

UNIVERSIDAD DE COSTA RICA

**ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA**

Metodología para medir el proceso de software

Profesora: Gabriela Salazar Bermúdez

Versión 2

Año - 2006

TABLA DE CONTENIDO

CAPÍTULO 1	3
Metodología de Análisis de Puntos de Función	3
Paso 1. Determinar el tipo de conteo de puntos de función.	4
Paso 2. Identificar el alcance del conteo y la frontera de la aplicación.	5
Paso 3. Identificar todas las funciones de datos y su complejidad y determinar los puntos de función desajustados.	6
Paso 4. Identificar todas las funciones transaccionales y su complejidad y determinar los puntos de función desajustados.	13
Paso 5. Determinar el Factor de Ajuste de Valor (FAV) a través de las 14 Características Generales	31
Paso 6. Calcular el Punto de Función de Ajuste Final.	39
CAPÍTULO 2	40
Casos Prácticos de Estimación de Puntos de Función	40
Caso 1: Sistema de Planilla	40
Caso 2: Sistema de Administración de Contrataciones	51
Caso 3: Sistema de Administración de Requerimientos	57
Caso 4: Sistema de Control de Horas	67
CAPÍTULO 3	81
Estimación del costo y el esfuerzo con base en los Puntos de Función estimados	81
Procedimiento para estimar costo y esfuerzo cuando no contamos con datos históricos.	81
Procedimiento para estimar costo y esfuerzo cuando contamos con datos históricos.	84
REFERENCIAS BIBLIOGRÁFICAS	86

Resumen

El Análisis de Puntos de Función (FPA, en inglés “Function Point Analysis”) fue desarrollado por A. J. Albrecht y sus colegas de IBM a mediados de 1970, con su primera presentación pública en 1978. El propósito era solucionar algunos de los problemas asociados con el cálculo del tamaño en Líneas de Código (LOC, en inglés “Lines of code”) y las medidas de productividad, especialmente por las diferencias en los cálculos de LOC que resultaban de los diferentes niveles de lenguajes que se utilizaban. [1, pág 95] A él le interesaba medir la funcionalidad del software desde el punto de vista del usuario, independientemente de su implementación. Él introdujo los Puntos de Función como una medida del tamaño de una aplicación desde un punto de vista funcional o de usuario.

En 1984 la organización responsable de definir y aclarar las reglas para el cálculo de Puntos de Función fue dada a “International Function Point Users Group” (IFPUG). IFPUG además de aclararlas, modificó el método original de Albrecht, publicando el “Counting Practices Manuals” que provee los estándares para la aplicación del cálculo de los Puntos de Función. [1, pág. 456]

A través de los años la métrica de Puntos de Función ha ganado aceptación como una medida de productividad clave. El principal beneficio es que han dado a la industria de software la habilidad de realizar estudios económicos serios. Como permite medir todas las actividades del software (requerimientos, diseño, codificación, pruebas, documentación, administración, etc.) ha llegado a ser un estándar de línea base y “benchmarks” del software.

El estándar actual de Puntos de Función publicado por IFPUG en su versión 4.1, fue explicada por David Garmus y David Herron en su obra “Function Point Analysis: Measurement Practices for Successful Software Projects” [2]. Ellos han contribuido durante mucho tiempo con el IFPUG.

Este documento intenta explicar en idioma español y con casos prácticos, la metodología de Puntos de Función, utilizando como base los capítulos 7 al 10 de la obra de Garmus y Herron [2]. Es conveniente indicar que esta técnica no es trivial, que sus reglas son bastante complejas y que de acuerdo a Kan [1, pág 456] y a Garmus y Herron [2, pág xvi] requieren suficiente capacitación para entenderlas.

Para facilitar la comprensión de esta técnica en el capítulo 1 se presentan las reglas del IFPUG que permiten estimar el tamaño de una aplicación y en el capítulo 2 se muestran casos prácticos de cómo aplicarlas.

