

[Aplicaciones de Ingeniería de Software]

Métricas del software y
Estimaciones

[Calidad del Software]

- Cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas que se esperan de todo software desarrollado profesionalmente.

[Factores de calidad de McCall]

- **Corrección:** El grado en el que el programa cumple con su especificación y satisface los objetivos que propuso el cliente.
- **Confiabilidad:** El grado en que se esperaría que un programa desempeñe su función con la precisión requerida.

[Factores de calidad de McCall]

- **Eficiencia:** La cantidad de código y de recursos de cómputo necesarios para que un programa realice su función.
- **Integridad:** El grado de control sobre el acceso al software o los datos por parte de personas autorizadas.

[Factores de calidad de McCall]

- **Facilidad de uso:** El esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar su salida.
- **Facilidad de mantenimiento:** El esfuerzo necesario para localizar y corregir un error en un programa

[Factores de calidad de McCall]

- **Flexibilidad:** El esfuerzo necesario para modificar un programa en operación.
- **Facilidad de prueba:** El esfuerzo que demanda probar un programa con el fin de asegurar que realiza su función.

[Factores de calidad de McCall]

- **Portabilidad:** El esfuerzo necesario para transferir el programa de un entorno de hardware o software a otro.
- **Facilidad de reutilización:** El grado en que un programa (o partes de él) puede reutilizarse en otras aplicaciones (en relación con el empaquetamiento y el alcance de las funciones que realiza el programa).

[Factores de calidad de McCall]

- **Interoperabilidad:** El esfuerzo necesario para acoplar un sistema a otro.

[Medidas, métricas e indicadores]

- **Medición:** es el acto de determinar una medida.

Recopilación de uno o más puntos de datos. Ejem: se investigan varias revisiones de componentes y pruebas de unidad para reunir medidas del número de errores encontrados en cada uno.

[Medidas, métricas e indicadores]

- **Medida:** una indicación cuantitativa de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso.

Recopilar un sólo tipo de datos: número de errores descubiertos de un sólo componente de software

[Medidas, métricas e indicadores]

- **Métrica** (IEEE): medida cuantitativa del grado en que un sistema componente o proceso posee un atributo determinado.

Relaciona las medidas individuales:
el número promedio de errores encontrados en cada revisión o prueba de unidad

[Medidas, métricas e indicadores]

- **Indicador:** es una métrica o una combinación de métricas que proporcionan conocimientos acerca de los procesos de software, un proyecto de software o el propio producto.

Una ciencia tiene la misma madurez que sus herramientas de medición (Luis Pasteur)

[Proceso de recopilación de métricas de software]

- La recopilación de datos requiere una investigación histórica de los proyectos previos para reconstruir los datos requeridos.
- Una vez que se han recopilado las medidas es posible calcular las métricas.

[Proceso de recopilación de métricas de software]

- Las métricas deben evaluarse y aplicarse durante la estimación del trabajo técnico, el control del proyecto y la mejora del proceso.
- La evaluación de las métricas se centra en las razones de los resultados obtenidos y produce un conjunto de indicadores que guían el proyecto o el proceso.

[Principios de medición]

- **Formulación:** La derivación de medidas y métricas apropiadas para la representación del software que se considera.
- **Recolección:** El mecanismo con que se acumulan los datos necesarios para derivar las métricas formuladas.
- **Análisis:** El cálculo de las métricas y la aplicación de herramientas matemáticas.

[Principios de medición]

- **Interpretación:** La evaluación de las métricas en un esfuerzo por conocer mejor la calidad de la representación.
- **Retroalimentación:** Recomendaciones derivadas de la interpretación de las métricas del producto transmitidas al equipo de software.

[Principios de medición]

- *Una métrica debe tener propiedades matemáticas deseables. El valor de la métrica debe estar en un rango significativo.*
- *Cuando una métrica representa una característica de software que aumenta cuando se representan rasgos positivos o que disminuye al encontrar rasgos indeseables, el valor de la métrica debe aumentar o disminuir en el mismo sentido.*

[Principios de medición]

- *Cada métrica debe validarse empíricamente en una amplia variedad de contextos antes de publicarse o aplicarse en la toma de decisiones. Una métrica debe medir el factor de interés, independientemente de otros factores.*

Panorámica de las métricas del producto

■ Métricas para el modelo de análisis:

- Funcionalidad entregada: medida indirecta de la funcionalidad que se empaqueta con el software
- Tamaño del sistema: mide el tamaño, desde el punto de vista de la información disponible del modelo de análisis
- Calidad de la especificación: grado en que se ha completado la especificación de requisitos

Panorámica de las métricas del producto

■ Métricas para el modelo de diseño:

- Métricas arquitectónicas: proporciona un indicio de la calidad del diseño arquitectónico
- Métricas al nivel del componente: miden la principalmente complejidad de los componentes de software

Panorámica de las métricas del producto

■ Métricas para el modelo de diseño:

- Métricas de diseño de interfaz: facilidad de uso
- Métricas especializadas en diseño OO: miden características de clases, de comunicación y colaboración.

Panorámica de las métricas del producto

■ Métricas de código fuente:

- Métricas de Halstead: proporcionan medidas únicas de un programa de cómputo
- Métricas de complejidad: miden la complejidad lógica del código fuente
- Métricas de longitud: proporcionan un indicio del tamaño del software

Panorámica de las métricas del producto

■ Métricas para pruebas:

- Métricas de cobertura de instrucciones y ramas: lleva al diseño de casos de prueba que proporciona cobertura del programa
- Métricas relacionadas con los defectos: se concentran en encontrar los defectos

Panorámica de las métricas del producto

■ Métricas para pruebas:

- Métricas de efectividad de prueba: proporcionan un indicio en tiempo real de la efectividad de las pruebas aplicadas
- Métricas en el proceso: métricas relacionadas con el proceso que se determinan a medida que se aplican las pruebas

[Métricas para la calidad de la especificación]

- [Davis, 93] propone las siguientes características para evaluar el modelo de análisis y la especificación de requisitos:
 - Especificidad (falta de ambigüedad)
 - Grado de avance
 - Corrección
 - Facilidad de comprensión
 - Facilidad de verificación
 - Consistencia interna y externa

[Métricas para la calidad de la especificación]

- Facilidad para alcanzar los objetivos
- Facilidad para darle seguimiento
- Facilidad para modificarse
- Precisión
- Facilidad de reutilización

Métricas para la calidad de la especificación

- Supóngase que hay n_r requisitos en una especificación, de modo que

$$n_r = n_f + n_{nf}$$

Donde n_f es el número de requisitos funcionales y el de n_{nf} no funcionales

Métricas para la calidad de la especificación

- Para determinar la especificidad (falta de ambigüedad) de los requisitos, se sugiere una métrica basada en la consistencia de la interpretación de los revisores de cada requisito:

$$Q_1 = n_{ui} / n_r$$

Donde n_{ui} es el número de requisitos que todos los revisores interpretaron de la misma manera. Cuanto más cercano esté el valor de Q a 1, menor será la ambigüedad de la especificación.

[Cuál es tu interpretación del requisito-característica?]

- Estudio epidemiológico de casos

[Medición del software]

- Medidas Directas del :
 - Proceso:
 - Costo y esfuerzo aplicados
 - Producto:
 - Líneas de código producidas
 - Rapidez de ejecución
- Medidas Indirectas del producto:
 - Funcionalidad
 - Calidad
 - Complejidad, etc.

[Métricas orientadas al tamaño (Lines of Code LOC)]

- Se deben considerar las actividades de ingeniería de software (análisis, diseño, código y prueba)
- Se pueden desarrollar un conjunto de métricas orientadas al tamaño para cada proyecto:
 - Errores por KLOC (miles de líneas de código)
 - Defectos por KLOC
 - Costo por KLOC
 - Errores por persona-mes
 - KLDC por persona-mes
 - Costo por página de documentación

[Métricas orientadas al tamaño]

- Controversias
 - Depende del lenguaje de programación
 - El planificador debe estimar las LOC que se producirán mucho antes de que el análisis y el diseño se hayan completado
 - Mejores programadores producen menos código

[Métricas orientadas al tamaño]

Proyecto	LOC	Esfuerzo (p/m)	\$(000)	Pág.Doc.	Errores	Defectos
Alfa	12,100	24	168	365	134	29
Beta	27,200	62	440	1,224	321	86
Gamma	20,200	43	314	1,050	256	64

[Métricas basadas en la función]

- Utilizada para medir la funcionalidad que entrega un sistema [Albrecht, 79]
- Emplea datos históricos como valor de normalización, *Puntos de Función PF*
- El cálculo se basa en características del dominio de información y la complejidad del software.

