

Asignatura

Entornos de Desarrollo

UNIDAD 3

Diseño y realización de pruebas



UNIVERSAE
Instituto Superior de FP



Procedimientos y casos de prueba

El procedimiento de prueba consiste en la realización de una prueba y cómo se va a realizar dicha prueba.

A la hora de realizar una prueba contemplamos los posibles errores, rendimiento y que se cumplan todos los requisitos para los que se desarrolla el software.

Lo recomendable es que la realización de pruebas las realice personal externo.

Los planes de prueba son dónde se detallan las pruebas a realizar y cubre los siguientes aspectos:

- 1 – Introducción
- 2 - Módulos
- 3 - Características del software a probar
- 4 - Características del software que no ha de probarse
- 5 - Enfoque de pruebas
- 6 - Criterios de validez
- 7 - Proceso de pruebas
- 8 - Requerimientos del entorno
- 9 -Homologación



Casos de prueba

Descripción

En la fase de pruebas de un software diseñamos y preparamos los casos de prueba creados para encontrar fallos en el software desarrollado.

Las pruebas no pueden consistir en cosas complejas, sino es más difícil encontrar el origen del fallo.



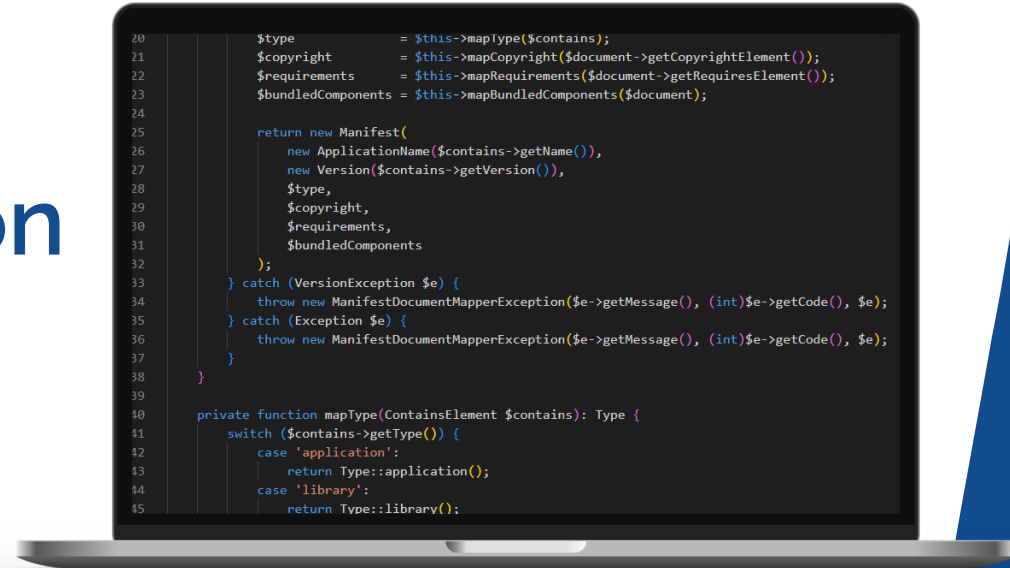
Codificación y ejecución de las pruebas

Descripción

Cuando se ha realizado el diseño de los casos de prueba, entonces se prepara el escenario para poder ejecutarla.

En la mayoría de los casos de ejecuta generando datos.

```
20     $type           = $this->mapType($contains);
21     $copyright       = $this->mapCopyright($document->getCopyrightElement());
22     $requirements    = $this->mapRequirements($document->getRequiresElement());
23     $bundledComponents = $this->mapBundledComponents($document);
24
25     return new Manifest(
26         new ApplicationName($contains->getName()),
27         new Version($contains->getVersion()),
28         $type,
29         $copyright,
30         $requirements,
31         $bundledComponents
32     );
33 } catch (VersionException $e) {
34     throw new ManifestDocumentMapperException($e->getMessage(), (int)$e->getCode(), $e);
35 } catch (Exception $e) {
36     throw new ManifestDocumentMapperException($e->getMessage(), (int)$e->getCode(), $e);
37 }
38
39
40 private function mapType(ContainsElement $contains): Type {
41     switch ($contains->getType()) {
42         case 'application':
43             return Type::application();
44         case 'library':
45             return Type::library();
```



