



UNIVERSIDAD
Popular del cesar

Ingeniería de Sistemas



INGENIERIA DE SOFTWARE II

UNIDAD No. 2

METRICAS DEL SOFTWARE

METRICAS DEL SOFTWARE

MODELOS DE MEDICION



Métricas del Software

- La **medición** es esencial para cualquier disciplina de ingeniería y la **ingeniería de software** no es una excepción.
- Las **métricas de software** se refieren a un amplio rango de **medidas para el software**, dentro del contexto de la planificación del proyecto de software, el proceso de desarrollo y de los productos finales.
- Siendo una **métrica una medida** estadística, estas medidas son aplicables a todo el **ciclo de vida del desarrollo**, desde la iniciación, cuando se estiman los costos, al seguimiento y control de la fiabilidad de los productos finales, y a la forma en que los productos cambian a través del tiempo debido a la aplicación de mejoras.
- Un ingeniero del Software **recopila medidas y desarrolla métricas** para obtener **indicadores**.

Métricas del Software

La medición del software persigue tres objetivos fundamentales:

- Entender qué ocurre durante el **desarrollo y el mantenimiento**
- Permitir controlar qué es lo que ocurre en los **proyectos**, y
- Mejorar los **procesos y productos**

(Fenton y Pfleeger, 1997).



Métricas del Software

ALCANCE DE LAS METRICAS

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

Las métricas deben ser:

- **Correctos:** la recogida debe hacerse de acuerdo a las reglas exactas de la definición o de la métrica
- **Exactas:** la diferencia entre el valor resultante de la medida y el valor real del dato debe ser lo mínima posible
- **Precisas:** el número de cifras utilizadas para expresarlos debe ser la apropiada.
- **Consistentes:** evaluaciones diferentes sobre los mismos datos deben dar los mismos resultados
- **Comparables:** Para ello, debe estar normalizada.

Métricas del Software

Etapas del proceso de medición

- **Formulación:** la obtención de medidas y métricas apropiadas para la representación del SW que se desea desarrollar.
- **Colección:** mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- **Análisis:** cálculo de las métricas.
- **Interpretación:** evaluación de los resultados.
- **Realimentación:** recomendaciones obtenidas a través de la interpretación de las métricas obtenidas por el equipo de desarrollo.

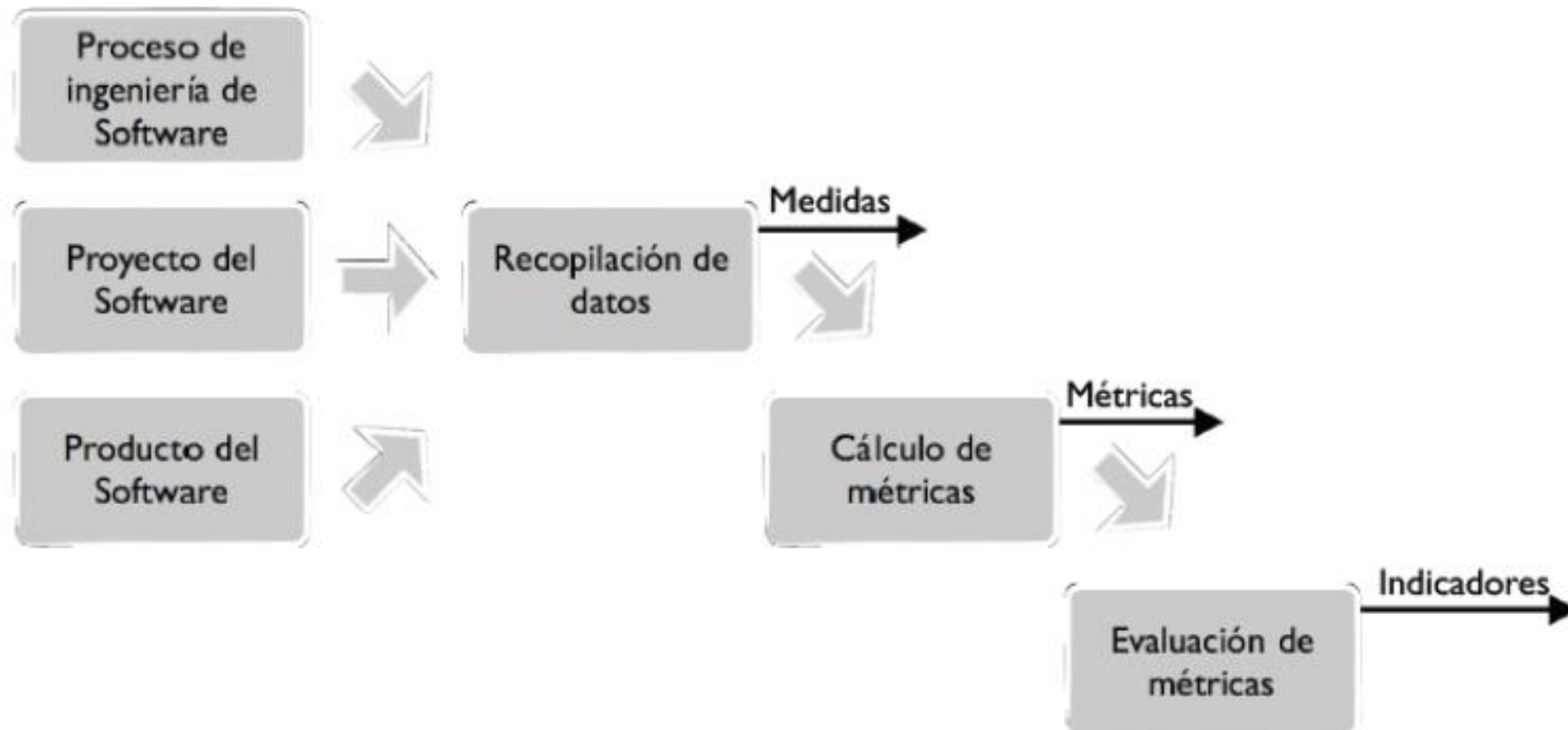
Métricas del Software

El ***proceso de medición*** incluye las siguientes actividades [McGarry et al., 2002]:

- **Planificación de las mediciones:** proporciona un método consistente para identificar las necesidades de información del proyecto, seleccionar y especificar las medidas e integrarlas en los procesos técnicos y de gestión del proyecto
- **Realización de las mediciones:** implica la recolección de datos de medida, el análisis de los datos y la presentación de resultados
- **Evaluación de las mediciones:** el proceso de medición y las medidas específicas se evalúan periódicamente y se mejoran si es necesario
- **Establecimiento y mantenimiento de un compromiso:** consiste en establecer los recursos, formación y herramientas para implementar un programa de medida efectivo, así como asegurar que existe el compromiso de usar la información producida

Métricas del Software

PROCESO DE RECOPIACIÓN DE MÉTRICAS



Métricas del Software

Recogida de datos métricos

■ Definición de los datos

Hay dos tipos de datos:

- ***Datos sin refinar*** resultantes de la medición inicial
- ***Datos refinados*** obtenidos extrayendo los elementos de datos relevantes de los datos sin refinar



Recolección de datos para la medición

GQM: Objetivos, Preguntas, Métricas

Modelo de medición

El **enfoque GQM** (*Goal-Question-Metric*) (Basili y Weiss, 1984) (Basili y Rombach, 1988) puede utilizarse para seleccionar e implementar métricas de una manera efectiva.

GQM se puede aplicar a todo el ciclo de vida del producto, procesos, proyecto y se puede alinear fácilmente con el ambiente organizacional. Tiene como principio básico que la medición debe ser realizada, siempre, orientada a un objetivo.