Propósito de las Métricas basadas en la función (PF)

- El conteo de puntos de función es utilizado como herramienta para la estimación, ejecución y control y cierre del proyecto:
 1. Estimar el esfuerzo, duración y personal requeridos por el proyecto de desarrollo, utilizando algún modelo de estimación estadístico y una Base de Datos de Proyectos Históricos.
 2. Controlar el crecimiento de tamaño a lo largo del ciclo de vida.

Propósito de las Métricas basadas en la función (PF)

3. Controlar la cantidad de producto que es generado a lo largo del proyecto.
4. Evaluar productividad (Puntos de Función / Personas-Mes) y Calidad (Defectos / Puntos de Función) para dar información a los procesos de Mejora Continua del Desarrollo de Software y alimentar la Base de Datos de Proyectos Históricos que podrá ser utilizada en futuras estimaciones.

Métricas basadas en la función

- Consideraciones a favor:
 - No depende del lenguaje de programación
 - Se basa en datos que es más probable que se conozcan temprano en la evolución de un proyecto

Métricas basadas en la función

- Consideraciones en contra:
 - No exista un estándar para la especificación de requisitos
 - Documentación inconsistente
 - Requisitos incompletos
 - No existe un estándar en la especificación de requisitos
 - Carencia de comprensión funcional
 - Desorganización

[Problemas con las medidas]

- Dificultad para cuantificar cualquier cosa
- Tiempo consumido
- Beneficios no son inmediatos
- Consistencia

[International Function Point User Group (IFPUG)]

- Existen diferentes métodos para calcular los PF basado en la propuesta inicial de Albrecht, y uno de los organismos internacionales encargados de revisarlas y establecer las reglas de su uso ha sido IFPUG

Componentes del Análisis de PF

■ Tipos de Función Transaccional

- External Inputs (EI)
- External Outputs (EO)
- External Inquiries (EQ)

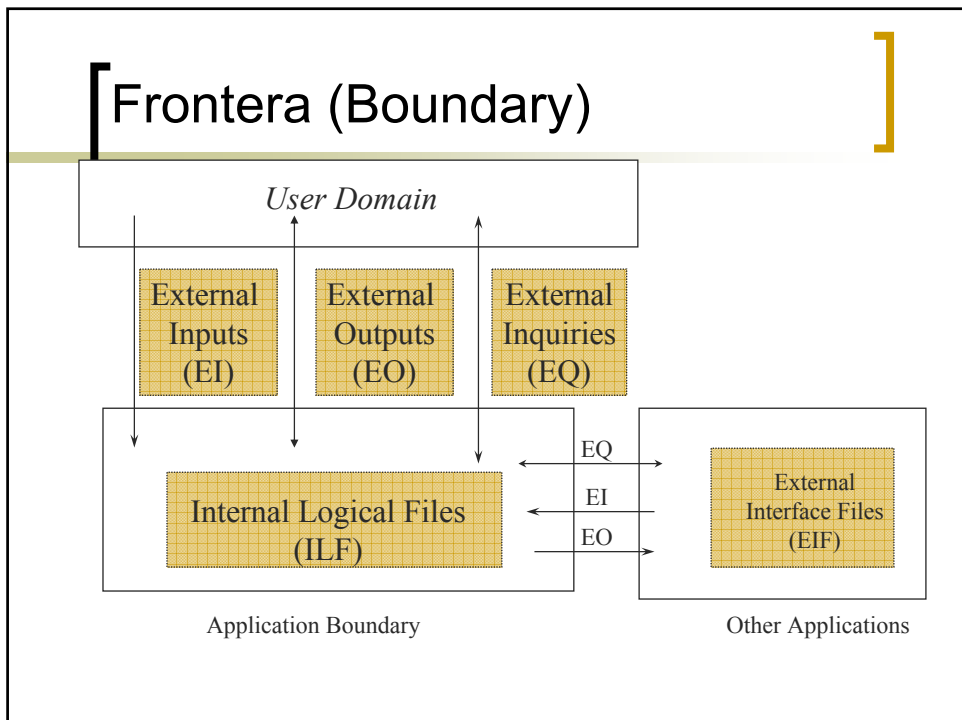
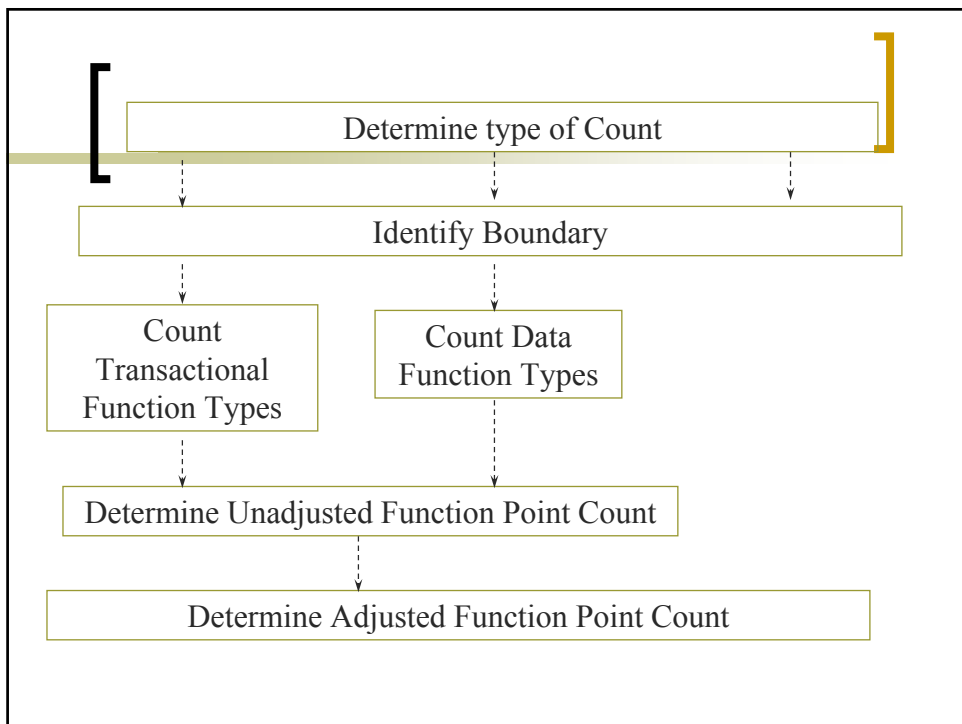
■ Tipos de Función de Datos

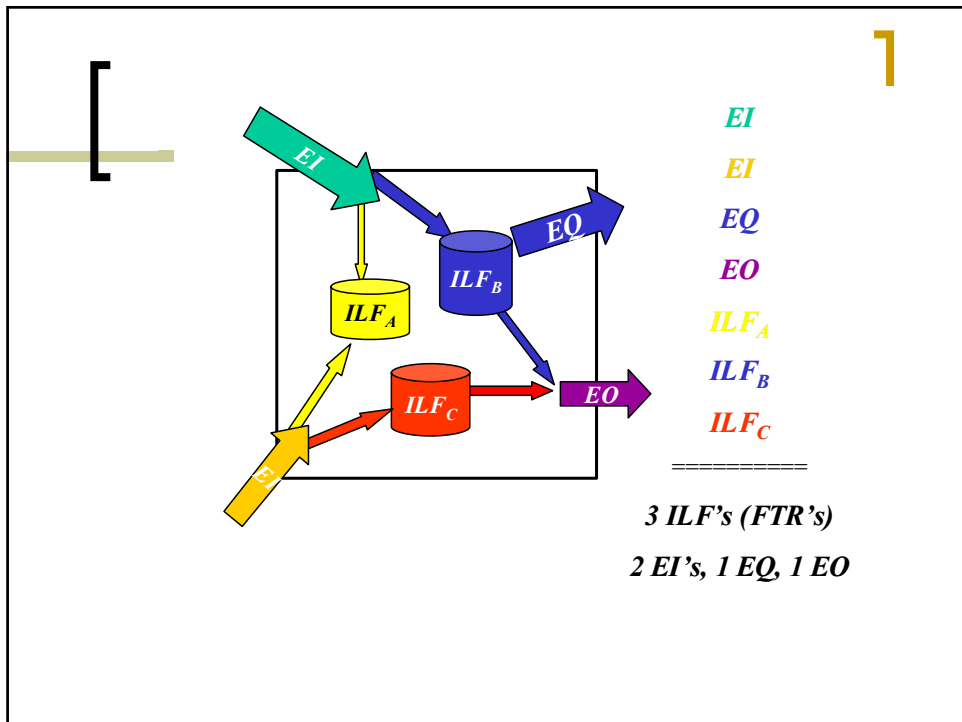
- Internal Logical Files (ILF)
- External Interface Files (EIF)

www.SoftwareMetrics.com

Procedimiento del conteo

- Paso 1 – Determinar el tipo de conteo
- Paso 2 – Establecer la frontera
- Paso 3 – Identificar y evaluar los tipos de función transaccional
- Paso 4 – Identificar y evaluar los tipos de función de datos
- Paso 5 -- Determinar el Valor de Factor de Ajuste
- Paso 6 – Determinar el conteo de los Puntos de Función Ajustados (Adjusted Function Point Count)





