



UNIVERSIDAD  
Popular del cesar

# Ingeniería de Sistemas



# **INGENIERIA DE SOFTWARE II**

## **UNIDAD No. 2**

**Métricas de procesos y de  
proyectos de software**

# Métricas del Software

## ALCANCE DE LAS METRICAS

Métricas para entender, controlar y mejorar

- ❑ **Proceso:** Cualquier actividad relacionada con la producción de software, todas las actividades del ciclo de vida del software
  - Requisitos, diseño, codificación, pruebas, mantenimiento, administración de configuraciones
- ❑ **Producto:** Cualquier artefacto, entregable o documentos que resultan de cualquiera de las actividades del proceso de software
  - Especificaciones de requisitos, plan, código, caso de prueba
- ❑ **Recurso/ proyecto:** Cualquier elemento que es necesario para realizar el proceso
  - Gente, tiempo, hardware, software, método

# Métricas del Software

Para un Producto, Proceso o Recurso se tiene:

- **Atributos externos**

- Pueden ser medidos únicamente con respecto a su interacción con el ambiente
- Por ejemplo: Confiabilidad

- **Atributos Internos**

- Pueden ser medidos en términos puramente de las entidades en si mismas.
- Por ejemplo, líneas de código

# Métricas de procesos del Software

## Métricas de Procesos

Los aspectos relacionados con el proceso de desarrollo de software que pueden medirse son:

### Atributos internos:

- Duración de un proceso o de una de sus actividades
- Esfuerzo asociado con el proceso o con una de sus actividades
- Número de incidentes de un tipo determinado que ocurren durante el proceso o una de sus actividades

### Atributos externos:

- Calidad
- Estabilidad

# Métricas de procesos del Software

Entidad	Interno	Externo
Especificar	tiempo, esfuerzo, número de cambios en los requerimientos	calidad, costo, estabilidad efectividad
Pruebas	tiempo, esfuerzo, número de defectos encontrados	costo, costo-efectividad
Planeación	tiempo, esfuerzo, error de estimación	precisión, costo



# Métricas de procesos del Software

## CATEGORÍAS

- Aunque se ha propuesto una gran variedad de taxonomías métricas, las siguientes atienden las áreas más importantes de las métricas.
  - Métricas para el modelo de análisis
  - Métricas para el modelo de diseño
  - Métricas para el código fuente
  - Métricas para pruebas

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE ANÁLISIS

- En esta fase se obtendrán los requisitos y se establece el fundamento para el diseño.
- En esta fase las métricas proporcionan una visión interna a la calidad del modelo de análisis.
- Estas métricas examinan el modelo de análisis con la intención de predecir el **tamaño del sistema resultante**.



# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE ANÁLISIS

- **Métricas basadas en la función:** Utilizada para medir el tamaño del sistema a construir.
  - Son medidas indirectas del software y del proceso.
  - Se centran en la funcionalidad o utilidad del programa.
  - Emplean como un valor de normalización una medida de la funcionalidad que entrega la aplicación.
  - La métrica orientada a la función utilizada con mayor amplitud es el **punto de función (PF)**, se miden en función de:
    - Número de entradas de usuario.
    - Número de salidas de usuario.
    - Número de peticiones de usuario.
    - Número de archivos.
    - Número de interfaces externas.

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE ANÁLISIS

- **Métrica Bang:** Utilizada para medir el tamaño del sistema a construir. Métrica que permite estimar el tamaño del producto de software desde el punto de vista del usuario e independientemente de su implementación
- Se calcula:
  - El numero de primitivas funcionales,
  - Elemento de datos - ED,
  - Objetos de datos - O,
  - Relaciones,
  - Estados,
  - Transiciones,
  - Primitivas modificadas (funciones externas adaptadas),
  - ED de entrada,
  - ED de salida,
  - ED grabados.

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE ANÁLISIS

- **Métrica de la calidad de la Especificación**
  - Valoración del modelo con la especificación de requisitos (completitud, consistencia, ambigüedad, comprensión, etc.).
  - Se plantea una fórmula para cada uno de los atributos de la especificación, incluyendo requisitos Funcionales y no funcionales, interpretación, etc.).

# Métricas de procesos del Software

## Métricas de requisitos

- Las medidas de requisitos son únicas porque permiten caracterizar el espacio del problema.
- Se encuentran disponibles en etapas tempranas del desarrollo.
- Son útiles en la realización de proyectos, en la predicción de alternativas, riesgos y resultados a obtener.

