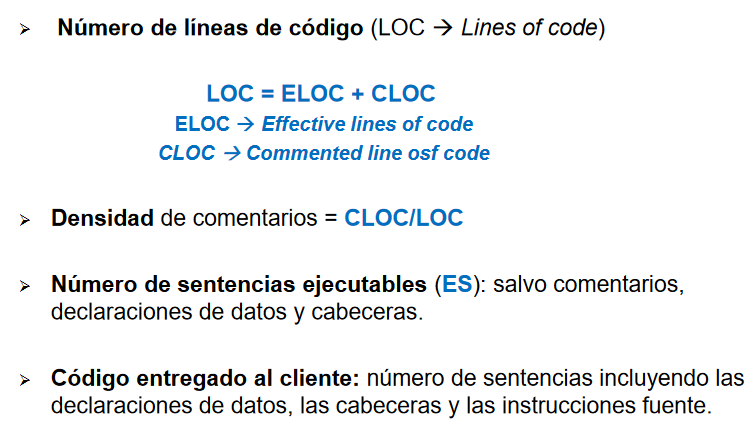
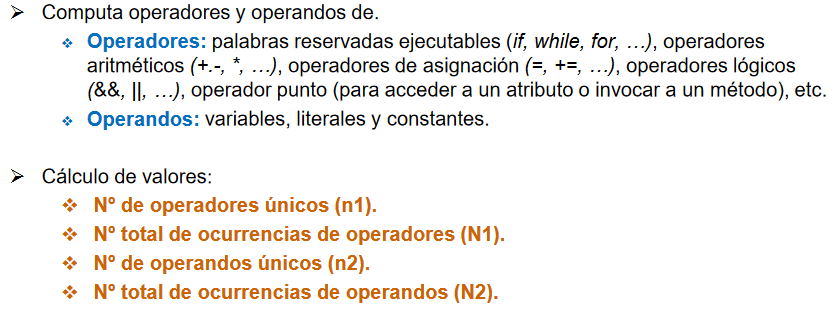
Resumen Tema 2 - Medición en el proceso software

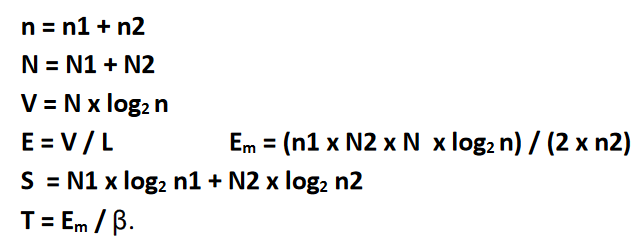
# Medición de código – Halstead



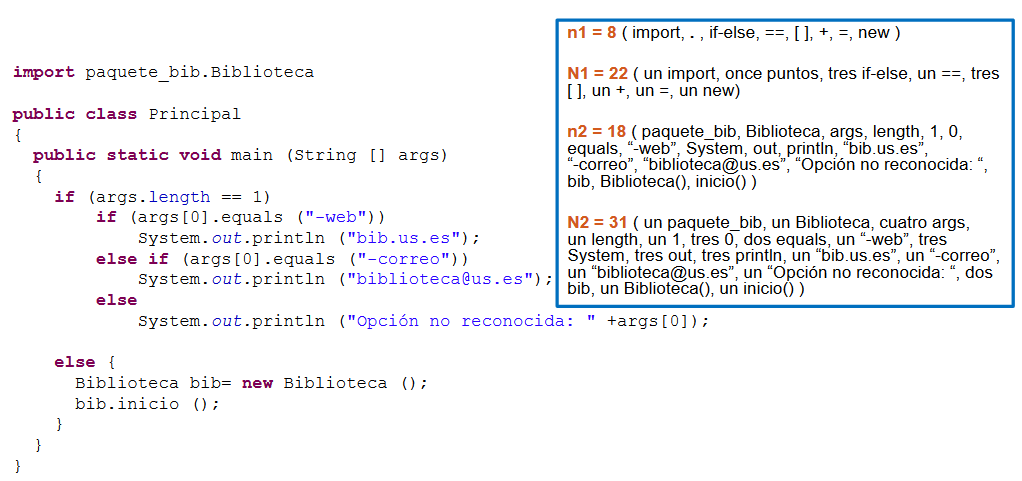
El número de líneas de código (LOC) es la medida más usada para medir la longitud del código fuente. Hay muchas propuestas para contarlas, pero la más extendida es la de HP, que no contabiliza las líneas comentadas ni en blanco. Este método se denomina con la abreviatura NCLOC o ELOC (*Effective lines of code*).

En el método anterior es útil medir por separado las líneas comentadas (CLOC) para calcular esfuerzo, productividad, etc. La longitud será: LOC = NCLOL + CLOC. A su vez, también se puede calcular la densidad de comentarios mediante CLOC/LOC.





## Ejercicio 1

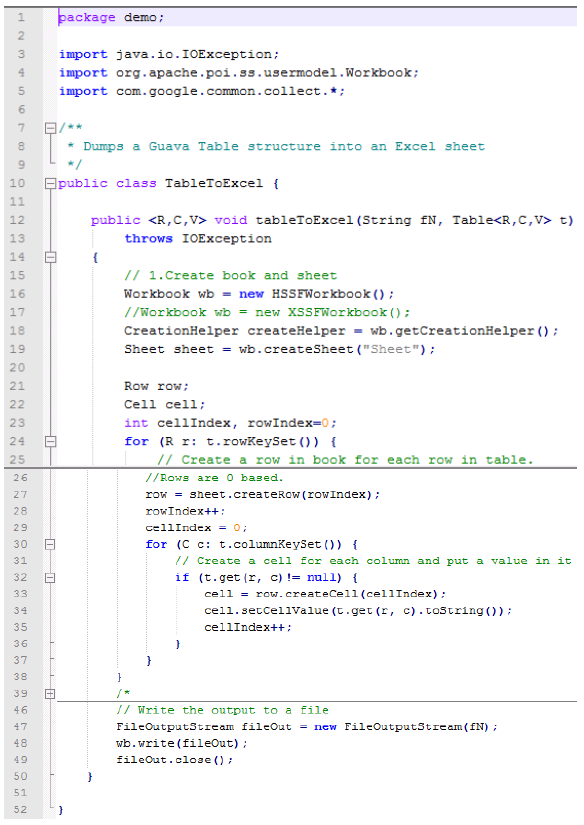


## Ejercicio 2

Dado el código que se adjunta, se ha de calcular la longitud del mismo, la densidad de

comentarios que presenta y el número de sentencias ejecutables. Las líneas en blanco se

ignoran.



