

## TEMA 8

### MÉTRICAS DEL SOFTWARE

#### 1. MÉTRICAS E INDICADORES DE LA CALIDAD

##### 1.1 Medida del tamaño

01 [Feb. 2005] ¿Cuál de las siguientes medidas sirven para cuantificar el tamaño de una aplicación?

- a) Errores.
- b) Documentación.
- c) Nivel del lenguaje.
- d) Funcionalidad. (pág. 222 y sig.)**

02 [Feb. 2004] [Sep. 2004] [Sep. 2005] [Feb. 2007] [Sep. 2007] ¿Cuál de las siguientes es una métrica de tamaño?

- a) Hombre-mes.
- b) Punto-función.
- c) Tokens.(pág. 222 y sig.)**
- d) Errores.

03 [Sep. 2004] [Sep. 2005] [Feb. 2007] [Feb. 2009] ¿Cuál de los siguientes factores NO está directamente relacionado con la medida del tamaño de la aplicación software?

- a) CLOC.
- b) Reusabilidad.
- c) Punto-función.
- d) Hombre-mes. (pág. 222 y sig.)**

**Nota común a 01 a 03:** Las métricas de tamaño son Líneas de código, Tokens, Funcionalidad y Reusabilidad. El análisis del punto función es una alternativa a la medida del tamaño.

04 [Sep. 2004] [Sep. 2005] Indicar que utilización es falsa:

- a) La medida tradicional del tamaño del software son las líneas de código (LOC).
- b)  $LOC = CLOC - NLOC$ . (pág. 223)**
- c) Normalmente las líneas de comentario (CLOC) se contabilizan como líneas de código.
- d) Las LOC son una medida sencilla de obtener pero presentar algunos inconvenientes.

**Nota:** La fórmula correcta es  $LOC = CLOC + LOC$

05 [Feb. 2009] La medida de la densidad de comentarios es:

- a) NLOC + CLOC.
- b) CLOC / LOC. (pág. 223)**
- c) CLOC / NLOC.
- d) NLOC / LOC.

**Nota:** La densidad de comentarios es el cociente entre el número de líneas de comentarios (CLOC) y el número de líneas totales (LOC).

06 [Feb. 2010] En la medida del tamaño de una aplicación, se denomina “token” a:

- a) Un elemento propio del lenguaje que posee significado por si mismo. (pág. 224)**
- b) Cualquier línea de texto del programa excluyendo comentarios y líneas en blanco.
- c) Un punto de ruptura de la secuencia del código.
- d) Un punto de entrada a una instrucción de iteración.

07 [Feb. 2004] [Sep. 2006] [Sep. 2008] Un elemento propio de un lenguaje que posee significado por sí mismo es un:

- a) Punto Función.
- b) Token. (pág. 224)**
- c) Indicador.
- d) Esfuerzo.

**Nota común a 06 a 07:** El enunciado de la pregunta 07 es la definición de muestra o token.

08 [Feb. 2005] [Feb. 2007] ¿Cuál de las siguientes ecuaciones tiene en cuenta el hecho de la reusabilidad según el modelo COCOMO?

- a)  $S_e = S_n + aS_u/100$ . (pág. 226)**
- b)  $S_e = S_n + S_u k$ .
- c)  $S_e = S_n + S_u/100$ .
- d)  $S_e = S_n + aS_u$ .

**Nota:**  $S_u$  es el código reutilizado y  $S_n$  es el código nuevo.

## 1.2 Medida de la Productividad

01 [Feb. 2005] [Sep. 2005] ¿Cuál de las siguientes medidas sirven para medir la productividad?

- a) Tokens.
- b) Puntos Función. (pág. 227)**
- c) Reusabilidad.
- d) Tasas.

02 [Feb. 2006] [Feb. 2007] ¿Cuál de los siguientes factores puede NO ser considerado en la medida de la productividad?

- a) Tokens. (pág. 227 a 230)**
- b) Tamaño del producto.
- c) Cantidad de documentación.
- d) Calidad del producto resultante.

