones, la API debe funcionar ante lo inesperado

Recursos accedidos por la API

Evaluar llamadas a la API que impliquen actualizaciones (registros, personalizar, entre otros), y verificar el comportamiento a partir del cambio

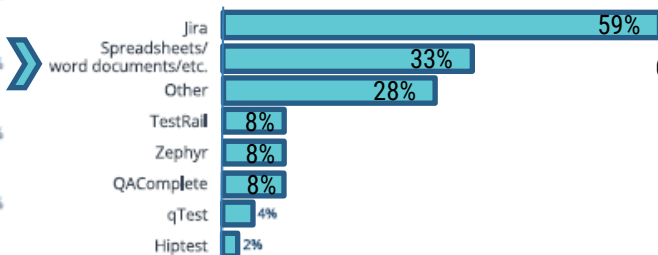
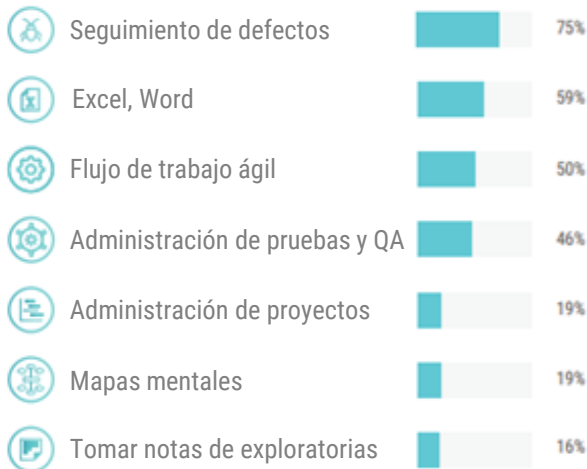
Instalación, configuración y ajustes

Selección de parámetros y valores de retorno

Ejecutar llamadas a funciones/métodos de la API aplicando una selección de datos y generando valores de respuesta (vacíos, null, uno, dos o más valores, entre otros)



HERRAMIENTAS



Seguimiento de defectos



Automatización



Ejecución de pruebas



Gestión de pruebas



Pruebas de desempeño



Trazabilidad de requerimientos



Diseño de pruebas



Pruebas unitarias



Análisis estático



Análisis dinámico



Otras

State of Testing Report © 2018 – PractiTest & Tea Time with Testers.
https://qablog.practitest.com/wp-content/uploads/2018/07/2018_state_of_testing_report_1.2.pdf

The State of Testing 2018, Smartbear
<https://smartbear.com/resources/ebooks/state-of-testing-report-2018/>

ISTQB @ Worldwide Software Testing Practices Report 2017 – 2018
<https://www.istqb.org/references/surveys/istqb@-worldwide-software-testing-practices-survey-2017-18.html>



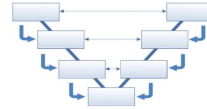
HERRAMIENTAS

ÁGIL



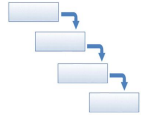
- » ~~Gestión de las pruebas~~
- » ~~Gestión de casos de prueba~~
- » Gestión de defectos
- » Herramientas de automatización

MODELO V



- » Gestión de las pruebas
- » Gestión de casos de prueba
- » Gestión de defectos
- » Herramientas de automatización

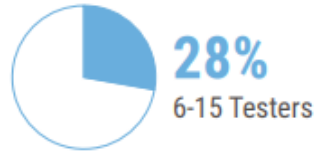
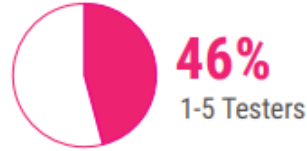
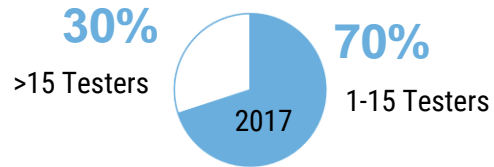
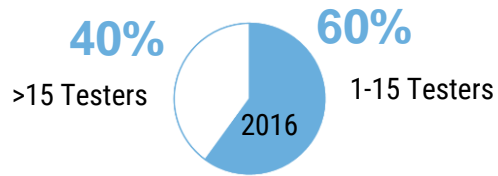
CASCADA