# Métricas de procesos del Software

## Medidas del proceso de requisitos

- Numero de requisitos nuevos o modificados por mes

$$\text{Porcentaje de requisitos} = \frac{\text{Cambios en los requisitos}}{\text{Requisitos totales}}$$

### Compleción de los requisitos

$$\text{Tasa de requisitos incompletos} = \frac{\text{Requisitos no definidos}}{\text{Requisitos totales}}$$

### Requisitos probados

$$\text{Tasa de requisitos probados} = \frac{\text{Requisitos probados}}{\text{Requisitos totales}}$$

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE DISEÑO

- Proporcionan al diseñador una mejor visión interna.
- Ayudan a que el diseño evolucione a un nivel superior de calidad.
- Éstas se concentran en las características de la estructura del programa dándole énfasis a la estructura arquitectónica y en la eficiencia de los módulos.



# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE DISEÑO

- **Métricas del Diseño Arquitectónico:** se centran en la arquitectura del programa y la eficiencia de los módulos. Son de caja negra.
  - Medidas de la Complejidad Estructural, de Datos, del Sistema.
- **Métricas a nivel de Componentes:** se centran en las características internas del módulo. Son de caja blanca. Necesitan el diseño procedimental.
  - Métricas de Cohesión
  - Métricas de Acoplamiento
  - Métricas de complejidad del flujo de control del programa
- **Métricas del Diseño de Interfaz.**

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE DISEÑO

### Métricas del Diseño Arquitectónico:

- Relacionadas con el control intramodular:
  - **Tamaño**: medida del número de nodos, número de arcos o combinación de ambos.
  - **Profundidad**: medida del número de niveles del camino más largo entre el nodo raíz y un nodo hoja.
  - **Anchura**: Medida del número máximo de nodos de un nivel.
  - **Relación entre nodos y arcos**: medida de la densidad de conectividad.

# Métricas de procesos del Software

## MÉTRICAS PARA EL MODELO DE DISEÑO

### Métricas del Diseño Arquitectónico:

- Métricas basadas en la estructura de herencia del código:
  - Profundidad media de la jerarquía de herencia
  - Cantidad promedio de subtipos
  - Profundidad del árbol de herencia (DIT)
  - Número de Hijos (NOC)

# Métricas de procesos del Software

**Métricas basadas en estructura de diseño:**

**Métricas a nivel de Componentes:**

**Acoplamiento:** *Grado de interdependencia entre módulos*

Tipos de acoplamiento entre módulos

- R0: relación sin acoplamiento
- R1: relación de acoplamiento de datos
- R2: relación de acoplamiento por estampado
- R3: relación de acoplamiento de control
- R4: relación de acoplamiento común
- R5: relación de acoplamiento por contenido

# Métricas de procesos del Software

## Métricas basadas en estructura de diseño

### Métricas a nivel de Componentes:

**Cohesión:** *Grado en que los componentes locales a un módulo colaboran para realizar una tarea concreta.*

### Niveles de cohesión:

- **Funcional**
- **Secuencial**
- **Comunicaciones**
- **Temporal**
- **Procedimental**
- **Lógica**
- **Por coincidencia**

# Métricas de procesos del Software

## Métricas basadas en estructura de diseño:

- **Cohesión funcional:** Los elementos del módulo están relacionados en el desarrollo de una única función.
- **Cohesión secuencial:** Un módulo realiza distintas tareas en secuencia, de forma que las entradas de cada tarea son las salidas de la tarea anterior.
- **Cohesión comunicacional:** El módulo realiza actividades paralelas usando los mismos datos de entrada y salida.
- **Cohesión procedimental:** El módulo tiene una serie de funciones relacionadas por un procedimiento efectuado por el código. Es similar a la secuencial, pero puede incluir el paso de controles. Será deseable que las funciones estén relacionadas o realicen tareas dentro del mismo ámbito (p.e. la biblioteca [string.h](#) de [C](#) contiene funciones para operar con cadenas de caracteres).
- **Cohesión temporal:** Los elementos del módulo están implicados en actividades relacionadas con el tiempo.
- **Cohesión lógica:** Las actividades que realiza el módulo tienen la misma categoría. Esto es: es como si se tuvieran partes independientes dentro del mismo módulo.
- **Cohesión casual o coincidente:** Los elementos del módulo contribuyen a las actividades relacionándose mutuamente de una manera poco significativa. Este tipo de cohesión viola el principio de independencia y de caja negra de los módulos.



# Métricas de procesos del Software

## MÉTRICAS PARA EL CÓDIGO FUENTE

- Estas métricas asignadas como cuantitativas por Halstead, se derivan después de que se ha generado el código o se estima una vez que el diseño esté completo.
- Métricas de construcción:
  - Código desarrollado
  - Código reutilizado
  - Código destruidos: Código eliminado, comentado
  - Complejidad del código: McCabe, Halstead
  - Estadísticas en la inspección del código: No de errores detectados.
  - Porcentaje de errores encontrados y reparados

# Métricas de procesos del Software

## **Métricas basadas en código fuente:**

- N° de líneas de código.
- N° de líneas de comentario.
- N° de instrucciones.
- Densidad de documentación.

