

**Curso de preparación
Tester Certificado
Nivel Básico**

Agenda



Capítulo 3. Técnicas Estáticas

- 3.1: Técnicas estáticas y el proceso de prueba
- 3.2 Procesos de revisión
- 3.3 Análisis estáticos por herramientas

3.1: Técnicas estáticas y el proceso de prueba

Términos

- ▶ **Pruebas dinámicas:** “dynamic testing” Pruebas que **implican la ejecución** del software, de un componente o sistema.
- ▶ **Pruebas estáticas:** “static testing” Pruebas de un componente o sistema a nivel de especificación o implementación, **sin ejecutar el software**, por ejemplo revisiones o análisis estático de código.
- ▶ **Análisis estático:** “static analysis” Análisis de los artefactos software, por ejemplo **requisitos o código**, llevado a cabo **sin la ejecución de estos artefactos** de software. Actividad basada en herramientas

Antecedentes Técnicas Estáticas

- ▶ Las pruebas estáticas se pueden **realizar con técnicas de análisis manual** (revisiones) y **análisis automatizados** (análisis estáticos) del código u otra documentación del proyecto.
- ▶ Se pueden ejecutar **antes** de la ejecución de **las pruebas dinámicas**.
- ▶ Los defectos encontrados durante la revisión al principio del ciclo de vida **son mas económicos de corregir**.

Capítulo 3 - Técnicas estáticas

Tema 3.1: Técnicas estáticas y el proceso de prueba

¿A qué productos de una fábrica de software se le pueden realizar pruebas estáticas?

- ▶ Cualquier producto de una fábrica de software puede ser revisado, entre los que se encuentran:

- * Especificaciones de requisitos

- * Código

- * Especificaciones de prueba

- * Scripts de prueba

- * Páginas Web

- * Especificaciones de diseño

- * Planes de prueba

- * Casos de prueba

- * Guías de usuario

Técnicas empleadas en el proceso de pruebas estáticas

- ▶ De verificación manual (Revisiones)
- ▶ Automatizadas (Análisis estáticos) de documentos o código.

Beneficios de las revisiones

- ▶ **Detección temprana de defectos** en el ciclo de desarrollo de software.
- ▶ Mejora la **productividad** del desarrollo.
- ▶ **Reduce la escala de tiempo** en la implementación del desarrollo.
- ▶ Se reduce en **costo y tiempo** las pruebas dinámicas y de UAT.
- ▶ **Reducción de defectos en etapas posteriores** del ciclo de vida del desarrollo.
- ▶ Comunicación Mejorada.
- ▶ Las revisiones pueden **encontrar omisiones**. Por ejemplo en los requisitos, donde hay omisiones, esto no se puede detectar en pruebas dinámicas.

Slide 8

EMC4

Describir la importancia y valor de considerar las técnicas estáticas para la evaluación de los productos de una fabrica de software.

Erika María Carvajal Fajardo, 01/05/2012

Pruebas estáticas VS pruebas dinámicas

Similitudes

- ▶ En las dos se detectan defectos.
- ▶ Son complementarias entre si. Encuentran diferentes tipos de defectos

Diferencias

Pruebas Estáticas	Pruebas Dinámicas
Se realizan en una fase temprana antes de que sean implementados en el código	Se realiza una vez se termina la codificación del software.
Se realizan sin ejecutar el código o software.	Se realizan ejecutando el código o software implementado.
Se enfocan en detectar causas de fallos (defectos) en vez de fallos en sí.	Se enfocan en defectos en la implementación del software.

Slide 9

EMC6

Explicar la diferencia entre las técnicas estáticas y dinámicas

Erika María Carvajal Fajardo, 01/05/2012

Objetivos de las revisiones y análisis estáticos

- ▶ Encontrar defectos del producto a implementarse, **en la etapa más temprana** posible del ciclo de desarrollo de software.
- ▶ Verificar o revisar que **el producto cumple** con los criterios predefinidos.
- ▶ Detectar **defectos típicos**, en las pruebas estáticas como:
 1. Desviaciones de **estándares**. EJ. Gestión de la configuración, Errores en los estándares de programación. (Ver eje. Herramienta)
 2. Defecto de **requisitos**
 3. Defectos de **Diseño y arquitectura** de software
 4. **Mantenibilidad** insuficiente.
 5. Especificaciones de **interfaces** Incorrectas.

Slide 10

EMC8

Describir los objetivos de las revisiones y análisis estáticos y compararlo contra las pruebas dinámicas