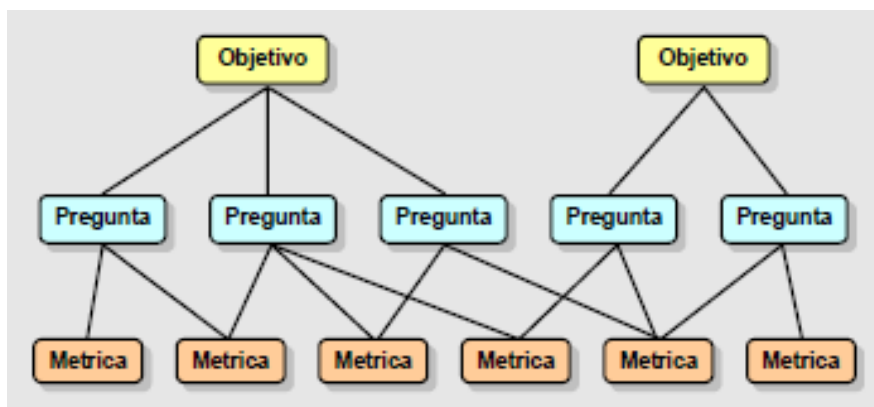
GQM: Objetivos, Preguntas, Métricas

- El método GQM ayuda en la definición de objetivos de una entidad.
- Una vez establecidos los objetivos, se pueden refinar a través de preguntas cuya respuesta permitirá concluir si los objetivos se cumplieron o no.
- Asociado a las preguntas se definen métricas cuyos valores ayudaran a contestar las preguntas.

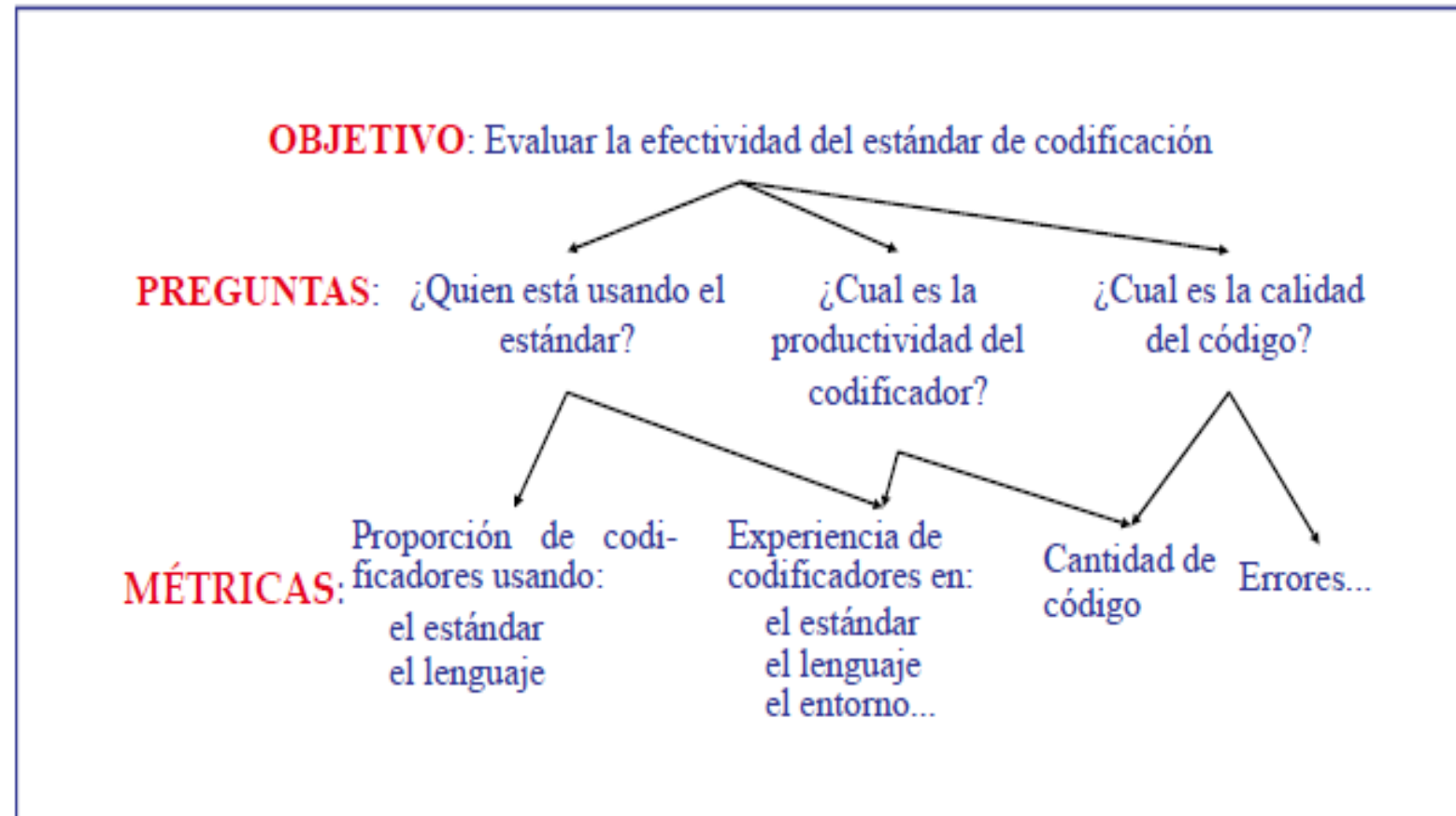
GQM: Objetivos, Preguntas, Métricas

El enfoque GQM se aplica en tres pasos:

- Lista de los objetivos principales del desarrollo y mantenimiento del proyecto.
- Para cada objetivo obtener las preguntas que deben contestarse para saber si se están cumpliendo los objetivos.
- Decidir qué medir para poder contestar las preguntas de forma adecuada.

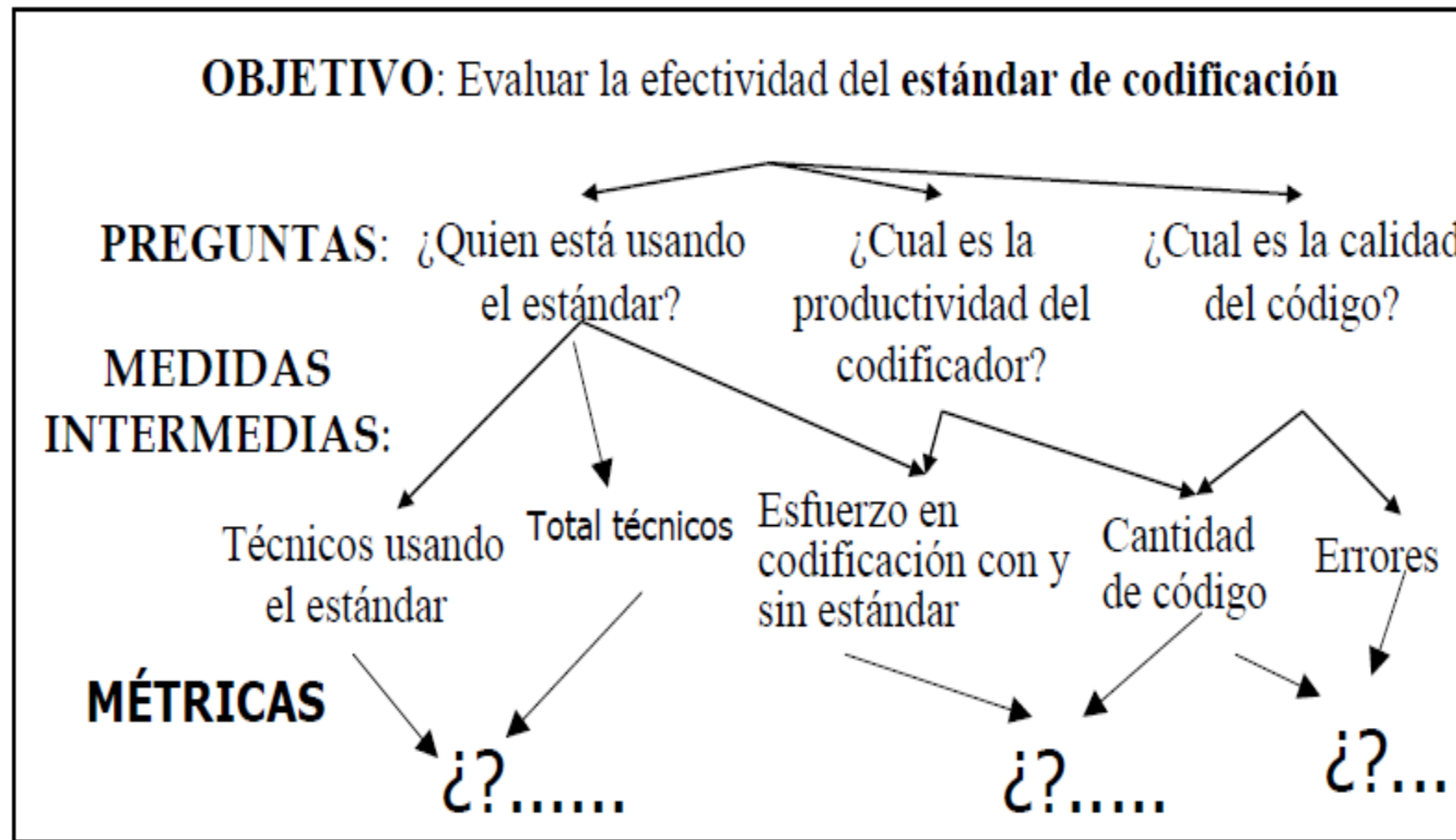


GQM: Objetivos, Preguntas, Métricas



Ejemplo de métricas derivadas de objetivos y preguntas (GQM)