El objetivo de este documento es que los estudiantes de pregrado y posgrado de la Escuela de Computación e Informática aprendan a cuantificar con mayor precisión el tamaño del software, y consecuentemente a derivar el cronograma y el presupuesto del proyecto a desarrollar. Actualmente la metodología que se utiliza es poco precisa, por estar basada solamente en la experiencia de los planificadores en proyectos similares. En cambio, la metodología de Puntos de Función logra estimar con mayor precisión el cronograma y el presupuesto con solo conocer su funcionalidad y complejidad de la aplicación. Otro objetivo de este documento es promover a través de los estudiantes, a que las empresas o instituciones públicas inicien su proceso de elaboración de métricas.

Los Puntos de Función son independientes del lenguaje del computador, de la metodología de desarrollo, de la tecnología o de la capacidad del equipo del proyecto usado para desarrollar la aplicación. Además pueden ser usados para estimar sistemas GUI (en inglés, “Graphical User Interfaces”), sistemas cliente/servidor o desarrollo orientado a objetos.

Capítulo 1

Metodología de Análisis de Puntos de Función

Los Puntos de Función deben representar la funcionalidad que el usuario del negocio requiere, por lo tanto nunca es muy temprano para realizar conteo de Puntos de Función. Tradicionalmente se cree que el conteo no puede ocurrir hasta que el diseño no se haya terminado. Las reglas y guías del IFPUG hacen posible realizar el conteo de Puntos de Función una vez que los requerimientos se hayan finalizado. Sin embargo hay heurísticas que permiten estimarlo aún más temprano en el ciclo de vida. Durante la obtención de los requerimientos para muchos proyectos se desarrolla el modelo de objetos del dominio, o el modelo de análisis los cuales generalmente, no tienen atributos ni está en tercera forma normal. Sin embargo en este punto el analista puede comenzar a identificar las entidades que necesita procesar para insertar, modificar, borrar y eliminar datos. En este momento se puede hacer una buena estimación del tamaño asumiendo que todas las transacciones identificadas tienen una complejidad media. Aún en este punto del ciclo de vida se pueden predecir con certeza el Valor de Factor de Ajuste (explicada en el paso 5 de este documento). Una vez que se han completado los requerimientos es posible contar acertadamente los Puntos de Función, porque ya las entidades tienen todos los atributos. Todas las interacciones de la aplicación, incluyendo pantallas y reportes deberían haber sido identificadas y los elementos de datos asociados con ellas especificados. No es necesario que las pantallas y reportes sean formateadas en este momento. Por supuesto, siguiendo con la fase de diseño el conteo se llega a perfeccionar, aún considerando en alguna medida cambios en los requerimientos. Se espera que ya entre el fin del diseño y el de las pruebas de aceptación no debería haber más cambios en el conteo.

El proceso de medición se puede resumir en los siguientes siete pasos:

1. Determinar el tipo de conteo de puntos de función.
2. Identificar el alcance del conteo y las fronteras de la aplicación.
3. Identificar todas las funciones de datos y su complejidad.
4. Identificar todas las funciones transaccionales y su complejidad.
5. Determinar los puntos de función sin ajustar.
6. Determinar el valor del Factor de Ajuste – basado en las 14 características generales del sistema.
7. Calcular los puntos de función ajustados.

El tiempo de conteo de los puntos de función varía dependiendo del estado particular de una aplicación. Típicamente hay menos información disponible en las etapas tempranas del desarrollo, y significativamente hay más información disponible conforme se desarrolla y entrega la aplicación. Al inicio del proceso la única información disponible podría ser verbal. Durante el desarrollo la información disponible para ayudar en el conteo se incrementa al incluir algunos de los siguientes documentos potenciales, los cuales ayudan a determinar el conteo de puntos de función:

- Propuesta del proyecto.
- Diagrama del sistema de alto nivel (mostrando relaciones con otras aplicaciones).
- Diagrama de Entidad-Relación.
- Modelo de datos lógico.