Erika María Carvajal Fajardo, 01/05/2012

3.2: Proceso de revisión

Términos

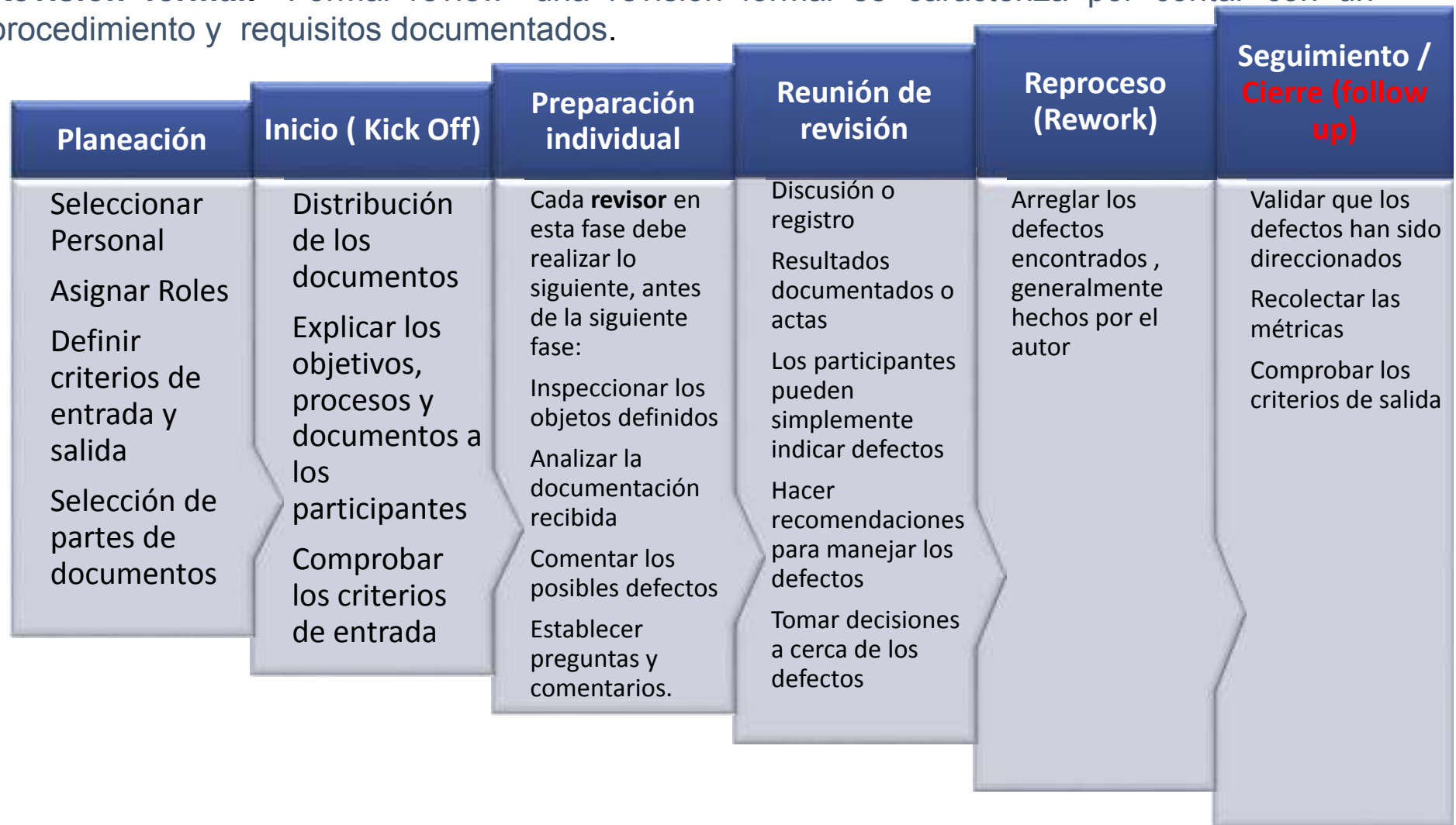
- ▶ **Criterio de entrada:** “Entry criteria” Conjunto de condiciones genéricas y específicas para **permitir que un proceso** prosiga con una tarea definida, por ejemplo la fase de pruebas. El objetivo de los criterios de entrada **es evitar que una tarea comience**, lo cual conllevaría un mayor esfuerzo que el necesario para eliminar los criterios de entrada fallidos. [Gilb y Graham]
- ▶ **Revisión por pares:** “Peer review” **Revisión entre dos personas**, por ejemplo dos probadores, un desarrollador y un probador o un usuario final y un probador, trabajando en conjunto con la finalidad de detectar defectos.

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Fases de una revisión formal

Revisión formal: “Formal review” una revisión formal se caracteriza por contar con un procedimiento y requisitos documentados.



Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Roles y Responsabilidades del proceso de revisión

Administrador - Director – Jefe de Proyecto - Gerente:

- ▶ Toma decisiones en la ejecución de revisiones
- ▶ Distribuye el Tiempo en los cronogramas del proyecto
- ▶ Determina si los objetivos de la revisión se han logrado

Moderador - Líder:

- ▶ Encargado de liderar la revisión del documento o conjunto de documentos
- ▶ Planear la revisión
- ▶ Realizar la reunión
- ▶ Hace seguimiento después de la reunión
- ▶ Mediar entre los diversos puntos. Es la persona en la que yace el éxito de la revisión.

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Roles y Responsabilidades

Autor:

- ▶ **Dueño del artefacto** objeto de la revisión, es el encargado de solucionar los defectos reportados, expone su trabajo ante la crítica.

Revisor: (Verificadores o Inspectores):

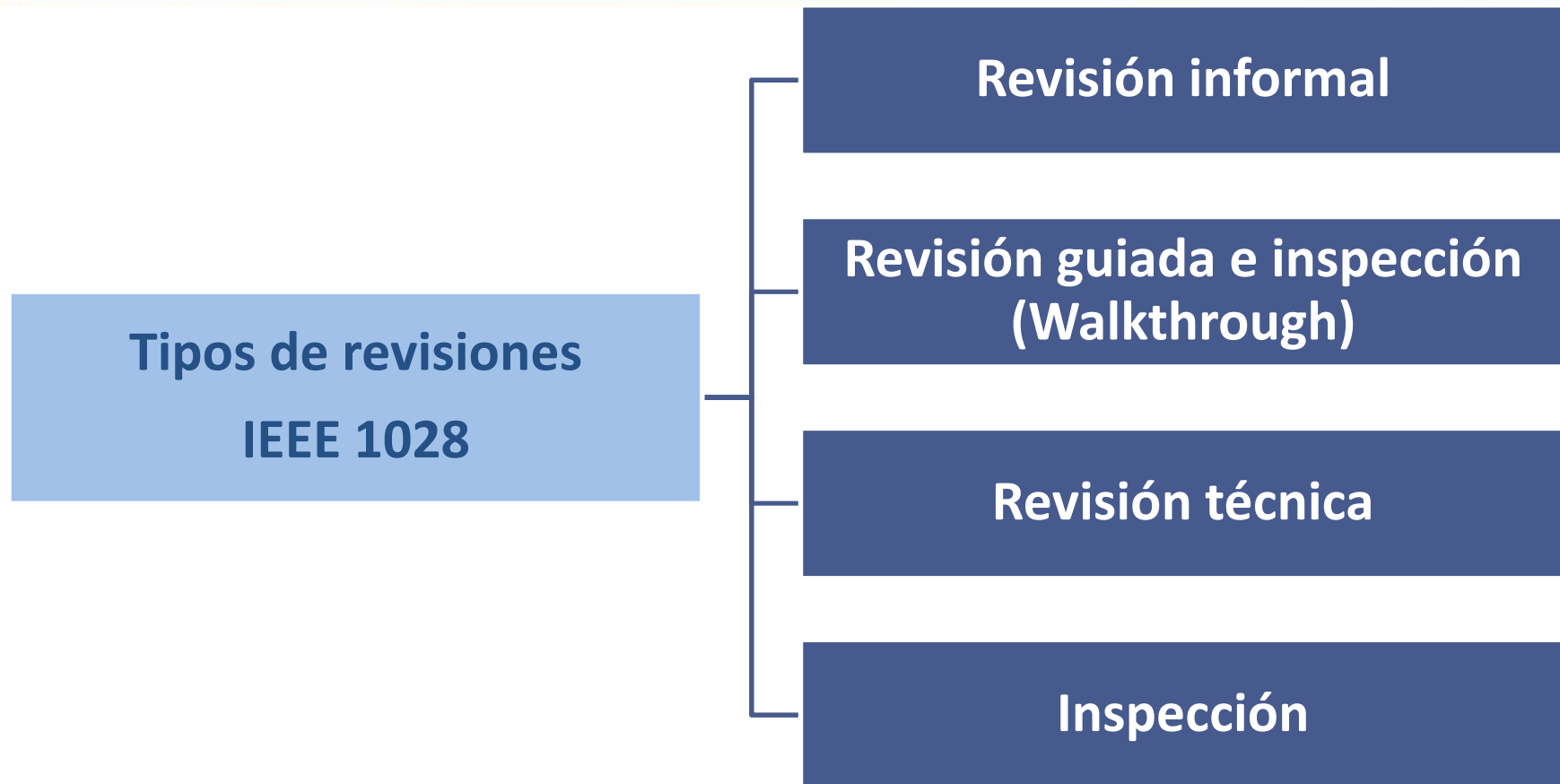
- ▶ Tienen una preparación necesaria para **identificar y describir los hallazgos** (Defectos) en el producto bajo revisión.
- ▶ Representar **diferentes perspectivas** y roles en el proceso de revisión. Utilizando listas de chequeo
- ▶ Deben **participar en todas las reuniones** de revisión.