Tipos de pruebas

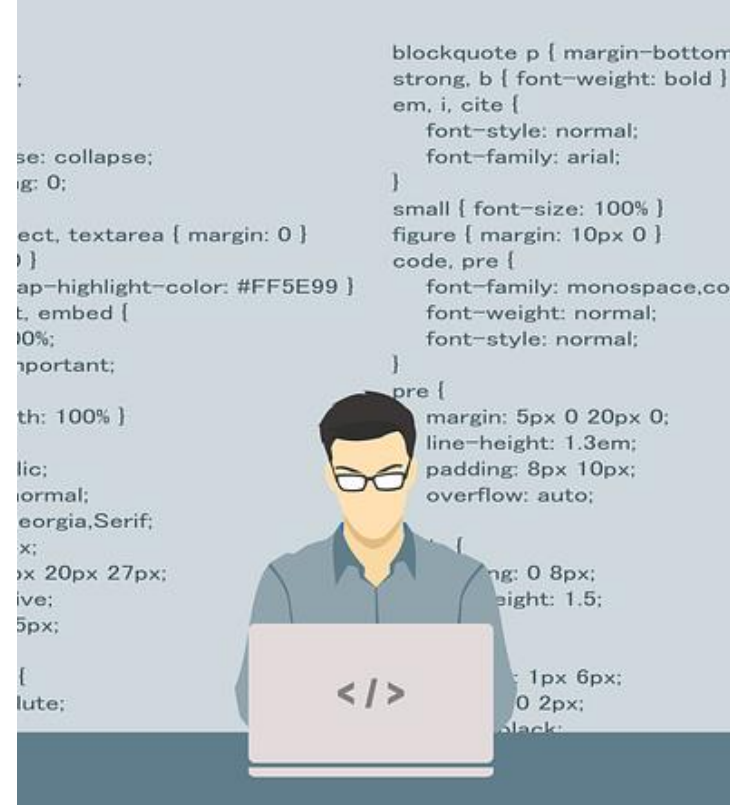
Las pruebas que se van a realizar son funcionales o características que ofrece el sistema.

El tester o encargado de realizar las pruebas, se basa en la documentación elaborada en base a la realización de las pruebas.

No se evalúa el funcionamiento interno del sistema sino las funciones que este realiza, esto es lo que se conoce como caja negra.

Las pruebas de seguridad también son funcionales.

Este tipo de pruebas son de estrés, de fiabilidad, etc.



Pruebas Funcionales y Estructurales

Pruebas caja blanca

Pruebas de cubrimiento

Consiste en ejecutar al menos una vez, todas las sentencias que se encuentren en el código del programa.



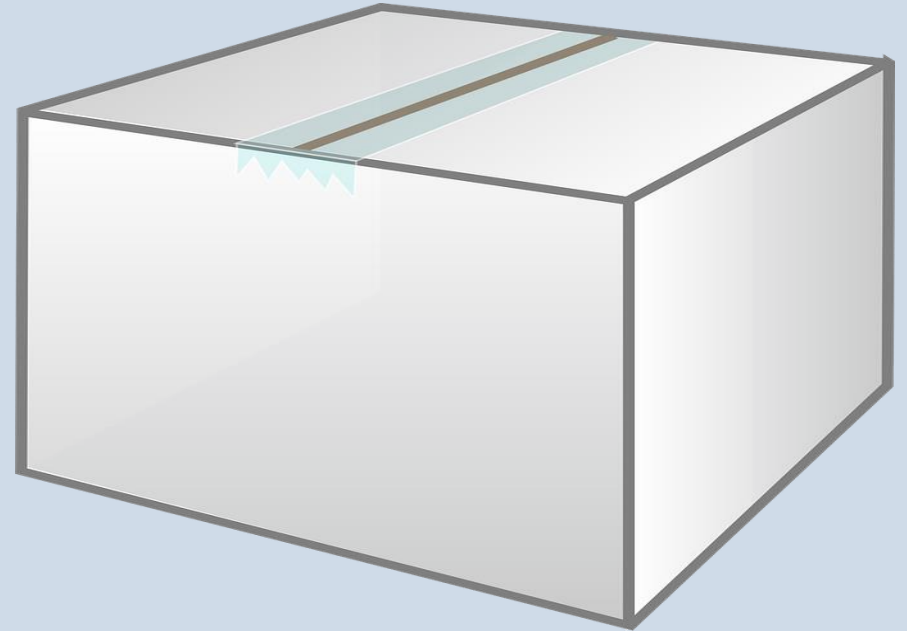
Prueba de condiciones

Cuando se produce una condición de sentencias las pruebas deben de ser múltiples.



