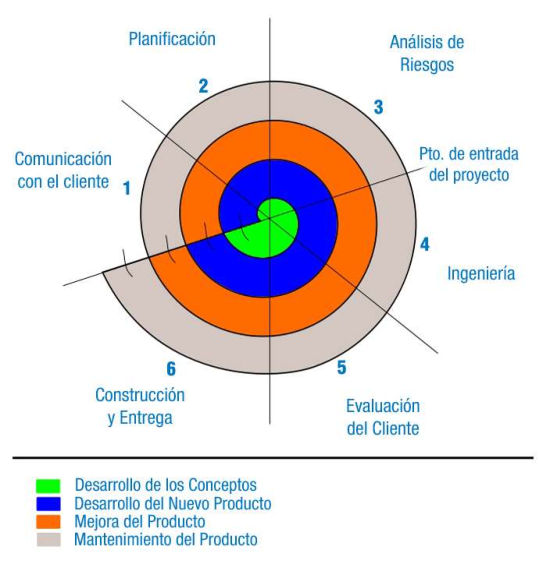
EEDD03

Diseño y realización de pruebas

# Planificación de las pruebas.



Siguiendo el modelo en espiral, se empieza con **pruebas unitarias** (probar cada función individualmente),seguidas de las **pruebas de integración** (comprobar que el conjunto funciona).

## Tipos de pruebas.

**Black box testing:** cuando se usa la aplicación con su interfaz externa. Le introducimos datos y comprobamos que los que salen son correctos.

**White box testing:** se comprueba el código de la aplicación para buscar estructuras incorrectas o ineficientes.

## Funcionales (pruebas de caja negra)

Hay tres tipos de pruebas:

1. **Particiones equivalentes:** Se pretende crear un conjunto de clases donde la prueba de un valor, en cuanto a verificación de errores, sería extrapolarle al que se conseguiría probando cualquier valor de la clase.
2. **Análisis de los valores límite:** Probar el valor máximo y mínimo.
3. **Pruebas aleatorias:** Se lanzan entradas aleatorias y a ver qué sale.

## Estructurales (pruebas de caja blanca)

Este tipo de pruebas se basan en criterios de cobertura lógica:

* **Cobertura de sentencias:** hay que generar pruebas suficientes para cubrir todas las instrucciones al menos una vez.
* **Cobertura de decisiones:** que cada opción se evalúe al menos una vez a true y otra a false.
* **Cobertura de condiciones:** que cada condición se evalúe al menos una vez a true y otra a false.
* **Cobertura de condiciones y decisiones:** crear las 2 anteriores a la vez.
* **Cobertura de caminos:** Se debe ejecutar al menos una vez cada secuencia de sentencias encadenadas.
* **Cobertura del camino de prueba:** Hay dos variantes, una en la que se ejecuta el camino una sola vez, y otra en la que se ejecuta tres veces: una sin entrar dentro, otra entrando una vez y otra entrando dos veces.

Se denomina cubrimiento al número de caminos que se comprueban.

Una herramienta para este tipo de pruebas es Clover para Java.

## Pruebas de regresión.

Las pruebas de regresión comprueban que los cambios en una aplicación no introducen un comportamiento no deseado.

Este tipo de pruebas repiten las pruebas hechas anteriormente. Se considera que la prueba ha tenido éxito si se encuentran errores.

El conjunto de pruebas de regresión tiene 3 clases distintas de prueba:

* Una muestra representativa que ejercite todas las funciones del software.
* Pruebas adicionales en las funciones que se van a ver afectadas por el cambio.
* Pruebas que se centran en los componentes del software que han cambiado.

# Procedimientos y casos de prueba.

Existen varios procedimientos para el diseño de casos de prueba:

* **Enfoque funcional o caja negra:** El programa recibe una entrada, produce una salida, y comprobamos que es correcto. Se aplican los **valores límite** y las **clases de equivalencia**.
* **Enfoque estructural o de caja blanca:** Nos centramos en la implementación interna del programa.
* **Enfoque aleatorio:** Se elaboran casos de prueba que prueben las entradas del programa. Se usan generadores automáticos de clases de prueba.

# Herramientas de depuración.

Los IDE normalmente cuentan con herramientas de depuración, como comprobación de sintaxis.

## Puntos de ruptura.

Son marcadores para líneas de código. Si se añaden, el programa se ejecuta hasta llegar al punto de ruptura, donde se detiene y hay que decirle que continúe hasta el siguiente punto de ruptura.