



UNIVERSIDAD  
Popular del cesar

# Ingeniería de Sistemas



# **INGENIERIA DE SOFTWARE II**

## **UNIDAD No. 3**

### **Estimación del software**

# Estimación del software



# ESTIMACION DEL SOFTWARE

## Introducción

El propósito de una estimación software es determinar si los objetivos, los tiempos de que se disponen para realizar el proyecto, son suficientemente realistas. Hacer una buena estimación software antes de ofertar un proyecto ayuda a detectar proyectos que conviene abordar y que son rentables.

Las estimaciones condicionan la planificación y guían la toma de decisiones a lo largo de todo el proyecto, por lo que su fiabilidad tiene una gran repercusión tanto en la duración y costo del proyecto como en la calidad del software producido.

Las variables susceptibles de estimación son muchas. El tamaño, esfuerzo o costo del proyecto son las más habituales, aunque la estimación de características de calidad del software está adquiriendo gran importancia.



# ESTIMACION DEL SOFTWARE

La gestión de todo proyecto de software comienza con la planificación de proyecto y sus actividades. Antes de que se empiece con el proyecto, el gestor y su equipo debe de hacer una estimación del proyecto, es decir, el trabajo, el esfuerzo, los recursos hardware y software que se necesitaran, el costo y el tiempo necesario para culminar el proyecto.

En la planificación del proyecto se determinara tareas y tiempo que se deben cumplir, así como también, los responsables de que se cumplan. La estimación del proyecto determinara casi con actitud el verdadero costo y el esfuerzo persona mes que se necesita de un proyecto.

La estimación de lo que costará el desarrollo de un software es una de las actividades de planeación que reviste especial importancia, ya que una de las características que debe tener un producto de software es que su costo sea adecuado, de lo contrario el proyecto puede fracasar.



# ESTIMACION DEL SOFTWARE

**Cuándo trabajar con estimaciones?**

**Etapas preliminares:**

- Para cotizar un contrato
- Para realizar estudios de factibilidad

**Durante el proyecto:**

- Un patrón contra el cual medir, ajustar el desempeño, y anticipar riesgos

**Al final del proyecto:**

- Extrapolar resultados a otros proyectos

# ESTIMACION DEL SOFTWARE

## Definición

- La estimación se define como el proceso que proporciona un valor a un conjunto de variables para la realización de un trabajo, dentro de un rango aceptable de tolerancia.
- Se define también como la predicción de **personal, del esfuerzo, de los costos y de la planificación** que se requerirá para realizar todas las actividades y construir todos los productos asociados con el proyecto.

# ESTIMACION DEL SOFTWARE

- En el mundo del desarrollo de software, Larry Putnam ha apuntado que la gestión del desarrollo de software considera la estimación como una actividad que permite obtener, principalmente, respuestas aproximadas a las siguientes preguntas
- ¿Cuánto costará ?
- ¿Cuánto tiempo llevará hacerlo?