Prueba de bucles

Son estructuras basadas en la repetición de sentencia.



Pruebas caja negra

Pruebas de equivalencia de datos

Se debe de crear un usuario y contraseña para acceso al sistema. Para probar esta funcionalidad se crean clases de equivalencia válidas.



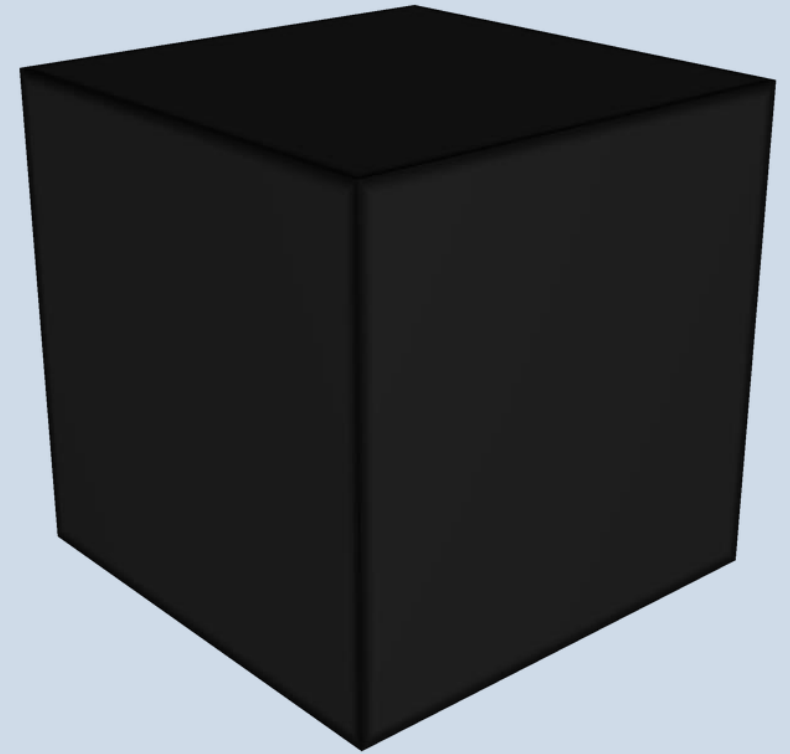
Prueba de valores límite

Su objetivo es comprobar que los valores que debe de recibir la aplicación son los correctos para comprobar su funcionamiento.



Prueba de interfaces

GUI (graphical user interface). Una interfaz es una serie de objetos con ciertas propiedades en conjunto.



Herramientas de depuración de código

Descripción

JUnit es un framework dónde se realizan pruebas de repetición.

Consiste en diseñar una prueba en concreto para cierta aplicación para ser ejecutada x veces en busca de distintos errores.