**Nota común a 01 y 02:** Las métricas de la productividad son el Hombre-Mes, los Puntos Función, los errores, la documentación y el nivel del lenguaje; pero además, hay que considerar la calidad del producto.

03 [Feb. 2006] [Sep. 2006] La aproximación a la productividad apoyada en el concepto *nivel del lenguaje* fue propuesta por:

- a) Halstead.
- b) Jones. (pág. 229)**
- c) Boehm.
- d) McCabe.

**Nota:** Capers Jones relaciona el nivel del lenguaje (menor número de estamentos requeridos para codificar un Punto Función) con la productividad y el tamaño del código.

04 [Sep. 2007] En la medida de la productividad del desarrollo del software:

- a) El KLOC se considera atributo externo.
- b) Se deben considerar factores como la calidad y la documentación. (pág. 229 y 230)**
- c) Sólo interviene el tamaño del producto resultante y el número de personas involucradas.
- d) El nivel del lenguaje empleado en la codificación influye linealmente.

**Nota;** En la medida de la productividad del software hay que tener en cuenta, además del control y la valoración del equipo de desarrollo (hombre-mes, puntos función, errores, documentación y nivel de lenguaje) la calidad del software resultante.

## 2. INDICADORES Y MÉTRICAS RELACIONADAS CON LA CALIDAD

### 2.1 Densidad de defectos e indicadores de calidad del proceso

01 [Feb. 2006] La “tasa de propagación de defectos” se define como:

- a) Número de defectos descubiertos dividido por el número total de defectos corregidos.
- b) Número de defectos ocasionados al corregir un defecto dividido por el número total de defectos corregidos. (pág. 231)**
- c) Número de defectos que se propagan dividido por el tamaño del programa.
- d) Número de defectos no corregidos dividido por el número de defectos detectados.

02 [Feb. 2008] [Feb. 2009] La relación entre el número de defectos ocasionados al corregir un defecto y el número de defectos corregidos se denomina:

- a) Tasa de propagación de defectos. (pág. 231)**
- b) Eficiencia en la eliminación de defectos.
- c) Fiabilidad en la corrección de defectos.
- d) Tasa de defectos correctivos.

**Nota común a 01 a 02:** La tasa de propagación de defectos tiene en cuenta los errores que se producen al corregir un defecto detectado.

03 [Sep. 2007] La densidad de defectos de un programa:

- a) Se puede estimar empleando el cálculo de probabilidades.
- b) Se mide empleando el tamaño del programa en KLOC o puntos función, (pág. 231)**
- c) Es un atributo interno medido mediante atributos externos.
- d) Está relacionado con el atributo Facilidad de uso.

**Nota:** En la medida de la tasa se puede emplear el tamaño del programa en KLOC o los puntos función.

04 [Feb. 2010] [Sep. 2010] El “(número de objetos probados al menos una vez/número de objetos totales)\*100” es:

- a) La eficiencia de la eliminación de defectos.
- b) La tasa de efectividad de las pruebas. (pág. 232)**
- c) La tasa d propagación de defectos.
- d) La exactitud de la densidad de defectos encontrados en las pruebas.

**Nota:** La tasa d efectividad de las pruebas es el tanto por ciento del número de objetos probados al menos una vez.

### 3. FIABILIDAD

01 [Feb. 2006] [Feb. 2007] [Sep. 2009] La fiabilidad se expresa como:

- a) El número de defectos encontrados dividido por el número de ejecuciones.
- b) La probabilidad de que aparezca un error en un tiempo determinado.
- c) El número de errores detectados dividido por el número de líneas de código.
- d) La probabilidad de que no ocurra un error en un determinado tiempo. (Pág. 232)**

**Nota:** La fiabilidad de un sistema está ligada al tiempo que funciona sin fallos.

02 [Feb. 2009] [Feb. 2010] [Sep. 2010] El tiempo medio entre fallos es:

- a) MTTF. (pág. 233)**
- b)  $R(t)$ .
- c)  $F(t)$ .
- d)  $H(t)$ .

