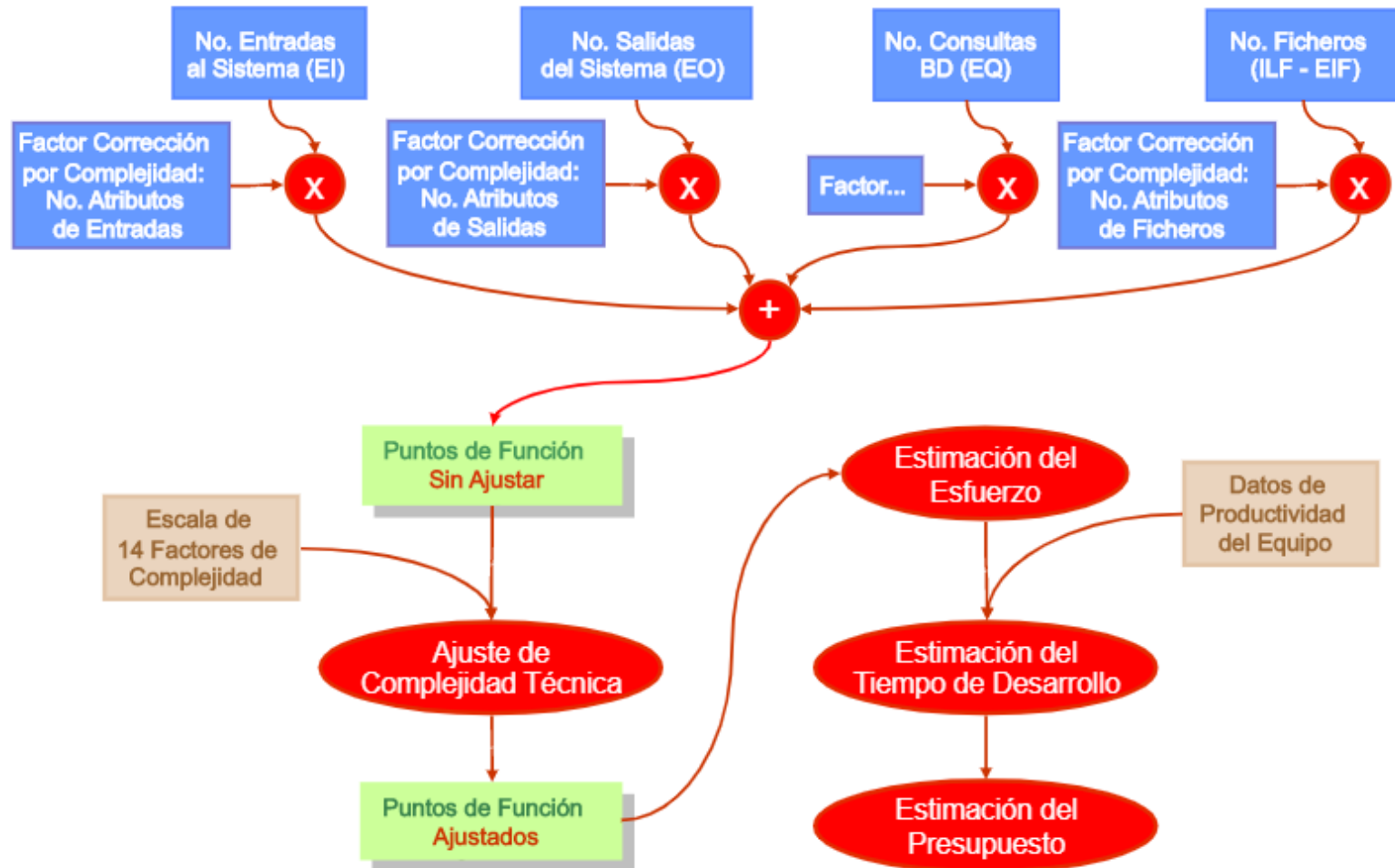


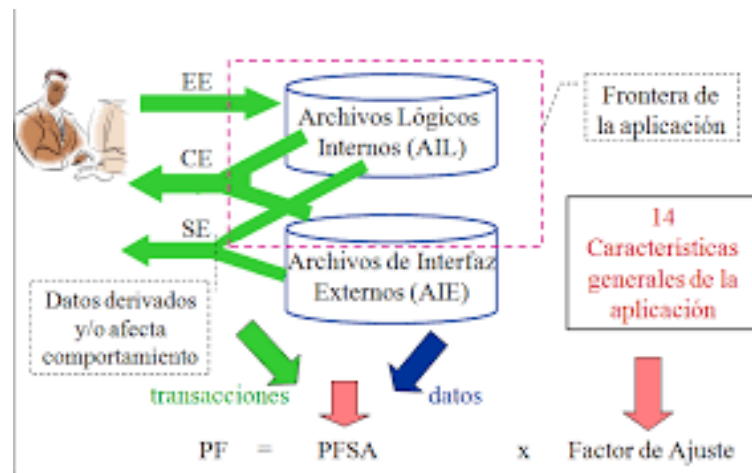
Estimación de puntos de función, esfuerzo, duración y presupuesto en proyectos de software



Resumen

- Aplicar método de estimación de puntos de función
- Estimar el esfuerzo en un proyecto informático (H/H)
- Estimar la duración de un proyecto de software (en meses)
- Estimar el presupuesto del proyecto informático

Estimación de puntos de función



Puntos de función

Todo proyecto de desarrollo de software comienza con una etapa de estimación y planificación, en la cual debemos determinar, a partir de los requerimientos del software, cuánto esfuerzo, personal, recursos materiales, tiempo y en última instancia dinero tomará construir el sistema o producto.

No solo los gerentes de proyectos estiman software, pues a cada desarrollador, analista de sistemas y software tester se le suele asignar un componente y se le pregunta cuánto tiempo le tomará elaborarlo.