## **Líneas de código por método**

- Número de constructores por tipo
- Número de campos por tipo
- Número de métodos por tipo
- Número de parámetros
- Líneas de código

# Métricas de procesos del Software

## MÉTRICAS PARA PRUEBAS

- La mayoría de las métricas se concentran en el proceso y no en producto. Debe apoyarse en las métricas del análisis y del diseño.
- Las métricas basadas en la función, pueden utilizarse para predecir el esfuerzo global de las pruebas.
- La métrica Bang puede proporcionar una indicación del número de los casos de prueba necesarios para examinar una medida primitiva.
- A medida que avanzan las pruebas, hay métricas que indican la completitud de las mismas:
  - Amplitud de las pruebas (cuantos requisitos se han probado).
  - Profundidad de las pruebas (% de los caminos básicos probados).
  - Perfiles de fallos (para dar prioridad y categorizar los errores encontrados).

# Métricas de procesos del Software

## Métricas de Pruebas

- Dirección de la prueba: Medida para ayudar a la planificación y diseño de pruebas. Métricas de tamaño, complejidad.
- Tendencias que muestran los casos de pruebas: Número de casos de pruebas sin resolver.
- Aumento de efectividad: Medida sobre los tipos de errores detectados, permite hacer una clasificación de acuerdo a la causa, etapa de desarrollo.
- Densidad de fallos: Relación de números de errores encontrados y el tamaño del programa.
- Conclusión de pruebas: Medida para determinar si una prueba se puede dar por terminada

# Métricas de proyecto

El objetivo de las métricas de proyecto es el de reducir el costo y el tiempo total de desarrollo del mismo. Estas métricas nos permiten:

- Evaluar el estado del proyecto en curso.
- Realizar un seguimiento de los riesgos potenciales en un proyecto.
- Detectar las áreas de problemas antes de que se conviertan en “críticas”. o Ajustar el flujo y las tareas de trabajo.
- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la ingeniería del software.

# Métricas de proyecto

- Los recursos incluyen cualquier entrada en la producción de software
  - ☐ Personal
  - ☐ Materiales
  - ☐ Herramientas
  - ☐ Métodos ...
- Las medidas de recursos ayudan a controlar el proceso indicando cómo el proceso está usando y transformando las entradas en salidas
- **Atributos internos**
  - ☐ Tamaño del equipo
  - ☐ Tiempo de experiencia
  - ☐ Madurez de las herramientas ...
- **Atributos externos**
  - ☐ Coste
  - ☐ Productividad ...

$$\text{productividad} = \text{cantidad de salida} / \text{esfuerzo de entrada}$$

La productividad combina una medida de proceso (entrada) con una medida de producto (salida)



# Métricas de proyecto

- Los recursos son las entidades que se requieren en las actividades de proceso.
- Los recursos incluyen cualquier entrada en la producción de software: **personal, materiales, herramientas, métodos, costo, productividad**
- El **costo** generalmente se mide, a partir del resto de los recursos, pudiéndose ver como el coste de las entradas afecta al coste de las salidas.
- La **productividad** es otro atributo externo importante que depende del proceso de desarrollo.

# Métricas de proyecto

En un proyecto de implantación, las métricas más interesantes a aplicar y que ayudan a controlar el flujo de trabajo y las tareas técnicas son las siguientes:

- **Cantidad de Funcionalidad.** Se puede obtener a partir de las métricas de tamaño LOC (Lines of Code) o basado en los Puntos de Función.
- **Esfuerzo.** Cantidad de trabajo en Personas/Mes.
- **Fiabilidad.** Expresada en ratio de defectos.
- **Productividad** =  $\text{Tamaño} / \text{Esfuerzo}$
- **Tiempo / Calendario.** Duración del proyecto.
- **Velocidad\_de\_entrega** =  $\text{Tamaño} / \text{Duración}$

# Métricas de proyecto

Entidad	Interno	Externo
Personal	edad, salario	productividad, experiencia
Software	precio, tamaño	uso (usability), confiabilidad
Oficinas	temperatura, tamaño, luz	confort, calidad

# Métricas de proyecto

## Productividad

- La productividad de un recurso de software se mide en función de la proporción entre lo que entra y sale de un proceso de producción de software

**Productividad = tamaño de software/ esfuerzo**

**Productividad = PF/ esfuerzo**

Unidades más utilizadas:

- Tamaño: líneas de código, Punto de función - PF
- Esfuerzo: persona-meses, persona- días