- » Gestión de las pruebas
- » Gestión de casos de prueba
- » Gestión de defectos
- » Herramientas de automatización



TAMAÑO DEL EQUIPO DE TRABAJO DE PRUEBAS



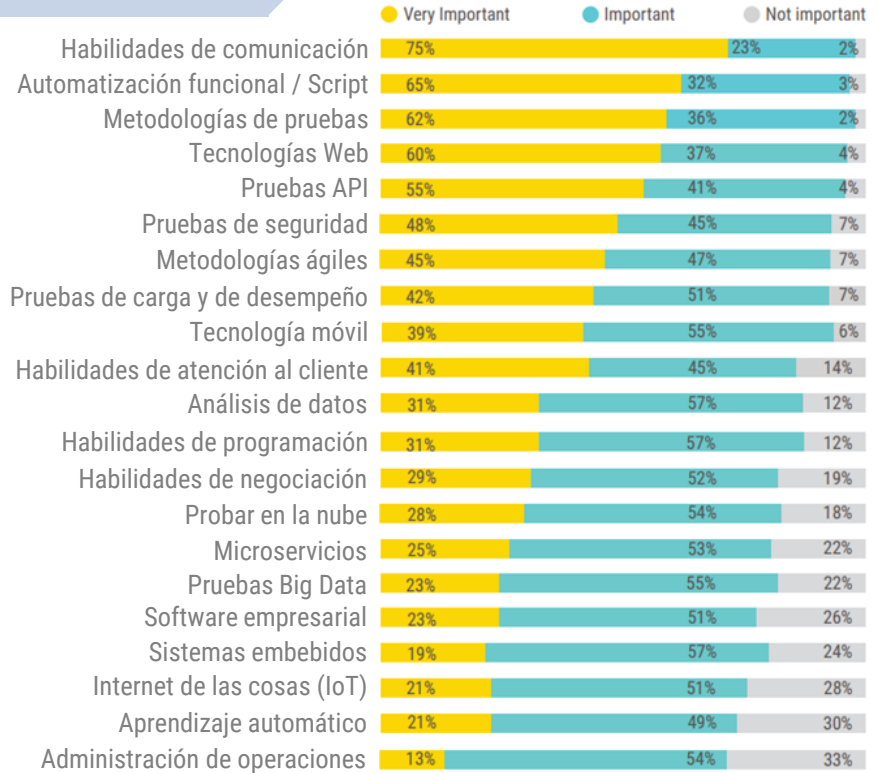
La tendencia indica que los equipos de pruebas se están reduciendo año tras año.



PERFIL DEL TESTER



Objetivo
Asertivo
Comunicador
Atención al detalle
Analítico
Curioso
Observador
Congruente
Crítico
Inquisitivo
Creativo
Perceptivo
Imparcial





HABILIDADES ESPERADAS DE PROBADORES ÁGILES





EQUIPO DE TRABAJO / NIVEL DE EXPERIENCIA

| | | |
|---|---|--|
| <p>ÁGIL</p>  <p>Testers dentro del equipo de trabajo Programador como tester Nivel técnico avanzado</p> | <p>MODELO V</p>  <p>Líder de pruebas (Experto) Equipo de testers (junior /senior)</p> | <p>CASCADA</p>  <p>Líder de pruebas (Experto) Equipo de testers (junior /senior)</p> |
| <p>1 a 2 probadores por 5 a 6 desarrolladores</p> | <p>1 Líder de pruebas, 1 probador por 10 desarrolladores</p> | |