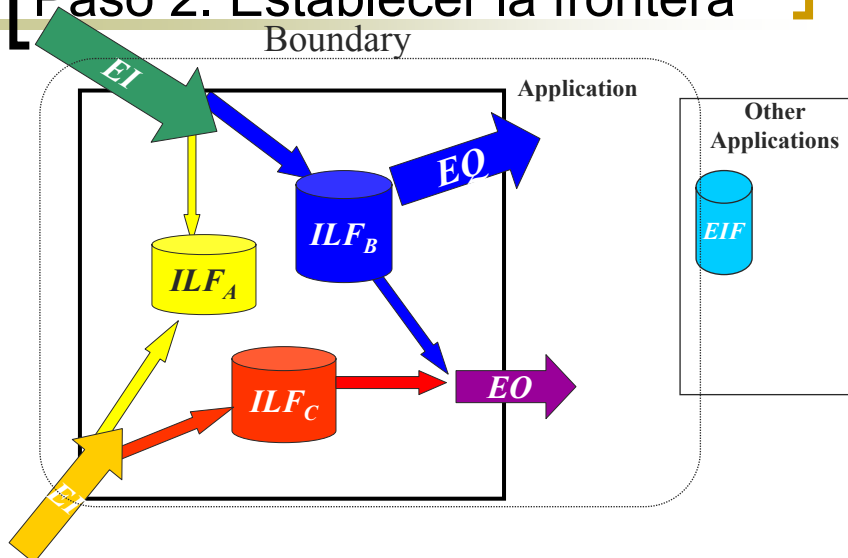
Paso 1. Determinar el tipo de conteo

- De nuevo desarrollo (Development)
- De proyectos de mantenimiento (Enhancement)
- De Aplicaciones (Application or Baseline)

[Paso 2. Establecer la frontera]

- La identificación del alcance, es normalmente establecida en la fase de inicio o planeación del proyecto, y típicamente está documentada en forma de requerimientos de alto nivel, especificaciones funcionales, casos de uso, etc.
- Los límites de la aplicación, establecerán la funcionalidad que será considerada externa a la aplicación. Para establecer dichos límites se tomará el punto de vista del usuario y nunca un criterio técnico o de implementación.

[Paso 2. Establecer la frontera]



Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- Este es uno de los pasos que tienen mayor impacto en el conteo de los Puntos de Función y comienza con la identificación de los Procesos Elementales.
- Un ***proceso elemental*** es la unidad mínima de actividad significativa al usuario (EI, EQ, EO)

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- **EI (External Input)** Si el propósito principal de la función es recibir información para administrar (*crear, modificar, eliminar*) un almacenamiento y cada una de estas funciones es considerada por sí misma una función transaccional.
 - Los datos deben venir de una entrada de pantalla o de otra aplicación.
 - Es un proceso elemental en el cual datos o controles cruzan la frontera de afuera hacia dentro.
 - Si los datos son información de control, éstos no tienen que mantener los ILF.

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- En una función transaccional por ejemplo **Registrar compra de artículos** se requerirá que los datos capturados sean guardados en un archivo de compra, además de actualizar el archivo de inventarios y enviar quizás un mensaje de notificación al cliente del envío. Aunque estos procesos parecieran funciones independientes, no tendrían sentido por si sólo que cada una de ellas fueran una transacción

Ejemplos de External Inputs

- **Datos del negocio:** nombre cliente, dirección, teléfono, etc., que actualizan un archivo.
- **Datos de control:** secuencias de ordenamiento, puerto impresor, o número de copias, que no actualizan un archivo.
- **Reglas del negocio:** número de días antes de que los clientes sean actualizados en un archivo.

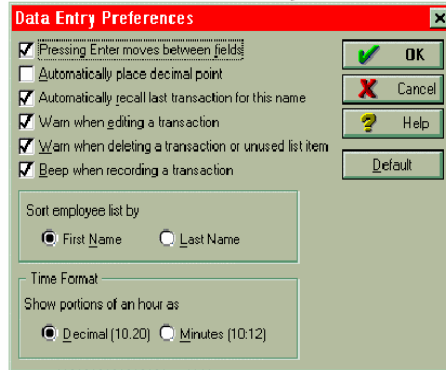
Ejemplos de External Inputs

Business Data: Customer Name, Address, Phone, and so on and so forth.

Control Data:

The data elements are those that invoke the transaction or change the behavior of the application. Each check box represents a data element. Additionally, the sort employee list radio buttons represents one data element as well as the time format radio buttons.

Control information changes or alters the state (or behavior) of the application. Control information specifies how, what, and when data will be processed.



Ejemplos de DET para una External Input

- **Data Input Fields:** nombre del cliente, información del negocio, etc.
- **GUI:** Radio buttons y check boxes
- **Calculated Values** que están almacenados
- **Messages**
 - No se pudo realizar la transacción
 - La información ha sido guardada exitosamente
 - Estás seguro que quieres borrar la información?

DET's para las GUI's

Radio Buttons	1 DET, pero cuando se puede seleccionar más de uno, entonces se cuenta 1 DET por cada uno
Check Boxes	Cada uno es un DET
Command Buttons	<ul style="list-style-type: none"> ■ Si hay un botón por Agregar, Modificar y Borrar de ejecución cada uno cuenta por un DET ■ Si hay un botón Ok y Cancel sólo se cuenta el OK, <i>el cancel no cuenta</i> pues es sólo de navegación ■ Los menús sólo cuentan cuando son resultado de consultar algún archivo, de otra manera son navegacionales.
Result of a Pick List Box	1 DET
Sound Bytes	1 DET
Photographic Images	1 DET

Ejemplos de DET para una External Input

- Si una función transaccional requiere de más de una pantalla o pestaña. Los DET son la suma de todos campos únicos requeridos en cada una de ellas.

The screenshot shows a 'New Customer' dialog box with multiple tabs: 'Customers', 'Address Info', and 'Additional Info'. The 'Address Info' tab is active, showing fields for 'Company Name', 'First Name', 'Last Name', 'Address', and 'Bill To'. The 'Additional Info' tab is also visible, showing fields for 'Credit Limit', 'Sales Tax Information', 'Custom Fields', and 'Opening Balance'. This illustrates how multiple screens or tabs contribute to the total DET count for a transaction.

[Paso 3. Identificar y evaluar los tipos de *Función Transaccional*]

- **EQ (External Inquiries)** el propósito principal es sólo presentar información y no realizar ningún procesamiento adicional.

[External Inquiries (EQ)]

- Un proceso elemental con ambos componentes de entrada y salida que resultan en datos recuperados de uno o mas archivos lógicos internos y/o archivos de interface externa.
- El proceso de entrada no mantiene ningún archivo lógico interno.
- El lado de la salida no contiene datos derivados.

[Ejemplos de EQ]

- Log sobre pantallas
- Solicitud de un registro específico
- Solicitud de ayuda y respuesta
- Reportes (Listado) de información
- Las listas (pick list box) son consideradas como EQ siempre y cuando la información que muestren sea tomada de un archivo

[Ejemplos de DET's para una External Inquiry]

- Input side
 - Un nombre de cliente para buscar sobre él
 - Un click sobre un scroll bar
- Output Side
 - Una lista de clientes por nombre
 - Mostrar un cliente en particular

Ejemplo de External Inquiry

Inventory Report University MousePads Inc.