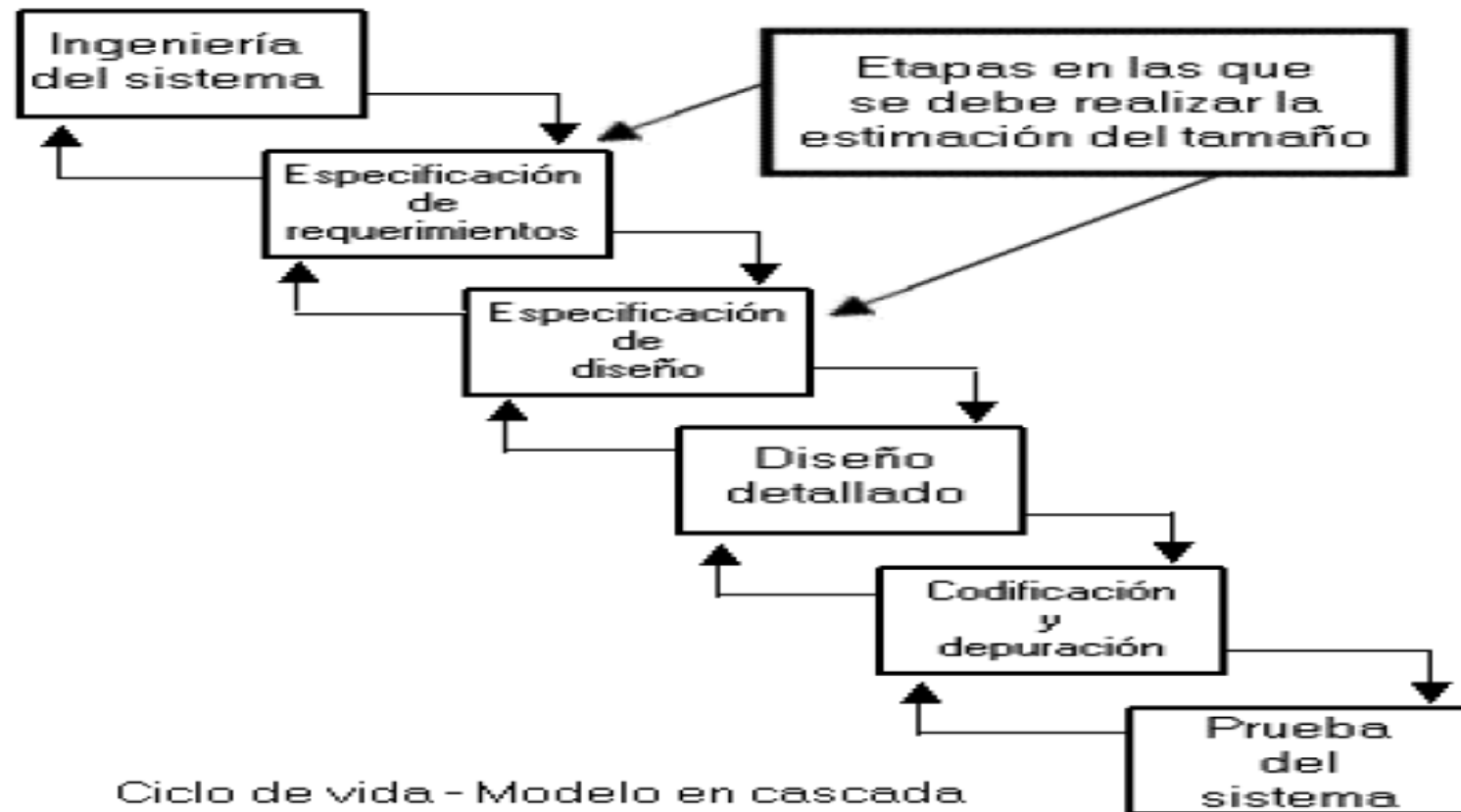


# ESTIMACION DEL SOFTWARE

- **Estimación de Tamaño:** ¿Cuanto (y cuan complejo) tengo que construir?
- **Estimación de Esfuerzo:** ¿Que tiempo me lleva construir lo estimado? ¿Cuantos trabajadores necesito?
- **Calculo de Costos :** Identificación de terceros recursos, calculo de costos de recursos y elaboración del presupuesto

# ESTIMACION DEL SOFTWARE

- Estimación de Tamaño



## ESTIMACION DEL SOFTWARE - TAMAÑO

- En varias disciplinas es usual la planeación basada en el tamaño de los proyectos.
- El tamaño del programa suele tener un buen grado de correlación con el esfuerzo necesario para implantarlo.
- Es más fácil estimar el tamaño de un programa que el tiempo de implantación.
- Para efectos de la planeación, normalmente se estima primero el tamaño y, con base en él, el tiempo.