Estas estimaciones suelen elaborarse de forma empírica, en base a la experiencia pasadas y conocimiento del trabajo a realizar, sin embargo, suelen ser imprecisas debido a que los requerimientos y sus implicaciones técnicas no se conocen en su totalidad y además ningún desarrollo de software es exactamente igual a uno anterior con el que se pueda comparar.

Frente a esto, se han desarrollado métodos como el de puntos de función, donde estimamos las funcionalidades usando fórmulas matemáticas basadas en parámetros como el tipo de componente, complejidad, factores del entorno, entre otros aspectos

Puntos de función

¿Qué es el método de puntos de función?

Es una técnica de estimación de software desarrollada originalmente por Allan Albrecht en 1979 mientras trabajaba para IBM, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado.

El trabajo de Albrecht fue continuado por el grupo internacional de usuarios de puntos de función, quienes plasmaron sus conceptos en el método **IFPUG-FPA** (Grupo Internacional de Usuarios de Puntos Función- Análisis en Puntos Función)

IFPUG-FPA realiza las valoraciones a partir de la funcionalidad del sistema, primero clasificándolas, luego asignando una complejidad y ponderación a cada una según unas tablas predefinidas, determinando así el valor de puntos de función.

Puntos de función

Sumando los puntos de todas las funcionalidades se obtiene la valoración de todo el proyecto y finalmente se puede aplicar un factor de ajuste, que puede depender de características generales del sistema como por ejemplo requerimientos no funcionales como el rendimiento, reusabilidad, facilidad de instalación y operación entre otros aspectos.

Los puntos de función permiten traducir el tamaño de funcionalidades de software a un número, a través de la suma ponderadas de las características que este tiene.

Una vez que tenemos los puntos de función, podemos traducirlos en horas hombre o días de trabajo, según factor de conversión que dependería de mediciones históricas de nuestra productividad. Con las horas hombre, podemos determinar el costo y presupuesto de los proyectos.

Puntos de función

¿Para qué se utiliza el método de puntos de función?

Estimación de proyectos de software

Una de las principales aplicaciones del método es en la determinación de valoraciones (estimaciones) del producto de software a desarrollar, que es una parte fundamental de todo proceso de ingeniería de software y de la gerencia de estos proyectos.

Dentro del ciclo de ingeniería de software, lo normal es que la estimación del proyecto ocurra después que la ingeniería de requerimientos produzca una primera versión de la especificación de requisitos con suficiente información para elaborar la estimación.

Puntos de función

Validar la calidad de las especificaciones funcionales

Si asignamos el mismo requisito a dos Analistas de sistemas y la estimación producida por los puntos de función difiere, esto nos puede servir para identificar ambigüedades en las especificaciones funcionales.

Seguimiento y control de proyectos

Otras aplicaciones de los puntos de función incluyen el seguimiento y control de proyectos, donde se puede usar para estimar el impacto de cambios en la funcionalidad solicitados durante el desarrollo y también para medir en puntos de función el avance del proyecto e inclusive la facturación en contratos de servicios.

Puntos de función

Medir la productividad y calidad de nuestro proyecto o servicio de ingeniería de software

Por otra parte, los puntos de función tienen aplicaciones en mediciones de productividad, por ejemplo si dividimos las horas hombre empleadas para producir cierta cantidad de puntos de función, obtenemos un indicador de productividad de nuestro equipo y de cómo esta puede mejorar en el tiempo.

Con esto adicionalmente podemos calibrar el factor de conversión entre puntos función y horas hombre que utilicemos para nuestras estimaciones.

Otros indicadores interesantes son número de incidencias por puntos de función producidos como medición de la calidad del proceso de desarrollo.

Puntos de función

¿En qué se diferencia estimar por puntos de función de estimar en base a la experiencia?

Los puntos de función permiten estandarizar las mediciones del tamaño del software, obteniendo estimaciones de mayor exactitud, frente a las que obtendríamos al basarnos solamente en nuestra experiencia y aproximación inexacta.

Diferentes Analistas de sistemas que apliquen el método utilizando los mismos parámetros deberían llegar a conclusiones similares (siempre y cuando los requerimientos estén definidos sin ambigüedades), por lo cual la medición del tamaño del software es más objetiva y auditable.

De esta forma, es posible por ejemplo comparar la complejidad de dos funcionalidades, módulos o proyectos enteros de forma más confiable, permitiendo tomar decisiones informadas sobre como priorizarlos.

A la hora de desarrollar software empresarial, el éxito o fracaso del proyecto suele medirse en función del desempeño de este frente a las expectativas establecidas inicialmente, de allí la importancia de la exactitud de las mediciones de tamaño del software que realizamos inicialmente.

Puntos de función

- Técnica de medición del tamaño funcional del software, desde el punto de vista del cliente.
- El análisis no considera ningún aspecto de implementación de la solución
- Método estándar ISO/IEC 20926 de medición de software que cuantifica los requisitos funcionales del usuario.

Puntos de función

- Antes de existir FPA, la métrica de comparación entre proyectos de software, eran las líneas de código.
- Métrica demasiado técnica. El usuario no puede entender que esta sucediendo

Puntos de función

- Antes de existir FPA, otra métrica de comparación entre proyectos de software, eran la cantidad de pantallas, informes o archivos que entregaba dicho software.
- PPF Toma esto y mide la función, no la cantidad de pantallas por esa función.