<i>Item</i>	<i>Description</i>
Hawk Pad	University of Iowa MousePad
JayPad	University of Kansas MousePad
HuskerPad	University of Nebraska MousPad

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- **EO (External Output)** Un proceso simple en cual los **datos derivados** cruzan la frontera de adentro hacia afuera.
- Los datos crean reportes o archivos de salida enviados a otra aplicación. Estos reportes y archivos son creados de uno o más archivos lógicos internos y/o archivos de interfaces externos.

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- Los datos derivados (como cálculos matemáticos, derivación de datos, etc.) son datos que son procesados para ser recuperados y editados de uno o más archivos lógicos internos y/o archivos de interfaces externos.

Ejemplos de EO

- EO's normalmente contienen datos del negocio.
- Reportes de texto
- Reportes Gráficos
- Reportes producidos sobre diferentes medios
- Salidas Electrónicas a otra aplicación
- Las listas (pick list box) son consideradas como EO siempre y cuando el resultado requiera hacer algún procesamiento adicional como cálculos a partir de un archivo

Ejemplos de DET para una External Output

- Mensajes
- Cálculo de valores sobre un reporte
- Valores sobre un reporte que son leídos de un ILF o EIF
- Valores no recursivos
- Generalmente, no se cuentan los encabezados de los reportes (literales) como elementos de datos a menos que sean dinámicos.

Ejemplo de External Output

Inventory Report University MousePads Inc.

<i>Item</i>	<i>Description</i>	<i>Quantity</i>
Hawk Pad	University of Iowa MousePad	1,250
JayPad	University of Kansas MousePad	500
HuskerPad	University of Nebraska MousPad	3,000
Total MousePads		4,750

Ejemplo de External Output

Actividad por día de la semana				
	Día	Aciertos	% del total	Sesiones
1	Dom	1004	8.58%	111
2	Lun	1887	16.13%	201
3	Mar	1547	13.22%	177
4	Mié	1975	16.88%	195
5	Jue	1591	13.6%	191
6	Vie	2271	19.41%	212
7	Sáb	1423	12.16%	148
	Total para los días hábiles	9271	79.25%	976
	Total para los fines de semana	2427	20.74%	259

Paso 3. Identificar y evaluar los tipos de *Función Transaccional*

- La complejidad de las funciones transaccionales depende del número de *Data Element Type* (DET) y *Record Element Type* (FTR).

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- Los **DET**, en las funciones transaccionales, son campos únicos de información que son identificables por el usuario y que entran o salen de la función transaccional.

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- Criterios de conteo aplicables a los DETs en las funciones transaccionales:
 - Para ser contado el DET debe entrar o salir de la aplicación, es decir un campo que sea utilizados internamente y que no entre o salga de la aplicación no es contado como DET.
 - Las etiquetas, nombres de campo, nombres de columna y variables de sistema como (fecha de sistema, número de página, número de columna, etc.) no son contados.

Paso 3. Identificar y evaluar los tipos de ***Función Transaccional***

- En aplicaciones on-line, contamos 1 DET para la capacidad de ejecución sin importar de cuantas formas distintas se pueda ejecutar el proceso elemental.
 - Por ejemplo, si podemos ejecutar la función de “Insertar Empleado” presionando el botón “Guardar” o presionando Ctrl-G o dando click en una opción de menú, sólo contaremos un DET por la capacidad para ejecutar dicha función.

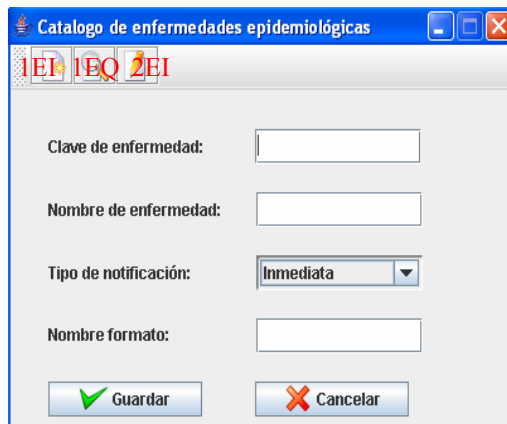
Paso 3. Identificar y evaluar los tipos de función transaccional

- En aplicaciones on-line, contamos 1 DET para la capacidad de envío de mensajes, sin importar cuantos mensajes sean enviados dentro de la misma función transaccional.
 - Por ejemplo, en la función “Insertar Empleado”, podrían hacerse diversas validaciones sobre cada uno de los datos capturados (longitud del campo, formato de la fecha, etc.), sin importar cuantas validaciones sean realizadas, sólo se contará un DET por la capacidad de generar mensajes en dicha función transaccional.

Paso 3. Identificar y evaluar los tipos de función transaccional

- En las listas (pick list box) cuando funcionan como EQ o EO, los elementos mostrados se cuentan cada uno como un DET, además de su capacidad de ejecución.

Qué tipo(s) de función transaccional encuentras?



Agregar

Consultar

Modificar

2EI

1EQ

Cuántos DET son en la función transaccional Agregar?

Los botones de menú son navegacionales y no variables, por lo tanto no cuentan como DET

Un solo DET por la capacidad de ejecución

5

Este botón es considerado sólo de navegación y no un DET

6

Un solo DET por la capacidad de enviar mensajes, no por cada mensaje

Cuántos DET son en la función transaccional Consultar y Modificar?

Consultar

Modificar

La suma en cada transacción sería igual a la de Agregar pues considera los mismos DET, es decir, en total son 6 por cada una.

Cuántos DET son en la función transaccional Agregar caso de influenza?

Sólo hay un botón de ejecución final que inserta todos los datos

La suma de todos los DET incluidos en cada pantalla (apartados)

Paso 3. Identificar y evaluar los tipos de *Función Transaccional*

- Por otro lado los **FTRs**, es un archivo referenciado interno (ILF) o externo (EIF) por una Función Transaccional.
- Con el número de DETs y FTRs y además el tipo de cada función transaccional, se consultará la complejidad

[Cúantos FTR están referenciados
en está Función Transaccional?]



[Tabla para determinar
complejidad de External Input (EI)]

	DET		
FTR	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Tabla para determinar complejidad de External Output (EO)

	DET		
FTR	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Tabla para determinar complejidad de External Inquiry (EQ)

	DET		
FTR	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- En este paso, primero debemos identificar los almacenamientos lógicos (también conocidos como entidades de información), en cuanto a que sean totalmente independientes y que tengan un significado para el usuario. Por ejemplo, índices, tablas temporales son considerados almacenamientos técnicos creados con propósitos de implementación y no contribuyen al tamaño funcional de la aplicación.

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- Catálogos que contienen sólo un ID y una descripción, son almacenamientos considerados como “Code Table” y tampoco tendrán una contribución en el tamaño funcional.

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- El punto clave para agrupar los almacenamientos lógicos, es verificando la dependencia entre ellos. Una pregunta clave, es respondernos si dicho almacenamiento puede existir de manera aislada, o si para tener significado debe formar parte de otro almacenamiento.

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- Por ejemplo: Si tenemos la tabla de empleado, y la tabla telefonos_empleado, **ambas son consideradas como un solo almacenamiento lógico**, ya que telefonos_empleado no tiene significado por si sola. Otro caso puede ser la de PacienteNacional y Teléfono



Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- Otra forma de ver esta independencia es preguntarnos: Si desaparece un registro de la tabla de empleado ¿tiene sentido para el negocio conservar el registro asociado en telefonos_empleado?. Si la respuesta es positiva, entonces se considera que cada entidad puede existir de manera **independiente** y por lo tanto serán dos almacenamientos lógicos. Por el contrario, si la respuesta es negativa, los almacenamientos no existen de manera independiente, y por lo tanto son un solo almacenamiento lógico.