FORMACIÓN



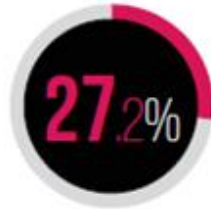
En el trabajo



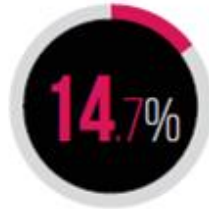
Certificación de competencias



Formación formal



En conferencias

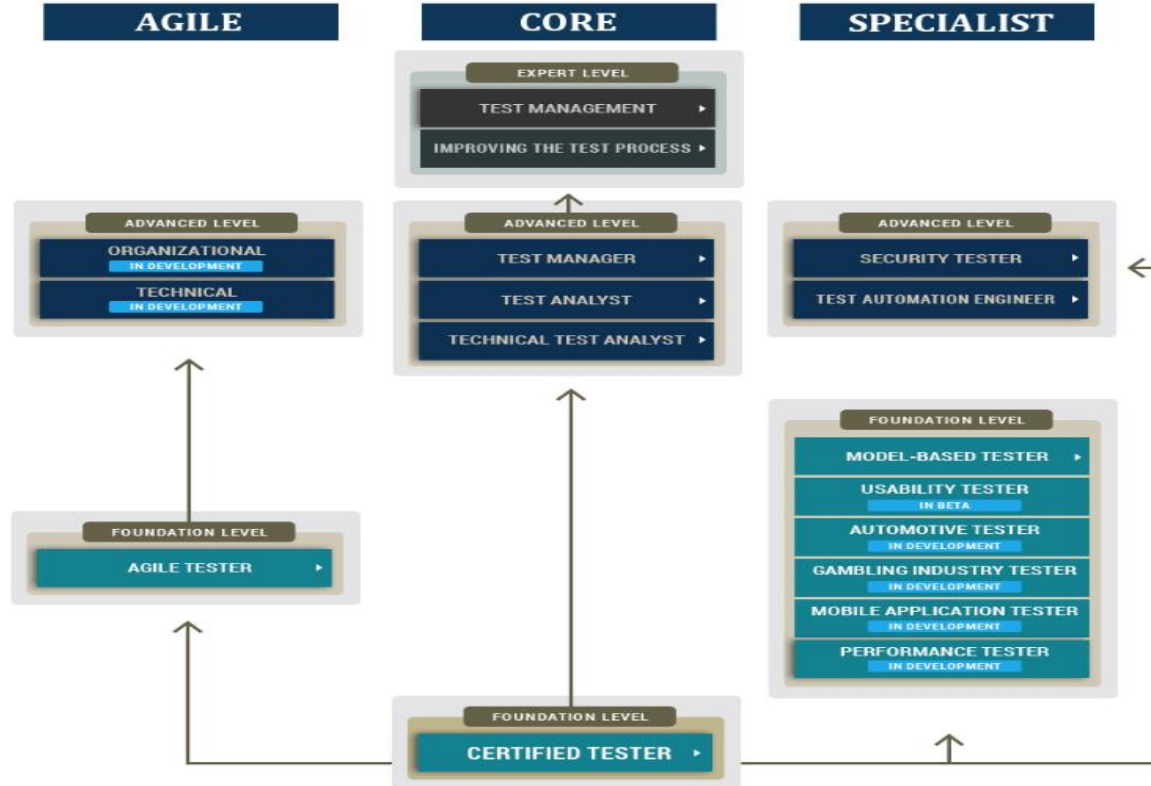


Ninguno





FORMACIÓN – CERTIFICACIÓN DE COMPETENCIAS





FORMACIÓN – CERTIFICACIÓN DE COMPETENCIAS

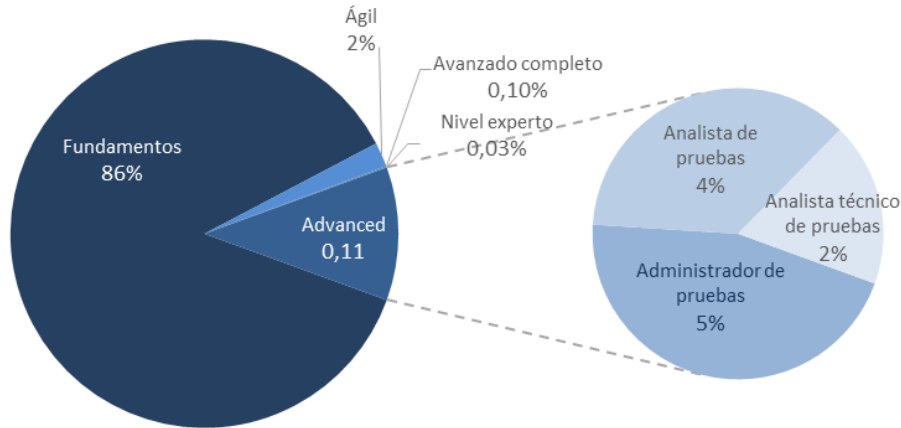


Internacional

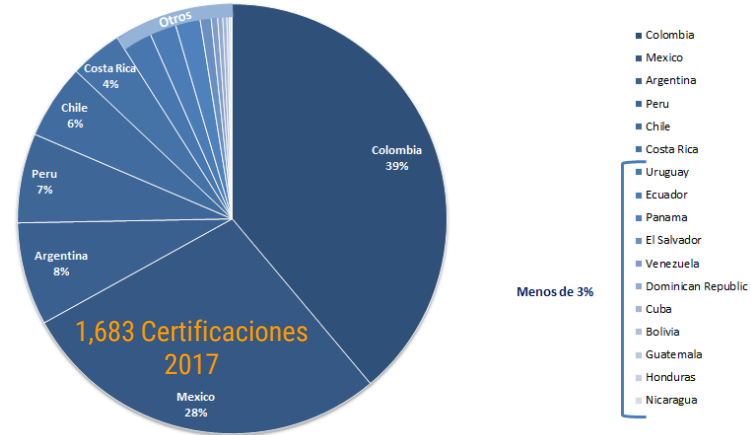
Hispanoamericana

Diciembre de 2018

Exámenes aplicados: **830,000+**
Certificaciones emitidas: **605,000+**




Distribución certificaciones ISTQB® en Hispanoamérica





FORMACIÓN FORMAL





















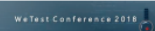














| Objetivos | Tester de Software | Tester profesional de Software | Líder de testing de Software |
|---|---|---|--|
| <p>Formar profesionales en testing de software capaces de ejecutar, diseñar y gestionar las pruebas de productos de software, así como liderar grupos de testing.</p> <p>Enseñar a “probar” el software y a “probar” las pruebas, conocer sus diferentes sabores, sus posibilidades y limitaciones, para poder así evaluar la calidad del software bajo prueba, y también, la calidad de las pruebas ejecutadas.</p> <p>Contribuir a generar una red regional de expertos en testing, orgullosos de su profesión, dispuestos a intercambiar experiencias, profundizar y actualizarse en esta disciplina.</p>  | <p>Introducción al testing</p> <p>Modulo 1 TS Introducción a la gestión de incidentes Introducción al testing funcional</p> <p>Documentación y reportes</p> <p>Modulo 2 TS Técnicas de testing funcional Introducción al testing de performance</p> <p>Un Tester de software graduado:</p> <ul style="list-style-type: none"> Comprende los conceptos básicos de la disciplina de testing, testing