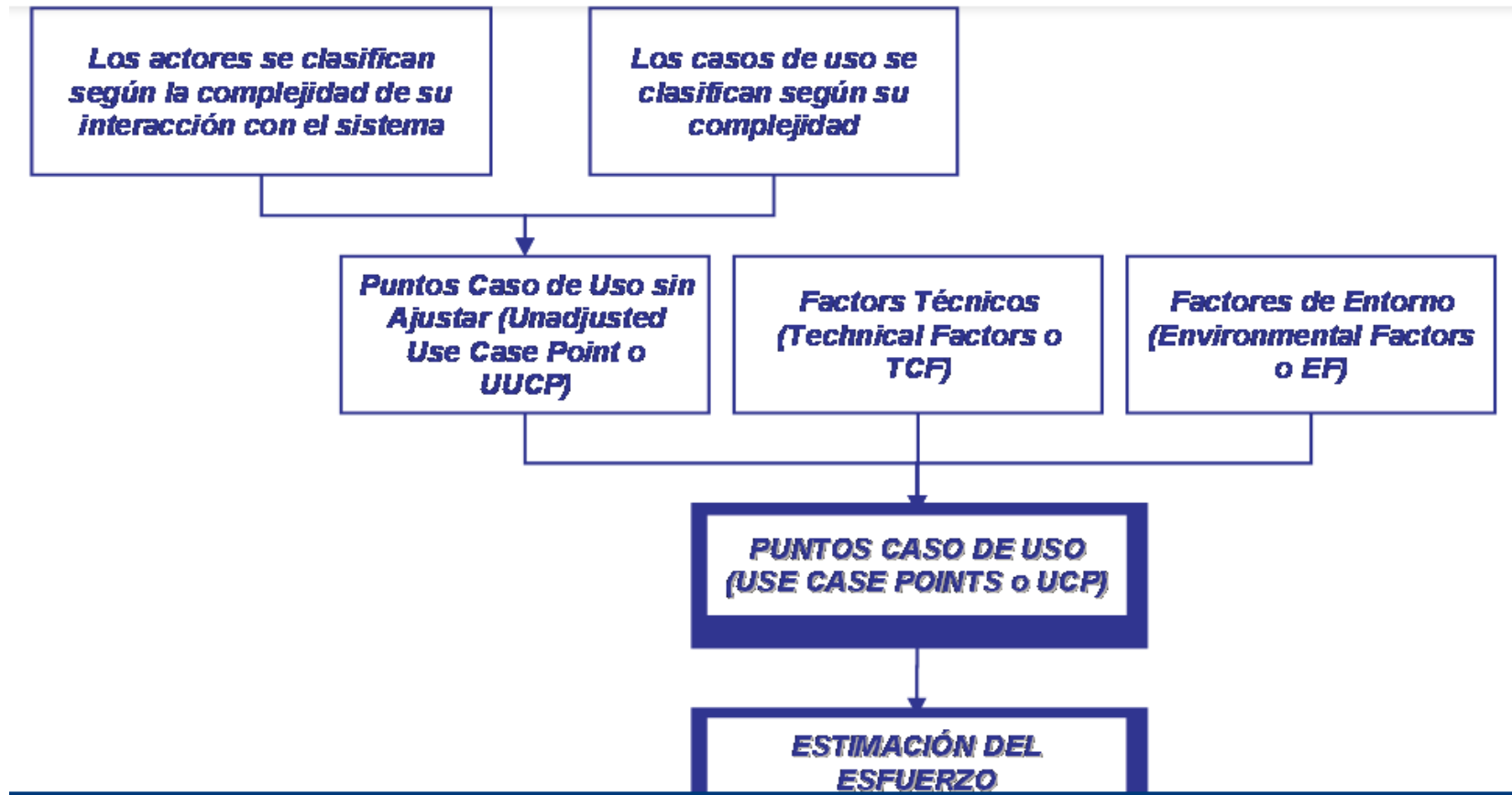
# ESTIMACION DEL SOFTWARE - Tamaño

- Medir un producto en función de la cantidad de algo físico de un artefacto software (ej. líneas de código, numero de atributos de una tabla).
- El tamaño se puede medir en líneas de código (LDC)
- El tamaño como puntos de función (PF).
- El tamaño por puntos de casos de usos
- El tamaño por puntos de objetos



# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos



# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### 1. Puntos de casos de usos - UCP

$$UCP = UUCP * TCF * ECF * PF$$

Donde:

- UUCP = Puntos de Caso de Uso sin ajustar
- TCF = Factor de complejidad técnica
- ECF = Factor de Complejidad del Medio Ambiente
- PF = Factor de productividad

# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

Cálculo de UUCP

$$UUCP = UAW + UUCW$$

### **Pesos de los Casos de Uso sin Ajustar (UUCW)**

- Basado en el número total de actividades (o pasos) contenidos en todos los escenarios del caso de uso.

### **Pesos de los Actores sin Ajustar (UAW)**

- Basado en la combinación de la complejidad de todos los actores en todos los casos de uso.

# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### UUCW (Pesos de los Casos de Uso sin Ajustar)

- Hay tres categorías de Casos de Uso:

Categoría de Caso de Uso	Descripción	Peso (factor)
Simple	Transacciones= 3 ó menos Clases= Menos de 5	5
Medio	Transacciones= 4 a 7 Clases= 5 a 10	10
Complejo	Transacciones= Más de 7 transacciones Clases= Más de 10 clases	15

$$UUCW = \sum (\text{Cantidad de un Tipo de Caso Uso} * \text{Factor})$$



# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### UAW (Pesos de los Actores sin Ajustar)

- Consiste en la evaluación de la complejidad de los actores con los que tendrá que interactuar el sistema.

Tipo de Actor	Descripción	Peso (factor)
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1
Medio	Otro sistema interactuando mediante un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3

$$UAW = \sum (\text{cantidad de un Tipo de Actor} * \text{Factor})$$

$$UUCP = UAW + UUCW$$

# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### 2. TCF (Factor de Complejidad Técnica)

- Compuesto por **13 puntos** que evalúan la complejidad de los módulos del sistema que se desarrolla.
- **Cada factor tiene:**
  - Un peso definido
  - Un factor de complejidad percibido subjetivamente y determinado por la percepción del equipo de desarrollo.

### 2. TCF (Factor de Complejidad Técnica)

Factor técnico	Descripción	Peso dado
T1	Sistema distribuido	2
T2	Rendimiento o tiempo de respuesta	1
T3	Eficiencia del usuario final	1
T4	Procesamiento interno complejo	1
T5	El código debe ser reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidad de uso	0.5
T8	Portabilidad	2
T9	Facilidad de cambio	1
T10	Concurrencia	1
T11	Características especiales de seguridad	1
T12	Provee acceso directo a terceras partes	1
T13	Se requiere facilidades especiales de entrenamiento a usuario	1

# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

- Cada uno de estos factores se deben evaluar según la siguiente escala:

Descripción	Valor
Irrelevante	0 a 2
Medio	3 a 4
Esencial	5

**Fórmula para calcular el TCF:**  
 $TCF = 0.6 + (0.01 * \text{Factor Total Técnico})$

### 2. TCF (Factor de Complejidad Técnica)

Factor técnico	Descripción	Peso	Impacto percibido	Factor calculado
T1	Sistema distribuido	2	1	2
T2	Rendimiento o tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	0	0
T6	Facilidad de instalación	0.5	0	0
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	0	0
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	0	0
T11	Características especiales de seguridad	1	0	0
T12	Provee acceso directo a terceras partes	1	3	3
T13	Se requiere facilidades especiales de entrenamiento a usuario	1	0	0
Factor Total Técnico				19.5

# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### 3. ECF (Factor de Complejidad Ambiental)

- Establece la experiencia del equipo de desarrollo.
- Los **factores** sobre los cuáles se realiza la evaluación están relacionados con las habilidades y experiencia del grupo de personas involucradas con el desarrollo del proyecto.
- Cada uno de estos factores se debe calificar con un valor de 0 a 5.  
Un valor de 1 significa que el factor tiene un fuerte impacto negativo para el proyecto, 3 es medio y 5 significa que tiene un fuerte impacto positivo.