**Nota:** El tiempo medio entre fallos es el MTTF (Mean Time To Failure).

### 4. COMPLEJIDAD DEL SOFTWARE

#### 4.1 La Ciencia del Software de Halstead

01 [Feb. 2008] Las métricas de Halstead se basan en:

- a) El número de operadores y operandos de un programa. (pág. 234)**
- b) El flujo de control de un programa.
- c) El flujo de información entre módulos de un programa.
- d) La independencia funcional y cohesión entre los módulos de un programa.

**Nota:** La ciencia de Halstead se basa en el conjunto de medidas número de operadores distintos en el programa, número de operandos distintos, número total de ocurrencia de operadores y número total de ocurrencia de operandos.

02 [Sep. 2006] En la ciencia del software de Halstead, el número mínimo de bits necesario para codificar un programa se denomina:

- a) Número ciclomático.
- b) Volumen del programa. (pág. 235)**
- c) Esfuerzo requerido.
- d) Ninguna de las anteriores.

**Nota:** Se define el volumen del programa como el número mínimo de bits necesario para codificarlo y viene dado por  $V = N \log (n_1 + N_2)$

## 4.2 La medida de la complejidad de McCabe

01 [Feb. 2006] La medida de la complejidad de McCabe se basa en:

- a) **La representación gráfica de un programa según su flujo de control. (pág. 235)**
- b) El flujo de información entre módulos.
- c) El flujo de información global.
- d) Los elementos sintácticos existentes en el programa.

02 [Feb. 2004] [Sep. 2004] La medida de la complejidad del software basada en el flujo de control del mismo es:

- a) La Ciencia del Software basada de Halstead.
- b) **La medida de la complejidad de McCabe. (pág.235)**
- c) La métrica de Henry y Kafura.
- d) El modelo de Constantine y Lockwood.

03 [Feb. 2005] [Sep. 2007] El valor de la complejidad de McCabe nos lo proporciona la ecuación:

- a)  **$V(G) = e - n + 2$ . (pag. 235)**
- b)  $V(G) = e.n + NI$ .
- c)  $V(G) = e.NI + 2$ .
- d)  $V(G) = n - 2$ .

04 [Sep. 2008] [Sep. 2009] [Sep. 2010] Podemos calcular el número de caminos independientes de un algoritmo con:

- a) La métrica de Henry y Kafura.
- b) La ciencia del software de Halstead.
- c) Las medidas de Nielsen.
- d) **El número ciclomático de McCabe. (pág. 237)**

05 [Sep. 2007] [Feb. 2008] Para realizar pruebas que aseguren que todas las sentencias de un programa se ejecuten al menos una vez:

- a) Nos basaremos en la teoría del esfuerzo requerido. (Halstead)
- b) Se recomienda utilizar las métricas de flujo de Henry y Kafura.
- c) **Podemos usar el número ciclomático de Mc Cabe. (pág. 237)**
- d) Hay que establecer un grafo de secuencias identificables contiguas.

**Nota común a 01 a 05:** La medida de la complejidad de McCabe o complejidad ciclomática se basa en la representación gráfica de un programa según el flujo del mismo y viene expresado por  $V(G) = e.n + 2$ , donde e es el número de flechas de conexión que implica una transferencia de control de un nodo a otro y n el número de nodos. El número ciclomático de Mc Cabe indica el número de pruebas necesario para que todas las sentencias se ejecuten al menos una vez.

### 4.3 La métrica de Henry y Kafura

01 [Feb. 2005] La medida de la complejidad que se basa en el flujo de información entre módulos la define la métrica de:

- a) McCabe y Kafura.
- b) Henry y Fenton.
- c) Henry y Kafura. (pág. 237)**
- d) McCabe y Halstead.

02 [Sep. 2004] [Sep. 2005] [Feb. 2007] La métrica de Henry y Kafura se basa en:

- a) La representación gráfica de un programa según su flujo de control.
- b) El flujo de información entre módulos. (pág. 237)**
- c) El flujo de información global.
- d) Los elementos sintácticos existentes en el programa.

**Nota común a 01 a 02:** La métrica de Henry y Kafura mide la complejidad basándose en el flujo de información entre módulos.