funcional y de performance. Conoce de gestión de incidencias, comprende la importancia de la documentación en el testing. A partir de lo aprendido, puede hacer experiencia en la industria, trabajando. Consideramos que está apto para desempeñarse como tester, ejecutando y comenzando a diseñar pruebas. | <p>Introducción a la automatización del testing funcional</p> <p>Modulo 1 TPS Taller de documentación y reportes Taller de técnicas de testing funcional - I</p> <p>Taller de testing de performance</p> <p>Modulo 2 TPS Taller de técnicas de testing funcional - II Taller de automatización de testing funcional Metodología y gestión de testing - I</p> <p>Un Tester profesional de software graduado:</p> <ul style="list-style-type: none"> Perfeccionó los conocimientos y habilidades adquiridas, su capacidad de diseño y de selección de la mejor estrategia y técnicas en un contexto determinado. Es capaz de planificar los recursos disponibles para lograr los objetivos de las pruebas, responsabilizarse por proyectos de testing de mediano porte y proponer mejoras en los procesos involucrados. | <p>Testing en el desarrollo de software</p> <p>Modulo 1 LTS Información para la toma de decisiones Optativas I, II</p> <p>Metodología y gestión de testing II</p> <p>Modulo 2 LTS Operativas III, IV Proyecto final del diploma</p> <p>Un Tester profesional de software graduado:</p> <ul style="list-style-type: none"> Coordina y dirige las actividades de testing de la organización y brega por la mejora permanente de los procesos involucrados. Promueve la jerarquización del testing sobre la base de dar visibilidad a la información que aporta y mostrar cómo puede ser utilizada para tomar decisiones. Entrena y promueve la capacitación permanente de los integrantes del grupo de testing. |