1. NCLOC( líneas de código útiles) = 31

CLOC( líneas comentadas) = 16

LOC(longitud) = NCLOC + CLOC = 47

1. Densidad = CLOC/LOC = 0,34
2. Líneas de código = 52

Líneas comentadas = 16

Espacios en blanco = 5

Declaraciones de datos y variables = 9

Cabecera = 6

Nº de sentencias = 52-16-5-9-6 = 16

## Ejercicio 3

Dado el código del ejercicio anterior, se ha de calcular el esfuerzo desplegado para su

desarrollo, la longitud del mismo y el tiempo de desarrollo necesario según Halstead.

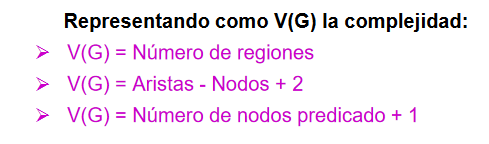
-n1 = import / \* / {} / throw / new / = / . / for / : / ++ / != / null / if = 13

-N1 = 3+1+5+1+2+7+222+2+2+2+1+1+1 = 50

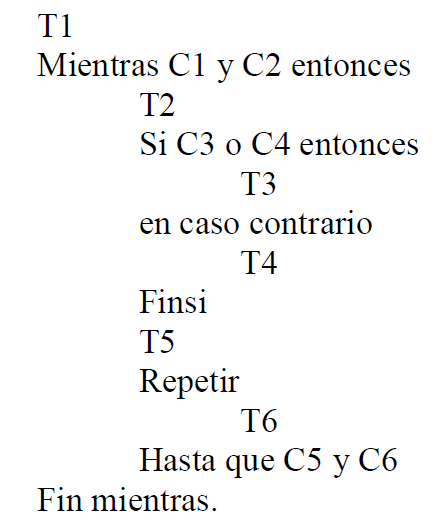
-n2 =

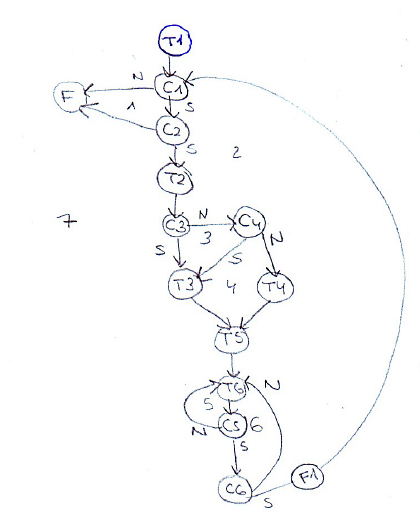
# McCabe





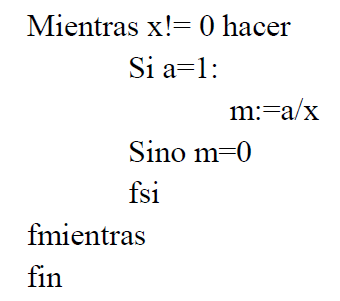
## Ejemplo 1

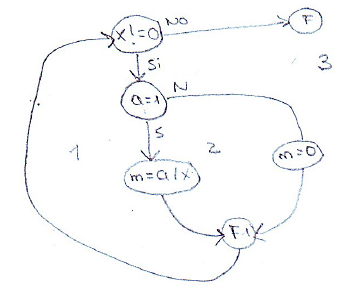




V(G) = 7

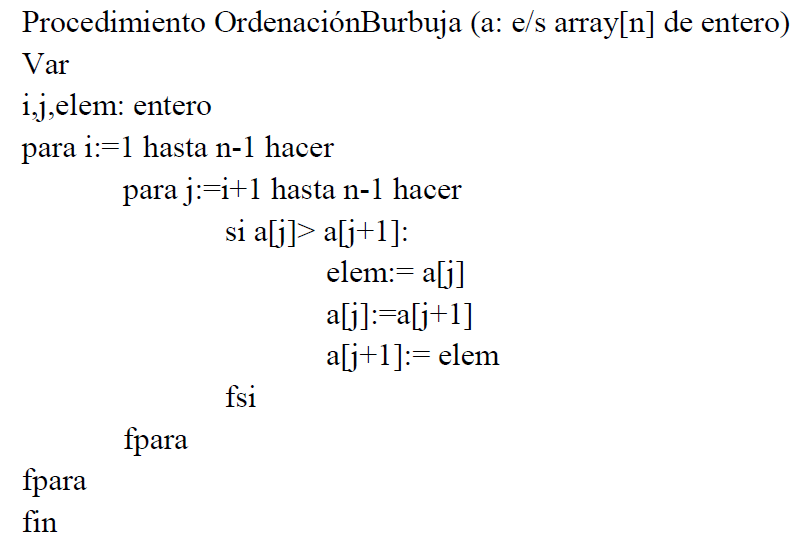
## Ejercicio 2

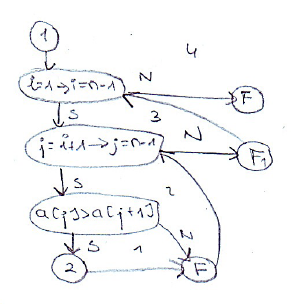




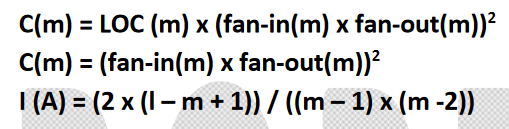
V(G) = 3

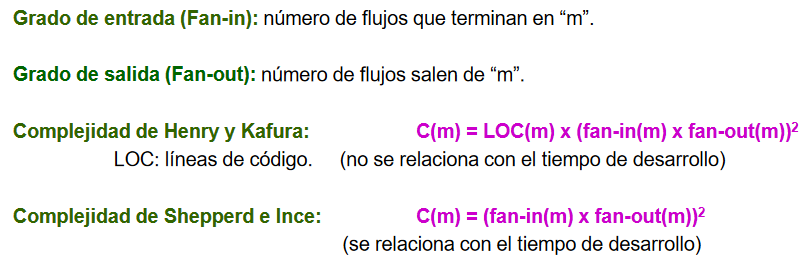
## Ejercicio 3

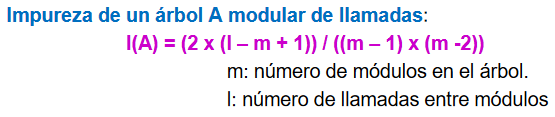




## Métricas de estructura del diseño

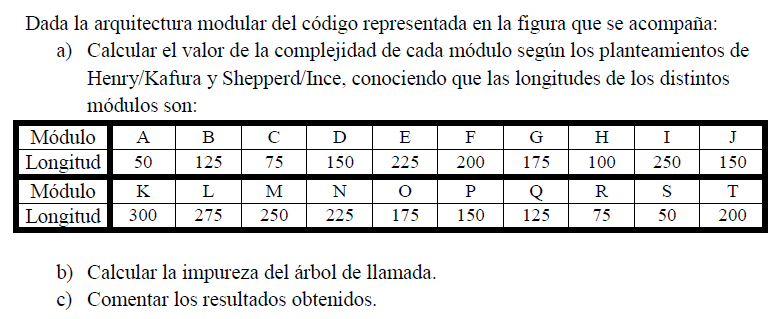


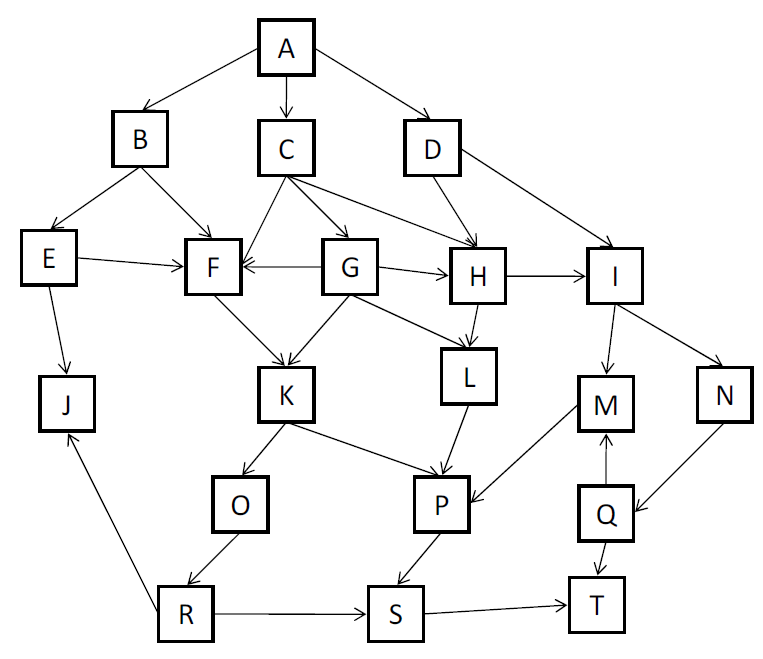


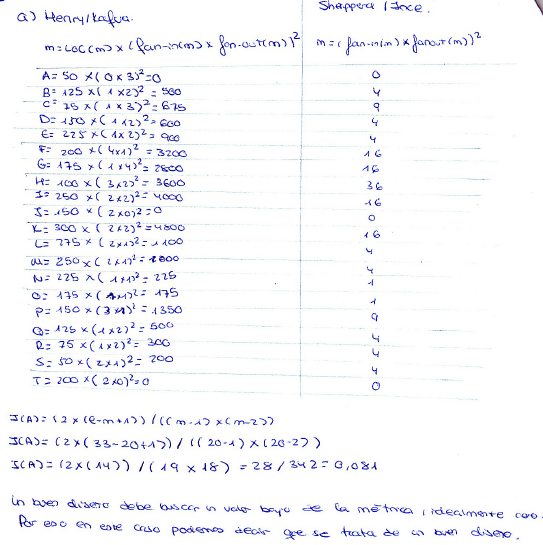