- Diagramas de flujos de datos.
- Modelos de objetos.
- Modelos de procesos.
- Documento de requerimientos del sistema.
- Prototipos.
- Especificación funcional.
- Casos de Uso.
- Especificación del Sistema.
- Especificación del diseño detallado.
- Modelos de diseño físicos.
- Modelos operacionales.
- Especificación de módulos/programas.
- Pantallas para sistemas en línea.
- Diseño de archivos.
- Copias de reportes o diseño de reportes.
- Manuales de usuario.
- Bases de datos.
- Casos de prueba.
- Manuales de usuario y documentación técnica.
- Materiales de capacitación.
- Ayuda del sistema.

A continuación se explica el detalle de cada uno de los siete pasos de la metodología tomando como base los capítulos 6 al 9 de [2].

Paso 1. Determinar el tipo de conteo de puntos de función.

Existen tres posibles tipos de conteo que se describen a continuación:

1. Conteo de puntos de función para Proyectos en Desarrollo.

Mide la funcionalidad provista a los usuarios finales en la primera instalación de la aplicación. Incluye la funcionalidad que se contará de la nueva aplicación y la requerida por los usuarios, para convertir los datos que residen en los archivos de datos viejos, hacia los nuevos archivos de datos. Conforme avanza el desarrollo del proyecto, a menudo hay que actualizar el conteo. Los siguientes conteos deben validar la funcionalidad identificada previamente y capturar la funcionalidad agregada. Los conteos podrían ocurrir durante las siguientes fases, ya sea en un enfoque de cascada o un proceso iterativo. Ver figura 1.

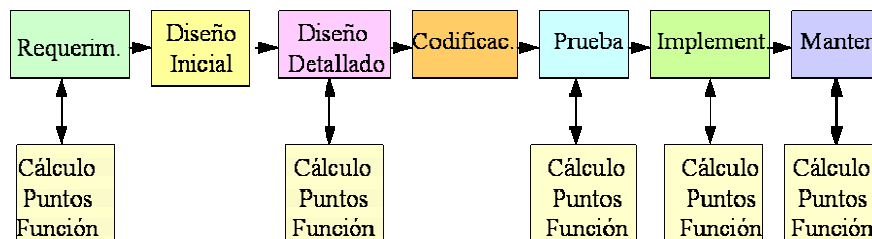


Figura 1. Conteo durante el desarrollo.

2. *Conteo de puntos de función para proyectos en mantenimiento.*

Mide las modificaciones a una aplicación existente e incluye la funcionalidad provista a los usuarios al agregar nuevas funciones, borrar funciones viejas y cambiar las funciones existentes. La funcionalidad por conversión también podría existir en un proyecto de mantenimiento. Después de realizarse el mantenimiento se debe revisar el conteo para reflejar los cambios apropiados en la funcionalidad de la aplicación actual.

3. *Conteo de puntos de función para una aplicación desarrollada.*

Mide una el tamaño de una aplicación ya instalada. Es conocido como “conteo de línea base o instalado” y provee el total de la funcionalidad provista a los usuarios finales. Típicamente representa las aplicaciones que actualmente están siendo utilizadas y mantenidas.

Paso 2. Identificar el alcance del conteo y la frontera de la aplicación.

El alcance del conteo se determina por el propósito del conteo. Identifica los sistemas, aplicaciones o subconjuntos de una aplicación que será medida. Podría incluir las funciones que serán satisfechas por la compra de un paquete; todas las aplicaciones que serán contratadas; o podría restringir las funciones dentro de una aplicación con un propósito específico como por ejemplo: los reportes. La frontera de la aplicación es el límite entre la aplicación que está siendo medida y las aplicaciones externas al dominio del usuario.