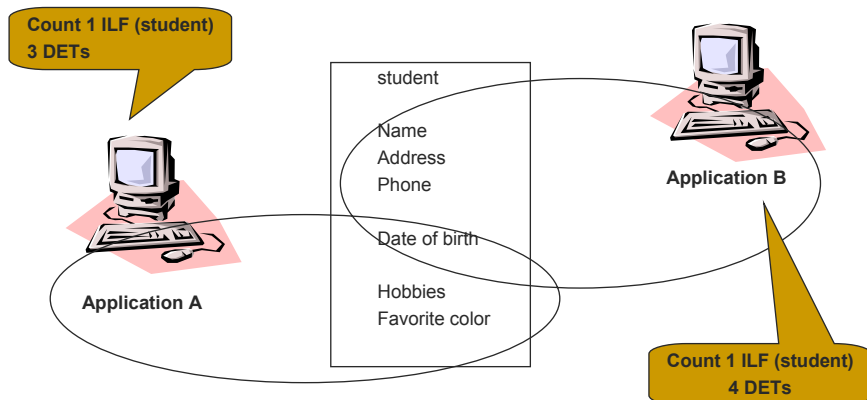
Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- **ILF (Internal Logical File)** el almacenamiento es mantenido por al menos una función transaccional dentro de la aplicación que contamos
- **EIF (External Logical File)** el almacenamiento es únicamente referenciado y no mantenido por la aplicación que estamos contando.

Ejemplos de ILF

- **Datos de negocio** (nombre cliente, dirección)
- **Datos de Control** (color, copias, puerto impresor)
- **Reglas basadas en datos** (criterios de pago, zona de impuestos)

DET's & ILF's compartidos



Paso 4. Identificar y evaluar los tipos de *Función de Datos*

- La complejidad de las Funciones de Datos de los ILF o EIF depende del número de DETs y RETs.
- Los **DET**s son los campos de datos únicos e identificables por el usuario

Enfermedad
ClaveEnfermedad
NombreEnfermedad
NombreFormato
TipoNotificacion

4 DETS

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

Los DET llave se cuentan una sola vez, al ser un solo archivo

Paciente Nacional	Telefono
ClavePaciente	ClavePaciente
NumDeExpediente	DescripcionTelefono
ApellidoPaterno	Telefono
ApellidoMaterno	
Nombre	
FechaDeNacimiento	
LugarDeNacimiento	
Calle	
Numero	
EdadAnios	
EdadMeses	
EdadDias	
Sexo	
EstadoCivil	
Escolaridad	
ClaveOcupacion	
JurisdiccionSanitaria	
UnidadDeAdscripcion	
ClaveColonia	

19 DET Paciente Nacional

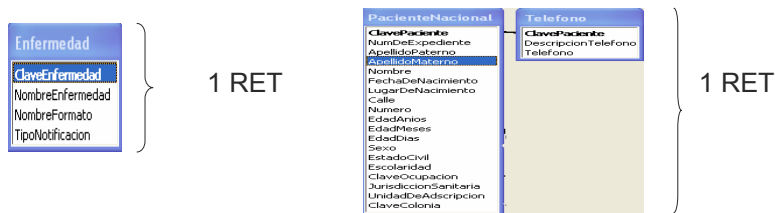
2 DET Teléfono

21 DET

Los DET que son llave foránea se cuentan cuando no se consideran un solo archivo

Paso 4. Identificar y evaluar los tipos de *Función de Datos*

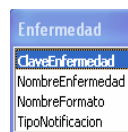
- Los **RET**, son los grupos de datos en el mismo almacenamiento y que son identificables por el usuario, cuando en el almacenamiento no hay subgrupos de datos, se considera que el almacenamiento tiene un sólo RET.



Qué tipo y cuántos archivos están referenciados?

The screenshot shows the 'Catalogo de enfermedades epidemiológicas' window. It contains the following fields and controls:

- Clave de enfermedad:** A text input field.
- Nombre de enfermedad:** A text input field.
- Tipo de notificación:** A dropdown menu with 'Inmediata' selected.
- Nombre formato:** A text input field.
- Buttons:** 'Guardar' (with a green checkmark icon) and 'Cancelar' (with a red X icon).



1 ILF con 4 DETS

Tabla para determinar complejidad de funciones de datos (ILF)

	DET		
RET	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Tabla para determinar complejidad de funciones de datos (EIF)

	DET		
RET	1 to 19	20 - 50	51 or More
1 RET	Low (5)	Low(5)	Average (7)
2 to 5 RET	Low (5)	Average (7)	High (10)
6 or More RET	Average (7)	High (10)	High (10)

Tabla para calcular la complejidad PF

Complejidad Funcional				
	Baja	Media	Alto	Total
External Input	X 3=	X 4 =	X 6 =	
External Output	X 4=	X 5 =	X 7 =	
External Inquiry	X 3=	X 4 =	X 6 =	
Internal Logical File	X 7=	X 10 =	X 15 =	
External Logical File	X 5=	X 7 =	X 10 =	
Puntos de Función no Ajustados				
Multiplicado por el Valor de Factor de Ajuste				
Total Puntos de Función Ajustados				

Repasando el ejercicio del Módulo de Enfermedades

- **Paso 1 – Identificar el Tipo de conteo**
 - Nuevo Desarrollo
- **Paso 2- Identificando la frontera**
 - Las funcionalidades que se van analizar son las correspondientes a SICE, en particular la del mantenimiento de catálogo de enfermedades

[Repasando el ejercicio]

■ Paso 3 – Identificar y evaluar los tipos de función transaccional

Nombre	Tipo	DET	FTR	Complejidad
Agregar enfermedad	EI	6	1	Baja (3)
Modificar enfermedad	EI	6	1	Baja (3)
Consultar enfermedad	EQ	6	1	Baja (3)

[Repasando el ejercicio]

■ Paso 4 – Identificar y evaluar los tipos de función de datos

- Para la funciones transaccionales de agregar, modificar y consultar el archivo afectado es el de Enfermedad

Nombre	Tipo	DET	RET	Complejidad
Enfermedad	ILF	4	1	Baja (7)

Tabla para calcular la complejidad PF de Manejo de Enfermedades

Complejidad Funcional				
	Baja	Media	Alto	Total
External Input	2 X 3 =	X 4 =	X 6 =	6
External Output	X 4 =	X 5 =	X 7 =	
External Inquiry	1 X 3 =	X 4 =	X 6 =	3
Internal Logical File	1 X 7 =	X 10 =	X 15 =	7
External Logical File	X 5 =	X 7 =	X 10 =	
Puntos de Función no Ajustados				16
Multiplicado por el Valor de Factor de Ajuste				
Total Puntos de Función Ajustados				

Caso Práctico

- Resolver Caso de Influenza

Paso 5 -- Determinar el Valor de Factor de Ajuste

- Este paso ayuda a determinar un valor de ajuste que puede aumentar o disminuir el valor de los puntos de función en un +/- 35%
- Las características evaluadas son 14 y refieren a requerimientos o restricciones no funcionales.

Características Generales del Sistema (GSC)

- Si la calificación de estos factores ambientales es mínimo, el valor de ajuste será de 0.65. Por el contrario si los factores ambientales son los más complicados, el valor de ajuste será de 1.35
- Cuando se calculan PF para realizar estimaciones mediante el uso de herramientas estadística, se alimentan únicamente los PF no ajustados.

[Escala de valores de GSC]

- 0 Not present or no influence
- 1 Incidental influence
- 2 Moderate influence
- 3 Average influence
- 4 Significant influence
- 5 Strong influence throughout

[Valor del Factor de Ajuste(Value Adjustment Factor)]

- Una vez que todas las características han sido contestadas asignándoles un valor, son tabuladas utilizando la Ecuación de Valor de Ajuste del IFPUG *Value Adjustment Equation* (VAF) --

$$\text{VAF} = 0.65 + [(\sum C_i) / 100]$$

Where:

C_i = grado de influencia de cada GSC

i = representa de cada una de las GSC (1 a la 14)
representing each GSC.

\sum = es la suma de las 14 GSC.

Valor del Factor de Ajuste (Value Adjustment Factor)

No.	General System Characteristic	Value
1	Data communications	
2	Distributed data processing	
3	Performance	
4	Heavily used configuration	
5	Transaction rate	
6	On-Line data entry	
7	End-user efficiency	
8	On-Line update	
9	Complex processing	
10	Reusability	
11	Installation ease	
12	Operational ease	
13	Multiple sites	
14	Facilitate change	
	Value Adjustment Factor	$\sum C_i$

GSC's at a Glance

GSC		Description
1.	Data Communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
2.	Distributed DP	How are distributed data and processing functions handled?
3.	Performance	Did the user require response time or throughput?
4.	Heavily Used Config.	How heavily used is the current hardware platform where the application will be executed?
5.	Transaction Rate	How frequently are transactions executed daily, weekly, monthly, etc.?
6.	Online Data Entry	What percentage of the information is entered online
7.	End-User Efficiency	Was the application designed for end-user efficiency?
8.	Online Update	How many ILF's are updated by On-Line transaction?
9.	Complex Processing	Does the application have extensive logical or mathematical processing?