GQM: Objetivos, Preguntas, Métricas





GQM: Objetivos, Preguntas, Métricas

Ejercicio

OBJETIVO: Mejorar el rendimiento del sistema

Preguntas:

Medidas

Métricas



Métricas de producto

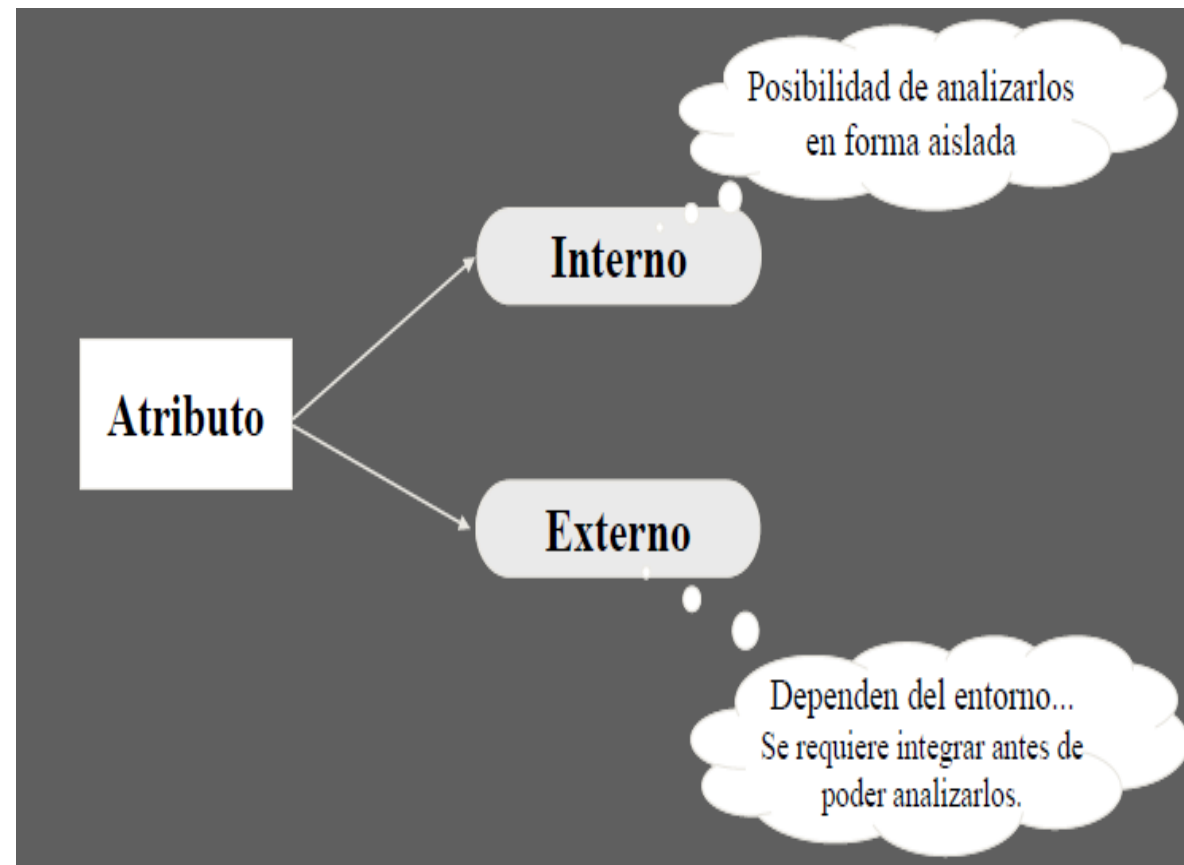
RAZONES PARA MEDIR UN PRODUCTO

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación.
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Métricas de producto

Un **Producto** se mide de acuerdo a sus **atributos**:

- **Atributos Internos:** Pueden ser medidos en términos puramente de las entidades en si mismas.
- **Atributos externos:** Pueden ser medidos únicamente con respecto a su interacción con el ambiente.



Métricas de producto

- Las características medibles del producto son:
 - **Atributos externos:**
 - Usabilidad
 - Integridad
 - Eficiencia
 - testeabilidad
 - Reusabilidad
 - Portabilidad
 - ...
 - **Atributos internos:**
 - Tamaño del producto
 - Longitud de las especificaciones
 - Modularidad del diseño
 - Acoplamiento y cohesión
 - Complejidad del código
 - ...

Métricas de producto

Entidad	Interno	Externo
Especificaciones	tamaño, reutilización, modularidad, corrección sintáctica	entendible, mantenibilidad
Diseños	tamaño, reuse, modularidad, acoplamiento, funcionalidad	calidad, complejidad mantenibilidad
Código	tamaño, reuse, complejidad algorítmica, flujo de control estructuración	confiabilidad, mantenibilidad

Medidas del producto: Atributos internos

Los atributos internos describen los productos de software de forma que dependen únicamente del producto mismo.

El producto puede ser descrito en función de:

- Su ***tamaño***
- Sus ***propiedades estructurales***

Medidas del producto: Atributos internos

Las **propiedades estructurales** del software son atributos internos relacionados con la calidad del producto. Los tipos de medidas estructurales son:

- **Flujo de control:** secuencia en que se ejecutan las instrucciones.
- **Flujo de datos:** seguimiento de cómo los datos se crean y se manejan por un programa.
- **Estructura de los datos:** organización de los datos independiente del programa.

Medidas del producto: Atributos internos

Los atributos usados para medir el **tamaño** del software:

- **Longitud:** Tamaño físico del producto.
- **Funcionalidad:** Funciones que proporciona el producto al usuario.
- **Complejidad:** Del problema, Algorítmica, Estructural, Cognitiva

Medidas del producto: Atributos internos

Tamaño - Longitud

- El tamaño se determina habitualmente tomando como referencia el programa en código fuente.
- El tamaño es una medida empleada por tres razones:
 - Es **fácil de obtener** una vez que el programa ha sido completado,
 - es uno de los factores **más importantes** en los métodos de desarrollo,
 - y la **productividad se expresa tradicionalmente con el tamaño del código.**
- Los principales productos que es útil medir son **la especificación, el diseño y el código.**

Medidas del producto: Atributos internos

Tamaño - Longitud

Código

- El **numero de líneas de código** (LOC) es la medida más usada para medir la longitud del código fuente.
- Para programas grandes es más adecuado el uso de **KLOC** (miles de líneas de código).
- Se han realizado muchas propuestas para contarlas. La más extendida es la de HP que **contabiliza**:
 - **Líneas de código efectivas** (ELOC: *effective lines of code*), que corresponden a las líneas **no comentadas** (NCLOC).
 - **Líneas de código comentadas** (CLOC).