Cuanto más se acerque la impureza a 0 mejor

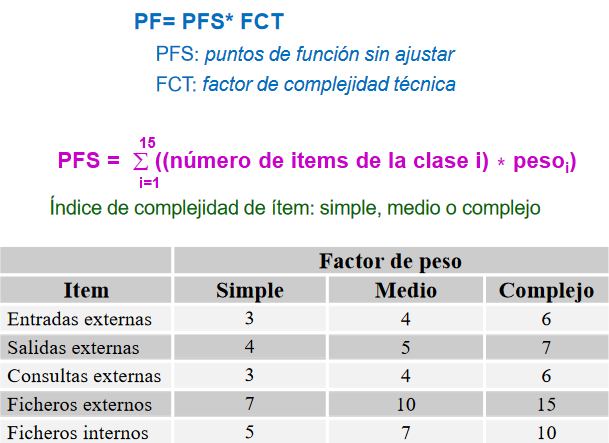
## Ejercicio 1

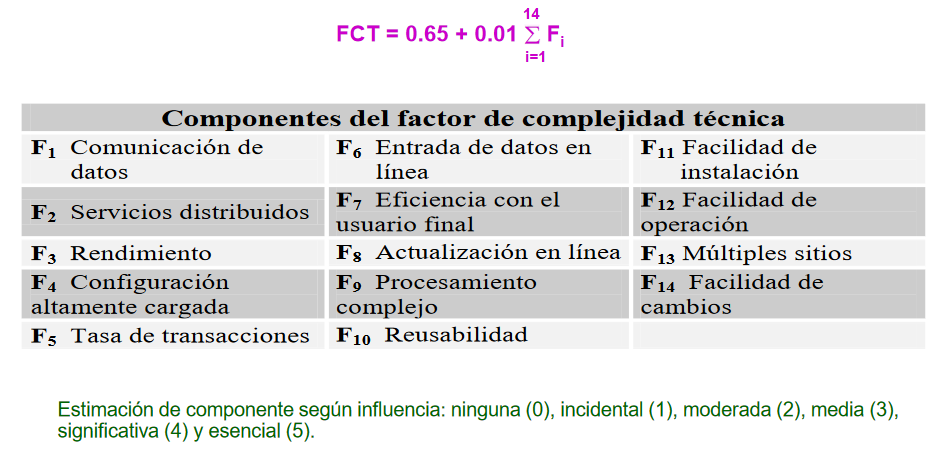


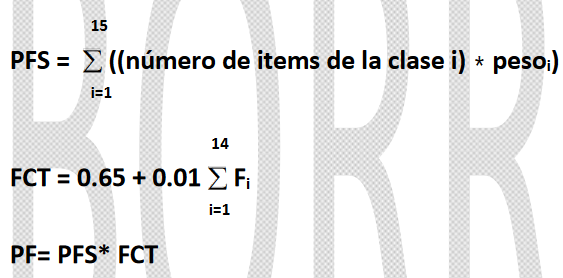




# Puntos de función (PF) (Albrecht)







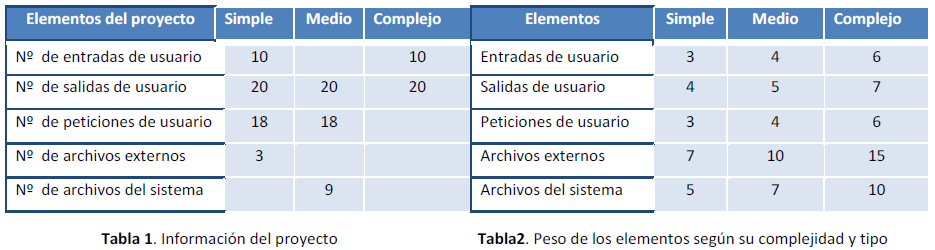
## Ejercicio 1

Se desea estimar el coste en euros de un proyecto, cuya información se muestra en la

tabla 1, utilizando el método de puntos de función. En la tabla 2 se recoge el peso de

cada tipo de elemento según la complejidad asignada, de acuerdo con la teoría de puntos

de función.



Se ha estimado que el proyecto presenta unas exigencias incidentes en relación a la

facilidad de operación; moderadas, respecto a la comunicación de datos, a la tasa de

transacciones y respecto a la diversidad de sitios en los que opera; medias, en lo que

afecta a la distribución de servicios, la entrada de datos en línea, la eficiencia con el usuario final y la complejidad de procesamiento; significativas, respecto al rendimiento

exigido, a la alta carga de la configuración y a la facilidad de cambios, así como

esencial en lo que atañe a la reusabilidad.

También se ha de considerar que el proyecto se implementará en JAVA y que como

valor medio, un punto funcional en JAVA corresponde a 63 líneas de código. De igual