Escriba (o grabador) :

- ▶ Responsable de **documentar los problemas**, dificultades y **puntos abiertos** que fueron identificados durante la reunión.

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión



- Un documento puede ser sujeto a mas de 1 revisión
- Si se utilizan diferentes tipos de revisión, su orden puede variar

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Revisión formal

Características principales

- ▶ Se formaliza como un evento público programado.
- ▶ Se informa por escrito los resultados de las mismas
- ▶ Todos los participantes, son responsables de la calidad de la revisión.

Ventaja

- ▶ Hace que los participantes en la revisión, tengan que prepararse previamente a la misma.

Desventaja

- ▶ Por el formalismo de la revisión, hace que se torne impersonal.

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Revisión informal

“Informal review” Es una revisión que no se basa en un procedimiento formalmente documentado

Características principales

- ▶ Proceso no formal
- ▶ Comprobación de escritorio (desk checking)
- ▶ Puede ser iniciada por el autor
- ▶ Revisión por pares (peer review), o una guía técnica revisando los diseños de códigos

Ventajas y desventajas de una revisión informal

Ventajas:

La mas rápida, simple y económica, baja presión en los resultados de las mismas

Desventajas:

No existe un formalismo de la revisión, puede estar documentado, no documenta defectos, no se realiza reunión.

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Revisión guiada (Walkthrough)

Características principales

- ▶ Liderada por el autor
- ▶ Se realiza la revisión mediante el proceso de pares
- ▶ Opcionalmente se preparan antes de la reunión de los revisores, el reporte de revisión, lista de hallazgos por el escriba (no es el autor).
- ▶ En la practica puede variar de muy informal a muy formal.
- ▶ El objetivo principal es aprendizaje, ganar conocimiento, encontrar defectos durante la revisión.

Ventajas y Desventajas Revisión guiada (Walkthrough)

Ventajas :

No es preparado, es programada con poco tiempo de anticipación.

Desventajas :

El autor del artefacto influye el resultado de las pruebas, en donde puede omitir puntos críticos o importantes; dado que él modera la revisión.

Slide 21

EMC34

Explicar las diferencias entre los diferentes tipos de revisión: revisión informal, revisión técnica, revisión guiada e inspección

Erika María Carvajal Fajardo, 03/05/2012

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Revisión técnica

Características principales

- ▶ Proceso documentado y de detección de defectos definido que incluye revisión de pares y **expertos técnicos**
- ▶ Puede ser realizado por revisión de pares **sin la participación de la gerencia.**
- ▶ Guiado por el **moderador entrenado (no el autor)**
- ▶ Preparación antes de la reunión
- ▶ **Uso opcional** de listas de chequeo, reporte de revisión, lista de hallazgos y participación de la **gerencia.**
- ▶ Puede variar de muy informal a muy formal
- ▶ **Objetivos:** Discutir, tomar decisiones, evaluar alternativas, encontrar defectos, resolver problemas técnicos, y verificar la conformidad de especificaciones y estándares

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Inspección

Características principales

- ▶ Guiado por **un moderador entrenado** y documentado por el escriba
- ▶ Usualmente revisión de pares
- ▶ Roles definidos
- ▶ Métricas Incluidas
- ▶ Procesos formales basados en reglas y listas de chequeo con cada criterio de entrada y de salida
- ▶ Preparación antes de la reunión
- ▶ Reporte de inspección, lista de hallazgos
- ▶ Proceso de seguimiento formal
- ▶ Opcionalmente, procesos de mejora y lectura de datos
- ▶ Objetivo Principal: Encontrar errores

Capítulo 3 - Técnicas estáticas

Tema 3.2: Proceso de revisión

Factores de éxito de una revisión

- ▶ Para cada revisión se **define un alcance** y un objetivo claro y trazable.
- ▶ Están involucradas las **personas adecuadas** para la revisión de los objetivos
- ▶ **Los defectos son bienvenidos** y expresados objetivamente
- ▶ Los problemas de las personas y los aspectos psicológicos son afrontados (**experiencia positiva para el autor**).
- ▶ Las técnicas de revisión **son aplicadas para que se ajusten** al tipo y nivel de los **productos** de una fabrica de software y a los **revisores**.

Factores de éxito de una revisión

- ▶ Listas de chequeo y roles son utilizadas, para incrementar la efectividad de la identificación de defectos.
- ▶ Capacitación en técnicas de revisión, especialmente las técnicas mas formales como la inspección.
- ▶ La gerencia es un soporte de un buen proceso de revisión (ejemplo, incorporar el tiempo adecuado en los cronogramas del proyectos para las actividades de revisión).
- ▶ Existe un énfasis en el aprendizaje y el mejoramiento de procesos.

3.3: Análisis estático por herramientas

Términos

- ▶ **Compilador:** “Compiler” Herramienta software que **traduce programas expresados en un lenguaje de alto nivel** a su lenguaje máquina equivalente. [IEEE 610].
- ▶ **Complejidad:** “complexity” Grado en el cual un componente o sistema, tiene un diseño y/o estructura interna, que es difícil de comprender, mantener y verificar.