Medidas del producto: Atributos internos

Tamaño - Longitud

Código

- Es útil medir por separado las **líneas comentadas** (CLOC) de las no comentadas para calcular esfuerzo, productividad, etc. La longitud total será:

$$\text{LOC} = \text{NCLOC} + \text{CLOC}$$

- También puede ser útil calcular la **densidad de comentarios**:

$$\text{DC} = \text{CLOC} / \text{LOC}$$

Medidas del producto: Atributos internos

Tamaño - Longitud

Código

- Para propósitos tales como la prueba es importante conocer cuanto código ejecutable se produce, para ello se mide el **número de sentencias ejecutables** (ES), ignorando los comentarios, declaraciones de datos y cabeceras
- Otra propuesta consiste en contabilizar únicamente el **código entregado al cliente**. Se cuenta el número de DSI (*delivered source instruction*) que incluye las declaraciones de datos, las cabeceras y las instrucciones del código fuente

Medidas del producto: Atributos internos

Tamaño - Longitud

Métricas de Halstead

- Considera un programa P como un conjunto de ***tokens*** que se pueden clasificar como operandos y operadores
- Parecidos a los tokens que un compilador puede distinguir en ese código.
- Se basa en computar los operadores y operandos de un programa.
- Las métricas básicas para los tokens son:
 - n1**: número de operadores únicos
 - n2**: número de operandos únicos
 - N1**: número total de ocurrencias de operadores
 - N2**: número total de ocurrencias de operandos

Medidas del producto: Atributos internos

Tamaño - Longitud

Métricas de Halstead

- Los **operadores** son las **palabras reservadas del lenguaje**, tales como IF-THEN, READ, FOR, ...;
- Los **operadores aritméticos** +, -, *, los de asignación y los operadores lógicos AND, EQUAL TO,-
- Los **operandos** son las **variables**, **literales** y las **constantes** del programa.
- Halstead distingue entre el número de operadores y operandos **únicos** y el número total de **ocurrencias** de operadores y operandos.

Medidas del producto: Atributos internos

Tamaño - Longitud

Métricas de Halstead

Por ejemplo, un programa puede tener un READ, siete asignaciones y un WRITE; por lo tanto tiene tres operadores únicos, pero nueve ocurrencias en total de los operadores, y de manera idéntica se procede con los operandos.

Medidas del producto: Atributos internos

Tamaño - Longitud

Métricas de Halstead: Métricas compuestas para un programa:

- **El tamaño** (N) o longitud de un programa:

$$N = N1 + N2$$

- **Vocabulario** (n) de un programa:

$$n = n1 + n2$$

- **Longitud estimada:**

$$L = N1. \log_2(n1) + N2. \log_2(n2)$$

- **Volumen:**

$$V = N \log_2(n)$$

$$\text{donde } N = N1 + N2 \text{ y } n = n1 + n2$$

- **Esfuerzo de implementación:**

$$E = (n1. N2. N. \log_2(n)) / (2. n2)$$

- **Tiempo de desarrollo de un programa:**

$$T = E / B$$

B : nº de discriminaciones mentales por segundo. Lo fija en 18.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

- Para realizar estimaciones de esfuerzo y duración es mejor estimar el tamaño del producto en base a su funcionalidad.
- La mayoría de los enfoques miden la funcionalidad con los documentos de especificación.
- El PF es una Medida de funcionalidad propuesta por Albrecht, 1979.
- Es una medida del producto y del proceso que se sigue para desarrollarlo.
- Está centrado en la “funcionalidad” o “utilidad” del producto.
- Los puntos de función son un *número positivo* con escala arbitraria que refleja el *tamaño de la funcionalidad del sistema* (a mayor tamaño mayor número de puntos de función)

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

Los Puntos de Función se caracterizan por:

1. Ser un **método independiente** de las herramientas de análisis, diseño y programación, debido a que se preocupa sólo de la complejidad de las funciones a implementar.
2. **Requerir de una descomposición funcional** del proyecto de software a realizar, en términos tales que se detecten todas las piezas elementales que componen el producto final. (Funciones elementales).
3. **Estimar la "cantidad de Puntos de Función"** de las funciones medidas, se realiza contando la cantidad de entradas, salidas, archivos, consultas e interfaces que utiliza. A mayor cantidad, mayor es el "peso de complejidad" que se le asignará.
4. **Ajustar la estimación del esfuerzo requerido**, por la vía de determinar la presencia de ciertos elementos que dificultan el desarrollo del proyecto. 5.
5. **Permitir realizar una estimación del esfuerzo requerido** (y por ende de la duración del proyecto) en etapas tempranas del proyecto (Etapa de Análisis).

Los beneficios de utilizar esta técnica son evidentes, permite realizar la estimación del trabajo requerido en una etapa temprana del proyecto y es independiente del entorno tecnológico a utilizar.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

- Utilizada para medir el tamaño del sistema a construir.
- Se calcula teniendo en cuenta las siguientes medidas:
 - El numero de entradas,
 - El numero de salidas,
 - El numero de peticiones - consultas,
 - El numero de archivos, y
 - El numero de interfaces externas
- Asociados a un numero de complejidad (simple, medio, complejo), más un factor de complejidad.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

El **PF** es una métrica que se calcula en función de la siguiente medidas:

- *Puntos de función Sin Ajuste (PSF)*
- *Factor de Complejidad de Procesamiento (FCP)*

$$PF = PFS * FCP$$

donde,

$$PFS = \Sigma((\text{número de ítems de la clase } i) * \text{pesoi})$$

$$FCP = 0.65 + (0.01 \times \text{Puntos de Complejidad de Procesamiento})$$

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

- ***Puntos de función Sin Ajuste (PSF)*** se obtiene utilizando una relación **empírica** basada en ítems del producto y valoraciones subjetivas de la complejidad del mismo.
- Se determinan los siguientes elementos de alguna representación del software:
 - **Entradas:** Entradas de usuario que proporcionan datos a la aplicación.
 - **Salidas:** Salidas que proporcionan información al usuario.
 - **Consultas:** Peticiones interactivas que requieren una respuesta.
 - **Interfaces con otros sistemas :** Interfaces con otros sistemas legibles por la máquina.
 - **Almacenamientos lógicos** (logical files): Ficheros maestros lógicos del sistema.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Entradas: Los datos se pueden dar por pantallas, formularios, cuadros de dialogo, controles o mensajes, a través de los cuales un usuario final o cualquier otro programa pueda añadir, borrar o cambiar datos de un programa.

Salidas: Pantallas, informes, gráficos o mensajes que el programa genera para el usuario final o cualquier otro programa. Esto incluye cualquier salida que tenga formato diferente o requiera un procesamiento diferente a otros tipos de salida.

Consultas: Una consulta está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva. Recuperación de datos de uno o más archivos lógicos internos y de archivos externos de la interfaz

Almacenamientos lógicos: Los principales grupos lógicos de datos de usuarios finales o información de control que están completamente controlados por el programa. Un archivo lógico podría constar de un único archivo plano o de una sola tabla en una base de datos relacional.

Archivos de interfaz externos: Archivos controlados por otros programas, con los que el programa va a interactuar. Esto incluye cada uno de los principales grupos de datos lógicos o información de control que entre o salga en el programa.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Puntos de función (PF)

Para cada ítem se le asigna un índice de complejidad (simple, medio o complejo) y un factor de peso en función del índice, cuyos valores se recogen de la tabla siguiente:

Parámetros de medición	Cuenta	Factor de ponderación					
		Simple	Medio	Complejo			
Número de entradas de usuario	<input type="text"/>	×	3	4	6	=	<input type="text"/>
Número de salidas de usuario	<input type="text"/>	×	4	5	7	=	<input type="text"/>
Número de peticiones de usuario	<input type="text"/>	×	3	4	6	=	<input type="text"/>
Número de archivos	<input type="text"/>	×	7	10	15	=	<input type="text"/>
Número de interfaces externas	<input type="text"/>	×	5	7	10	=	<input type="text"/>
Cuenta total							<input type="text"/>

$$PFS = \Sigma((\text{número de ítems de la clase } i) * \text{peso } i)$$

Simple: Contiene pocos datos/tipos/campos, pocos archivos internos se ven afectados en su procesamiento, y/o su interfaz de usuario es simple/fácil de desarrollar

Promedio: No es claramente ni simple ni complejo

Complejo: Contiene muchos datos/tipos/campos, muchos archivos internos se ven afectados en su procesamiento, y/o su interfaz de usuario es compleja/difícil de desarrollar

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Factor de Complejidad de Procesamiento