forma se considerará que la productividad de los integrantes del equipo de desarrollo es

de 50 líneas de código a la hora, que el equipo de desarrollo está formado por tres

personas, que cada persona trabaja de forma efectiva cien horas de trabajo al mes y que

el sueldo bruto al mes de un técnico de la organización que desarrolla el proyecto es de

2.000 euros.

PFS = 10\*3+10\*6+20\*4+20\*5+20\*7+18\*3+18\*4+3\*7+9\*7 = 720

FCT = 0,65 + 0,01\*(1\*1+2\*3+3\*4+4\*3+5\*1) = 1,01

PF = PFS\*FCT = 727,2

1PF = 63 líneas

1 persona hace 50 líneas/hora, trabaja 100 horas al mes y cobra 2000€ al mes

Equipo está compuesto por 3 personas

Líneas en total del proyecto = 727,2\*63 = 45.813,6 líneas

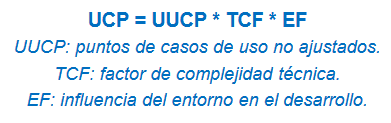
Si una persona hace 50 líneas en una hora, el equipo hace 150 líneas a la hora

Para terminar el proyecto necesito las siguientes horas = 45.813,6/150 = 305,424 horas

Cuantos meses dura el proyecto = 305,424/100 = 3,05424 meses

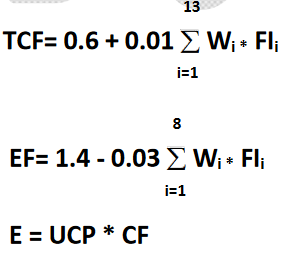
Y va a costar = 3,05424\*2000\*3 = 18.325,44€

# Puntos de caso de uso (UCP) (Karner)



E: Esfuerzo de desarrollo en horas-técnico

UUCP: suma ponderada del número de actores y del número de casos de uso de la especificación.