# Métricas de proyecto

Además, se pueden calcular otras métricas interesantes: ·

**Productividad** = KLDC/ persona-mes

**Calidad** = errores / KLDC

**Documentación** = páginas de documentación / KLDC

# Métricas de proyecto

## Medición del personal

- La estructura del equipo del proyecto es un factor clave en la productividad
- El factor particular que afecta a la productividad es la **complejidad de las comunicaciones**: *complejidad causada por el número de individuos implicados y el método de comunicación requerido entre los miembros de un proyecto* [Grady y Caswell, 1987]
  - ❑ **Tamaño**: número de individuos del equipo.
  - ❑ **Densidad de comunicación**: Es la proporción entre el tamaño del equipo y el número de comunicaciones que se producen entre ellos.
  - ❑ **Nivel de comunicación**: Es un indicador que mide el grado de comunicación entre los miembros.
  - ❑ **Nivel de comunicación individual**: Es el número de individuos con que se comunica un determinado miembro del equipo
  - ❑ **Promedio del nivel de comunicación individual**: Es la medida de los niveles individuales



# Métricas de proyecto

## Equipos

- Se representa la estructura del equipo mediante un grafo donde los nodos son los miembros del equipo y los arcos los caminos de comunicación directa entre ellos, se pueden establecer las siguientes medidas:
  - ❑ **Tamaño:** número de nodos del grafo (**n**)
  - ❑ **Densidad de comunicación:** relación entre arcos (**e**) y nodos (**n**)  
Densidad de Comunicaciones =  $e/n$
  - ❑ **Nivel de comunicación:** medida de la impureza del árbol  
 $m(G) = 2.(e - n + 1) / (n - 1)(n - 2)$
  - ❑ **Nivel de comunicación individual (NCI):** Numero de comunicaciones del nodo
  - ❑ **Promedio del nivel de comunicación individual** = Suma NCI / n

# Métricas del Software

ENTIDADES	ATRIBUTOS	
	Internos	Externos
<b>Productos</b> Especificaciones, diseño, código...	Tamaño, reusabilidad, modularidad, funcionalidad, acoplamiento, complejidad...	Comprensión, mantenibilidad, calidad, fiabilidad ...
<b>Procesos</b> Realización de la especificación, del diseño, del código...	Tiempo, esfuerzo, cambios en requisitos, fallos en la especificación	Calidad, coste, estabilidad
<b>Recursos</b> Personal, equipos, hardware, software...	Edad, precio, tamaño del equipo, velocidad, tamaño de memoria	Productividad, experiencia, calidad, usabilidad, fiabilidad

# Ejemplo de Métrica de Madurez

Nombre:	Suficiencia de las pruebas
Propósito:	Cuántas de los casos de prueba necesarios están cubiertos por el plan de pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.
Medición, fórmula:	$X = A/B$ A = número de casos de prueba en el plan B = número de casos de prueba requeridos
Interpretación:	$0 \leq X$ Entre X se mayor, mejor la suficiencia.
Tipo de escala:	absoluta
Tipo de medida:	$X = \text{count/count}$ A = count B = count
Fuente de medición:	A proviene del plan de pruebas B proviene de la especificación de requisitos
ISO/IEC 12207 SLCP:	Aseguramiento de Calidad Resolución de problemas Verificación
Audiencia:	Desarrolladores Mantenedores

# Ejemplo de Métrica de Entendibilidad

Nombre:	Funciones evidentes
Propósito:	Qué proporción de las funciones del sistemas son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula:	$X = A/B$ A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mejor.
Tipo de escala:	absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A = count B = count
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Requeridores Desarrolladores



# Ejemplo de Conformidad de la Transportabilidad

<b>Nombre:</b>	Conformidad de transportabilidad
<b>Propósito:</b>	Qué tan conforme es la transportabilidad del producto con regulaciones, estándares y convenciones aplicables.
<b>Método de aplicación:</b>	Contar los artículos encontrados que requieren conformidad y comparar con el número de artículos en la especificación que requieren conformidad.
<b>Medición, fórmula:</b>	$X = A/B$ A = número de artículos implementados de conformidad B = total de artículos que requieren conformidad
<b>Interpretación:</b>	$0 \leq X \leq 1$ Entre más cercano a 1, más completa.
<b>Tipo de escala:</b>	absoluta
<b>Tipo de medida:</b>	$X = \text{count/count}$ A = count B = count
<b>Fuente de medición:</b>	Especificación de conformidad y estándares, convenciones y regulaciones relacionados. Diseño Código fuente Informe de revisión
<b>ISO/IEC 12207 SLCP:</b>	Verificación Revisión conjunta
<b>Audiencia:</b>	Requeridores Desarrolladores



**UNIVERSIDAD**  
Popular del cesar