- Son una *medida del grado de influencia en la complejidad del sistema*, de una serie de factores preestablecidos
- Se consideran 14 factores diferentes
- Cada factor se evalúa en una escala de 0 (no tiene influencia) a 5 (influencia muy fuerte). 0=sin influencia, 1=accidental, 2=moderado, 3=medio, 4=significativo, 5=esencial.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Factor de Ajuste		Descripción	Peso
1	Comunicación de Datos	¿Cuántas facilidades de comunicación hay disponibles para ayudar con el intercambio de información con la aplicación o el sistema?	4
2	Procesamiento distribuido de los datos	*Distribuida" significa que los componentes (o los datos) de la aplicación están distribuidos en dos o más procesadores diferentes (esto también incrementa el factor anterior). ¿Cómo se manejan los datos y las funciones de procesamiento distribuido?	3
3	Rendimiento	¿Existen requerimientos de velocidad o tiempo de respuesta?	4
4	Configuraciones fuertemente utilizadas	¿Qué tan intensivamente se utiliza la plataforma de hardware donde se ejecutará la aplicación o el sistema?	2
5	Tasas de Transacción	¿Con qué frecuencia se ejecutan las transacciones? diarias, semanales, mensuales...	4
6	Entrada de datos On-line	¿Qué porcentaje de la información se ingresa on-line?	0
7	Diseño para la eficiencia de usuario final	¿Se designa la aplicación para maximizar la eficiencia del usuario final?	4
8	Actualizaciones on-line	¿Cuántos archivos lógicos internos se actualizan por una transacción on-line?	0
9	Procesamiento complejo	¿Hay procesamientos lógicos o matemáticos intensos en la aplicación?	1
10	Reusabilidad	La aplicación se desarrolla para suplir una o muchas de las necesidades de los usuarios	5
11	Facilidad de instalación	¿Es muy difícil la instalación y la conversión al nuevo sistema?	4
12	Facilidad de operación	¿Cómo de efectivos y automatizados son los procedimientos de arranque, parada, backup y restore del sistema?	2
13	Puestos Múltiples	¿La aplicación fue concebida para su instalación en múltiples sitios y organizaciones?	5
14	Facilidad de cambio	La aplicación fue concebida para facilitar los cambios sobre la misma	2
GRADO TOTAL DE INFLUENCIA (TDI)			40

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Factor de Complejidad de Procesamiento (FCP)

Se calcula con la fórmula:

$$FCP = 0.65 + (0.01 \times \text{Puntos de Complejidad de Procesamiento})$$

- Su rango de valores está entre 0.65 (0 puntos de complejidad) hasta 1.35 (70 puntos de complejidad)

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Conteo	Elementos		Peso		Resultado
10	Entradas	x	4	=	40
10	Salidas	x	5	=	50
10	Consultas	x	4	=	40
1	Archivos Lógicos	x	10	=	10
1	Archivos de Interfaz	x	7	=	7
Total					147

Factor de Influencia	Puntuación
Comunicaciones de datos	0
Procesamiento distribuido	0
Objetivos de rendimiento	4
Configuración de uso intensivo	3
Tasas de transacción rápidas	3
Entrada de datos en línea	4
Amigabilidad en el diseño	4
Actualización de datos en línea	2
Procesamiento complejo	3
Reusabilidad	0
Facilidad de instalación	4
Facilidad operacional	4
Multiplicidad de emplazamientos	5
Versatilidad	4

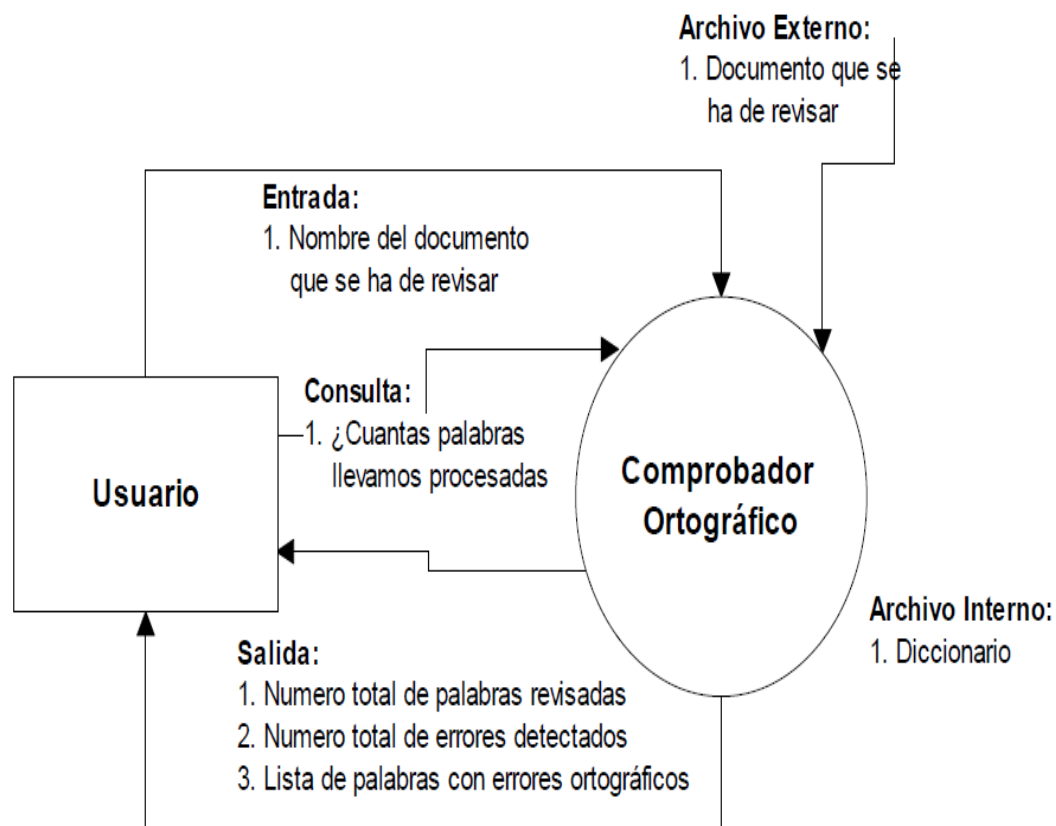
PFS = $\Sigma((\text{número de items de la clase } i) * \text{peso})$

FCP = $0.65 + (0.01 \times PCP)$

PF = $PFS * FCP$

Medidas del producto: Atributos internos

Tamaño - Funcionalidad



- Para este sencillo programa, el número de elementos es $1+3+1+1+1=7$.

Medidas del producto: Atributos internos

Tamaño - Funcionalidad

Líneas de **código KLOC** se determinan, teniendo en cuenta los puntos de función **PF**:

KLOC= (PF * Líneas de código por cada PF)/1000

Lenguaje	Lineas de codigo / Puntos de Funcion			
	Pro	Med	Min	Máx
--				
ASP	56	50	32	106
Assembler	209	203	91	320
C	148	107	22	704
C++	59	53	20	178
C#	58	59	51	66
FoxPro	36	35	34	38
J2EE (Java)	57	50	50	67
Java	55	53	9	214
JavaScript	54	55	45	63
JSP	59	-	-	-
.NET	60	60	60	60
Perl	57	57	45	60
PL/SQL	47	39	16	78

Lenguaje	Correlación Código Fuente por PF (aprox)
Assembler	320
C	128
ALGOL	105
FORTRAN	105
PASCAL	91
RPG	80
PL/I	80
Modula-2	80
Prolog	64
LISP	64
BASIC	64
4GL para BD	40
APL	32
Smalltalk	29
Query	13
Spreadsheet	6
Sql	13
VB	24
Java	46
Html	14
Delphi	118
C++	53
COBOL	107

<http://www.sc.ehu.es/jiwdocoj/mmis/fpa.htm>

Medidas del producto: Atributos internos

Tamaño - Complejidad

Métricas de complejidad McCabe:

La complejidad ciclomatica se basa en la cuenta del numero de caminos lógicos individuales contenidos en un programa.

$$V(G) = a - n + 2$$

Donde

a : # de arcos o aristas del grafo.

n : # de nodos.

Complejidad Ciclomática	Complejidad del código
1-10	Simple, sin riesgos
11-20	Algo complejo, riesgo moderado
21-50	Complejo, riesgo elevado
51+	Muy difícil de probar, riesgo muy alto

Medidas del producto: Atributos internos

Especificaciones y diseño

- Los documentos de especificación de requisitos y de diseño tienen representaciones de muchos tipos (texto, gráficos, símbolos...)
- La medición del atributo longitud exige la identificación de elementos atómicos que puedan contarse. Ejemplo:
- Diagramas de flujo de datos: procesos, entidades externas, almacenes de datos y flujo de datos
 - Salidas, funciones, operaciones y axiomas, líneas de especificación (declaraciones y predicados), páginas de documentación como objetos atómicos.