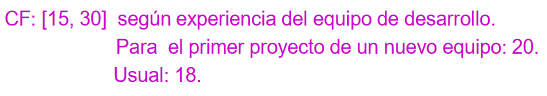


Peso para actores:

Simple = 1, Medio = 2, Complejo = 3

Peso para casos de uso:

Simple = 5, Medio = 10, Complejo=15



# Ejercicio 1

Calcule los *puntos de caso de uso* y el *esfuerzo de desarrollo* de un proyecto cuya

especificación de requisitos contiene 25 casos de uso de los que 10 son complejos, 16

son de complejidad media y 9 simples. En cuanto a los actores, 1 es complejo, 6 tienen

complejidad media y 2 son simples. Respecto a la *complejidad técnica del proyecto*

(Tabla1) las variables que tienen incidencia son: el rendimiento (incidencia = 4), la

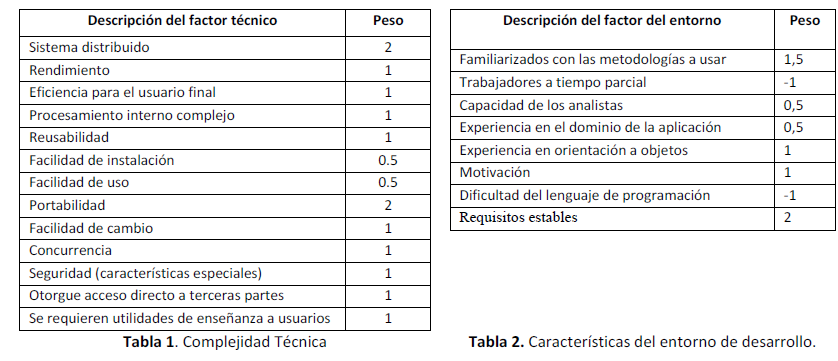
portabilidad (incidencia= 2) y la reusabilidad (incidencia 1). Para calcular TCF use la

fórmula TCF= 0,6 + 0,01 Wi \* FIi. Sobre las peculiaridades del *entorno de desarrollo*

(Tabla 2), se sabe que sí están familiarizados con las metodologías (impacto=4) y que

hay varios trabajadores a tiempo parcial (impacto= 2). Para calcular EF use la fórmula

# EF= 1.4 + (-0.03) Wi \* FIi..



Peso de casos de uso= 10\*15+16\*10+9\*5 = 355

Peso de actores = 1\*3+3\*2+2\*1 = 17

UUCP = 355+17 = 372

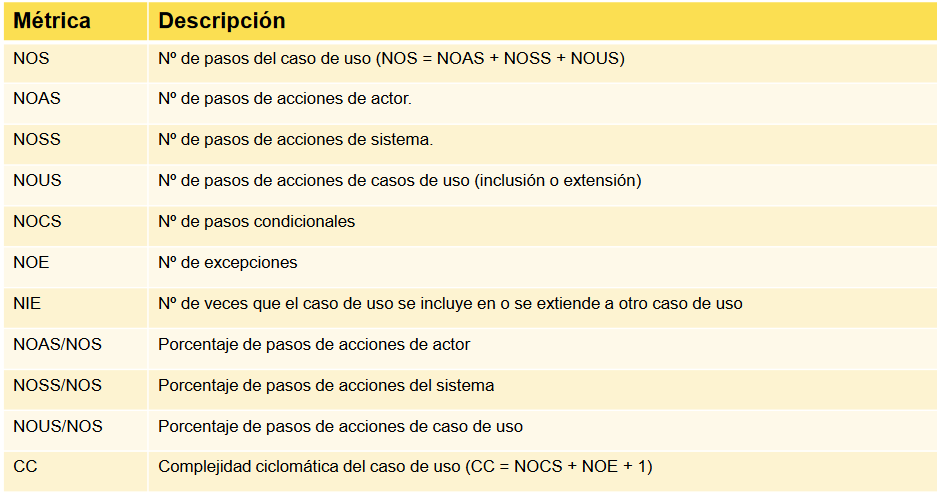
TCF = 0,6 + 0,01(1\*4+2\*2+1\*1) = 0,69

EF= 1,4 – 0,03(1,5\*4-2)= 1,28

UCP = 372\*0,69\*1,28 = 328,5504

E = UCP \* CF = UCP\*20= 6.571,008

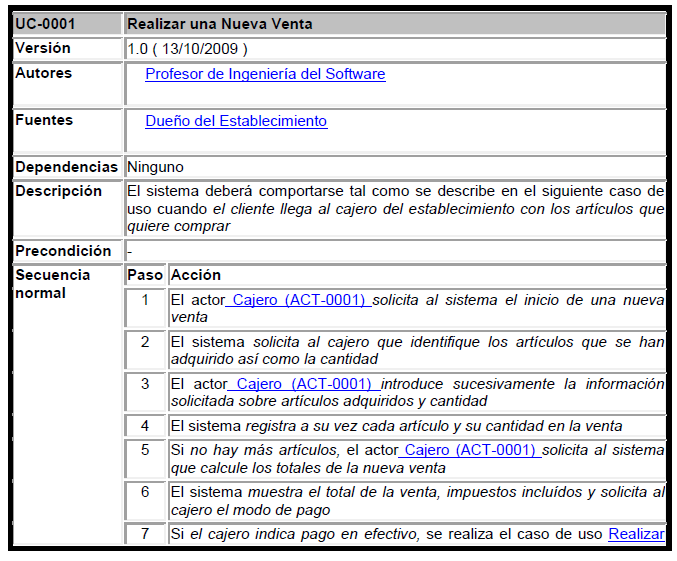
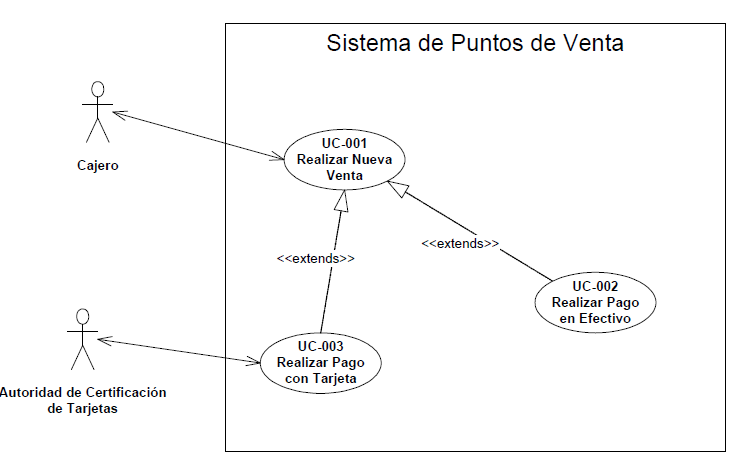
# Probabilidad de defectos de los casos de uso (Bernárdez y Durán)