Objetivos del proceso de medición

- Ser una medida consistente
- Consistente: Dos profesionales analizando un mismo proyecto llegan al mismo resultado
- Simple para minimizar el esfuerzo de la medición

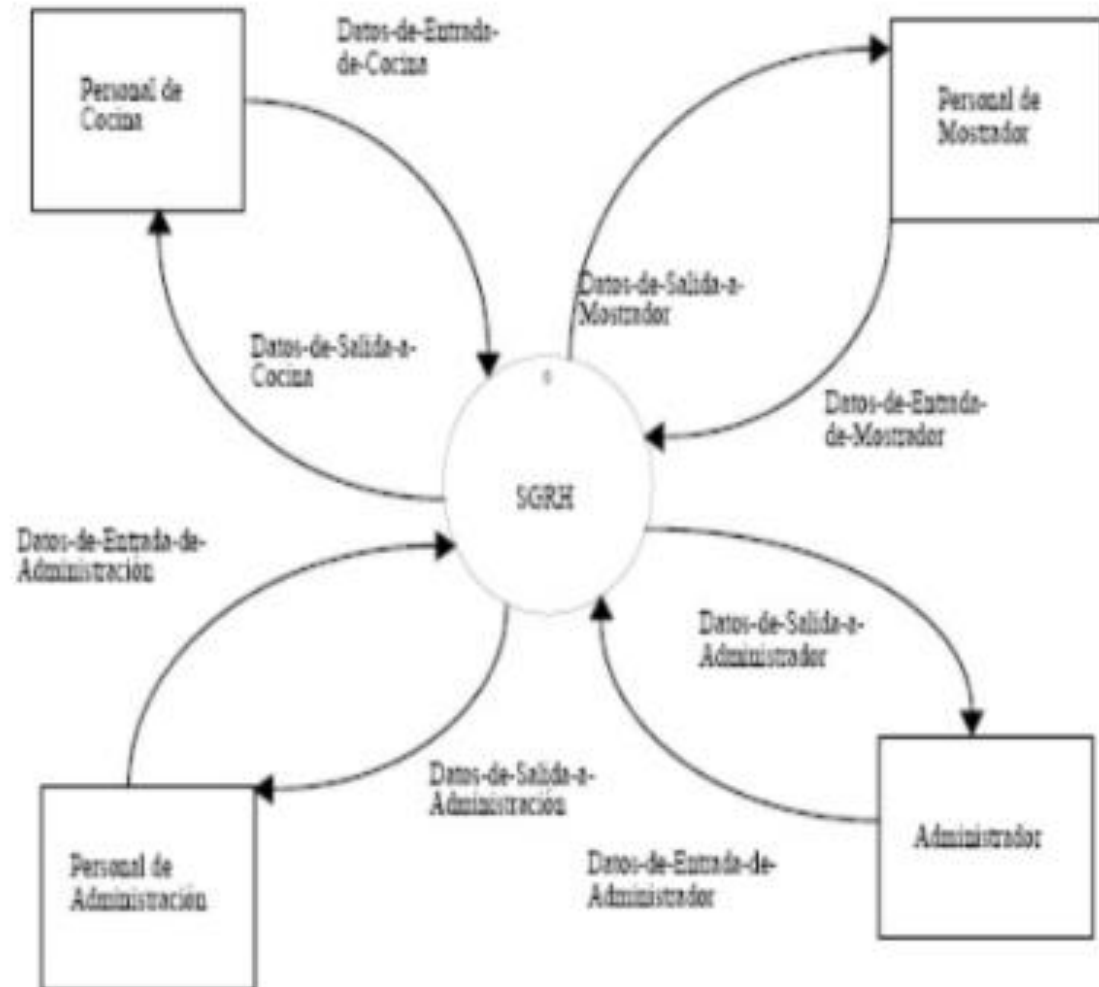
Procedimiento de estimación de los puntos de función