IFPUG ha definido las siguientes reglas para identificar las fronteras:

- **Las fronteras se basan en el punto de vista del usuario.** El usuario debe ser capaz de definir el alcance de la aplicación y la funcionalidad del negocio en el lenguaje del usuario.
- **Las fronteras entre aplicaciones relacionadas se basan en la funcionalidad individual del negocio y no en consideraciones técnicas.** Por ejemplo: Microsoft Office actualmente consiste de Word, Excel, Power Point y Access, cada una es una aplicación individual dentro de Microsoft Office. Otro ejemplo es System Alpha que consiste de un proceso de entrada de datos en línea, seguido por un proceso de actualización en lote; ésta es una aplicación simple que incluye ambos procesos en línea y en lote.
- **Las fronteras establecidas para una aplicación no influyen en el alcance del conteo.** La frontera de una aplicación que está en mantenimiento permanece como estaba, excepto por supuesto si se le agrega funcionalidad que pueda expandir la frontera y borrar funcionalidad que pueda reducir la frontera. Una mejora no define su propia frontera de aplicación, pero esta es definida por el alcance del conteo. Los proyectos en desarrollo y mantenimiento a menudo incluyen más de una aplicación simple. En estos casos, las fronteras de la aplicación son identificadas dentro del alcance del conteo, pero se cuentan en forma separada. En la figura 2 se presenta en forma gráfica el conteo de puntos de función.

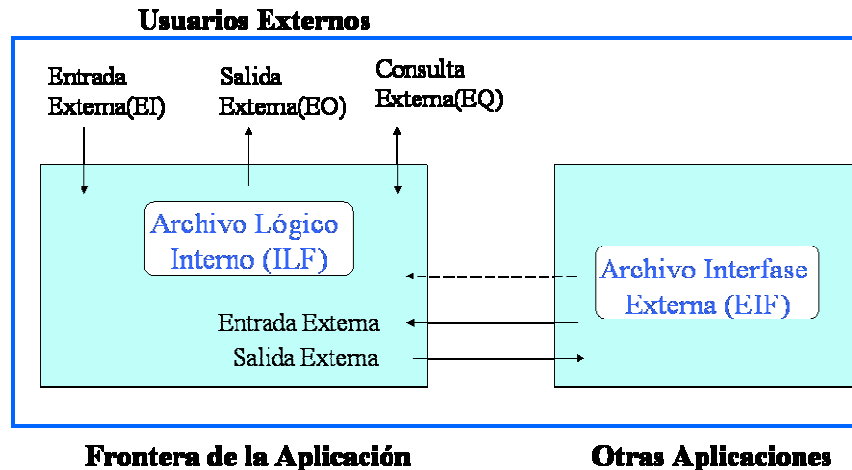


Figura 2. Funcionalidad reconocida en el conteo de puntos de función.

Existe mucha confusión en los usuarios respecto a la definición de las fronteras. Es extremadamente importante que el conteo se realice consistentemente y en forma correcta. El conteo tiene poco beneficio si es consistente pero incorrecto, especialmente cuando las fronteras de la aplicación se están identificando. Las fronteras no deben ser establecidas a nivel de programas o de módulos, sino al nivel más alto de la aplicación.

Paso 3. Identificar todas las funciones de datos y su complejidad y determinar los puntos de función desajustados.

Existen dos tipos de funciones de datos que se deben identificar los cuales son:

- los Archivos Lógicos Internos (ILF, en inglés “Internal Logical Files”) y
- los Archivos de Interfases Externas (EIF, en inglés “External Interface Files”).

1. Definición de los Archivos Lógicos Internos (ILF, en inglés “Internal Logical Files”)

Un ILF es un grupo identificable por el usuario, de datos relacionados lógicamente o información de control administrada dentro de las fronteras de la aplicación. El propósito principal de un ILF es mantener los datos administrados a través de uno o más procesos elementales, de la aplicación que se está contando.

A continuación se describe cada término.

Identificables por el usuario se refiere a requerimientos definidos para procesos y o grupos de datos que son acordados y entendidos por el usuario y el desarrollador. Por ejemplo: un registro de cuentas de cheques que podría requerirse en una aplicación financiera.

Los *datos* se refieren a un conjunto de hechos administrados dentro de la aplicación. Por ejemplo: número de cheque, cantidad, fecha, pago, etc. que se pueden administrar en un registro de cuentas de cheques.