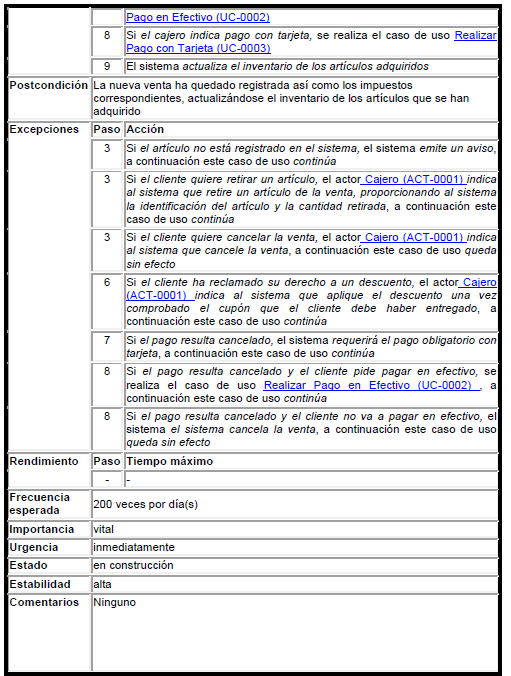


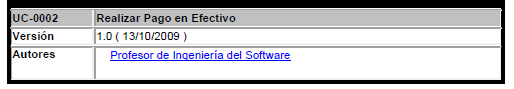
## Ejercicio 1

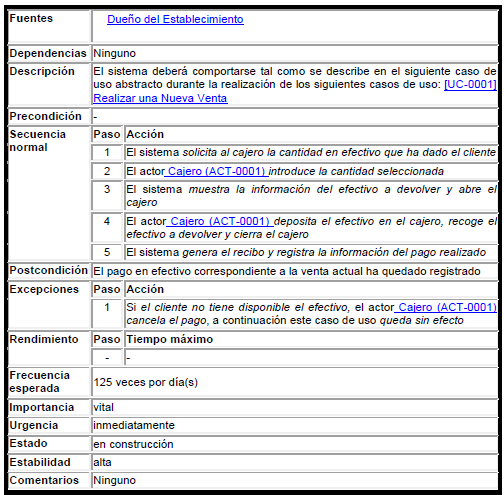
Se describe a continuación un sistema de gestión de los puntos de venta de una cadena

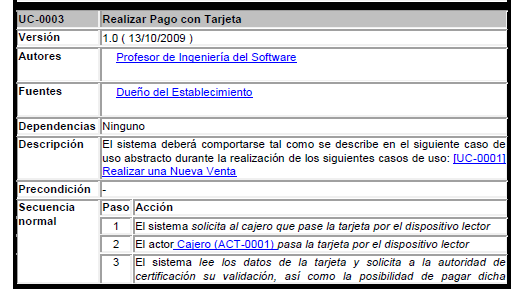
de establecimientos.

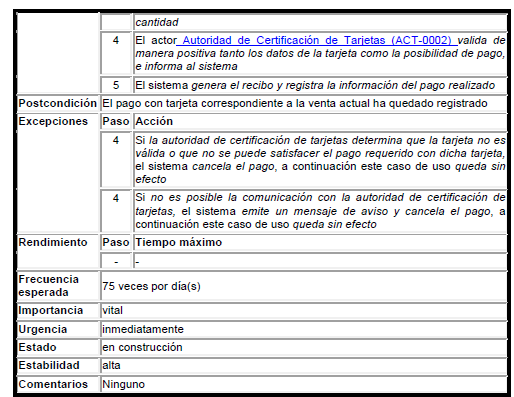










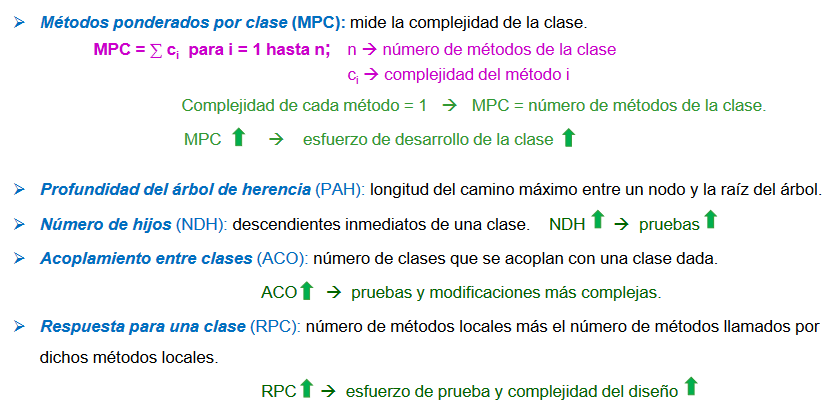


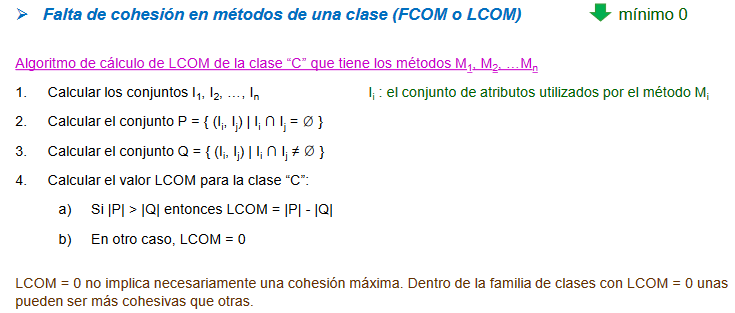
Se desea calcular los valores que adquieren las métricas de probabilidad de defectos de

los casos de uso, de Bernárdez y Durán, en el supuesto descrito anteriormente.