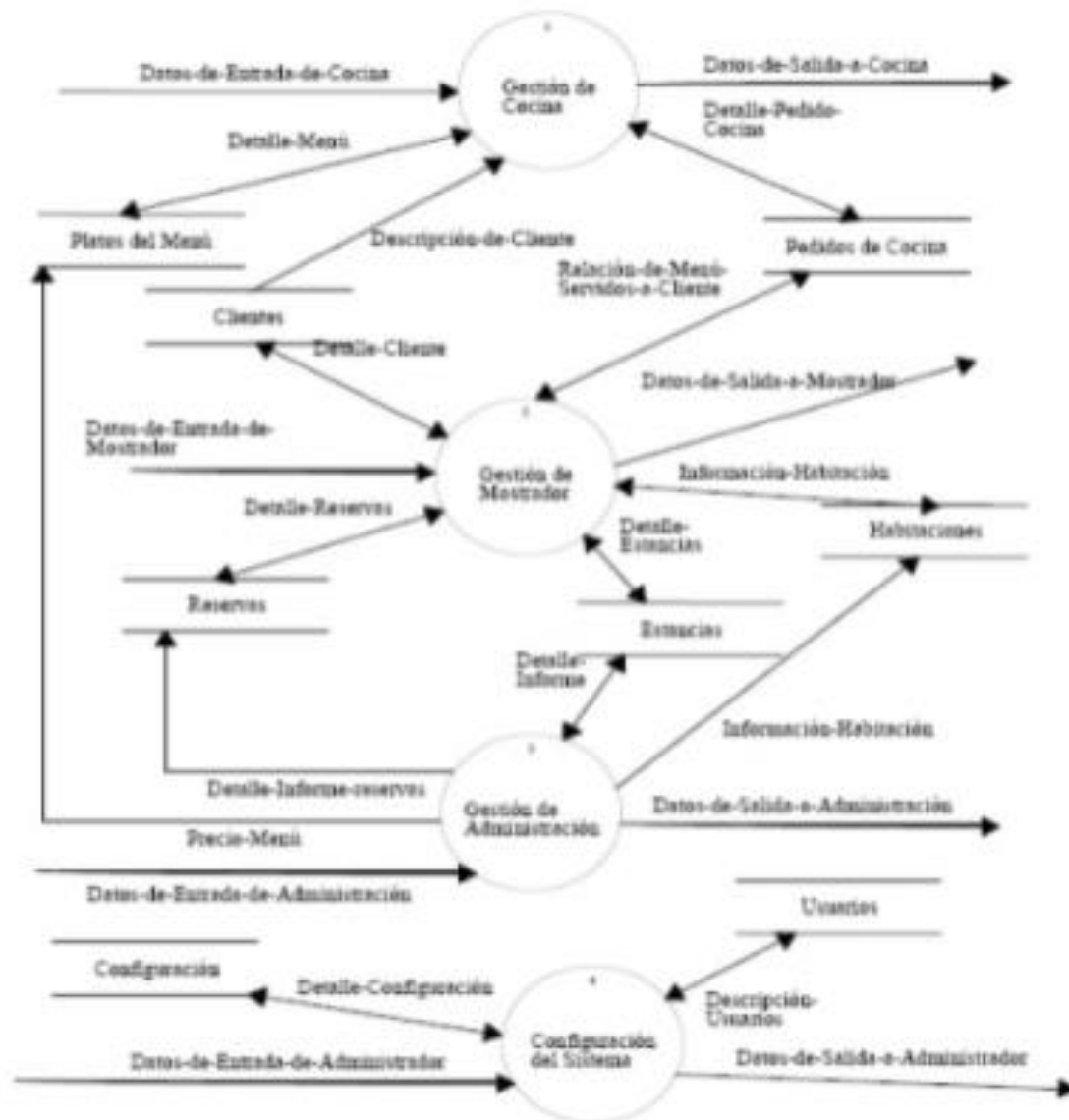
1. Obtener información del sistema

- Se requiere conocimiento global del sistema y construir un Modelo de entidades primarias:
 - Diagrama de Contexto o Diagrama de Flujo de Datos (Programación Estructurada)
 - Diagrama de Casos de Uso (UML)

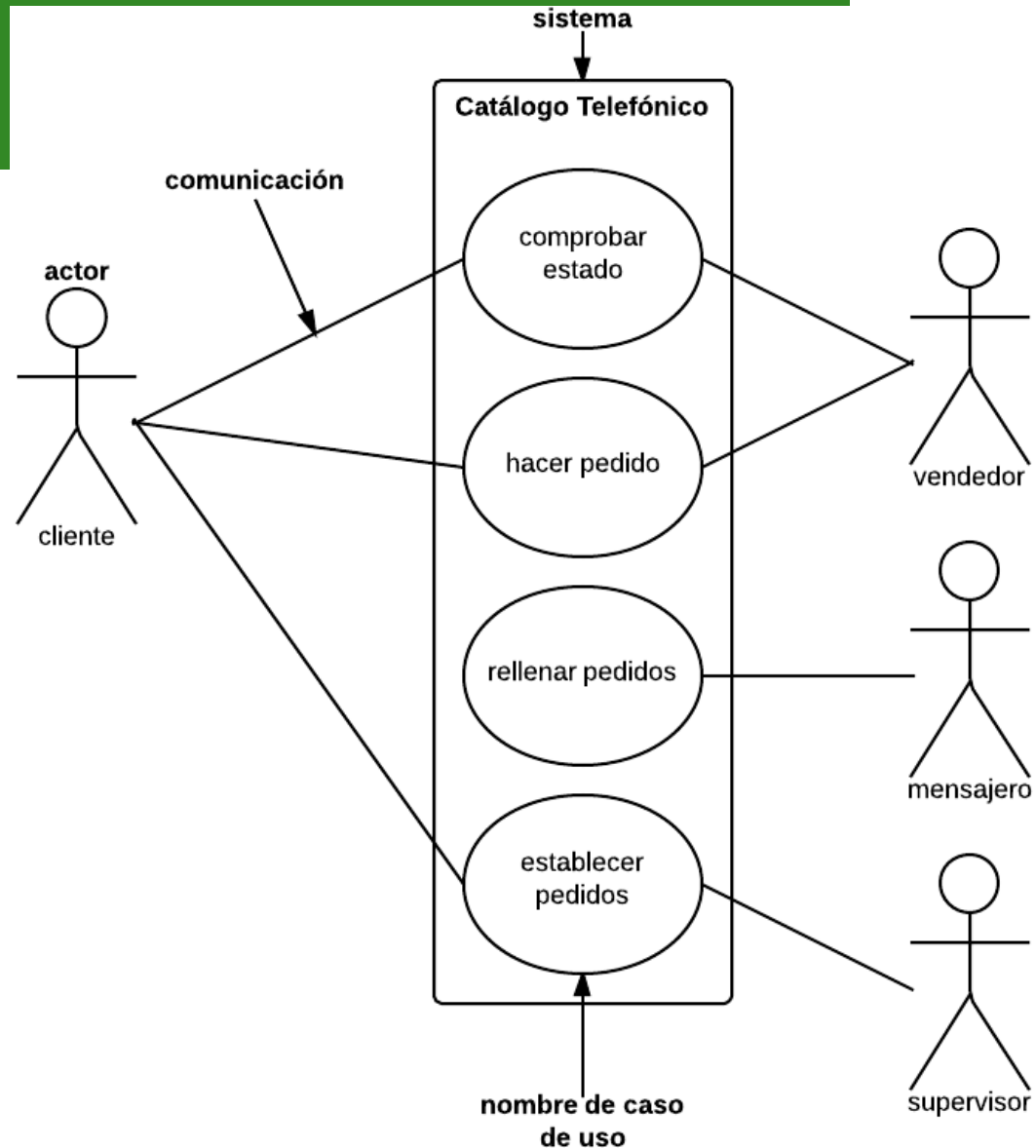
1. Obtener información del sistema



1. Obtener información del sistema



1. Obtener información del sistema



2. Identificar los componentes del sistema

A partir de los diagramas anteriores identificar los componentes del sistema:

- Interacción Función de transacción (Interacción con el usuario)
- Almacenamiento Función de datos

2. Identificar los componentes del sistema

– Interacción Función de transacción

- **Entrada externa** (EI -> External Input)
 - Pantallas donde el usuario ingresa datos
 - Altas, bajas, cambios
- **Salida externa** (EO -> External Output)
 - Producen valor agregado (por ejemplo: agrupan datos)
 - Informes, gráficos, reportes
- **Consulta externa** (EQ -> External Query)
 - Toman el contenido de archivos internos y lo presentan (por ejemplo un listado)
 - Recuperar y mostrar datos al usuario (**Buscar**)

2. Identificar los componentes del sistema

– Almacenamiento Función de datos

- **Archivo lógico interno** (ILF -> Internal Logical File)
 - Archivo del punto de vista lógico
 - Las tablas en la base de datos
- **Archivo de interfaz externo** (EIF -> External Interface File)
 - Datos referenciados a otros sistemas
 - Datos mantenidos por otros sistemas, pero usados por el sistema actual

Ejemplo

- Buscar (**EQ**)
- Actualizar (**EI**)
- Insertar (**EI**)
- Listar (**EQ**)
- Eliminar (**EI**)
- Informes o reportes (**EO**)
- Tablas de BD (**ILF**)

Ejemplo

Componente	
Ingreso de cliente	EI
Modificación de cliente	EI
Listado de clientes	EQ
Reporte de clientes por país	EO
Tabla de clientes	ILF

Ejemplo

- 1 Registro de Equipos de futbol (EI)
- 1 Registro de partidos (EI)
- 1 Buscar partido por fecha (EQ)
- 1 Actualización de datos del equipo (EI)
- 1 Eliminar equipos (EI)
- 1 reporte con gráficos de equipos(EO)
- 1 reporte de los equipos registrados por rango de fechas (EO)
- 1 reporte de partidos (EO)
- 4 Tablas en BD (ILF)

3. Calcular No. de elementos y su complejidad

- Contar los elementos de cada componente y definir su complejidad.
- Para la complejidad se utilizan las siguientes tablas para asignar pesos en función del número de atributos que tengan y el número de archivos a los que afecte.
- Son valores estándar de la **IFPUG**

Ejemplo

- 1 Registro de Equipos de futbol (EI)
- 1 Registro de partidos (EI)
- 1 Buscar partido por fecha (EQ)
- 1 Actualización de datos del equipo (EI)
- 1 Eliminar equipos (EI)
- 1 reporte con gráficos de equipos(EO)
- 1 reporte de los equipos registrados por rango de fechas (EO)
- 1 reporte de partidos (EO)
- 4 Tablas en BD (ILF)

3. Calcular No. de elementos y su complejidad

CLASIFICACION DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

Tabla 1

CLASIFICACION DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

Tabla 2

3. Calcular No. de elementos y su complejidad

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

Tabla 3

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

Tabla 4

Ejemplo

Tipo / Complejidad	Baja	Media	Alta
(EI) Entrada externa	3 PF	4 PF	6 PF
(EO) Salida externa	4 PF	5 PF	7 PF
(EQ) Consulta externa	3 PF	4 PF	6 PF
(ILF) Archivo lógico interno	7 PF	10 PF	15 PF
(EIF) Archivo de interfaz externo	5 PF	7 PF	10 PF

Valores estándar (IFPUG) International Function Point Users Group

Ejemplo

Para el ejemplo, se considerará que todas las funciones identificadas serán de **complejidad media**.

- 1 Registro de Equipos de futbol (EI 4 PF)
- 1 Registro de partidos (EI 4 PF)
- 1 Buscar partido por fecha (EQ 4 PF)
- 1 Actualización de datos del equipo (EI 4 PF)
- 1 Eliminar equipos (EI 4 PF)
- 1 reporte con gráficos de equipos (EO 5 PF)
- 1 reporte de los equipos registrados por rango de fechas (EO 5 PF)
- 1 reporte de partidos (EO 5 PF)
- 4 Tablas en BD (ILF 40 PF)

4. Cálculo de los puntos de función sin ajustar (PFSA)

Tipo / Complejidad	Baja	Media	Alta	TOTAL
(EI) Entrada externa	3 PF	4 x 4 PF	6 PF	16
(EO) Salida externa	4 PF	3 x 5 PF	7 PF	15
(EQ) Consulta externa	3 PF	1 x 4 PF	6 PF	4
(ILF) Archivo lógico interno	7 PF	4 x 10 PF	15 PF	40
(EIF) Archivo de interfaz externo	5 PF	0 x 7 PF	10 PF	0
			PFSA	75

5. Cálculo de los puntos de función ajustados (PFA)

Factor de Ajuste

El cálculo del factor de ajuste está basado en 14 características generales de los sistemas, que miden la funcionalidad general de la aplicación. A cada característica se le atribuye un peso que varía de 0 a 5 e indica el grado o nivel de influencia que cada característica tiene en la aplicación que está siendo estudiada.

Para calcular el factor de ajuste hay que seguir estos pasos:

1. Evaluar el impacto de cada una de las 14 características generales del sistema, atribuyendo un peso de 0 a 5 para cada característica.
2. Calcular el nivel de influencia, que se obtiene sumando los pesos de cada una de las 14 características.
3. Calcular el factor de ajuste, aplicando la fórmula correspondiente:

$$\text{Factor de ajuste} = (\text{Nivel de influencia} * 0.01) + 0.65$$

Niveles de influencia

1. Comunicación de datos

Los datos e informaciones de control utilizados por la aplicación son enviados o recibidos a través de recursos de comunicación de datos. Terminales y estaciones de trabajo son algunos ejemplos. Todos los dispositivos de comunicación utilizan algún tipo de protocolo de comunicación.