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Análisis Estáticos:

“static analysis” Análisis de los artefactos del software, llevado a cabo, **sin la ejecución de estos**. Por ejemplo requisitos o código.

- ▶ Las herramientas de análisis estático son normalmente **usadas por los desarrolladores**.
 - Verifican contra reglas predefinidas o estándares de programación.
 - Se utilizan antes y después de las pruebas de componentes o de integración
- ▶ Las herramientas de análisis estático son utilizadas **por diseñadores durante el modelo de software**
- ▶ Las herramientas pueden producir una **gran cantidad de mensajes de advertencia**, que necesitan ser bien gestionadas para permitir el uso mas efectivo de la herramienta

Objetivo del Análisis Estático:

- ▶ Encontrar defectos en el **código fuente y modelos del software**.
- ▶ Son realizados **sin ejecutar el software**, a revisar por la herramienta
- ▶ Encuentran **defectos en vez de fallos**
- ▶ Las herramientas **también analizan datos de salida** generados como HTML o XML

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Comparación de las pruebas estáticas VS dinámicas

Pruebas estáticas

Aspectos positivos

- ▶ Detección **temprana de defectos**
- ▶ Sirven para **verificar** los siguientes entregables (**requisitos, diseño, código, casos de prueba**, etc.)
- ▶ Conclusiones de validez general
- ▶ **Identificación de defectos** que no se encuentran fácilmente por las pruebas dinámicas
- ▶ **Mantenibilidad** mejora de código y diseño

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Comparación de las pruebas estáticas VS dinámicas

Pruebas estáticas

Aspectos negativos

- ▶ Sujeto a los errores **de acuerdo al razonamiento** del tester
- ▶ No se usa para validación (**solo verificación**)
- ▶ No consideran el hardware o el software de base



Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Pruebas dinámicas

Aspectos positivos

- ▶ Las pruebas sirven tanto para **verificar como para validar**
- ▶ Para las pruebas se **tiene en cuenta el ambiente** donde se va a usar el software.
- ▶ Sirve para **probar otras características** además de funcionalidad

Aspectos negativos

- ▶ Su generalidad no es siempre clara (solo se aseguran los casos probados)
- ▶ Está atado al contexto donde es ejecutado
- ▶ Solo se **prueba el software** construido

Estos son los defectos mas comunes en el análisis estático

- ▶ Referenciar una variable con un valor indefinido
- ▶ Interface inconsistente entre los módulos y componentes
- ▶ Detectar dependencias e inconsistencias en modelos de software, por ejemplo interfaces.
- ▶ Variables que nunca son utilizadas
- ▶ Código inaccesible (muerto)
- ▶ Violaciones de estándares de programación
- ▶ Vulnerabilidad de seguridad
- ▶ Violaciones de sintaxis de código y modelos de software

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Dentro del control estático automatizado, se cuentan con herramientas como compiladores y analizadores

Compiladores

- ▶ Su función principal es verificar el código fuente y lo hace coincidir con las reglas de sintaxis del lenguaje y señalan las variables no utilizadas.
- ▶ Detecta defectos sintácticos e incompatibilidades de tipo y otros defectos semánticos.
- ▶ Ejemplo de error de sintaxis: Confundir un cero (0) por una O

Analizadores

- ▶ Analiza estándares
- ▶ Métricas de complejidad
- ▶ Acople de objetos

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Análisis de flujo de control

Flujo de control: “control flow” **Secuencia de eventos** (caminos) en la ejecución a través de un componente o sistema.

Se basa en una **representación gráfica del programa**, en un grafo de flujo, representado en tres elementos, nodos, aristas y regiones.

Estos **grafos se utilizan para identificar caminos**, con el fin de analizar el comportamiento del programa y situar puntos de ruptura por anomalías en el uso de variables.(ejemplo ramas muertas)

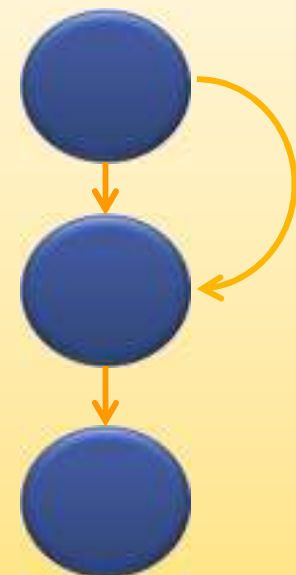
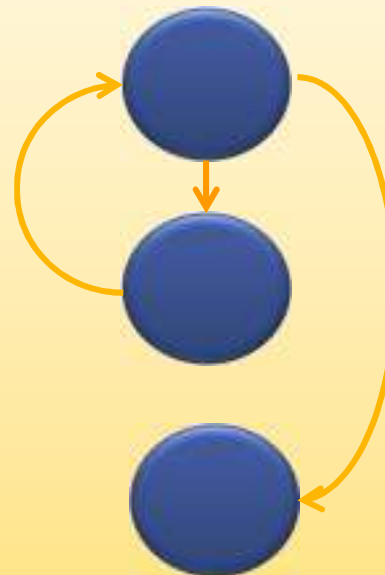
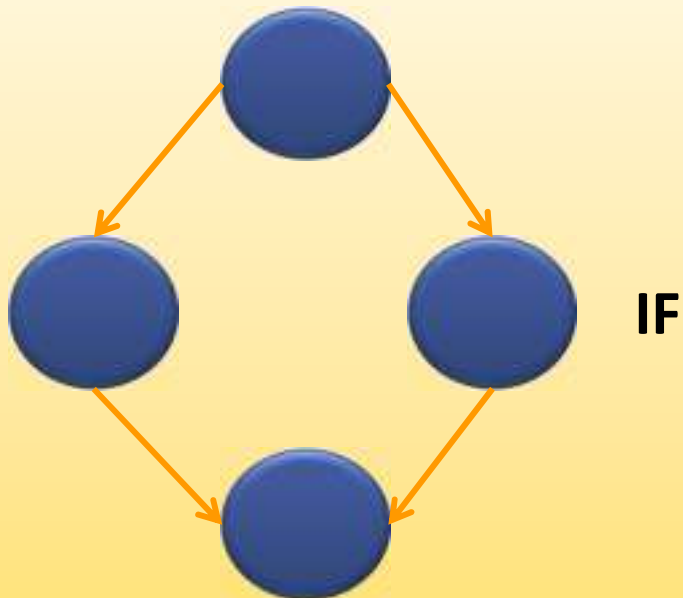
- ▶ **Nodos:** representan cero, una o varias sentencias en secuencia. Cada nodo **comprende como máximo una sentencia de decisión** (bifurcación).