GSC's at a Glance - continued

	GSC	Description
10.	Reusability	Was the application developed to meet one or many user's needs?
11.	Installation Ease	How difficult is conversion and installation?
12.	Operational Ease	How effective and/or automated are start-up, back up, and recovery procedures?
13.	Multiple Sites	Was the application specifically designed to be installed at multiple sites for multiple organizations?
14.	Facilitate Change	Was the application specifically designed to facilitate change?

Data Communications GSC

Score As	Descriptions to Determine Degree of Influence
0	Application is pure batch processing or a standalone PC.
1	Application is batch but has remote data entry <i>or</i> remote printing.
2	Application is batch but has remote data entry <i>and</i> remote printing.
3	Application includes online data collection or TP (teleprocessing) front end to a batch process or query system.
4	Application is more than a front-end, but supports only one type of TP communications protocol.
5	Application is more than a front-end, and supports more than one type of TP communications protocol.

Example Rating:

An application that allows query of application via a web based solution and local access would receive a value of 3.

An application that allows for the update of ILF's via the Internet and local update would receive a value of a 5.

Distributed DP GSC

Score As	Descriptions to Determine Degree of Influence
0	Application does not aid the transfer of data or processing function between components of the system.
1	Application prepares data for end user processing on another component of the system such as PC spreadsheets and PC DBMS
2	Data is prepared for transfer, then is transferred and processed on another component of the system (not for end-user processing).
3	Distributed processing and data transfer are online and in one direction only.
4	Distributed processing and data transfer are online and in both directions.
5	Processing functions are dynamically performed on the most appropriate component of the system.

Example Rating:

- * Copying files from a mainframe to a local PC or copy files from an Internet or intranet would receive a value of 2.
- * Reading via a client or via Internet or intranet would receive a value of 3.
- * Reading and updating via Internet or intranet would receive a value of 4.
- * Depending on available resources, the application processes either local, on server, on intranet or Internet application would receive a value of 5.

Performance GSC

Score As	Descriptions to Determine Degree of Influence
0	No special performance requirements were stated by the user.
1	Performance and design requirements were stated and reviewed but no special actions were required.
2	Response time or throughput is critical during peak hours. No special design for CPU utilization was required. Processing deadline is for the next business day.
3	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
4	In addition, stated user performance requirements are stringent enough to require performance analysis tasks in the design phase.
5	In addition, performance analysis tools were used in the design, development, and/or implementation phases to meet the stated user performance requirements.

Heavily Used Config. GSC

Score As	Descriptions to Determine Degree of Influence
0	No explicit or implicit operational restrictions are included. Some security or timing considerations are included.
1	Operational restrictions do exist, but are less restrictive than a typical application.
2	No special effort is needed to meet the restrictions.
3	Specific processor requirements for a specific piece of the application are included.
4	Stated operation restrictions require special constraints on the application in the central processor or a dedicated processor.
5	In addition, there are special constraints on the application in the distributed components of the system.

Transaction Rate GSC

Score As	Descriptions to Determine Degree of Influence
0	No peak transaction period is anticipated.
1	Peak transaction period (e.g., monthly, quarterly, seasonally, annually) is anticipated.
2	Weekly peak transaction period is anticipated.
3	Daily peak transaction period is anticipated.
4	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase.
5	High transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks and, in addition, require the use of performance analysis tools in the design, development, and/or installation phases.

[Online Data Entry GSC]

Score As	Descriptions to Determine Degree of Influence
0	All transactions are processed in batch mode.
1	1% to 7% of transactions are interactive data entry.
2	8% to 15% of transactions are interactive data entry.
3	16% to 23% of transactions are interactive data entry.
4	24% to 30% of transactions are interactive data entry.
5	More than 30% of transactions are interactive data entry.

[End User Efficiency GSC]

End User Design Components:

- Navigational aids (for example, function keys, jumps, dynamically generated menus)
- Menus
- Online help and documents
- Automated cursor movement
- Scrolling
- Remote printing (via online transactions)
- Assigned function keys
- Batch jobs submitted from online transactions
- Cursor selection of screen data
- Heavy use of reverse video, highlighting, colors underlining, and other indicators
- Hard copy user documentation of online transactions
- Mouse interface
- Pop-up windows.
- As few screens as possible to accomplish a business function
- Bilingual support (supports two languages; count as four items)
- Multilingual support (supports more than two languages; count as six items) .

[End User Efficiency GSC]

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	One to three of the above.
2	Four to five of the above.
3	Six or more of the above, but there are no specific user requirements related to efficiency.
4	Six or more of the above, and stated requirements for end-user efficiency are strong enough to require design tasks for human factors to be included (for example, minimize key strokes, maximize defaults, use of templates).
5	Six or more of the above, and stated requirements for end-user efficiency are strong enough to require use of special tools and processes to demonstrate that the objectives have been achieved.

[Online Update GSC]

Score As	Descriptions to Determine Degree of Influence
0	None.
1	Online update of one to three control files is included. Volume of updating is low and recovery is easy.
2	Online update of four or more control files is included. Volume of updating is low and recovery easy.
3	Online update of major internal logical files is included.
4	In addition, protection against data lost is essential and has been specially designed and programmed in the system.
5	In addition, high volumes bring cost considerations into the recovery process. Highly automated recovery procedures with minimum operator intervention are included.

Complex Processing GSC

Complex Processing Components:

- Sensitive control.
- Extensive logical processing.
- Extensive mathematical processing.
- Much exception processing.
- Complex processing to handle multiple I/O.

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	Any one of the above.
2	Any two of the above.
3	Any three of the above.
4	Any four of the above.
5	All five of the above.

Reusability GSC

Score As	Descriptions to Determine Degree of Influence
0	No reusable code.
1	Reusable code is used within the application.
2	Less than 10% of the application considered more than one user's needs.
3	Ten percent (10%) or more of the application considered more than one user's needs.
4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
5	The application was specifically packaged and/or documented to ease re-use, and the application is customized for use by means of user parameter maintenance.

Installation Ease GSC

Score As	Descriptions to Determine Degree of Influence
0	No special considerations were stated by the user, and no special setup is required for installation.
1	No special considerations were stated by the user <i>but</i> special setup is required for installation.
2	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of conversion on the project is not considered to be important.
3	Conversion and installation requirements were stated by the user, and conversion and installation guides were provided and tested. The impact of conversion on the project is considered to be important.
4	In addition to 2 above, automated conversion and installation tools were provided and tested.
5	In addition to 3 above, automated conversion and installation tools were provided and tested.

Operational Ease GSC

Score As	Descriptions to Determine Degree of Influence
0	No special operational considerations other than the normal back-up procedures were stated by the user.
1-4	<ul style="list-style-type: none"> * One, some, or all of the following items apply to the application. Select all that apply. Each item has a point value of one, except as noted otherwise. * Effective start-up, back-up, and recovery processes were provided, but operator intervention is required. * Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items). * The application minimizes the need for tape mounts. * The application minimizes the need for paper handling.
5	The application is designed for unattended operation. Unattended operation means <i>no operator intervention</i> is required to operate the system other than to start up or shut down the application. Automatic error recovery is a feature of the application.

Multiple Sites GSC

Score As	Descriptions to Determine Degree of Influence
0	User requirements do not require considering the needs of more than one user/installation site.
1	Needs of multiple sites were considered in the design, and the application is designed to operate only under identical hardware and software environments.
2	Needs of multiple sites were considered in the design, and the application is designed to operate only <i>under similar</i> hardware and/or software environments.
3	Needs of multiple sites were considered in the design, and the application is designed to operate <i>under different</i> hardware and/or software environments.
4	Documentation and support plan are provided and tested to support the application at multiple sites and the application is as described by 1 or 2.
5	Documentation and support plan are provided and tested to support the application at multiple sites and the application is as described by 3.