# ESTIMACION DEL SOFTWARE - Tamaño

## Puntos de casos de usos

### 3. ECF (Factor de Complejidad Ambiental)

Factor Ambiental	Descripción	Peso
E1	Familiaridad con el modelo de proyecto utilizado Familiaridad con UML	1.5
E2	Personal tiempo parcial	-1
E3	Capacidad del analista líder	0.5
E4	Experiencia en la aplicación	0.5
E5	Experiencia en orientación a objetos	1
E6	Motivación	1
E7	Dificultad del lenguaje de programación	-1
E8	Estabilidad de los requerimientos	2

Factor Ambiental	Descripción	Peso	Impacto percibido	Factor calculado
E1	Familiaridad con el modelo de proyecto utilizado Familiaridad con UML	1.5	5	7.5
E2	Personal tiempo parcial	-1	0	0
E3	Capacidad del analista líder	0.5	5	2.5
E4	Experiencia en la aplicación	0.5	0	0
E5	Experiencia en orientación a objetos	1	5	5
E6	Motivación	1	5	5
E7	Dificultad del lenguaje de programación	-1	0	0
E8	Estabilidad de los requerimientos	2	3	6
11/04/2011 Factor Ambiental Total				26

**Fórmula para calcular ECF:**

$$\text{ECF} = 1.4 + (-0.03 * \text{Factor Ambiental Total})$$

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

- PROBE sirve de apoyo para la estimación basada en PROxy.
- PROBE usa proxies para estimar el tamaño del programa y el tiempo de desarrollo.
- Un buen proxy ayudará a realizar estimaciones precisas.

# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

Ejemplo Proxies Producto

Clases, funciones, y procedimientos Elementos del producto

- Elementos de la base de datos
- Pantallas, reportes, scripts, archivos
- Capítulos del libro

# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

La forma de estimar con este método es:

- Descomponer el producto a construir en distintas piezas.
- Hacer una relación de estas piezas con otras desarrolladas anteriormente.
- Utilizar el tamaño de las piezas desarrolladas anteriormente, para estimar el tamaño de nuevas piezas.



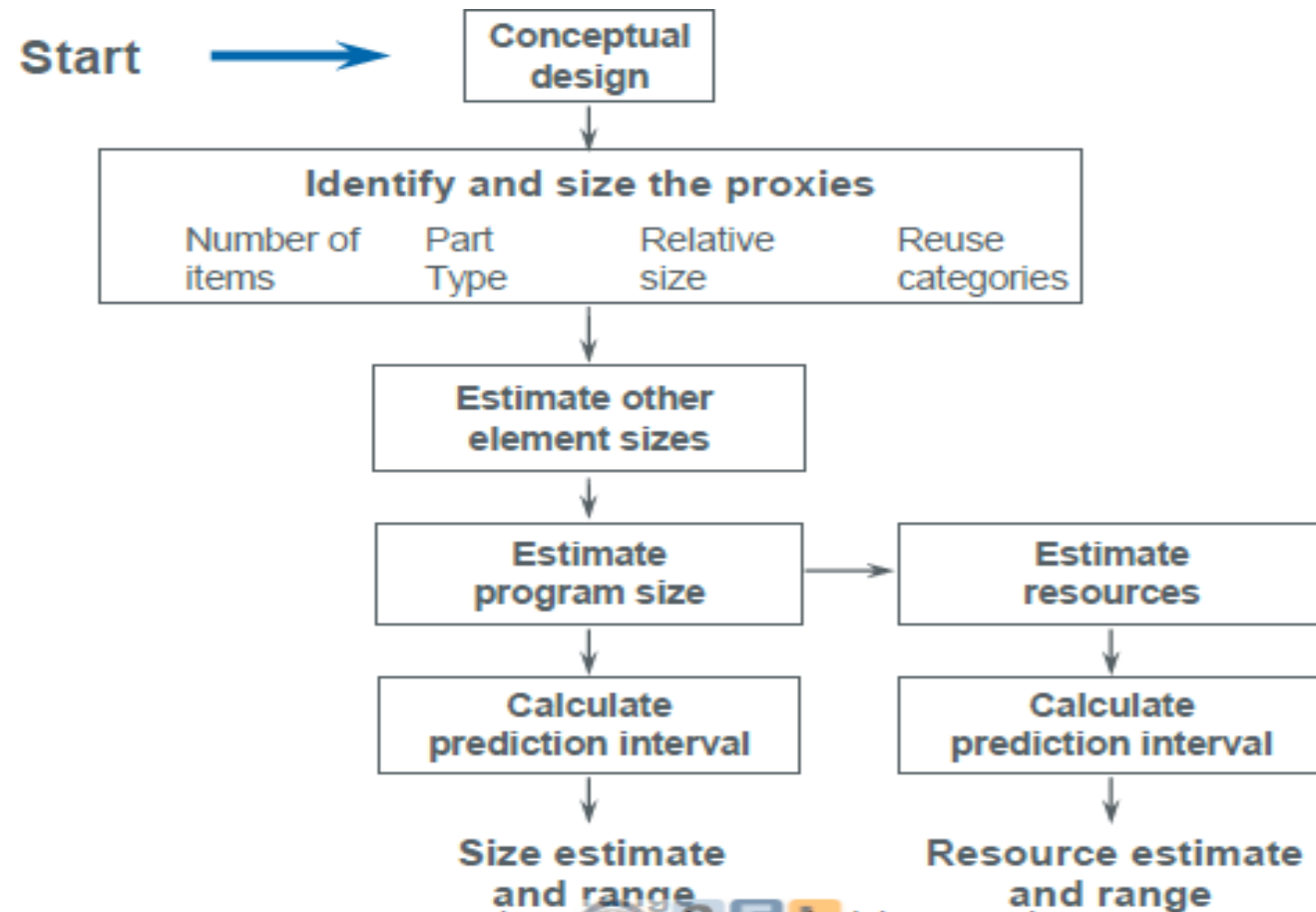
# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