- ▶ **Aristas:** líneas que **unen dos nodos** ósea la transferencia del flujo, o secuencia, ejemplo(representa los if, while, repeat)
- ▶ **Nodos predicado:** cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR, ...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina **nodo predicado**.

Capítulo 3 - Técnicas estáticas

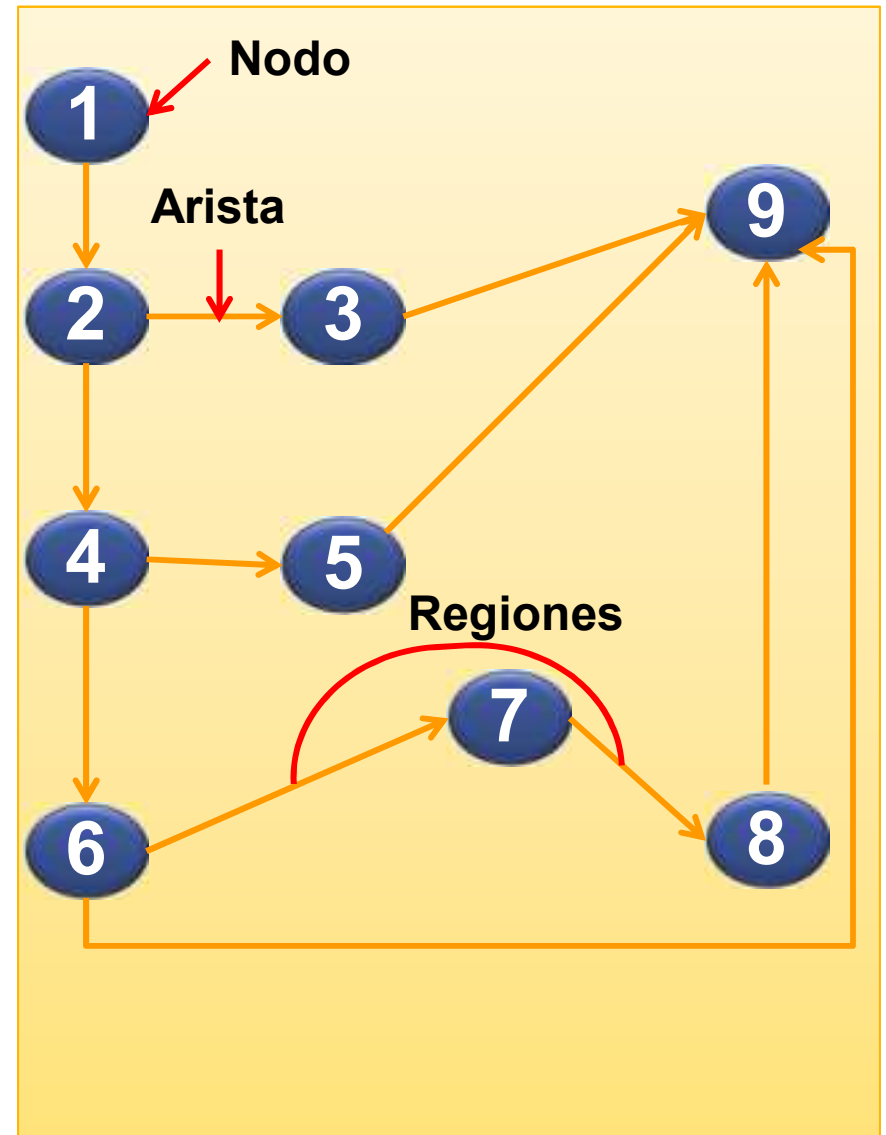
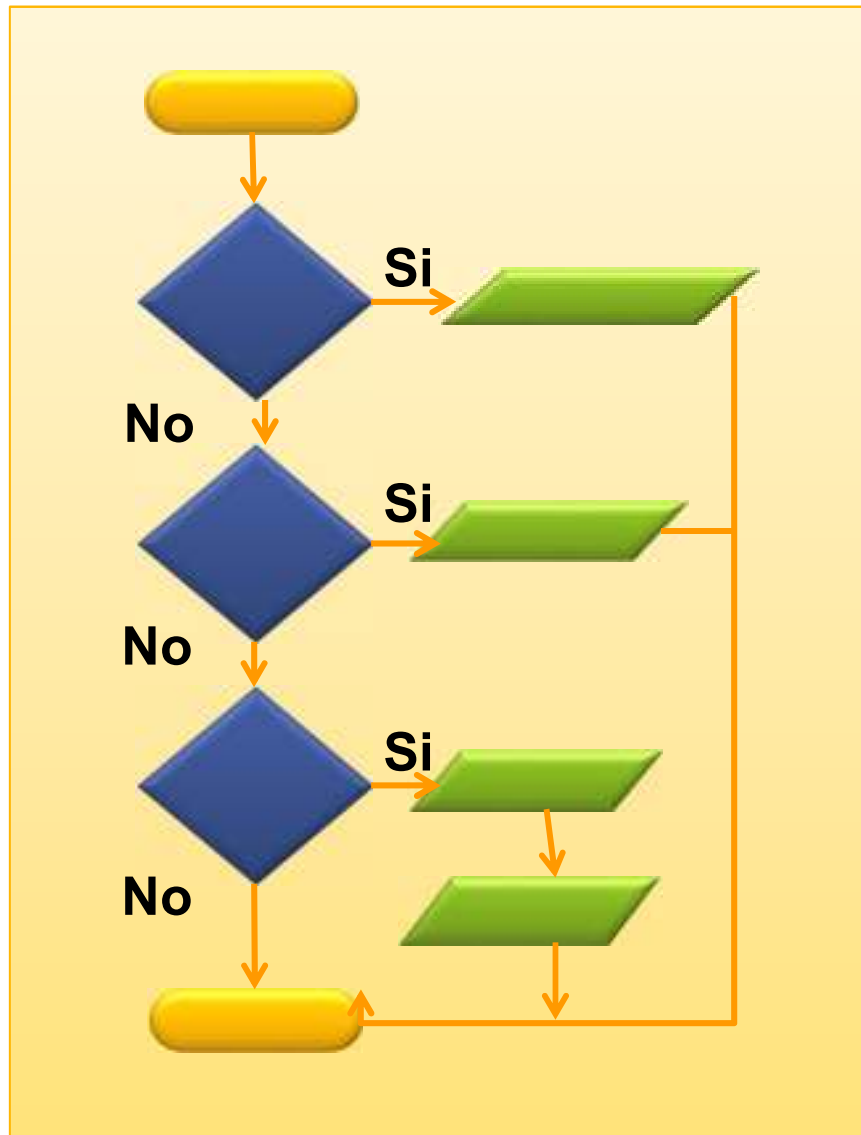
Tema 3.3: Análisis estático por herramientas