Vista	Diagrama	Objetos atómicos
Funcional	Diagrama de flujo de datos Diccionario de datos	Burbujas Elementos de datos
Datos	Diagrama entidad relación	Objetos, relaciones
Estado	Diagrama de transición de estados	Estados, transiciones

Medidas del producto: Atributos internos.

METRICAS ORIENTADAS A CLASES

La **clase** es la unidad principal de todo sistema **orientado** a objetos. Las **características** principales del enfoque orientado a objetos son:

- **Localización:** La localización es una característica del software que indica la forma que se concentra la información dentro de un programa.
 - Dado que las clases constituyen la unidad básica de los sistemas OO, la localización está basada en los objetos. Por tanto, las métricas deberían de ser aplicables a la clase (objeto) como si se tratara de una entidad completa.
- **Encapsulamiento:** Inclusión dentro de un mismo objeto de los datos (información que describe su estado) y los métodos (mecanismos de transformación o de acceso al estado de dichos objetos).
 - El encapsulamiento influye en las métricas cambiando el objetivo de la medida, que pasa de ser un único módulo a ser un paquete de datos (atributos) y de módulos de procesamiento (operaciones).

Medidas del producto: Atributos internos.

METRICAS ORIENTADAS A CLASES

Características:

- **Ocultamiento de información:** Suprimir los detalles operacionales de un componente de un programa.
 - Un sistema OO bien diseñado debería de impulsar al ocultamiento de información. Por tanto, aquellas métricas que proporcionen una indicación del grado en que se ha logrado el ocultamiento proporcionarán una indicación de la calidad del diseño OO.
- **Herencia:** El mecanismo que permite que las responsabilidades de un objeto puedan ser propagadas a otros objetos.
 - La herencia se basa en una relación jerárquica entre varias clases de tal forma que se pueden compartir atributos y operaciones en la subclase de la superclase de la que hereda ciertas propiedades y añade propiedades particulares.

Medidas del producto: Atributos internos.

METRICAS ORIENTADAS A CLASES

Características:

- **Abstracción:** Mecanismo que permite a los diseñadores enfocarse en los detalles esenciales de un componente del programa con un poco conocimiento de los detalles de bajo nivel.
 - Dado que una clase es una abstracción que se puede visualizar con muchos niveles distintos de detalles, y de muchas maneras diferentes, las métricas OO representan la abstracción en términos de medidas de una clase (p.ej.: número de instancias por clase por aplicación).

Medidas del producto: Atributos internos.

METRICAS ORIENTADAS A CLASES

Serie de métricas CK (Chidamber/Kemerer)

- Estas métricas propuestas de diseño basadas en clases
 - ❖ Métodos ponderados por clase
 - ❖ Árbol de profundidad de herencia
 - ❖ Número de descendientes
 - ❖ Acoplamiento entre clases
 - ❖ Respuesta para una clase
 - ❖ Carencia de cohesión en los métodos

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Serie de métricas CK (Chidamber/Kemerer)

Métodos ponderados por clase (WMC: Weighted Methods per Class). Calcula la suma de la complejidad ciclomática de los métodos de una clase:

$$WMC = \sum C_i \quad i=1.....n$$

- Siendo **C_i** la complejidad ciclomática del método **i**. La complejidad está relacionado con el esfuerzo requerido para implementar y probar una clase.

- El WMC debe ser lo más bajo posible. Cuanto más alto es el valor WMC, más complejo el árbol de herencia y menos reutilizable. Las principales interpretaciones de esta métrica son las siguientes:
 - El número y la complejidad de los métodos son indicadores del tiempo necesario para desarrollar/mantener la clase.
 - Cuanto mayor sea el nº de métodos mayor impacto potencial tendrá en los hijos, sus herederos potenciales.
 - Las clases con gran nº de métodos serán de aplicación específica, y por lo tanto más difíciles de reutilizar.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

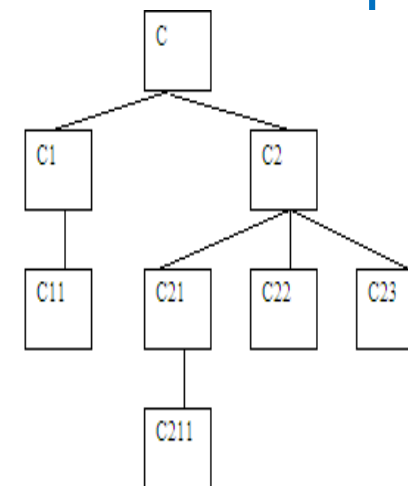
Serie de métricas CK (Chidamber/Kemerer)

Profundidad en el árbol de herencia (DIT: Depth Inheritance Tree). Es la distancia desde una clase a la raíz del árbol de herencia.

- Cuanto más alto es el mayor valor de DIT, mayor complejidad hay en el diseño
- Cuanto más alto sea el valor de DIT de una clase más posibilidades existen de que reutilice/refine métodos heredados.

Las principales interpretaciones de esta métrica son las siguientes:

- A mayor profundidad de la clase, más métodos puede heredar y es más difícil de explicar su comportamiento.
- A mayor profundidad de una clase, mayor posibilidad de reutilización de métodos heredados.



Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Serie de métricas CK (Chidamber/Kemerer)

Número de hijos inmediatos en el árbol de herencia (NOC: Number Of Children).

- Cuanto más alto es el valor de NOC, una clase es más reutilizable pero también la probabilidad de que se hayan hecho extensiones no apropiadas de la clase es mayor.

Las principales interpretaciones de esta métrica son las siguientes:

- Cuanto mayor sea NOC más reutilización habrá por herencia.
- Si NOC es muy grande hay un fallo en la abstracción de la clase padre, falta algún nivel intermedio.
- NOC da una idea del peso que la clase tiene en el diseño, y de los recursos que se deben dedicar a probar sus métodos.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Serie de métricas CK (Chidamber/Kemerer)

Acoplamiento entre clases (CBO: Coupling Between Object Classes).

- Es el número de clases acopladas a una clase.
- Dos clases están acopladas cuando los métodos de una de ellas usan variables o métodos de una instancia de la otra clase.
- Si existen varias dependencias sobre una misma clase es computada como una sola.

Las interpretaciones de esta métrica son las siguientes:

- Cuanto mayor es CBO, peor es la modularidad y la reutilización.
- Cuanto mayor es CBO, peor es el encapsulamiento y más cuesta mantenerlo.
- No es deseable que $CBO > 14$. Cuanto más alto es el más difícil será el mantenimiento y el reuso y en general el código será más propenso a fallos.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Serie de métricas CK (Chidamber/Kemerer)

Respuesta de una clase - RFC (Response for a class).

- Es el conjunto de métodos que potencialmente se ejecutan en respuesta de un mensaje recibido por un objeto de esa clase.
- Se suman los métodos públicos de la clase más los métodos que son llamados dentro del código interno de la clase.
- Aumenta la dificultad de las pruebas.
- Aumenta la complejidad del diseño.

- El conjunto respuesta para la clase puede ser expresado de la siguiente manera:

$RS = \{M\} \cup \{R_i\}$, donde $\{R_i\}$ es el conjunto de métodos llamados por el método i ; y $\{M\}$ es el conjunto de todos los métodos en la clase.

- Cuanto mayor es **RFC**, más complejidad tiene el sistema ya que se puede invocar un mayor número de métodos como respuesta a un mensaje.
- A mayor respuesta de una clase, mayor complejidad de esta.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Serie de métricas CK (Chidamber/Kemerer)

Falta de cohesión de los métodos LCOM (Lack of cohesion in methods)

- Cuando un método de una clase accede a uno o más atributos de la clase.
- LCOM es el número de métodos que acceden uno o más atributos iguales.
- Dos métodos son similares si comparten al menos un atributo de la clase.