Se divide en varias fases:

- Diseño conceptual
- Clasificación de los objetos
- Cálculo de LOC modificadas y agregadas
- Estimación del tamaño del programa
- Cálculo del intervalo de predicción

# ESTIMACION DEL SOFTWARE - Tamaño



# ESTIMACION DEL SOFTWARE - Tamaño

## Diseño Conceptual

El primer paso de estimación es hacer un diseño conceptual.

- Relacionar los requisitos para el producto
- Definir los elementos del producto que producirán las funciones deseadas
- Estimar el tamaño de lo que se planea construir
- Busca identificar los objetos que tendrían que conformar la aplicación
- El criterio es: *¿Cuáles objetos harían falta para poder construir la aplicación?*

# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

Se cuenta con la siguiente información

- clase A, tres ítems (o métodos), 39 total LOC
- clase B, cinco ítems, 127 total LOC
- clase C, dos ítems, 64 total LOC
- clase D, tres ítems, 28 total LOC
- clase E, un ítem, 12 LOC
- clase F, dos ítems, 21 total LOC

El LOC por ítem es 13, 25.4, 32, 9.333, 12, 10.5.

# ESTIMACION DEL SOFTWARE - Tamaño

## Método PROBE

### Clasificación de los objetos

- Los objetos identificados en el diseño conceptual deben clasificarse
- La clasificación se hace según dos conceptos:
- Tipo
- Tamaño relativo de los métodos (identificado con base en información histórica)

# ESTIMACION DEL SOFTWARE - Tamaño

## Ejemplo Rangos de tamaño de clase en C++

Tipo	LOC por item				
	VS	S	M	L	VL
Cálculo	2.34	5.13	11.25	24.66	54.04
Datos	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Lógica	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación del esfuerzo

Las tareas a realizar en un proyecto de software se miden en términos de **esfuerzo requerido en horas/hombre**, por lo que poseen un **costo económico** y forman parte del presupuesto.

En el vínculo contractual que une al cliente con la empresa encargada de desarrollar el software, se acuerdan tres variables importantes:

- Alcance del producto a desarrollar,
- Precio,
- Tiempo incurrido para el desarrollo.

La **estimación del esfuerzo** de cada tarea debe ser lo más precisa posible, ya que de esto depende, en gran parte, el éxito del proyecto. La entrega del producto en el tiempo estipulado, el uso eficiente del presupuesto asignado y el cumplimiento de las funcionalidades requeridas, inciden en la calidad del producto y muestran la eficacia del proceso de desarrollo.

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación del esfuerzo

El desarrollo de software está sujeto a restricciones técnicas y económicas, las cuales se expresan en el plan de proyecto en forma explícita o implícita.

El plan de proyecto se compone de actividades y tareas, definidas en función de los requerimientos definidos. Estas actividades y tareas tienen un esfuerzo requerido para su realización

La estimación de esfuerzo se mide en términos del esfuerzo requerido por persona/mes, normalmente expresado en términos de Horas/Hombre (HH).

El esfuerzo es la magnitud de costo de elaboración de un proyecto, y se expresa mediante unidades tales como personas-mes o personas-año

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Técnicas de Estimación de Esfuerzos

1. Juicio Experto
2. Estimación por Analogía
3. Modelos Paramétricos o Algorítmicos
4. Ajuste del Producto al Precio.
5. Otras técnicas

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación por Juicio Experto

1. Diversos expertos elaboran estimaciones individuales.
2. Estas estimaciones se comparan y analizan.
3. Se repite el proceso hasta alcanzar un resultado satisfactorio.

## Ventajas e Inconvenientes

- (+) Los expertos valoran factores subjetivos (ej. estado anímico del equipo) que no se computan mediante otros métodos.
- (-) Precisa de expertos (personas), difícilmente reemplazables.
- (-) No reproducible, resultados dispares en diferentes aplicaciones.