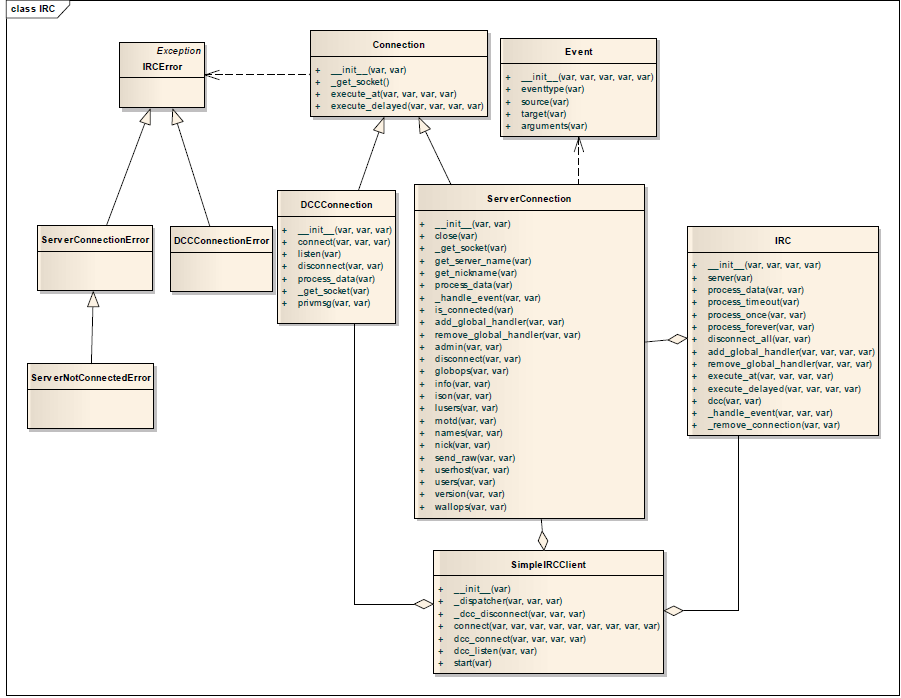
|  |  |  |  |
| --- | --- | --- | --- |
|  | UC-1 | UC-2 | UC-3 |
| NOS | 9 | 5 | 5 |
| NOAS | 3 | 2 | 2 |
| NOSS | 4 | 3 | 3 |
| NOUS | 2 | 0 | 0 |
| NOCS | 3 | 0 | 0 |
| NOE | 7 | 1 | 2 |
| NIE | 0 | 1 | 1 |
| NOAS/NOS | 0,33 | 0,4 | 0,4 |
| NOSS/NOS | 0,44 | 0,6 | 0,6 |
| NOUS/NOS | 0,22 | 0 | 0 |
| CC | 11 | 2 | 3 |

# CK (Chidamber/Kemerer)





## Ejercicio 1



NDH = 5

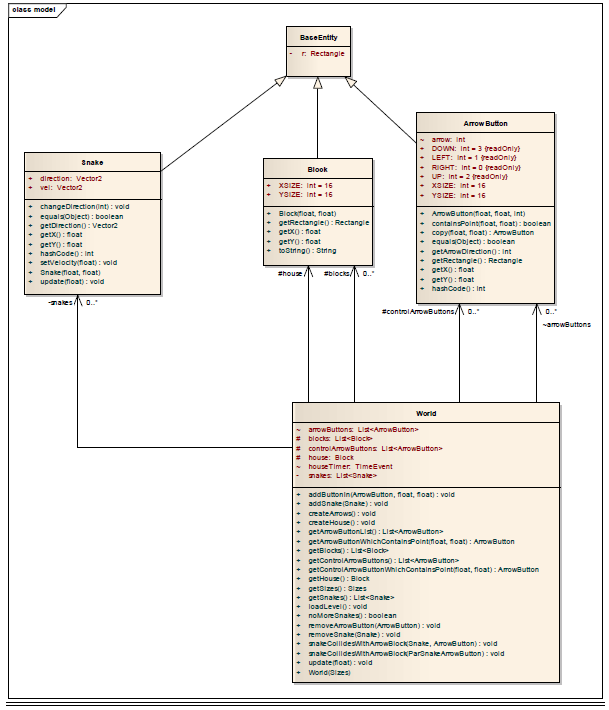
ServerConenectioError y DCCConecctionError son clases herencia de IRCError, ServerNotConnectionError es hija de ServerConnectionError y DCCConnection y ServerConnection son hijos de Connection.

PAH =3

La longitud de herencia más larga es la entre IRCError y ServerNotConnectionError

ACO =6

A IRC se acopla ServerConnection, a SimpleIRCClient se le acoplan tres clases, a Event una y a la clase error otra.

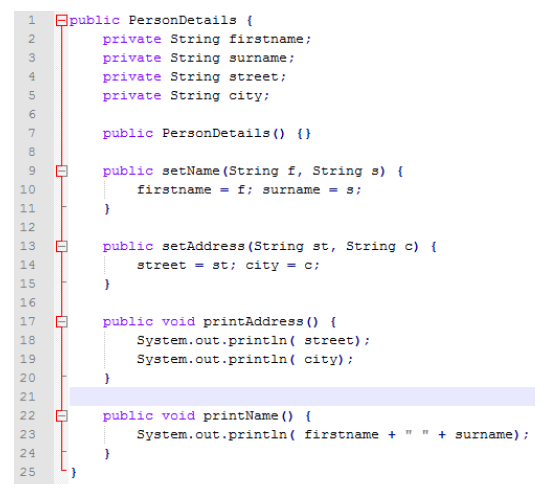


NDH = 3

PAH = 2

ACO = 5

## Ejercicio 2



|  |  |  |
| --- | --- | --- |
| Conjunto | Método | Atributos |
| I1 | PersonDetails | {} |
| I2 | setName | {firstname,surname} |
| I3 | setAdresse | {Street,city} |
| I4 | printAdresse | {Street,city} |
| I5 | printName | {firstname,surname} |