Facilitate Change GSC

Facilitate Change Characteristics:

- Flexible query and report facility is provided that can handle simple requests.
- Flexible query and report facility is provided that can handle requests of average complexity.
- Flexible query and report facility is provided that can handle complex requests.
- Business control data is kept in tables that are maintained by the user with online interactive processes, but changes take effect only on the next business day.
- Business control data is kept in tables that are maintained by the user with online interactive processes, and the changes take effect immediately (count as two items).

[Facilitate Change GSC]

Score As	Descriptions to Determine Degree of Influence
0	None of the above.
1	Any one of the above.
2	Any two of the above.
3	Any three of the above.
4	Any four of the above.
5	All five of the above.

[Paso 5 -- Determinar el Valor de Factor de Ajuste]

■ **$FP = UAF * VAF$**

○ Donde:

- UAF = Unadjusted Function Points
- VAF = Value Adjustment Factor

Tabla para calcular la complejidad PF de Manejo de Enfermedades

Si consideramos el mínimo valor de ajuste para SICE entonces el Total de PF sería el siguiente:

Complejidad Funcional				
	Baja	Media	Alto	Total
External Input	2 X 3 =	X 4 =	X 6 =	6
External Output	X 4 =	X 5 =	X 7 =	
External Inquiry	1 X 3 =	X 4 =	X 6 =	3
Internal Logical File	1 X 7 =	X 10 =	X 15 =	7
External Logical File	X 5 =	X 7 =	X 10 =	
Puntos de Función no Ajustados				16
Multiplicado por el Valor de Factor de Ajuste				.65
Total Puntos de Función Ajustados				10.4

Calculando Adjusted FP para Desarrollo

- Cuando se trata de una aplicación que va a reemplazar a otra, los datos deben ser convertidos, de ahí que alguna veces se necesite hacer un desarrollo para asistir la conversión.
- Por lo tanto se propone agregar estos puntos de función al desarrollo total del proyecto.

Calculando PF para Nuevo Desarrollo + PF de Aplicaciones base

■ $DFP = (UFP + CFP) * VAF$

○ Donde:

- DFP total de PF para el desarrollo del proyecto
- UFP es el conteo total de los PF no ajustados
- CFP son los puntos de función agregados por la aplicación de conversión
- VAF es el valor del factor de ajuste (asume que es = 1)

Estimación

- *La estimación del costo, del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.*

[Problemática en la estimación de proyectos software]

1. No existe un modelo de estimación universal o una formula que pueda ser usada para todas las organizaciones.
2. Hay muchas personas implicadas en los proyectos que necesitan de estimaciones.
3. La utilidad de una estimación también dependerá de la etapa de desarrollo en la que nos encontremos.

[Problemática en la estimación de proyectos software]

4. Generalmente, la estimación se hace superficialmente, sin apreciar el esfuerzo requerido para hacer un trabajo.
5. Las estimaciones claras, completas y precisas son difíciles de formular, especialmente al inicio del proyecto.
6. La rapidez con la que cambia la tecnología de la información y las metodologías de desarrollo de software son un problema para la estabilización del proceso de estimación.

Definiciones asociadas a las estimaciones

Proyecto Cualquier esfuerzo planeado que tiene productos a ser generados, compromisos de entrega preestablecidos y limitaciones de recursos y presupuesto.

En general tiene las siguientes características:

- Fecha de inicio y fin
- Objetivos
- Uso de recursos restringidos a límites
- Una estructura jerárquica de actividades

Tamaño medida de que tan grande es el producto

Esfuerzo número de horas necesarias para completar una actividad.

Indicadores de Productividad

Tasa de entrega de proyecto = FP/h

Mide la tasa de entrega de proyectos.

FP es el valor de puntos de función no ajustados de un proyecto

h indica todo el esfuerzo aplicado al proyecto.

Tasa de soporte = h/FP período

Refleja el trabajo realizado sobre las aplicaciones sin crear nuevas funcionalidades (reparación de defectos, conversiones o mantenimiento preventivo).

h es el tiempo invertido en estas actividades durante un determinado período.

FP es el tamaño del proyecto en PFs. Esta métrica se calcula anualmente o trimestralmente.

[Indicadores de Calidad]

Tasa de Costo de Reparación = Costo / FP

Cuantifica el costo de reparar los defectos del software excluyendo los costos de prevención y detección de dichos errores. Se debe calcular mensualmente por un periodo no inferior a los primeros seis meses después de la implementación del proyecto.

costo es el tiempo total de reparación en horas multiplicado por la tasa de reparación por hora del personal.

FP son los PFs totales de la aplicación que está siendo reparada.

[Indicadores de Calidad]

Tasa de Estabilidad = 1 - (#cambios / FP)

Proporciona un indicador de que tan bien una mejora o una aplicación cumplió las expectativas del usuario.

#cambios son los cambios solicitados durante el primer trimestre (90 días) después de la implementación.

FP es el tamaño de la aplicación en PFs.

[Indicadores de Calidad]

Tasa de Defectos = #defectos / FP

Relaciona el numero de defectos con el tamaño en PFs de una aplicación.

#defectos es el total de incidencias en las que la aplicación no cumplió las especificaciones.

FP son los PFs de la aplicación mantenida. Se debe calcular mensualmente solo durante los primeros seis meses después de la implementación del proyecto.

[Indicadores de Calidad]

Destreza en Testeo = #defectos/FP

Es la tasa de defectos durante la fase de pruebas. Una tasa alta indica o bien poca calidad o unos procedimientos de pruebas muy efectivos. Se debe comparar con la tasa de defectos observada en la aplicación después de la implementación. Se calcula para cada ciclo de pruebas.

Fiabilidad = 1 - (#fallos/FP)

Considera el número de *fallos* de la aplicación desde que se puso en marcha. Múltiples fallos causados por el mismo defecto se cuentan repetidamente. **FP** es el total de PFs de la aplicación que está siendo medida. Se debe calcular mensual o trimestralmente.

ISBSG International Software Benchmarking Standards Group, Australia.

■ Método de Estimación (Ecuaciones):

- **Tasa de Entrega** (persona hora por PF)
- **Esfuerzo** (persona hora)
- **Duración** (horas transcurridas)
- **Tiempo de Entrega** (FP entregado por mes) de un proyecto.

ISBSG

- Se basa en los datos del repositorio ISBSG.
- Las ecuaciones pueden usar el tamaño del proyecto, el tamaño máximo del equipo de desarrollo o la combinación de ambos.
- Las ecuaciones son proporcionadas para:
 - Plataforma de desarrollo (mainframe, mid-range, PC y multiplataforma)
 - Tipo de lenguaje (3GL, 4GL y generador de aplicaciones)
 - Combinación de plataforma y lenguaje.

[ISBSG]

- Ejemplo: Proyecto software desarrollado en plataforma mainframe con 460 PFs.

Tasa de Entrega (TE)	$TE = 14.35 \times \text{Tamaño}^{-0.072}$	= 9.2 h/PF
Esfuerzo Proy(PWE)	$PWE = C \times \text{Tamaño}^E$	= 4,245 h
Duración (D)	$D = 1.081 \times \text{Tamaño}^{0.344}$	= 8.9 meses
Tiempo de Entrega (TE)	$TE = 0.925 \times \text{Tamaño}^{0.410}$	= 51.63 PF/mes

PWD (Project Work Effort, Hour)

[ISBSG]

Esfuerzo por plataforma

- **$PWE = C \times \text{Tamaño}^E$**

Plataforma	Lenguaje	C	E	Num proyectos	Ratio fiabilidad
Mainframe		14.35	0.928	384	0.631
Mini		14.89	0.900	108	0.540
PC		3.78	1.016	116	0.473
	3GL	13.34	0.973	285	0.633
	4GL	10.07	0.899	290	0.492
	ApGen	39.54	0.716	46	0.672

ISBSG

Esfuerzo por plataforma

$$PWE = C \times \text{Tamaño}^E$$

Plataforma	Lenguaje	C	E	Num proyectos	Ratio fiabilidad
Mainframe	3GL	15.10	0.950	193	0.650
Mainframe	4GL	12.47	0.896	112	0.618
Mainframe	ApGen	33.81	0.753	42	0.714
Mini	3GL	7.24	1.068	40	0.727
Mini	4GL	18.07	0.851	58	0.483
PC	3GL	0.587	1.390	10	0.955
PC	4GL	3.38	0.974	81	0.519