FORMACIÓN – CONFERENCIAS / EVENTOS

TESTING CONFERENCES AND EVENTS WORTH VISITING!

SOMETHING FOR EVERYONE, ALMOST EVERYWHERE

| | | | | | | |
|--|--|---|---|---|--|--|
|  OnlineTestConf |  ASTQB Summit |  JaSST |  STP Conf |  Selenium Conference |  HUSTEF |  CAST |
|  TestBash |  StarEast |  StarWest |  AgileTestingDays |  Agile DevOps East |  Automation Guild |  ANZTB |
|  StarCanada |  SauceCon |  QualityJam |  Eurostar |  Let's Test South Africa |  heisenbug-moscow |  WeTest Conference |
|  Romanian Testing Conference |  COMAQA |  QA&Test |  Expo QA |  GTAC |  QA or Highway Conference |  Nordic Testing Days |
|  qa: challengeaccepted |  Tabara de testare |  QA Fest |  Testistanbul |  Test Leadership Congress |  PNSQC |  copenhagen Context |


National Software Testing Conference, (UK)

WHAT'S IN IT FOR ME?

Look for events that are close to you and free of charge, you'll be amazed by the high quality information you will get from them! There are also a number of Online events that make conferences really accessible to all.



The screenshot shows the website for 'argenTesting' (Congreso Argentino de Testing). The header is blue with the logo. Below the header, there are navigation links: HOME ARGENTINA, EAGLE 2017, MENU, AUTORES, FAQ, INQUIRY, and social media icons. The main content area is titled 'Congreso Argentino de Testing' and features several categories with icons and brief descriptions: Charlas (Presentations), Talleres (Workshops), Expositores (Exhibitors), Comités revisor (Review Committees), Investigador (Researchers), Oportunidades laborales (Job Opportunities), Sponsors, Encuestas (Surveys), and Colaboradores (Collaborators).

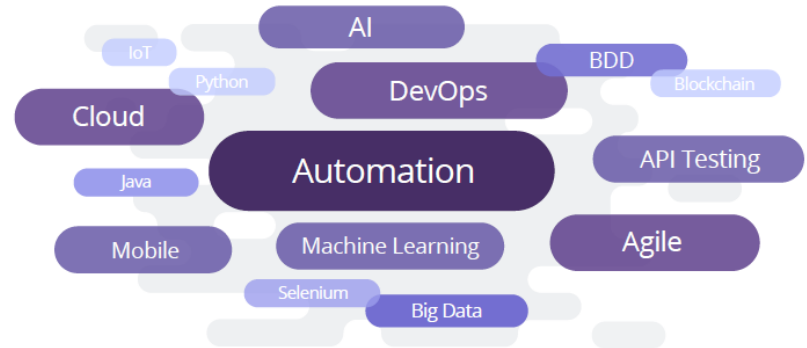


TENDENCIAS DE PRUEBAS



Cloud Testing ofrece distintos tipos de pruebas, un nuevo enfoque para preparar de forma rápida, flexible, escalable

Top 3 practices and technologies users expect to see more of in the next two years.