Relacionados lógicamente se refiere al requerimiento que cada grupo de datos unidos lógicamente debe satisfacer de acuerdo a las descripciones provistas. Un ILF no debe

depender ni ser atributo de otro ILF para mantener su existencia. Los grupos deben ser mezclados tanto como sea necesario, particularmente aquellos creados por razones de rendimiento o de implementación. Típicamente los ILF deben presentarse en segunda o tercera forma normal. Un ejemplo puede ser una tabla de direcciones que probablemente pertenece a un nivel superior, tal como un archivo de clientes, un archivo de pagos, un archivo de ubicación de inventario o un archivo de empleados.

Información de control es un dato usado por la aplicación para influir en un proceso elemental de la aplicación que se está contando. Éste especifica qué, cuándo y cómo los datos van a ser procesados. En el caso de los ILF estos datos, reglas o parámetros se almacenan y actualizan dentro de la aplicación. Algunos ejemplos son: un dato de control para administrar la impresión, la fecha y la hora administrada por el usuario para establecer el tiempo de ciertos eventos.

Administrados se refiere a la capacidad de modificar datos a través de un proceso elemental. Los datos o la información de control pueden ser administrados a través de transacciones como: **agregar, cambiar, borrar, enviar, transferir, evaluar, mantener, revisar, actualizar**, etc.. Un ILF se puede administrar y contar como un ILF dentro de más de una aplicación; sin embargo se cuenta solo una vez para cada aplicación donde éste es administrado.

Un proceso elemental es la unidad de actividad más pequeña que tiene significado para el usuario. Ejemplo: Un problema de stock en un almacén podría descomponerse en varios subprocesos tales como: Crear, Leer, Actualizar y Desplegar de manera que el problema crea una cantidad dada, lee un archivo para validar si es por crédito y actualiza la cantidad del stock disponible. El problema del stock es un proceso elemental, no es un subproceso subordinado y puede actualizar más de un ILF a través de la misma transacción.

1.1 Reglas para contar los ILF

IFPUG ha definido las siguientes reglas para que datos o información de control sean contados como un ILF:

- El grupo de datos o información de control debe estar relacionado lógicamente y debe ser reconocible por el usuario.
- El grupo de datos debe ser administrado a través de un proceso elemental dentro de las fronteras de la aplicación que está siendo contada.

Una vez que se ha identificado un ILF dentro de una aplicación, este no puede ser contado como un EIF para la misma aplicación, aún si éste es usado por referencia por otras transacciones, ni puede ser contado como EIF durante un proyecto de mantenimiento para esa aplicación.

Algunos ejemplos adicionales de ILF son:

- Datos transaccionales tales como: registros de inventarios, registros de capacitación de empleados, registro de planilla, transacciones de tarjetas de crédito, ventas de productos.
- Datos de seguridad o “password” administrados dentro de la aplicación.
- Datos de ayuda administrados dentro de la aplicación.
- Datos de edición administrados dentro de la aplicación.

- Datos de parámetros administrados dentro de la aplicación.
- Archivos de error y su descripción administrados dentro de la aplicación.

Algunos ILF erróneos que no deben ser contados son:

- Archivos temporales o iteraciones del mismo archivo.
- Archivos de trabajo.
- Archivos de ordenamiento.
- Archivos de vistas que contienen datos extraídos de otros ILF o EIF antes de desplegarlos o imprimirlos, como parte del procesamiento necesario para producir una salida externa (EO, en inglés “External Output”) o una consulta externa (EQ, en inglés “External Inquiry”).
- Copias del mismo archivo ordenados en forma diferente o mantenidos en localizaciones adicionales.
- Archivos introducidos por problemas de tecnología.
- Archivos de índices, uniones, relaciones o conexiones, a menos que ellos contengan atributos no claves administrados independientemente.
- Datos de auditoría o históricos, estos deben ser contados junto con los datos transaccionales de la aplicación.
- Archivos mantenidos por otras aplicaciones y leídos o referenciados solamente, estos deben contarse como EIF.
- Datos de respaldo usados como backup y recuperación de la empresa, esta capacidad se reconoce dentro de las 14 características solamente.

