



## ADMINISTRACIÓN Y GESTIÓN DE PROYECTOS DE SOFTWARE - (7500)

### ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE - (7502)

Segundo cuatrimestre de 2018

#### TRABAJO PRÁCTICO 9

#### Métricas II: Atributos Internos - Monitoreo

#### Tamaño - Líneas de código

1. Para el programa que se presenta, evaluar :
  - a. Líneas de Código
    - Cantidad de líneas físicas
    - Cantidad de líneas físicas excepto líneas en blanco
    - Cantidad de líneas físicas excepto líneas en blanco y comentarios
    - Cantidad de sentencias (excepto comentarios). Una sentencia puede abarcar más de una línea
    - Cantidad de sentencias ejecutables (sin considerar condiciones de excepción)
  - b. Densidad de comentarios:
    - Densidad de comentarios=Cantidad de líneas de comentario/Cantidad de líneas de código

---

```
with TEXT_IO; use TEXT_IO;
procedure Main is

    --This program copies characters from an input
    --file to an output file. Termination occurs
    --either when all characters are copied or
    --when a NULL character is input

    Nullchar, Eof: exception;
    Char: CHARACTER;
    Input_file, Output_file, Console: FILE_TYPE;

Begin
    loop
        Open (FILE => Input_file, MODE => IN_FILE,
              NAME => "CharsIn");
        Open (FILE => Output_file, MODE =>OUT_FILE,
              NAME => "CharOut");
        Get (Input_file, Char);
        if END_OF_FILE (Input_file) then
            raise Eof;
        elsif Char = ASCII.NUL then
            raise Nullchar;
        else
            Put(Output_file, Char);
        end if;
    end loop;
exception
    when Eof => Put (Console, "no null characters");
    when Nullchar => Put (Console, "null terminator");
end Main
```

---

## Medidas Morfológicas

2. Un *módulo* se puede definir como *una secuencia contigua de sentencias de programa, acotadas por elementos de contorno y que tienen un identificador agregado* (Yourdon y Constantine, 1979).

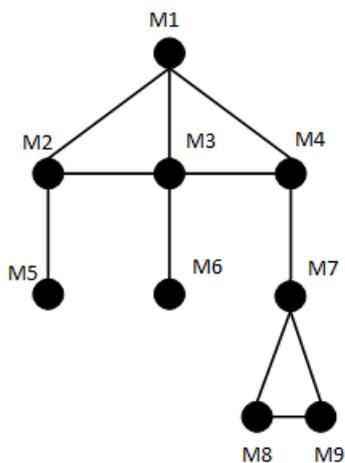
Para describir *aspectos inter-modulares* se construyen modelos que permiten representar las relaciones entre módulos, por ejemplo, grafos de dependencias entre módulos.

a. Para analizar la *Modularidad* existen distintos enfoques:

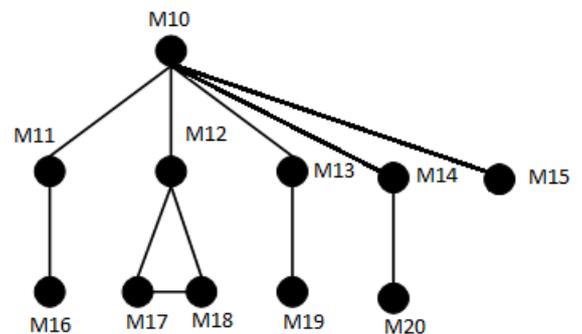
- Longitud promedio de los Módulos (Boehm)
- $M1 = \text{módulos} / \text{procedimientos}$  (Hausen)
- $M2 = \text{módulos} / \text{variables}$

b. Para analizar la "forma" de la estructura general del sistema (conformado por sus módulos) nos focalizamos en los aspectos morfológicos y medimos los atributos:

- **tamaño:** medido como cantidad de nodos, cantidad de arcos.
- **profundidad:** medido como longitud del camino más largo desde el nodo raíz a un nodo hoja.
- **ancho:** medido como el máximo número de nodos de cualquier nivel.
- **relación arco-nodo:** puede ser considerado como una medida de densidad de conectividad.



DISEÑO 1



DISEÑO 2

2.1 Evalúe los aspectos morfológicos de los diseños 1 y 2.

2.2 Evalúe longitud promedio de los módulos, M1 y M2 del diseño 1 en base a la información de la siguiente tabla:

Módulo	Longitud	Procedimientos	Variabes
M1	530	5	9
M2	760	4	10
M3	1200	8	20
M4	1378	2	18
M5	900	4	11
M6	600	4	4
M7	1150	8	7
M8	1400	7	16
M9	1550	15	12

### Reuso

3. Investigue :
  - a. ¿Qué información aporta medir el reuso de un sistema?
  - b. ¿En qué etapas/tareas del desarrollo del software es posible introducir el reuso?
  - c. Analice y describa cómo se relaciona el reuso con la medición de tamaño y productividad.
  - d. Diferencie reuso y reuso interno. Proponga y defina métricas para cada uno.

### Otras métricas

4. El Centro de Tecnología de Seguridad del Software de la NASA (NASA-SATC) propone entre otras métricas la que llama "Porcentaje de Comentarios" y la define como:  
el número de comentarios dividido por el total de líneas de código menos el número de líneas en blanco.  
Donde el número de comentarios está definido como el número de líneas de código que tienen comentarios: tanto los comentarios que comparten la línea con código como los comentarios que ocupan una línea en sí mismos.  
¿Qué significado tiene un número alto de esta métrica? ¿Qué características del código analizado refleja?
5. La métrica para proyectos orientados a objetos "Número de guiones de escenario", ¿qué idea nos puede dar sobre un proyecto/programa?

### Monitoreo

6. Para el ejercicio 7 del Trabajo Práctico 6, considere que inicialmente se habían definido los siguientes Hitos:

Hito 1	Fin Semana 5	Fin Diseño
Hito 2	Fin Semana 5	Fin Documentación del Producto
Hito 3	Fin Semana 8	Fin Implementación del SW
Hito 4	Fin Semana 10	Fin Implementación e Integración
Hito 5	Fin Semana 11	Fin Sistema ABC

Considerando la sucesión de eventos indicada y las acciones que usted propuso realizar se solicita:

- a. Graficar la Carta de tendencia de Hitos
- b. Graficar el costo de la planificación inicial vs. el costo real correspondiente a cada semana (Eje X: semanas; Eje Y: Costo)

## Referencias

- [1] T. DeMarco. *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1986.
- [2] Norman E. Fenton and Shari L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. 2 edition.