Calificar el nivel de influencia en la aplicación de acuerdo con la siguiente tabla:

Grado	Descripción
0	Aplicación puramente <i>batch</i> o funciona en una computadora aislada
1	La aplicación es <i>batch</i> , pero utiliza entrada de datos remota o impresión remota
2	La aplicación es <i>batch</i> , pero utiliza entrada de datos remota e impresión remota
3	La aplicación incluye entrada de datos <i>on-line</i> vía entrada de video o un procesador <i>front-end</i> para alimentar procesos <i>batch</i> o sistemas de consultas.
4	La aplicación es más que una entrada <i>on-line</i> , y soporta apenas un protocolo de comunicación
5	La aplicación es más que una entrada <i>on-line</i> y soporta más de un protocolo de comunicación

Niveles de influencia

2. Procesamiento distribuido

Datos o procesamiento distribuidos entre varias unidades de procesamiento (*CPUs*) son características generales que pueden influenciar en la complejidad de la aplicación.

Grado	Descripción
0	La aplicación no contribuye en la transferencia de datos o funciones entre los procesadores de la empresa
1	La aplicación prepara datos para el usuario final en otra <i>CPU</i> de la empresa
2	La aplicación prepara datos para transferencia, los transfiere y entonces son procesados en otro equipamiento de la empresa (no por el usuario final)
3	Procesamiento distribuido y la transferencia de datos son <i>on-line</i> , en apenas una dirección
4	Procesamiento distribuido y la transferencia de datos son <i>on-line</i> , en ambas direcciones
5	Las funciones de procesamiento son dinámicamente ejecutadas en el equipamiento más adecuado

Niveles de influencia

3. *Objetivos de Rendimiento*

Los objetivos de rendimiento del sistema, establecidos y aprobados por el usuario en términos de respuesta, influyen o podría influenciar el proyecto, desarrollo, implementación o soporte de la aplicación.

Grado	Descripción
0	Ningún requerimiento especial de performance fue solicitado por el usuario
1	Requerimientos de performance y de diseño fueron establecidos y previstos, sin embargo ninguna acción especial fue requerida
2	El tiempo de respuesta y el volumen de datos son críticos durante horarios pico de procesamiento. Ninguna determinación especial para la utilización del procesador fue establecida. El intervalo de tiempo límite para la disponibilidad de procesamiento es siempre el próximo día hábil
3	El tiempo de respuesta y volumen de procesamiento son items críticos durante todo el horario comercial. Ninguna determinación especial para la utilización del procesador fue establecida. El tiempo limite necesario para la comunicación con otros sistemas es un aspecto importante
4	Los requerimientos de performance establecidos requieren tareas de análisis de performance en la fase de análisis y diseño de la aplicación
5	Además de lo descrito en el ítem anterior, herramientas de análisis de performance fueron usadas en las fases de diseño, desarrollo y/o implementación para atender los requerimientos de performance establecidos por el usuario

Niveles de influencia

4. Configuración del equipamiento

Esta característica representa la necesidad de realizar consideraciones especiales en el diseño de los sistemas para que la configuración del equipamiento no sea sobrecargada

Grado	Descripción
0	Ninguna restricción operacional explícita o implícita fue incluida
1	Existen restricciones operacionales leves. No es necesario un esfuerzo especial para resolver estas restricciones
2	Algunas consideraciones de ajuste de performance y seguridad son necesarias
3	Son necesarias especificaciones especiales de procesador para un módulo específico de la aplicación
4	Restricciones operacionales requieren cuidados especiales en el procesador central o procesador dedicado
5	Además de las características del ítem anterior, hay consideraciones especiales en la distribución del sistema y sus componentes

Niveles de influencia

5. Tasa de transacciones

El nivel de transacciones es alto y tiene influencia en el diseño, desarrollo, implementación y mantenimiento de la aplicación

Grado	Descripción
0	No están previstos periodos picos de volumen de transacción
1	Están previstos picos de transacciones mensualmente, trimestralmente, anualmente o en un cierto periodo del año
2	Se prevén picos semanales
3	Se prevén picos diariamente
4	Alto nivel de transacciones fue establecido por el usuario, el tiempo de respuesta necesario exige un nivel alto o suficiente para requerir análisis de performance y diseño
5	Además de lo descrito en el ítem anterior, es necesario utilizar herramientas de análisis de performance en las fases de diseño, desarrollo y/o implementación

Niveles de influencia

6. *Entrada de datos en línea*

Esta característica cuantifica la entrada de datos *on-line* provista por la aplicación

Grado	Descripción
0	Todas las transacciones son procesadas en modo <i>batch</i>
1	De 1% al 7% de las transacciones son entradas de datos <i>on-line</i>
2	De 8% al 15% de las transacciones son entradas de datos <i>on-line</i>
3	De 16% al 23% de las transacciones son entradas de datos <i>on-line</i>
4	De 24% al 30% de las transacciones son entradas de datos <i>on-line</i>
5	Más del 30% de las transacciones son entradas de datos <i>on-line</i>

Factor de ajuste

7. Interfase con el usuario

Las funciones *on-line* del sistema hacen énfasis en la amigabilidad del sistema y su facilidad de uso, buscando aumentar la eficiencia del usuario final. El sistema posee:

- Ayuda para la navegación (teclas de función, accesos directos y menús dinámicos)
- Menús
- Documentación y ayuda *on-line*
- Movimiento automático del cursor
- *Scrolling* vertical y horizontal
- Impresión remota (a través de transacciones *on-line*)
- Teclas de función preestablecidas
- Ejecución de procesos batch a partir de transacciones *on-line*
- Selección de datos vía movimiento del cursor en la pantalla
- Utilización intensa de campos en video reverso, intensificados, subrayados, coloridos y otros indicadores
- Impresión de la documentación de las transacciones *on-line* por medio de *hard copy*
- Utilización del *mouse*
- Menús *pop-up*
- El menor número de pantallas posibles para ejecutar las funciones del negocio
- Soporte bilingüe (el soporte de dos idiomas, cuente como cuatro items)
- Soporte multilingüe (el soporte de más de dos idiomas, cuente como seis items)