TENDENCIAS DE PRUEBAS



CROWD TESTING

- Externalización de los servicios de Testing y QA
- Voluntariamente hacen las pruebas y suele pagarse por bug encontrado
- Consigue grandes volúmenes de probadores por todo el mundo – objetividad
- Considerar confidencialidad / derechos de autor

CROWDSOURCE TESTING

- Tercerización masiva o subcontratación voluntaria
- Externalización de tareas a un grupo numeroso de personas o comunidad vía internet que voluntariamente prueban una aplicación
- Se prueba en muchos entornos y plataformas
- Se centran en problemas y usabilidad



SALARIOS DE LOS PROBADORES



| | > 1 year | 1-2 | 2-5 | 5-10 | 10+ |
|----------------------|----------|-----|-----|------|-----|
| Africa | 5 | 16 | 16 | 26 | 46 |
| Asia | 5 | 22 | 34 | 40 | 53 |
| China | NA | 5 | 13 | 41 | 57 |
| India | 10 | 19 | 19 | 34 | 51 |
| Latin America | 9 | NA | 18 | 32 | 48 |
| East Europe / Russia | 10 | 13 | 19 | 28 | 34 |
| Western Europe | 37 | 34 | 50 | 62 | 80 |
| Middle East | 9 | 28 | 55 | 58 | 103 |
| Aus / NZ | 40 | 40 | 68 | 79 | 110 |
| USA / Canada | 40 | 48 | 68 | 87 | 107 |

Salarios en miles de dólares, incluyen bonos y beneficios en caso de haber (por año)
Se carece de datos suficientes para proporcionar información significativa

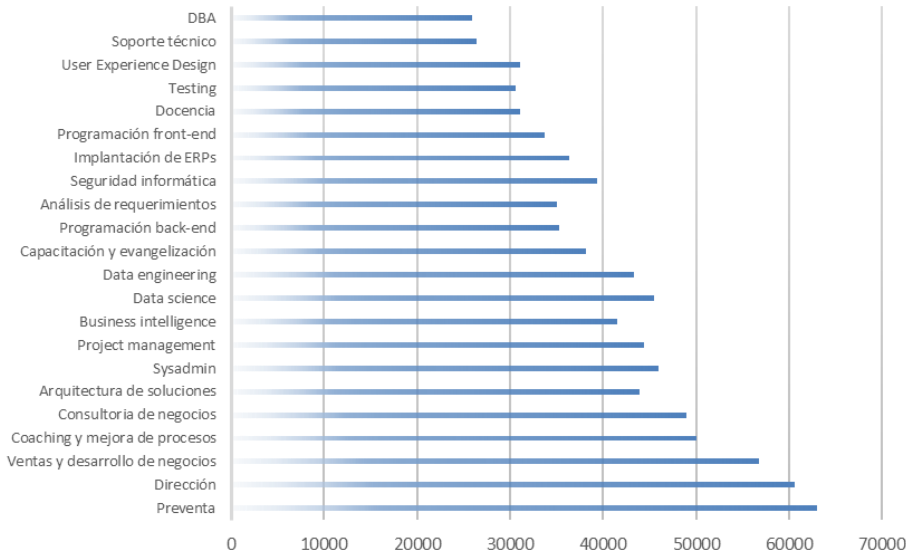


SALARIOS DE LOS PROBADORES



Encuesta realizada en noviembre de 2018, se consideraron 2,215 respuestas válidas, de las cuales 2,075 son de personas que indicaron residir en México