* Conjunto P (sin concordancias):

I1I2,I1I3,I1I4,I1I5,I2I3,I2I4,I3I5 y I4I5

P=8

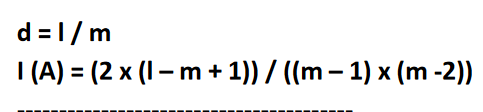
* Conjunto Q:

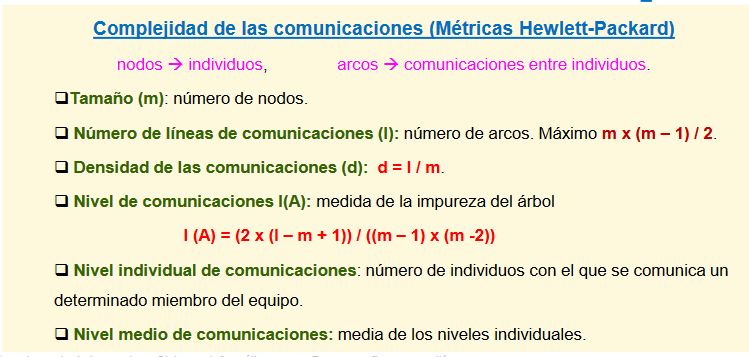
I2I5 y I3I4

LCOM = P-Q = 6

b) RPC = 5 métodos + 3 syso (métodos llamados por los anteriores) = 8

# Medición de atributos de los recursos





## Ejercicio 1

Una organización dedicada a la producción de software se estructura de manera que,

dependiendo del director, existe un director técnico, un director de logística , un

coordinador de calidad y un director de recursos humanos, los cuales participan de

forma colaborativa entre sí en el día a día de la organización.

El director técnico tendrá a su cargo a los directores de los diferentes proyectos que

desarrolla la organización, aunque circunstancialmente la organización solamente se

encuentra desarrollando un proyecto, pero de una importancia muy significativa para la

misma.

Cada proyecto cuenta con un responsable de desarrollo y un responsable de pruebas, y

se conoce que para el proyecto en cuestión el equipo de trabajo está constituido por dos

ingenieros de requisitos, un arquitecto software y dos técnicos implementadores. A su

vez, el equipo de pruebas está integrado por dos técnicos cualificados.

En el funcionamiento diario de la organización y como consecuencia de la organización

del trabajo en la misma, se sabe que los directores de proyecto deben resolver las

necesidades de recursos que se le planteen con el director de logística de la empresa, así

como deberá hacerlo con el director de recursos humanos para las necesidades que se le

planteen en relación con el personal asignado al proyecto. Por su parte, los aspectos

relacionados con el aseguramiento de la calidad en relación al proyecto deberá tratarlos

con el coordinador de calidad.

Los ingenieros de requisitos interactuarán para llevar a cabo su labor y asistirán

indistintamente a los requerimientos que les puedan plantear los técnicos

implementadores, pero éstos siempre plantearán sus requerimientos a los ingenieros de

requisitos a través del arquitecto de software. A su vez, los técnicos implementadores

también desarrollan su labor trabajando de forma colaborativa.

El arquitecto software además de estar al tanto de la labor realizada por todos y cada

uno de los demás miembros del equipo de trabajo, debe coordinarse con el responsable

de pruebas para realizar las pruebas de los productos obtenidos. Para la realización de

estas pruebas el responsable de pruebas debe contar también con las aportaciones del

responsable de desarrollo.

Teniendo en cuenta la descripción anterior y considerando que las relaciones de

dependencias descritas implican también la necesidad de colaboración entre los agentes

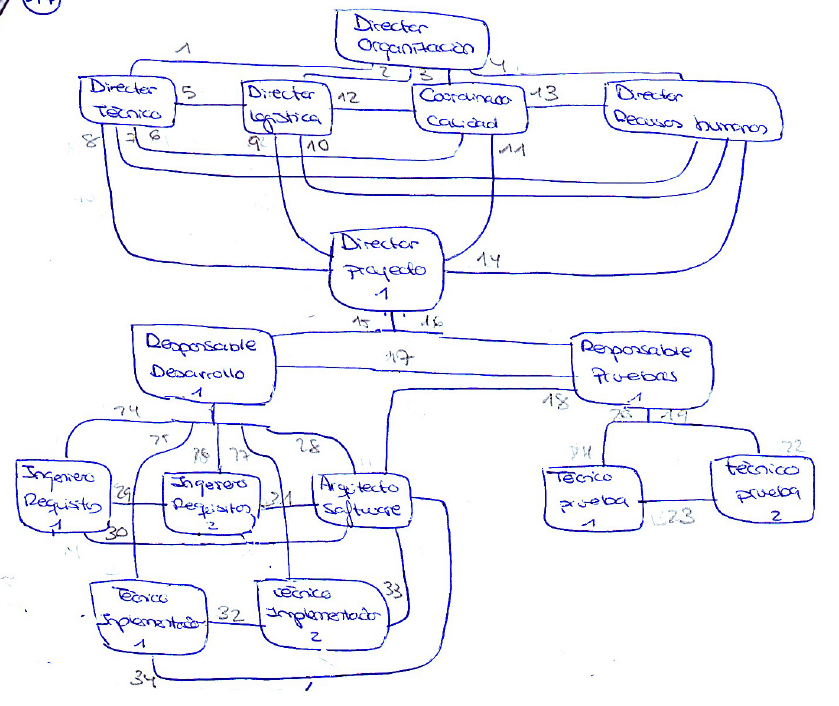
implicados:

1. Aplicar las métricas de Hewlett-Packard para medir la complejidad de las

comunicaciones.

2. Analizar la proximidad o no, del grafo de comunicaciones que presenta la

organización, a una estructura de árbol.



Número de nodos = 15

Número de líneas = 32

Densidad 32/15 = 2,13

I(A) = (2\*(32-15+1))/((15-1)\*(15-2)) = 0,17