- A mayor número de atributos similares, mayor cohesión hay en la clase.
 - Si ningún método accede a sus atributos, LCOM=0.
 - Si una clase tiene 6 métodos y 4 de ellos tienen un atributo en común, LCOM=4

Las interpretaciones de esta métrica son las siguientes:

- A mayor cohesión mayor encapsulamiento.
- Un valor grande puede indicar que la clase debe dividirse.
- Baja cohesión indica alta complejidad y alta probabilidad de error en el desarrollo.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Métricas propuestas por Lorenz y Kidd

Lorenz y Kidd dividen las métricas basadas en clases en cuatro categorías:

- Tamaño
- Herencia
- Valores internos
- Valores externos.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Métricas propuestas por Lorenz y Kidd

Métricas de Tamaño: Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema orientado a objetos en su totalidad. Las métricas correspondientes al tamaño son:

- **PIM:** Número de métodos de instancia públicos.
- **NIM:** Todos los métodos de una instancia.
- **NIV:** Todas las variables de una instancia.
- **NCM:** Todos los métodos de una clase.
- **NIM:** Todas las variables de una clase.
- Si existen valores grandes de Tamaño éstos mostrarán que una clase puede tener demasiada responsabilidad, lo cual reducirá la reutilización de la clase y complicará la implementación y su comprobación.
- Cuanto menor sea el valor medio para el tamaño, más probable es que las clases existentes dentro del sistema se puedan reutilizar ampliamente.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Métricas propuestas por Lorenz y Kidd

Métricas de **herencia**: Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones a lo largo y ancho de la jerarquía de clases.

- **NMO**: Número de métodos sobrecargados.
- **NMI**: Número de métodos heredados.
- **NMA**: Número de métodos añadidos, número total de métodos que se definen en una subclase.
- **IE**: Índice de especialización para cada clase. Cómo una subclase redefine el comportamiento de superclase.

$$IE = (\text{Número de métodos sobrescritos} * \text{nivel de anidamiento jerarquía}) / \text{número total de métodos}$$

- Cuanto más elevado sea el valor de IE es más probable que la jerarquía de clases tenga clases que no se ajustan a la abstracción de la superclase.

Medidas del producto: Atributos internos

Métricas Orientadas A Clases

Métricas propuestas por Lorenz y Kidd

- Las métricas para **valores internos** de clase examinan la cohesión y asuntos relacionados con el código.
- Las métricas orientadas a **valores externos** examinan el acoplamiento y la reutilización.

Medidas del producto: Atributos internos.

METRICAS ORIENTADAS A CLASES

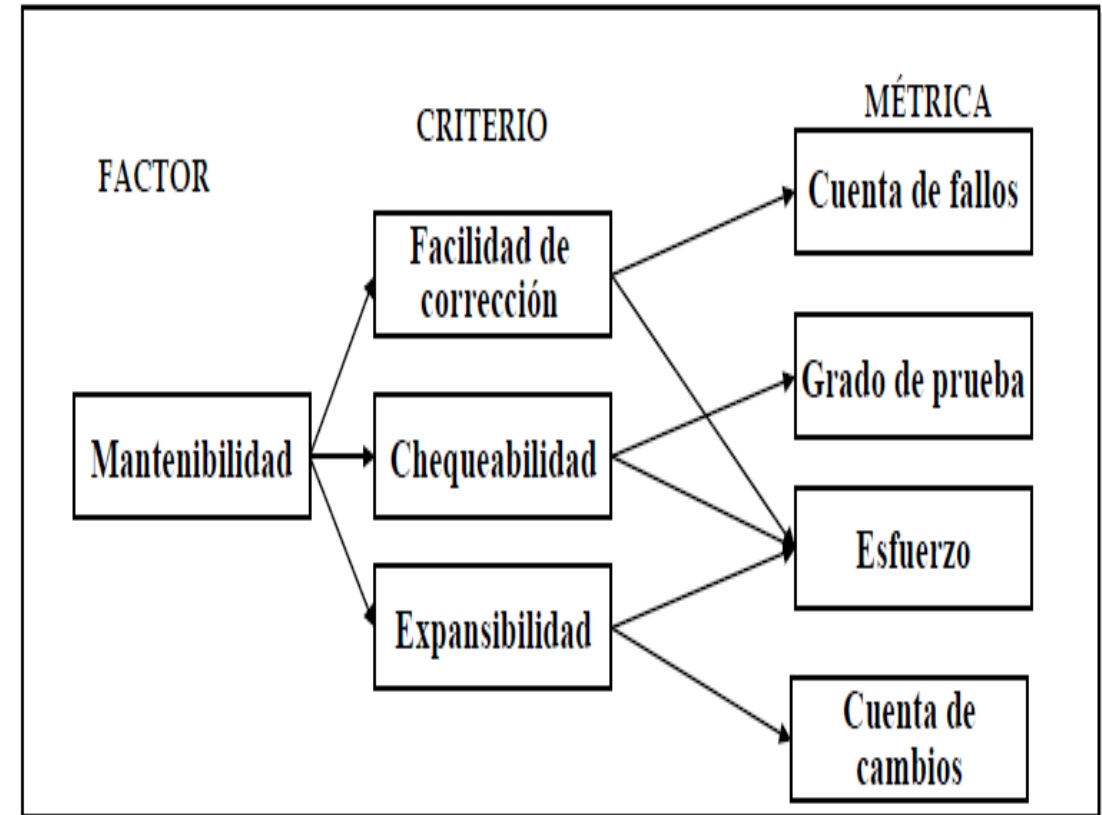
- **Colección de métricas MDOO**
 - Factor de herencia de métodos
 - Factor de acoplamiento
 - Factor de polimorfismo

Medidas del producto: Atributos Externos

- Los atributos externos de un producto son aquellos que pueden medirse únicamente con respecto a cómo el producto se relaciona con su **entorno**.
- Los atributos externos sólo son **medibles** cuando el producto está **completo**.
- La mayoría están relacionados con algún aspecto de la **calidad**.

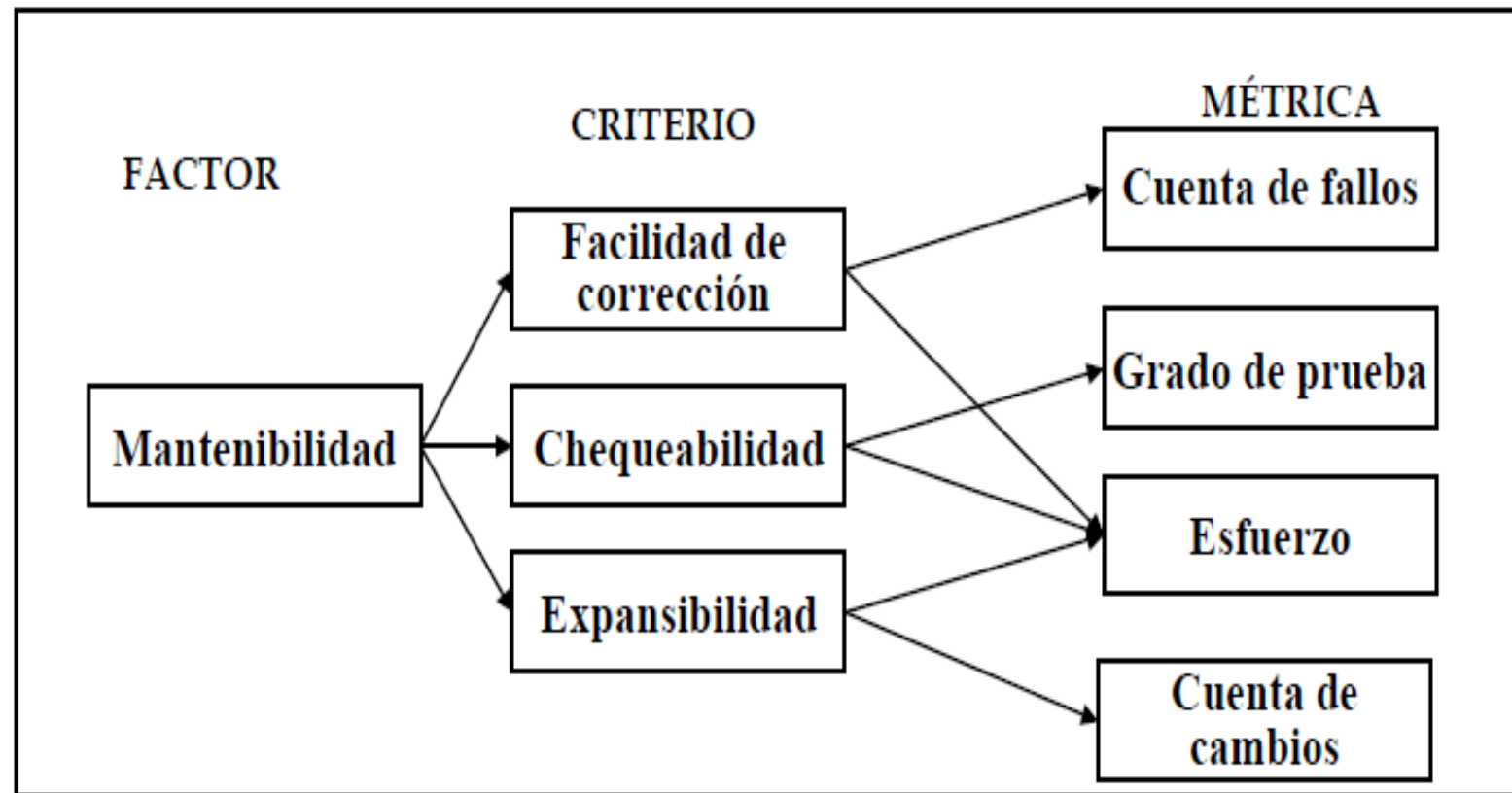
Medidas del producto: Atributos Externos