Análisis de flujo de control



Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

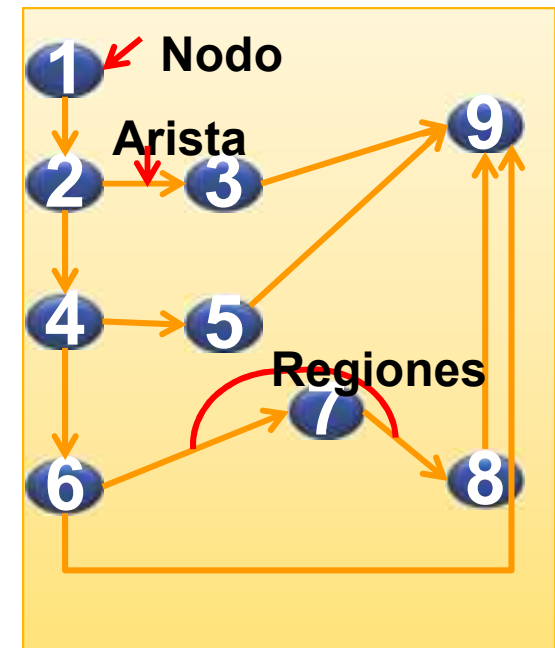


Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

El grafo de flujo permite observar los cuatro caminos básicos de ejecución:

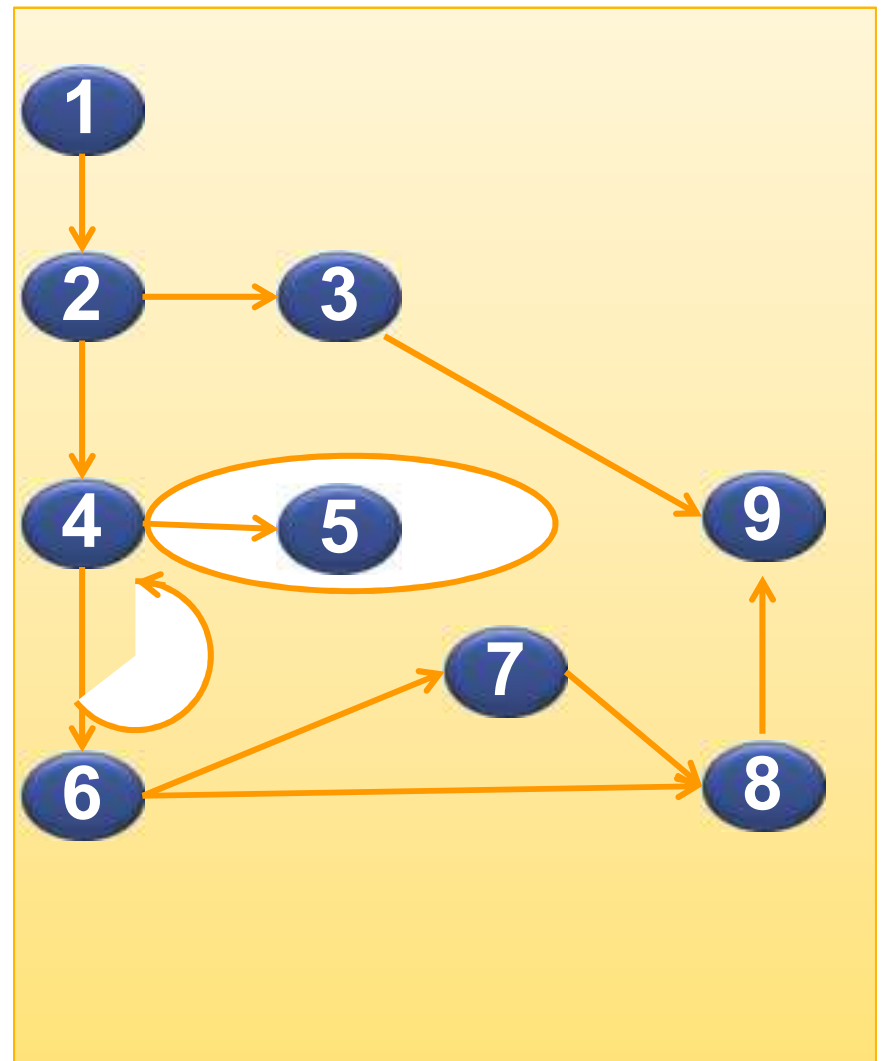
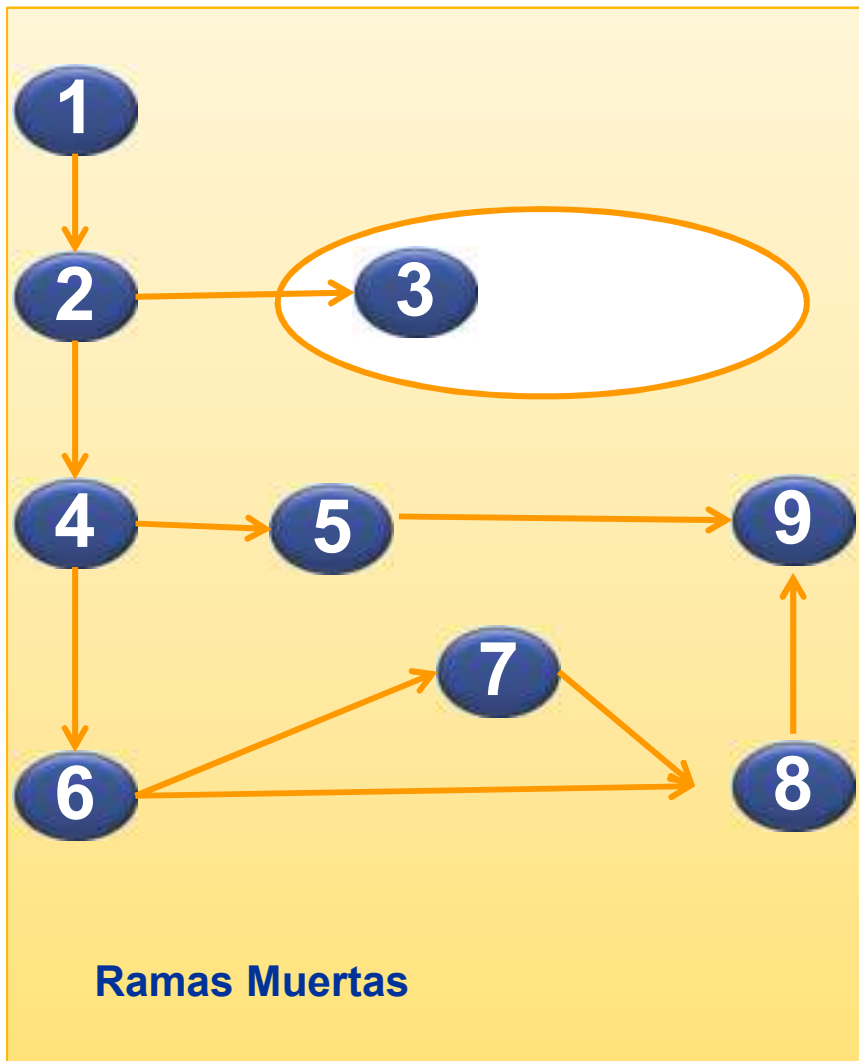
- ▶ Primer camino: 1-2-3-9
- ▶ Segundo camino: 1-2-4-5-9
- ▶ Tercer camino: 1-2-4-6-7-8-9
- ▶ Cuarto camino: 1-2-4-6-9