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación por Juicio Experto: Técnica Delphi

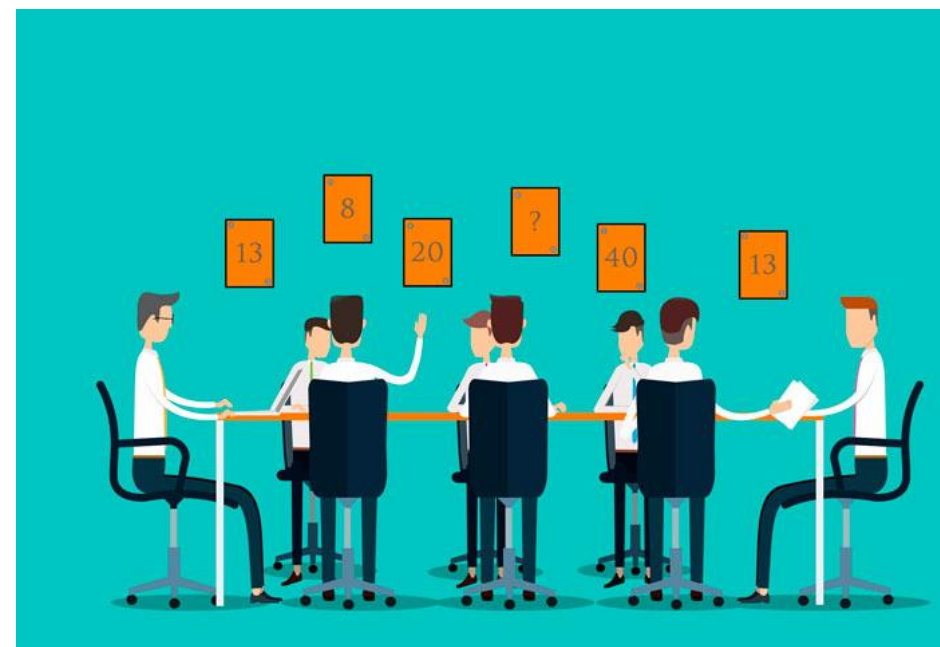
1. Se forma un grupo de expertos con un coordinador designado.
2. El coordinador proporciona una descripción del sistema y formularios para la Estimación.
3. Se discuten en grupo cuestiones generales sobre el sistema.
4. Cada experto hace su propia Estimación de forma anónima.
5. El coordinador procesa las estimaciones individuales.
6. En caso de discrepancias no triviales, se discuten en grupo de nuevo.
7. Se vuelve al punto 4 hasta alcanzar el deseado grado de convergencia.



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación por Juicio Experto: Planning Poker

- La técnica de estimación empírica conocida como “**Planning Poker**”, donde un grupo de expertos discuten y evalúan el esfuerzo requerido para cada tarea del proyecto.
- Se utiliza la opinión de expertos como referencia ya que su experiencia les permite adaptarse a nuevas tareas en base a su similitud.
- Esta técnica se utiliza en metodologías ágiles para estimar cada historia de usuario

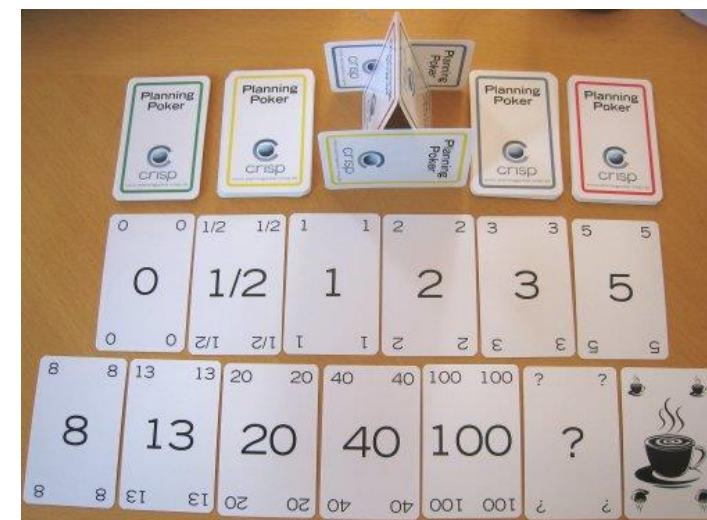




# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación por Juicio Experto: Planning Poker

- La técnica consiste en crear una reunión de trabajo con el grupo de expertos.
- Se requiere de un moderador que dirigirá la actividad.
- Primero se expone el proyecto que se va a desarrollar.
- Luego se determinan las actividades requeridas para ejecutar el proyecto, y entonces para cada una de las actividades los distintos miembros del equipo realizan una estimación empírica y se conjuntan y discuten los resultados con el fin de cerrar la brecha y converger en un valor estimado para cada tarea.
- La información obtenida con la estimación de cada tarea se utiliza para realizar el plan general del proyecto y conocer el esfuerzo total.



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Estimación por Juicio Experto: Planning Poker

### El proceso

- El moderador lee y explica una historia o requerimiento ante los miembros del equipo. Se podrán hacer preguntas para dejar claro qué se desea alcanzar.
- Los participantes anotan en una tarjeta su estimación para dicha tarea y la entregan al moderador de manera anónima, es decir que no se sepa qué tarjeta corresponde a qué persona.
- El moderador captura los resultados en un pizarrón o en alguna herramienta de software.
- Una vez que se conoce los integrantes deberán discutirlos, haciendo énfasis en el razonamiento detrás de los valores extremos (el más bajo y el más alto).
- Tomando en cuenta el conocimiento obtenido por esta discusión, cada participante hace una nueva estimación para la historia y la anota en una tarjeta.
- El moderador captura las estimaciones ajustadas y en caso de no haber consenso se vuelven a discutir.