Niveles de influencia

Grado	Descripción
0	ningún de los items descritos
1	De uno a tres de los items descritos
2	De cuatro a cinco de los items descritos
3	Más de cinco de los items descritos, no hay requerimientos específicos del usuario en cuanto a amigabilidad del sistema
4	Más de cinco de los items descritos, y fueron descritos requerimientos en cuanto a amigabilidad del sistema suficientes para generar actividades específicas incluyendo factores tales como minimización de la digitación
5	Más de cinco de los items descritos y fueron establecidos requerimientos en cuanto a la amigabilidad suficientes para utilizar herramientas especiales y procesos especiales para demostrar anticipadamente que los objetivos fueron alcanzados

Niveles de influencia

8. Actualización en línea

La aplicación posibilita la actualización *on-line* de los archivos lógicos internos

Grado	Descripción
0	Ninguna
1	Actualización <i>on-line</i> de uno a tres archivos lógicos internos
2	Actualización <i>on-line</i> de más de tres archivos lógicos internos
3	Actualización <i>on-line</i> de la mayoría de los archivos lógicos internos
4	Además del ítem anterior, la protección contra pérdidas de datos es esencial y fue específicamente proyectado y codificado en el sistema
5	Además del ítem anterior, altos volúmenes influyen en la las consideraciones de costo en el proceso de recuperación. Procesos para automatizar la recuperación fueron incluios minimizando la intervención del operador

Niveles de influencia

9. Procesamiento complejo

El procesamiento complejo es una de las características de la aplicación, los siguientes componentes están presentes:

- Procesamiento especial de auditoria y/o procesamiento especial de seguridad
- Procesamiento lógico extensivo
- Procesamiento matemático extensivo
- Gran cantidad de procesamiento de excepciones, resultando en transacciones incompletas que deber ser procesadas nuevamente. Por ejemplo, transacciones de datos incompletas interrumpidas por problemas de comunicación o con datos incompletos
- Procesamiento complejo para manipular múltiples posibilidades de entrada/salida. Ejemplo: multimedia

Niveles de influencia

Grado	Descripción
0	Ninguno de los items descritos
1	apenas uno de los items descritos
2	Dos de los items descritos
3	Tres de los items descritos
4	Cuatro de los items descritos
5	Todos los items descritos

Niveles de influencia

10. Reusabilidad del código

La aplicación y su código serán o fueron proyectados, desarrollados y mantenidos para ser utilizados en otras aplicaciones.

Grado	Descripción
0	No presenta código reutilizable
1	Código reutilizado fue usado solamente dentro de la aplicación
2	Menos del 10% de la aplicación fue proyectada previendo la utilización posterior del código por otra aplicación
3	10% o más de la aplicación fue proyectada previendo la utilización posterior del código por otra aplicación
4	La aplicación fue específicamente proyectada y/o documentada para tener su código fácilmente reutilizable por otra aplicación y la aplicación es configurada por el usuario a nivel de código fuente
5	La aplicación fue específicamente proyectada y/o documentada para tener su código fácilmente reutilizable por otra aplicación y la aplicación es configurada para uso a través de parámetros que pueden ser alterados por el usuario

Factor de ajuste

11. Facilidad de implementación

La facilidad de implementación y conversión de datos son características de la aplicación. Un plan de conversión e implementación y/o herramientas de conversión fueron provistas y probadas durante la fase de prueba de la aplicación

Grado	Descripción
0	Ninguna consideración especial fue establecida por el usuario y ningún procedimiento especial fue necesario en la implementación
1	Ninguna consideración especial fue establecida por el usuario, más procedimientos especiales son requeridos en la implementación
2	Requerimientos de conversión e implementación fueron establecidos por el usuario y rutinas de de conversión e implementación fueron proporcionados y probados. el impacto de conversión en el proyecto no es considerado importante
3	Requerimientos de conversión e implementación fueron establecidos por el usuario y rutinas de de conversión e implementación fueron proporcionados y probados. el impacto de conversión en el proyecto es considerado importante
4	Además del ítem 2, conversión automática y herramientas de implementación fueron proporcionadas y probadas
5	Además del ítem 3, conversión automática y herramientas de implementación fueron proveídas y

Niveles de influencia

12. Facilidad de operación

La facilidad de operación es una característica del sistema. Procedimientos de inicialización, respaldo y recuperación fueron proveídos y probados durante la fase de prueba del sistema. La aplicación minimiza la necesidad de actividades manuales, tales como montaje de cintas magnéticas, manejo de papel e intervención del operador.

Grado	Descripción
0	Ninguna consideración especial de operación, además del proceso normal de respaldo establecido por el usuario
1 - 4	Verificar cuáles de las siguientes afirmaciones pueden ser identificadas en la aplicación. Cada ítem vale un punto, excepto se defina lo contrario: <ul style="list-style-type: none">• Fueron desarrollados procedimientos de inicialización y respaldo, siendo necesaria la intervención del operador• Se establecieron procesos de inicialización, respaldo y recuperación sin ninguna intervención del operador (contar como 2 ítems)• La aplicación minimiza la necesidad de montaje de cintas magnéticas• La aplicación minimiza la necesidad de manejo de papel
5	La aplicación fue diseñada para trabajar sin operador, ninguna intervención del operador es necesaria para operar el sistema, excepto ejecutar y cerrar la aplicación. La aplicación posee rutinas automáticas de recuperación en caso de error

Niveles de influencia

13. Instalaciones Múltiples

La aplicación fue específicamente proyectada, diseñada e mantenida para ser instalada en múltiples locales de una organización o para múltiples organizaciones.