ESTUDIOS DE SALARIOS SG 2019



Consideraciones: no se especifica nivel (junior, senior, Lead) - experiencia, tipo de pruebas, manejo de herramientas, si cuenta o no con certificaciones



RETOS DE LOS PROBADORES



Actualizaciones constantes y nuevos ciclos de pruebas



Número de proyectos a atender

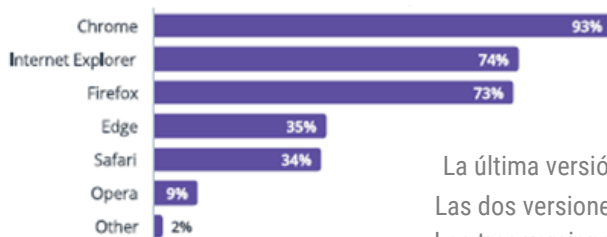


Tiempo asignado para pruebas

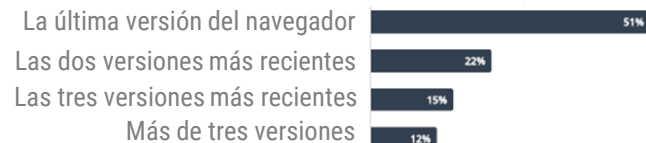


Diversidad de plataformas, herramientas, funcionalidad, necesidades

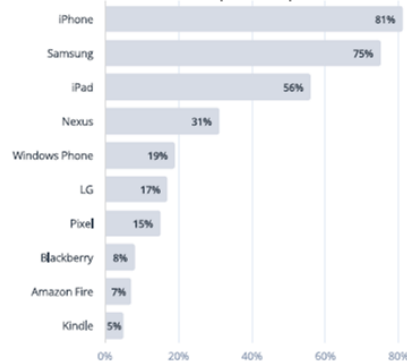
Navegadores donde se aplican pruebas



Número de versiones de navegador se prueban

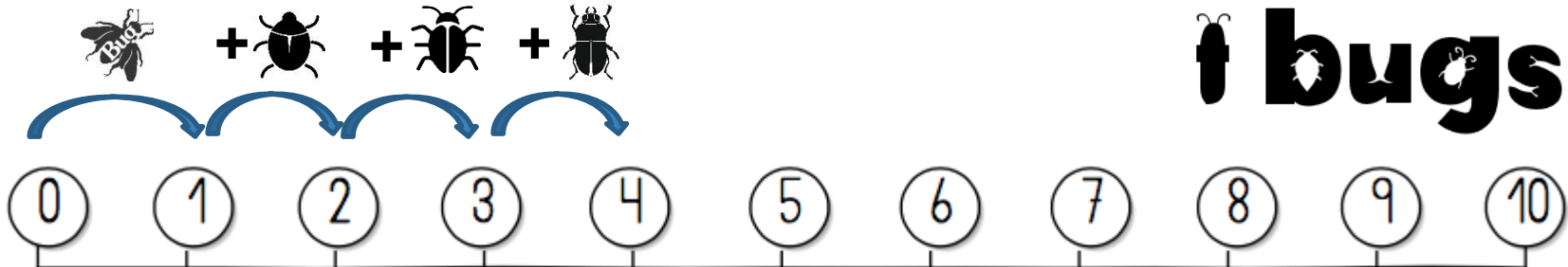


Dispositivos móviles donde se aplican pruebas





¿CÓMO SE SIENTE EL PRBADADOR?



t bugs d

| | |
|---|---|
| <ul style="list-style-type: none"> > calidad del software > probabilidad de corregir todos los defectos | <ul style="list-style-type: none"> < calidad del software < probabilidad de corregir los defectos > probabilidad de que el cliente identifique defectos |
|---|---|

**Probar poco no agrega valor al negocio,
pero probar mucho tampoco**



¡Gracias!

¿Tienen preguntas?

lilianarc@unam.mx