1. Se dispone de datos fiables sobre proyectos realizados en el pasado.
2. Se identifican atributos del proyecto.
3. Se buscan proyectos con similares atributos.
4. Se ajustan las estimaciones de acuerdo a las características propias.
5. Se requiere del uso de expertos para los 3 puntos anteriores.

- (+) Basado en experiencias pasadas reales que incluyen factores subjetivos (ej. riesgos).
- (-) Precisa de la existencia de proyectos pasados similares.
- (-) ¿Cómo se mide el grado de similitud entre proyectos?
- (-) Lo que era válido en el pasado puede no ser válido ahora

Tamaño del Programa	Software de Sistemas		Software de Gestión		Software "a medida"	
(LDC)	Duración (meses)	Esfuerzo (personas-mes)	Duración (meses)	Esfuerzo (personas-mes)	Duración (meses)	Esfuerzo (personas-mes)
10.000	10	48	6	9	7	15
15.000	12	76	7	15	8	24
20.000	14	110	8	21	9	34
25.000	15	140	9	27	10	44
30.000	16	185	9	37	11	59
35.000	17	220	10	44	12	71
40.000	18	270	10	54	13	88
45.000	19	310	11	61	13	100
50.000	20	360	11	71	14	115
60.000	21	440	12	88	15	145
70.000	23	540	13	105	16	175
80.000	24	630	14	125	17	210
90.000	25	730	15	140	17	240
100.000	26	820	15	160	18	270
120.000	28	1.000	16	200	20	335
140.000	30	1.200	17	240	21	400
160.000	32	1.400	18	280	22	470
180.000	34	1.600	19	330	23	540
200.000	35	1.900	20	370	24	610
250.000	38	2.400	22	480	26	800
300.000	41	3.000	24	600	29	1.000
400.000	47	4.200	27	840	32	1.400
500.000	51	5.500	29	1.100	35	1.800

# ESTIMACION DEL SOFTWARE - Esfuerzo

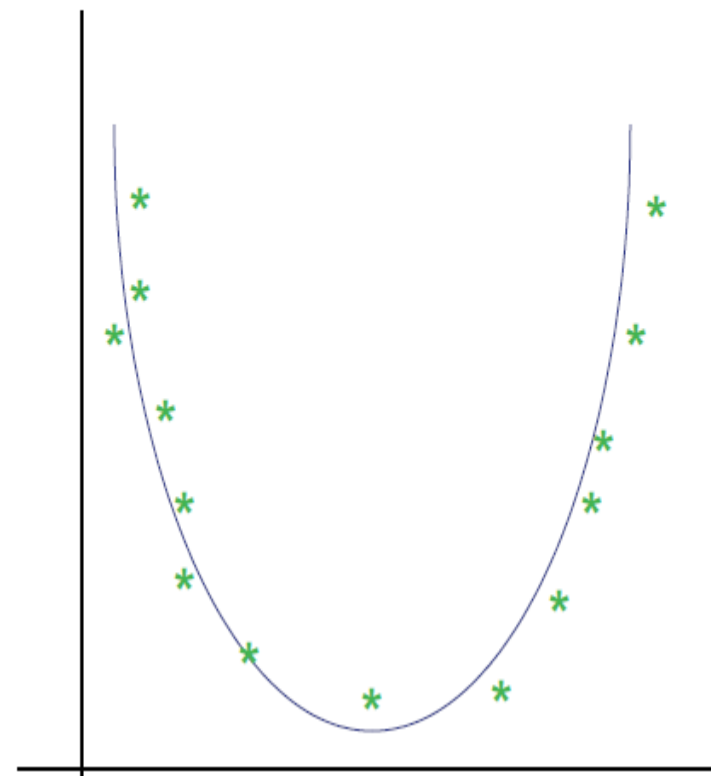
## Estimación Algorítmica o Paramétrica

1. Basados en construir funciones predictivas que estimen el costo de un proyecto.
2. Estas funciones predictivas se calculan mediante ajustes, en función de diversos parámetros, sobre datos de proyectos anteriores.

### Ventajas e Inconvenientes

- (+) Cálculos reproducibles y que permiten refinamiento.
- (+) El problema se reduce a estimar el valor de ciertos parámetros (ej. tamaño, complejidad).
- (+) Basado en sólidas experiencias pasadas estadísticamente significativas.
- (-) Lo que era válido en el pasado puede no ser válido ahora.
- (-) Dificultad de adaptación a circunstancias o eventos no excepcionales.

## Estimación Algorítmica o Paramétrica



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Ajuste del Coste al Precio Esperado (Pricing to Win)

- 1.El cliente tiene un presupuesto fijo.
- 2.El cliente no puede proporcionar una descripción clara y detallada de lo que quiere.
- 3.Se establece como costo del producto el precio que el cliente está dispuesto a pagar.
- 4.A continuación, se identifican los requisitos del sistema software.
- 5.Se implementan los requisitos de forma que el proyecto se ajuste al costo.
- 6.Se pueden dejar requisitos sin implementar.