03 [Feb. 2009] El cociente entre el número de módulos con cohesión y el número total de módulos es la medida de cohesión entre módulos de:

- a) Henry y Kafura.
- b) Halstead.
- c) Piattini.
- d) Fenton. (pág. 237)**

**Nota:** Fenton propuso como medida de cohesión entre módulos el cociente indicado.

04 [Feb. 2005] En la métrica de Henry y Kafura el número de flujos locales hacia el procedimiento A más el número de estructuras de datos en las que el procedimiento A busca información es:

- a) El fan-out del procedimiento A.
- b) El flujo global de información.
- c) El fan-in del procedimiento A. (pág. 239)**
- d) El flujo local de información.

05 [Feb. 2006] En la métrica de Henry y Kafura, el número de flujos locales que vienen del procedimiento A más el número de estructuras de datos actualizadas por dicho procedimiento es:

- a) El fan-in del procedimiento A.
- b) El fan-out del procedimiento A. (pág. 239)**
- c) La complejidad del procedimiento A.
- d) El flujo global de información del procedimiento A.

**Nota común a 04 a 05:** La métrica de Henry y Kafura requiere la medida de los flujos global, local y locales directo e indirecto de información, el fan-in y el fan-out.

## 5. MÉTRICAS PARA MODELOS DE DATOS

### 5.1 Métricas a nivel de tabla

### 5.2 Métricas a nivel de estrella

01 [Sep. 2009] ¿Cuál de las siguientes es una métrica a nivel de estrella?

- a) **NFK(S). (pág. 240)**
- b) NFK(T).
- c) NFT(Sc).
- d) NASD(Sc)

**Nota:** La métrica a nivel de estrella es la NFK(S) o Número de claves ajenas de una estrella.

### 5.3 Métricas a nivel de esquema

01 [Feb. 2009] En el modelo de datos, el número de atributos de las tablas de dimensión compartida es:

- a) NFT(Sc).
- b) NASDT(Sc). (pág. 241)**
- c) NADT(Sc).
- d) NA(Sc).

**Nota:** El número de atributos de la tabla de dimensión compartida es  $NASDT(Sc) = \sum NA(DT_i)$  ( $i = 1$  a  $NSDT$ ).

### 5.4 Calidad de los propios datos

## 6. MEDIDAS DE FACILIDAD DE USO DE LAS INTERFACES DE USUARIO

### 6.1 Clasificación de los métodos

01 [Sep. 2009] [Feb. 2010] [Sep. 2010] ¿Cuántos criterios distinguen Wixon y Wilson para la clasificación de los métodos de medida de la facilidad de uso de los interfaces de usuario?

- a) 3.
- b) 4.
- c) **5. (pág. 243)**
- d) 6.

**Nota:** Wixon y Wilson distinguen 5 criterios de clasificación (Métodos formativos vs Métodos sumativos, Métodos de descubrimiento vs. Métodos de decisión, Métodos formales vs. Métodos informales, Métodos que involucran al usuario vs. Métodos que no involucran al usuario y Métodos completos vs. Métodos componentes).

### 6.2 Algunos métodos de evaluación

01 [Sep. 2009] [Sep. 2010] Para medir la satisfacción de los usuarios se utiliza el método:

- a) SUMI.
- b) WAMMI.
- c) SUS.
- d) **MUMMS. (pág. 244)**

**Nota:** El MUMMS (Measuring the Usability of Multi-Media Systems) mide la satisfacción de los usuarios en el uso de los interfaces multimedia.

02 [Sep. 2004] [Sep. 2005] [Feb. 2007] Dentro de los métodos de evaluación de la calidad de un interfaz de usuario tenemos:

- a) CALDEA.
- b) Interface Measure Method (IMM).
- c) SSE-CMM.
- d) **Usability Inspection. (pág. 244)**

**Noa:** Los métodos se clasifican en user testing (los participantes son los usuarios) y usability inspection (los protagonistas son los expertos).