Estimación del Esfuerzo

ISBSG

Para proyectos en plataforma PC y lenguajes 4GL como los lenguajes de script utilizados en el desarrollo de aplicaciones Web (o lenguajes como Visual Basic, ASP, y Java), la ecuación es:

$$\text{Esfuerzo} = 3,38 * (\text{Tamaño en PF})^{0.974}$$

Suponiendo que son 137 PF

$$\text{Esfuerzo} = 3,38 * (137)^{0.974}$$

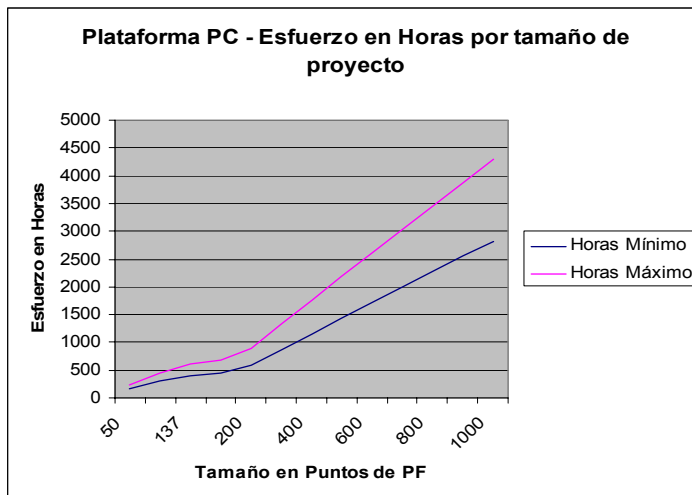
$$\text{Esfuerzo} = 407,45 \text{ horas de trabajo} = 2,9 \text{ horas/PF}$$

Estas **407,45 horas** de trabajo equivalen a aproximadamente **2,9 horas** de trabajo por puntos de función (h/PF).

Interpretación del esfuerzo por rangos de PF

Tamaño PF	Rango de Esfuerzo en Horas	
	Mínimo	Máximo
50	152,66	231,88
100	299,86	455,49
137	407,46	618,93
150	445,07	676,06
200	589,01	894,70
300	874,24	1327,98
400	1156,97	1757,44
500	1437,85	2184,09
600	1717,26	2608,52
700	1995,46	3031,10
800	2272,62	3452,11
900	2548,88	3871,74
1000	2824,34	4290,17

Interpretación del esfuerzo por rangos de PF



Estimación del Esfuerzo considerando el tamaño del equipo de proyecto ISBSG

- Teniendo en cuenta que el proyecto está siendo desarrollado por dos personas, la siguiente Ecuación puede ser aplicada:

$$\text{Esfuerzo} = 9,484 * (\text{Tamaño en PF})^{0.585} * (\text{TamEquipo})^{1.038}$$

$$\text{Esfuerzo} = 9,484 * (137)^{0.585} * (2)^{1.038}$$

$$\text{Esfuerzo} = 346,28 \text{ horas de trabajo} = 2,5 \text{ horas/PF}$$

Estimación Duración ISBSG

- La estimación de la Duración en plataforma de PC es:

$$\text{Duración} = 0.38 * (\text{Esfuerzo})^{0.37}$$

$$\text{Duración} = 0.38 * (346,25 \text{ horas})^{0.37}$$

$$\text{Duración} = 3.3 \text{ meses}$$

[SPR (Caper Jones) Software Productivity Research Inc., USA]

- Métodos de Estimación
 - Repositorio con 8.000 proyectos
 - Ecuaciones para estimar esfuerzo (persona hora), duración de un proyecto.
 - Tablas de Lenguajes de Programación
 - Tabla 1: Relación entre Nivel del Lenguaje y Productividad
 - Tabla 2: Lenguaje de Programación y Niveles
 - Herramienta: SPR KnowledgePLAN™
 - Soporta la herramienta Function Point WORKBENCH™ de la Charismatek Software Metrics.

[SPR (Caper Jones)]

- Ecuaciones:
 - **Estimación Indicativa o 'Ball-park':** técnica de Macro-Estimación que se utiliza habitualmente en situaciones de falta de información sobre el proyecto.
 - Ecuación para determinar el esfuerzo de desarrollo de un proyecto:

$$Esfuerzo = \left(\frac{TamañoPF}{150} \right) * TamañoPF^{0.4}$$

[Estimación del Esfuerzo SPR (Caper Jones)]

Para un proyecto de 137 PF, el esfuerzo sería:

$$Esfuerzo = \left(\frac{\text{Tamaño en PF}}{150} \right) * \text{Tamaño en PF}^{0,4}$$

$$Esfuerzo = \left(\frac{137}{150} \right) * 137^{0,4}$$

$$Esfuerzo = 6,4 \text{ meses de trabajo}$$

Estos **6,41 meses** de trabajo suponen unas **896 horas** de desarrollo, suponiendo una jornada laboral de **35 horas** semanales. Es decir, una única persona trabajando en el desarrollo del proyecto debería invertir **896 horas** hasta su finalización.

[SPR (Caper Jones)]

■ Ecuaciones:

- Ecuación para determinar la duración de un proyecto:

$$Duración = \text{Tamaño PF}^{0,4}$$

[SPR (Caper Jones)]

- Para el proyecto de 137 PF la duración sería:

$$\text{Duración} = (\text{Tamaño en PF})^{0,4}$$

$$\text{Duración} = (137)^{0,4}$$

$$\text{Duración} = 7,1 \text{ meses}$$

[SPR (Caper Jones)]

Tabla 1: Relación entre Nivel del Lenguaje y Productividad

Nivel del Lenguaje	Productividad Media por persona mes
1 – 3	5 a 10 PFs
4 – 8	10 a 20 PFs
9 – 15	16 a 23 PFs
16 – 23	15 a 30 PFs
24 – 55	30 a 50 PFs
Arriba de 55	40 a 100 PFs

SPR (Caper Jones)

Tabla 2: Lenguaje de Programación y Niveles (Resumen)

■ La relación entre líneas de código y puntos de función depende del lenguaje de programación en que se implementan el software y la calidad del diseño.

Lenguaje	Nivel	Horas / PF	Promedio de sentencias del lenguaje por PFs
2GL	3.00		107
3GL	4.00	17.60	80
4GL	16.00	7.65	20
5GL	70.00		5
Java	6.00	11.73	53
Visual Basic 5.0	11.00	8.80	29

Líneas de código y PF por otra consultora

<http://www.qsm.com/FPGearing.html>

*** Indicates languages which were present before but have been adjusted due to new information

Language	QSM SLOC/FP Data				David Consulting Data
	Avg	Median	Low	High	
Access	35	38	15	47	-
Ada	154	-	104	205	-
Advantage	38	38	38	38	-
APS	86	83	20	184	-
ASP	69	62	32	127	-
Assembler**	172	157	86	320	575 Basic/400 Macro
C **	148	104	9	704	225
C++ **	60	53	29	178	80
C#	59	59	51	66	-
Clipper	38	39	27	70	60
COBOL **	73	77	8	400	175
Ideal	66	52	34	203	-
IEF/Cool:Gen	38	31	10	180	-
Informix	42	31	24	57	-
J2EE	61	50	50	100	-
Java**	60	59	14	97	80
JavaScript**	56	54	44	65	50
JCL**	60	48	21	115	400
JSP	59	-	-	-	-

Estimación de la Duración (The David Consulting Group)

- Tabla 1: PFs / persona mes y Horas por PF en diferentes plataformas

Ciente-Servidor	18	9.8
Mainframe	12	14.7
Web	27	6.5
E-Business Web	16	11.0
Network Systems	24	7.3
Paquetes Software	21	8.4
Data Warehouse	8	22.0

Según la propuesta de *David Consulting*, una persona al mes puede producir **27 FPs** en el desarrollo de aplicaciones web.

Lo que supone según el cálculo realizado $137/27 = 5.07$ meses por una persona.

Considerando una jornada laboral de **35 horas**, las horas necesarias para desarrollar el proyecto Web son:
 $5.07 \times 140 = 709,8$ horas = **88,72 días** de trabajo

Referencias

- Pressman, R. Ingeniería de software: un enfoque práctico. 6ta Ed., McGrawHill, 2006.
- Aquiles, S. Caso Práctico Puntos de Función, Revista SoftwareGuru Conocimiento en Práctica.
- www.SoftwareMetrics.Com, 2007
- Abrahão, S. On the Functional Size Measurement of Object Oriented