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Técnicas descendentes (Top-down):

- Se evalúa el costo de un sistema desde sus funcionalidades globales.
- Permite estimar mejor aspectos globales como integración de subsistemas o coordinación entre equipos.
- Se puede aplicar con pocos detalles, en fases tempranas.
- Puede subestimar costos asociados a detalles internos de uno o más subsistemas.
- No resultan adecuadas para tomar decisiones sobre componentes individuales.

## Técnicas ascendentes (Bottom-up)

- Se estima el costo de cada uno de los componentes de un sistema software y luego se calcula el total.
- Permite analizar mejor detalles individuales de cada componente.
- Son en general más precisos, porque la media de los errores en las estimaciones individuales tiende a 0.
- Puede subestimar costes asociados a actividades globales.
- Precisa disponer de una estructura más o menos detallada del sistema y suelen ser más costosos.



# ESTIMACION DEL SOFTWARE - Esfuerzo

## Modelos basados en la Inteligencia artificial

Técnicas de minería de datos a problemas de IS en general y a la estimación en particular. Entre este tipo de técnicas se encuentran

- Redes neuronales.
- Técnicas de aprendizaje computacional.
- Lógica difusa.
- Sistemas expertos.
- Árboles de decisión
- Programación genética

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Sistemas basados en reglas:

- Basados en un conjunto de reglas de la forma

si [....] entonces [.....]

si  $(30 \leq \text{CasosUso} \leq 35)$  entonces esfuerzo = 50 per/mes

- Reglas generadas del histórico del proyecto mediante algoritmos de minería de datos, sirven para tomar acciones en un momento concreto.
- Funciona con lógica difusa.
- Si las reglas se incrementan su manejo y significado se torna difícil.

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Razonamiento basado en casos:

- Consiste en replicar el proceso que un experto llevaría a cabo para tomar una decisión basándose en la experiencia.
- Estimación puramente por el experto (la “base de datos” de los proyectos previos está en la cabeza del experto).
- Estimación por el experto, informalmente soportada por una base de datos conteniendo información acerca de proyectos concluidos.
- Estimación basada en el uso de algoritmos de clusterización para encontrar proyectos similares.

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Redes Neuronales:

- Usadas para el reconocimiento de patrones y clasificación.
- Usadas cuando las interacciones entre las entradas y las salidas y donde los datos de entrada son distorsionados por altos niveles de ruido.
- En el caso de estimación, entre la variable estimación del esfuerzo de un proyecto de software y las variables de entrada que han sido determinadas como características de los proyectos

# ESTIMACION DEL SOFTWARE - Esfuerzo

## Cálculo del Esfuerzo con los Puntos de Caso de Uso (UCP)

El valor de los Puntos de Caso de Uso (UCP) se calcula de la siguiente manera:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF}$$

[KARNER93] propone un factor de 20 horas / hombre por punto de caso de uso, resultado del valor medio obtenido en su investigación, para la estimación temprana del esfuerzo requerido para el desarrollo de un sistema de software.

Por lo tanto, la cantidad total de horas hombre estimada para un proyecto de software se calcula como:

$$\text{Total horas/hombre} = \text{UCP} * \text{MR (recursos necesarios por UCP)} = \text{UCP} * 20$$

# ESTIMACION DEL SOFTWARE - Esfuerzo

Los valores de los tres tipos de proyectos en COCOMO clásico:

MODELO COCOMO BASICO				
PROYECTO	a	b	c	d
ORGANICO	2.4	1.05	2.5	0.38
SEMIACOPLADO	3.0	1.12	2.5	0.35
EMPOTRADO	3.6	1.20	2.5	0.32

MODELO	ESFUERZO Hombre-mes	Tiempo de desarrollo (Meses)	No. Personas
BASICO	$E = a(KLOC)^b$	$T = c E^d$	$P = E/T$
INTERMEDIO	$E = a(KLOC)^b * FAE$	$T = c E^d$	$P = E/T$
AVANZADO	$E = a(KLOC)^b * FAE$	$T = c E^d$	$P = E/T$



# ESTIMACION DEL COSTO DEL SOFTWARE

## Cálculo de Costos

- ¿Cuanto cuesta un recurso (ej. unos auriculares bluetooth, un programador senior)?.
- Existen técnicas que calculan recursos y duraciones necesarias. Para poder multiplicar cantidad de recurso por costo de cada recurso.
- En muchas ocasiones para estimar también el costo de los recursos, se usan catálogos, tablas estandarizadas de precios, etc.

# ESTIMACION DEL COSTO DEL SOFTWARE

## Determinación del Precio Final

1. Precio final = costo total + beneficio.
2. El costo total incluye:
  - Gastos indirectos (ej. infraestructura, plan social).
  - Reserva para contingencias (ej. devaluaciones de divisas).
3. El beneficio se ajusta en función de (entre otros):
  - Oportunidad de mercado (nuevo segmento).
  - Incertidumbre en las estimaciones.
  - Términos contractuales (ej. obliga a permanencia).
  - Volatilidad de los requisitos (software gestión universitaria).
  - Estado financiero de la empresa.



**UNIVERSIDAD**  
Popular del cesar