Planificación de pruebas

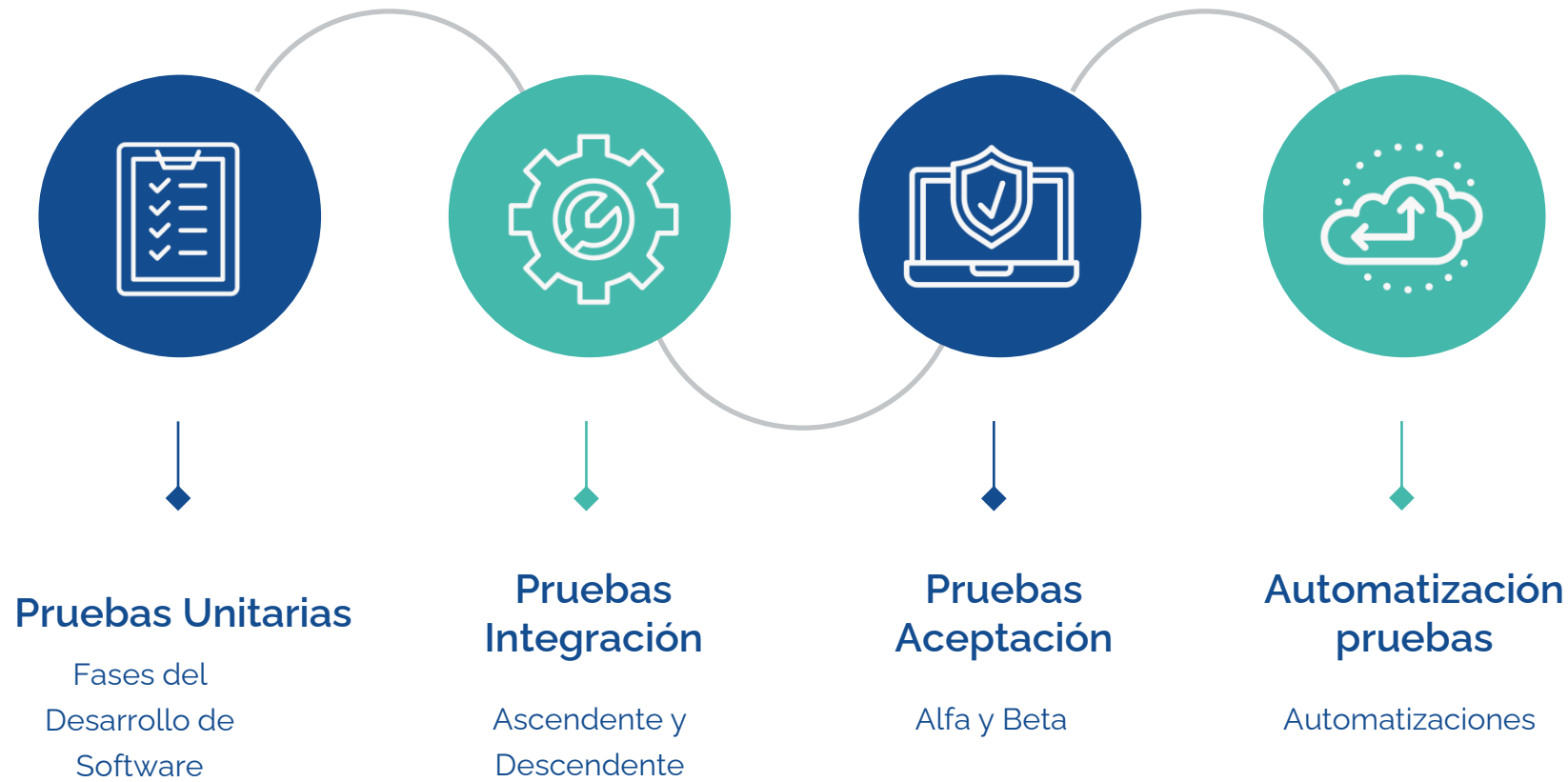
Funcionamiento

En el inicio del proceso del diseño del Software es necesario el planteamiento de pruebas a realizar. Dónde y cuándo.

Depende de la envergadura del proyecto, se debe de establecer las pruebas a elegir,



Planificación de pruebas



Calidad del Software



Desarrollo

La calidad del software es fundamental para medir nuestro software de cara al mercado.

Los sistemas que se desarrollan no deben de tener fallos y además deben de cumplir con las garantías solicitadas por el cliente.

Medidas o Métricas de calidad del software

Los criterios de calidad se definen al inicio del proyecto y no deben moverse.

Estos criterios pueden medirse y se determinan por el N.º de errores en líneas de código y revisiones realizadas.

The background is a solid blue color. Overlaid on this are several faint, light-blue geometric patterns. These include a grid of small squares that form larger, irregular shapes, and numerous small, light-blue arrows pointing in various directions. The overall effect is a sense of movement and digital connectivity.

UNIVERSAE

— CHANGE YOUR WAY —