03 [Feb. 2004] [Sep. 2006] [Sep. 2008] En el modelo de Constantine y Lockwood, la eficiencia se mide con la métrica:

- a) Essential Efficiency.
- b) Task Concordance. (pág. 245)**
- c) Layout Uniformity.
- d) Visual Coherence.

04 [Feb. 2006] En el método de Constantine y Lockwood la comprensibilidad se mide con la métrica:

- a) Comprehensibility.
- b) Task Concordance.
- c) Visual Coherence. (pág. 245)**
- d) Task visibility

**Nota común a 03 a 04:** Las características a medir y sus métricas en este modelo son: Simplicidad (Essential Efficiency), Eficiencia, Simplicidad (Task Concordance), Visibilidad (Task Visibility), Regularidad, Uniformidad (Layout Uniformity) y Comprensibilidad (Visual Coherence)

## 7. MEDIDAS DE SEGURIDAD

### 7.1 Un poco de historia

01 [Sep. 2009] El Common Criteria v. 1999 sirvió de base al estándar:

- a) CTCPEC.
- b) ISO 9126.
- c) ISO 27000.
- d) ISO 15408. (pág. 247)**

**Nota:** La norma ISO/IEC 15408 se basó en la versión de 1999 de los conocidos Common Criteria.

### 7.2 SSE-CMM

### 7.3 Métricas de eficacia de los algoritmos criptográficos

01 [Sep. 2004] [Sep. 2005] [Feb. 2008] [Feb. 2010] [Sep. 2010] ¿Cuál de los siguientes niveles no corresponde a los propuestos para la medida de fuerza de un algoritmo criptográfico?

- a) US.
- b) CS.
- c) CW. (pág. 249)**
- d) W.

02 [Sep. 2007] Un algoritmo criptográfico posee nivel CS si:

- a) No se puede decodificar mediante análisis computacional pero sí por ataque intensivo.
- b) Se puede decodificar en un tiempo razonable utilizando recursos computacionales elevados.
- c) Se puede decodificar mediante análisis computacional en tiempo razonable.
- d) No se puede decodificar mediante análisis sistemático en un período de tiempo lo suficientemente corto para que la información sea útil. (pág. 249)**

03 [Feb. 2004] [Sep. 2004] [Sep. 2005] [Feb. 2007] Un algoritmo criptográfico que puede descifrarse con claves no suficientemente largas tiene el nivel:

- a) CS.
- b) CCS. (pág. 249)**
- c) US.
- d) VW.

04 [Feb. 2004] [Sep. 2006] Un algoritmo criptográfico que puede descifrarse en un corto tiempo y a un bajo coste, tiene el nivel:

- a) US.
- b) CS.
- c) V.
- d) VW. (pág. 249)**

**Nota común a 01 a 04:** Los niveles son US (no es posible decodificar el contenido a partir de dicho contenido), CS (no se puede decodificar en un tiempo suficiente para que la información sea útil), CCS (se puede decodificar con claves que no son suficientemente largas), W (se decodifica mediante ataques de fuerza bruta en un tiempo razonable) y VW (puede decodificarse en un tiempo corto)

#### 7.4 Métricas de seguridad de red

01 [Sep. 2006] [Sep. 2008] ¿Cuál de las siguientes medidas de la seguridad en red NO es básica?

- a) Número de palabras claves de acceso caducadas.
- b) Número de usuarios externos al sistema. (pág. 251)**
- c) Frecuencia de las actualizaciones contra virus.
- d) Número de intrusiones detectadas en la red en un período de tiempo.

02 [Feb. 2006] ¿Cuál de las siguientes medidas de la seguridad en red es de proceso?

- a) Número de virus detectados en un período de tiempo.
- b) Porcentaje de sistemas con información sensible. (pág. 251)**
- c) Porcentaje de claves de acceso caducadas.
- d) Frecuencia de las actualizaciones antivirus.

**Nota común a 01 y 02:** Las métricas de proceso son número de usuarios externos, número de usuarios internos que salen a la red, número de firewalls, porcentaje de claves de entradas acorde con la política de seguridad, porcentaje de claves de entrada modificadas según la política y porcentaje de sistemas con información sensible.