Grado	Descripción
0	Los requerimientos del usuario no consideran la necesidad de instalación de más de un local
1	La necesidad de múltiples locales fue considerada en el proyecto y la aplicación fue diseñada para operar apenas sobre el mismo ambiente de hardware y software
2	La necesidad de múltiples locales fue considerada en el proyecto y la aplicación fue diseñada para operar en ambientes similares de software y hardware
3	La necesidad de múltiples locales fue considerada en el proyecto y la aplicación está separada para trabajar sobre diferentes ambientes de hardware y/o software
4	Plan de mantenimiento y documentación fueron proporcionados y probados para soportar la aplicación en múltiples locales, además los ítems 1 y 2 caracterizan a la aplicación
5	Plan de documentación e mantenimiento fueron proveídos y probados para soportar la aplicación en múltiples locales, además el ítem 3 caracteriza a la aplicación

Factor de ajuste

14. Facilidad de cambios

La aplicación fue específicamente proyectada y diseñada con vistas a facilitar su mantenimiento. Las siguientes características pueden ser atribuidas a la aplicación:

- Están disponibles facilidades como consultas e informes flexibles para atender necesidades simples (contar 1 ítem)
- Están disponibles facilidades como consultas e informes flexibles para atender necesidades de complejidad media (contar 2 ítems)
- Están disponibles facilidades como consultas e informes flexibles para atender necesidades complejas (contar 3 ítems)
- Datos de control son almacenados en tablas que son mantenidas por el usuario a través de procesos *on-line*, pero los cambios se hacen efectivos solamente al día siguiente
- Datos de control son almacenados en tablas que son mantenidas por el usuario a través de procesos *on-line*, pero los cambios se hacen efectivos inmediatamente (contar 2 ítems)

Niveles de influencia

Grado	Descripción
0	Ninguno de los items descritos
1	apenas uno de los items descritos
2	Dos de los items descritos
3	Tres de los items descritos
4	Cuatro de los items descritos
5	Todos los items descritos

Ejemplo

Características generales del sistema	Nivel de influencia
1- Comunicación de datos	4
2- Procesamiento distribuido	4
3- Perfomance (desempeño)	1
4-Configuración del equipamiento	1
5- Volumen de transacciones	3
6- Entrada de datos <i>on-line</i>	5
7- Interfase con el usuario	2
8- Actualización <i>on-line</i>	3
9- Procesamiento complejo	1
10- Reusabilidad	1
11-Facilidad de implementación	0
12- Facilidad de operación	1
13- Múltiples locales	2
14- Facilidad de cambios	4
Nivel de influecia Total	32

Factor de ajuste

El factor de ajuste se calcula mediante la fórmula:

$$\text{Factor de ajuste} = (\text{Nivel de influencia} * 0.01) + 0.65$$

Utilizando la fórmula en el ejemplo:

$$\text{Factor de ajuste} = (32 * 0.01) + 0.65 = 0.97$$

5. Cálculo de los puntos de función ajustados (PFA)

- $PFA = PFSA * \text{Factor de ajuste}$
- Donde:
 - **PFSA**: Puntos de función sin ajustar
 - **PFA** : Puntos de función ajustado

$$PFA = 75 * 0.97$$

$$PFA = 72.75 \rightarrow 73$$

6. Estimación del esfuerzo

- **PFA = 73**
- *El objetivo ahora es estimar la cantidad de esfuerzo necesario para desarrollar la aplicación. Este esfuerzo se mide en horas/hombre, meses/hombre o años/hombre. Los puntos de función en cierto modo son una medida subjetiva*

6. Estimación de esfuerzo

- *La cantidad de horas/hombre por punto de función es algo difícil e impreciso de valorar, de forma global. Esto es normal, lo contrario sería suponer que la productividad de todas las empresas de desarrollo de software es igual.*

6. Estimación de esfuerzo

Lenguaje	Horas PF promedio	Lineas de código por PF
Ensamblador	25	300
COBOL	15	100
Lenguajes 4ta Generación	8	20

$H/H = PFA * \text{Horas PF promedio}$

$H/H = 73 * 8$

$H/H = \mathbf{584 \text{ Horas Hombre}}$

7. Estimación de la duración del proyecto

H/H = **584 Horas Hombre**

Datos complementarios:

5 horas diarias de trabajo efectivo

1 mes = 20 días

Desarrolladores = **1**

Horas = $584 / 1 = 584$ horas (Duración del proyecto en horas)

$584 / 5 = 116.8$ días de trabajo

$116.8 / 20 = \mathbf{5.84}$ meses para desarrollar el software de lunes a viernes **5** horas diarias con **1** trabajador (**ESTIMACIÓN de duración del proyecto**)

7. Estimación de la duración del proyecto

H/H = **584 Horas Hombre**

Datos complementarios:

5 horas diarias de trabajo efectivo

1 mes = 20 días

Desarrolladores = **3**

Horas = $584 / 3 = 194.7$ horas (Duración del proyecto en horas)

$194.7 / 5 = 38.94$ días de trabajo

$38.94 / 20 = \mathbf{1,95}$ meses para desarrollar el software de lunes a viernes **5** horas diarias con **3** desarrolladores (**ESTIMACIÓN de duración del proyecto**)

8. Cálculo de presupuesto del proyecto

- Sueldo mensual desarrolladores: \$9,000.00
- Otros costos del proyecto: \$10,000.000
- Costo = (Desarrolladores * Duración meses * sueldos) + Otros costos
- Costo = (3 * 1.95 * 9,000.00) + 10,000.00 =
\$62,650.00 + IVA

Ejercicio

- Aplicar método de estimación de puntos de función
- Estimar el esfuerzo de su proyecto informático (H/H)
- Estimar la duración de su proyecto (en meses)
- Estimar el presupuesto de su proyecto informático