- Los modelos de calidad recogen atributos denominados **factores de calidad** (atributos de alto nivel).
- Los factores de calidad puede descomponerse en atributos de bajo nivel denominados **criterios de calidad**.
- Los criterios de calidad pueden asociarse con un conjunto de atributos de bajo nivel medibles directamente obteniendo las **métricas de calidad**.



Descomposición del factor "mantenibilidad"

Medidas del producto: Atributos Externos



Descomposición del factor “mantenibilidad”

Medidas del producto: Atributos Externos

Características operacionales

- **Fiabilidad**
 - **Corrección:** un programa es correcto cuando efectúa sin errores las tareas que le son encomendadas.
 - **Robustez:** capacidad de un programa para responder a situaciones no previstas.
- **Eficiencia:** ofrecer un máximo de prestaciones requiriendo un mínimo de recursos.
- **Facilidad de uso:** esfuerzo que se necesita para aprender un programa, para trabajar con él, preparar la entrada e interpretar su salida.
- **Integridad:** el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.

Medidas del producto: Atributos Externos

Características de modificación: capacidad de soportar cambios.

- **Facilidad de mantenimiento:** el esfuerzo requerido para modificar un programa operativo.
- **Extensibilidad:** adición de nuevas funcionalidades de manera sencilla.
- **Facilidad de prueba:** el esfuerzo requerido para probar un programa de forma que asegure que realiza su función requerida.

Medidas del producto: Atributos Externos

Características de transición: adaptabilidad a nuevos entornos.

- **Portabilidad:** capacidad de ejecutarse en diferentes plataformas sin cambios.
- **Reusabilidad:** propiedad de los componentes software por la que pueden aplicarse a tareas distintas de aquellas para las que se construyeron inicialmente.
- **Interoperabilidad:** los sistemas deben ser capaces de interoperar con otros componentes software.

Medidas del producto: Atributos Externos

Medición de la portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Medidas del producto: Atributos Externos

Medición de la portabilidad

- Se entiende por portabilidad la facilidad de mover una aplicación de un entorno a otro.
- La portabilidad se puede expresar como:

$$\text{portabilidad} = 1 - (ET/ER)$$

ET: medida de los recursos necesarios para mover el sistema a otro entorno

ER: medida de los recursos necesarios para crear el sistema en el entorno residente

Medidas del producto: Atributos Externos

Medición de la portabilidad

Característica: Portabilidad		
Subcaracterística: Facilidad de instalación		
Métricas		
<i>Nombre</i>	<i>Propósito de la métrica</i>	<i>Medición o fórmula de cálculo</i>
Facilidad de instalación	¿Puede el usuario o quien mantiene el software fácilmente instalar el software en un ambiente operacional?	$X = A / B$ A = Número de casos en que el usuario es exitoso en la operación de instalación. B = Número total de casos en que el usuario intenta ejecutar la operación de instalación.

Medidas del producto: Atributos Externos

Medición de la portabilidad

Coexistencia: capacidad de un producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes

$$\text{Grado de coexistencia} = 1 - (A/B)$$

A: No de errores de incompatibilidad en una prueba

B: No de pruebas totales

Reemplazabilidad: capacidad del software para ser usado en lugar de otro, para el mismo propósito y en el mismo entorno.

$$\text{Grado de reemplazabilidad} = 1 - (A/B)$$

A: funciones ejecutadas que no logran el propósito

B: No total de funciones ejecutadas

$$\text{Adaptabilidad} = 1 - (A/B);$$

A: tiempo medio del entorno nuevo;

B: tiempo medio del entorno anterior

Medidas del producto: Atributos Externos

Métricas de Usabilidad del producto

- Las métricas de **usabilidad** son una herramienta que ayudan a definir la intersección entre la efectividad, la eficiencia y la satisfacción en el contexto del uso.
- Consisten en lograr finalizar las tareas y en la satisfacción de los usuarios mientras las realizan. Como:
 - Si están consiguiendo realizar lo que desean,
 - no les cuesta demasiado tiempo hacerlo,
 - no cometen muchos fallos o errores,
 - y tienen un alto concepto de la aplicación después de usarla,entonces estamos hablando de un producto usable.

Algunas métricas de usabilidad:

- Tasa de éxito de ejecución de una tarea
- Tiempo que toma realizar una tarea.
- Número de clics necesarios para realizar una tarea
- Medición de satisfacción o frustración

Medidas del producto: Atributos Externos

Métricas de eficiencia del producto

- Para evaluar el rendimiento de un sistema informático se requiere un conjunto de variables cuyos valores definen el comportamiento del sistema. Estas variables pueden hacer referencia tanto al comportamiento del hardware o del software como al comportamiento que el usuario percibe del sistema.
- La **eficiencia** es una característica principal que se refiere a la capacidad del producto de software para proporcionar un desempeño apropiado, en relación con la cantidad de recurso utilizado, bajo condiciones establecidas en determinado momento del tiempo.

Para determinar qué tan **eficiente es un producto**, se ha subdividido en tres subcaracterísticas que abarcan los aspectos más relevantes que se deben tener en cuenta al respecto:

- **Comportamiento en el tiempo:** Esta subcaracterística se refiere a la capacidad del producto de software para proporcionar adecuados tiempos de respuesta, de procesamiento y de tasas de eficiencia en el desempeño de su función, bajo condiciones establecidas.
- **Consumo de Recursos:** Esta subcaracterística principal se refiere a la capacidad del producto de software para utilizar una apropiada cantidad y tipos de recursos cuando el software desempeña su función bajo condiciones establecidas. Los recursos humanos están incluidos como parte de la productividad.
- **Conformidad en la eficiencia:** Esta subcaracterística se refiere a la capacidad del producto de software para adherirse a estándares o convenciones definidos para que el software cumpla con los lineamientos establecidos.

Medidas del producto: Atributos Externos

Métricas de mantenibilidad del producto

Es el conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema de software. Las subcaracterísticas de este factor son:

- Facilidad de Análisis
- Facilidad de Cambio
- Estabilidad
- Facilidad de Prueba
- Conformidad
- **Facilidad para ser analizado:** Es la capacidad del producto software para ser diagnosticadas las deficiencias o causas de los fallos en el software o para identificar las partes que han de ser modificadas.

- **Facilidad para ser cambiado:** Es la capacidad del producto software que se permite que una determinada modificación sea implementada.
- **Facilidad de ser probado:** Es la capacidad del producto software que permite que el software modificado sea validado.
- **Estabilidad:** Es la capacidad del producto software para evitar efectos inesperados debido a modificaciones del software.
- **Conformidad:** Es la capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

Medidas del producto: Atributos Externos

Medida de la calidad basada en defectos

Medición de la densidad de defectos

- Para cualquier producto software se pueden considerar dos tipos de defectos:
 - Defectos conocidos
 - Defectos latentes
- La densidad de defectos se puede definir en función de los primeros:

número de defectos conocidos

Densidad de defectos = -----

tamaño del producto

Medidas del producto: Atributos Externos

Medida de la calidad basada en defectos

Se puede medir la calidad en función de la relación:

Tiempo empleado en la corrección de defectos

Tiempo total de desarrollo del sistema

Referencias

- <http://avellano.usal.es/~mmoreno/APITema2.pdf>



UNIVERSIDAD
Popular del cesar