Si hay una inconsistencia como un camino que no conduce a nada, un bucle abandonado, entre otros, estos se verán evidenciados en este flujo.

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas



Análisis de flujo de datos

Flujo de datos: “data flow” Representación abstracta de la **secuencia y posibles cambios de estado** de los objetos de datos, donde el estado de un objeto puede ser cualquiera de los siguientes: creación, uso o eliminación. [Beizer].

- ▶ Se analiza el efecto de cada instrucción
- ▶ Es mas fácil la detección de defectos en el lugar exacto.
- ▶ Se debe recorrer muchas sentencias para localizar los defectos.
- ▶ Iterativamente ir llevando el control de los datos a través del flujo de control, hasta que no existan mas cambios

Análisis de flujo de datos

- ▶ Las variables pueden tomar los siguientes estados:
 - Indefinido
 - Referenciada
 - Definida
 - No Utilizado

Ejemplo de Ciclo Infinito en programación C (WHILE)

Código

```
#include<stdio.h>
int main(void)
{
    int x = 0;
    while(x < 10)
    {
        if(x == 9)
            x = 0;
        x++;
        printf("No finaliza!\n");
    }
    return 0;
}
```

Análisis:

Se observa que la sentencia `printf("No finaliza\n");` siempre se ejecuta porque la condición del bucle `while()` siempre es cierta, no existe una condición de salida que obligue al bucle a finalizar.

Exactamente, no se alcanza la condición de salida.

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

La forma de que se salga del while y continúe normalmente es quitar las líneas de código del if, es decir:

```
#include<stdio.h>
int main(void)
{
    int x = 0;
    while(x < 10)
    {
        x++;
        printf("Si finaliza!\n");
    }
    return 0;
}
```


Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

Métricas

Métrica: “Metric” Escala de medida y el método utilizado para la medición. [ISO 14598]

- ▶ Se tiene diferentes medidas para el análisis estático por herramientas. Algunas de las métricas son:
 - Puntos de función
 - Estructuras de control de datos (por ejemplo métrica de Halstead (Halstead- Metric)).
 - Bang (De Marco), Complejidad del código fuente (como la complejidad ciclomática de los métodos)

Se encuentran herramientas para este análisis estático, ya sean libres o con licenciamiento.

Ver <http://www.euriko.com.mx/euriko-ia/39-productos/76-calidad-de-software.html>

Número Ciclomático

- ▶ Esta métrica fue propuesta por Thomas Mc Cabe en 1976. Se basa en el diagrama de flujo de control
- ▶ Define **el número de caminos independientes** dentro de un fragmento de código y determina **el número de pruebas que se deben realizar** para asegurar que se ejecuta una sentencia al menos una vez.
- ▶ Mc Cabe encontró que un **valor de 10 es un limite superior** practico para el tamaño del modulo. Si supera este valor se hace muy complejo probarlo, entenderlo y modificarlo.

Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

$$V(G) = A - N + 2p$$

A = Número de aristas del grafo. Una arista conecta 2 nodos

N= Número de nodos del grafo correspondiente a sentencias del programa

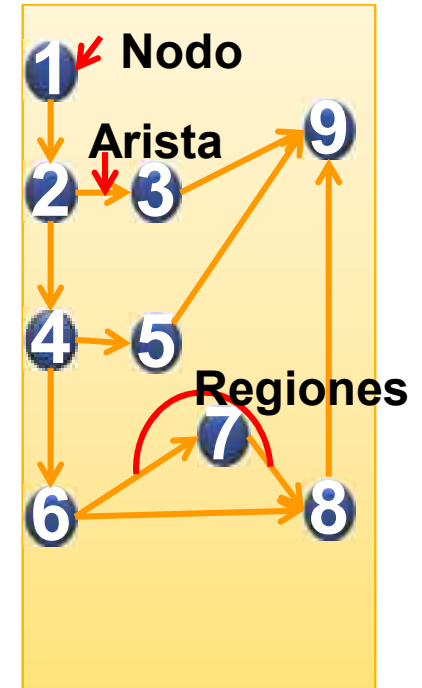
P= Numero de componentes o de partes independientes del programa.

Una Versión simplificada es:

$$V(G) = \text{Numero de condiciones} + 1$$

O $V(G) = \text{Numero de condiciones} + \text{Numero de retornos o salidas}$ (nunca es menor que 1)

Si estas dos formas dan resultados diferentes se debe a ramas superfluas o ramas faltantes.



Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas

$$V(G) = e - N + 2p$$

E= 11 aristas

N= 9

P= 1

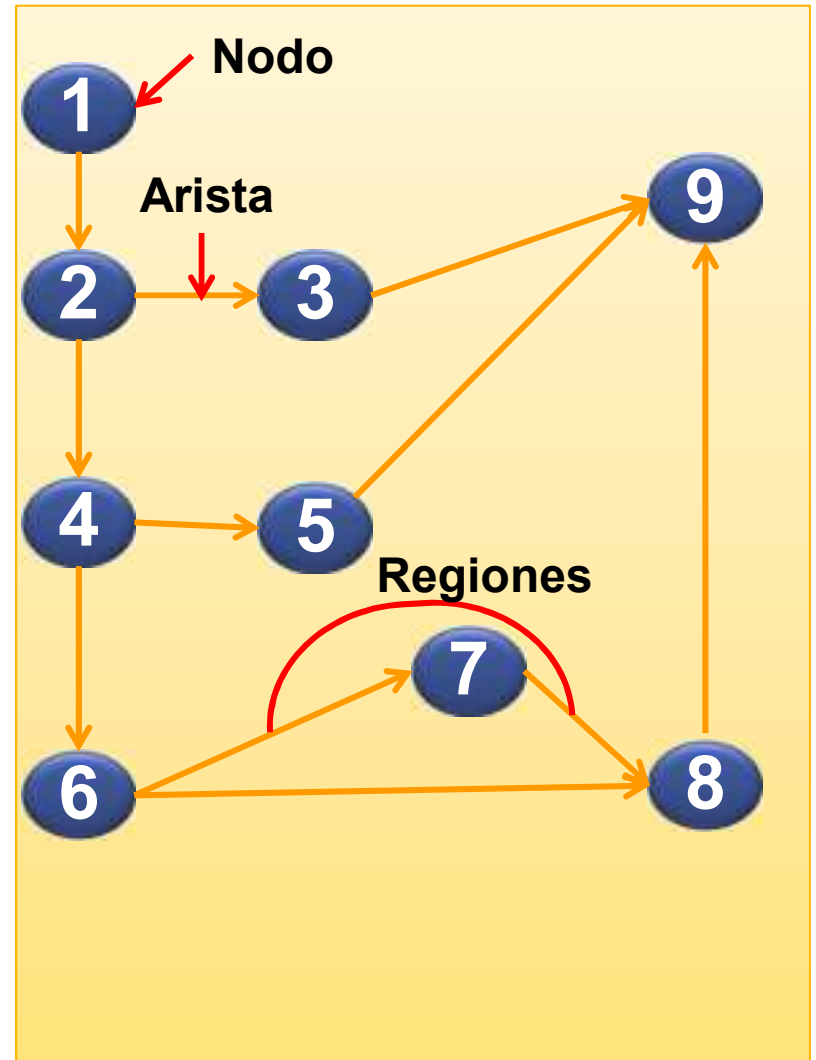
$$V(G) = 11 - 9 + 2(1)$$

$$V(G) = 4$$

$$V(G) = C + 1$$

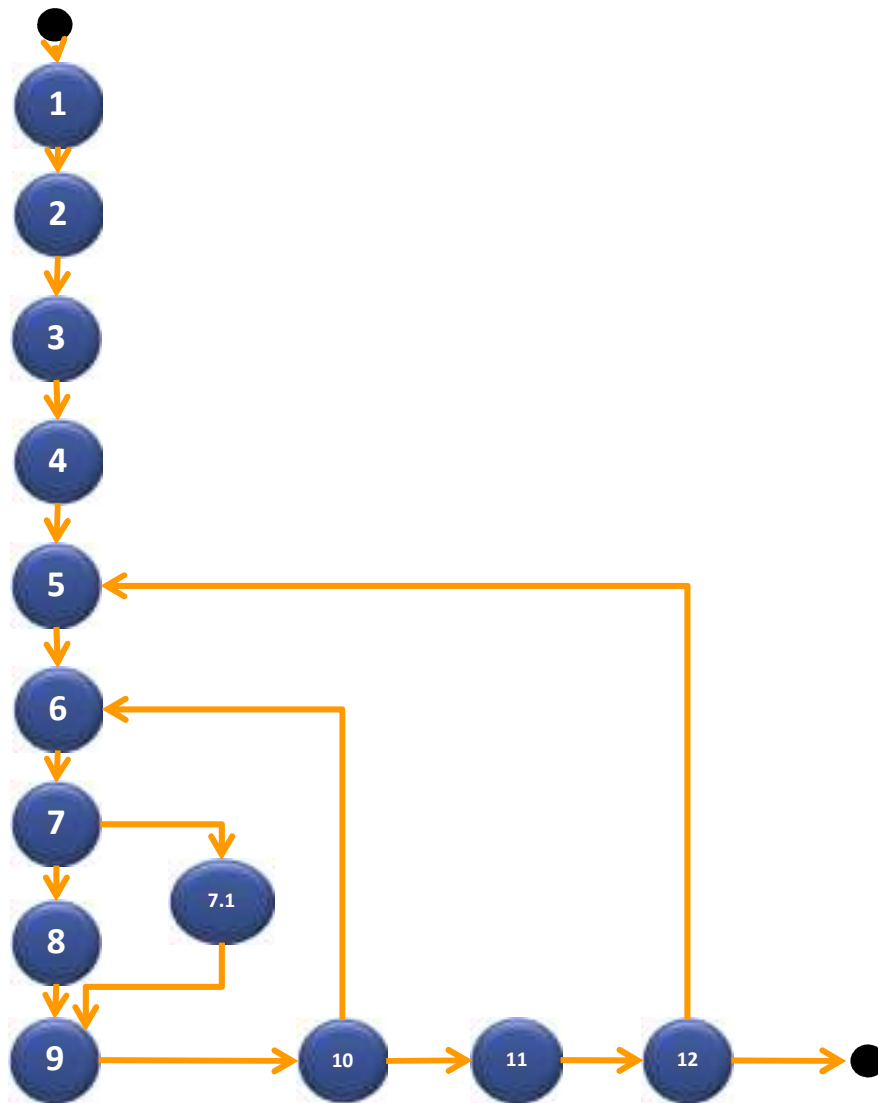
C= 3 condiciones

$$V(G) = 3 + 1 = 4$$



Capítulo 3 - Técnicas estáticas

Tema 3.3: Análisis estático por herramientas



Caminos identificados

Camino 1: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12

Camino 2: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 7.1 - 9 - 10 - 11 - 12

Camino 3: 1-2 - 3- 4- 5- 6 -7 - 8- 9- 10 - 6- 7- 8- 9- 10-11 -12

Camino 4: 1-2-3-4-5-6-7-8-9-10-11-12-5-6-7-8-9-10-11-12

Cálculo de la complejidad ciclomática

$$V(G) = A - N + 2$$

$$V(G) = 15 - 13 + 2$$

$$V(G) = 4$$

Donde: A = Aristas y N = Nodos