2. Definición de los Archivos Interfases Externas (EIF, en inglés “External Interface Files”)

Un EIF es un grupo identificable por el usuario, de datos relacionados lógicamente o información de control referenciados por la aplicación que se está contando, pero administrados dentro de las fronteras de una aplicación diferente. El propósito principal de un EIF es mantener datos referenciados a través de uno o más procesos elementales dentro de la frontera de la aplicación que se está contando. Un EIF contado para una aplicación debe ser contado como ILF en otra aplicación

A continuación se describe cada término de esta definición. Las definiciones son muy similares a los de los ILF descritos en la sección anterior.

Identificables por el usuario se refiere a requerimientos definidos para procesos y grupos de datos que son acordados y entendidos por el usuario y el desarrollador. Por ejemplo: un registro de cuentas de cheques que podría ser usado para la validación en una aplicación, el cual lee los datos solamente mientras valida una transacción no relacionada.

Los *datos* se refieren a un conjunto de hechos administrados por otra aplicación. Por ejemplo: número de cheque, cantidad, fecha, pago, etc. que se pueden administrar en un registro de cuentas de cheques.

Relacionados lógicamente se refiere al requerimiento que cada grupo de datos unidos lógicamente debe satisfacer de acuerdo a las descripciones provistas. Un EIF no debe depender ni ser atributo de otro EIF para mantener su existencia. Los grupos deben ser

mezclados tanto como sea necesario, particularmente aquellos creados por razones de rendimiento o de implementación. Típicamente los EIF deben presentarse en segunda o tercera forma normal. Un ejemplo puede ser una tabla de direcciones que probablemente pertenece a un nivel superior, tal como un archivo de clientes, un archivo de pagos, un archivo de ubicación de inventario o un archivo de empleados.

Información de control es un dato usado por la aplicación para influir en un proceso elemental de la aplicación que se está contando. Éste especifica qué, cuándo y cómo los datos van a ser procesados. En el caso de los EIF estos datos, reglas o parámetros se almacenan y se actualizan dentro de la aplicación. Algunos ejemplos son: un dato de control para administrar la impresión, la fecha y la hora administrada por el usuario para establecer el tiempo de ciertos eventos.

Administrados se refiere a la capacidad de modificar datos a través de un proceso elemental de otra aplicación.

Un proceso elemental es la unidad de actividad más pequeña y significativa para el usuario. Por ejemplo: una pantalla de stock en un almacén podría descomponerse en varios subprocesos, de manera que un archivo pueda ser leído para determinar la cantidad de stock y otro archivo independiente pueda ser leído para determinar la descripción de los ítems. El proceso elemental puede leer más de un EIF a través de la misma transacción.

2.1 Reglas para contar los EIF.

IFPUG ha definido las siguientes reglas para que datos o información de control sean contados como un EIF:

- El grupo de datos o información de control está relacionado lógicamente y es identificable por el usuario.
- El grupo de datos es referenciado por una aplicación externa a la que se está contando.
- El grupo de datos no es administrado por la aplicación que se está contando.
- El grupo de datos es administrado en un ILF de otra aplicación.

Una vez que un grupo de datos se ha identificado como EIF dentro de una aplicación, el EIF no puede ser contado de nuevo dentro de la misma aplicación, aún si éste es usado por referencia por otras transacciones, o contiene diferentes datos del mismo archivo.

Algunos ejemplos comunes de EIF son:

- Datos de la aplicación extraídos o leídos de otra aplicación.
- Datos de seguridad o “password” administrados fuera de la aplicación.
- Datos de ayuda administrados fuera de la aplicación.
- Datos de edición administrados fuera de la aplicación.
- Datos de parámetros administrados fuera de la aplicación.
- Archivos de error y su descripción administrados fuera de la aplicación.

Algunos EIF erróneos son:

- Un dato recibido de otra aplicación que administran uno o más ILF’s dentro de la aplicación que se está contando. Esto es considerado como un dato transaccional y se deben contar como EI.

- Un dato que es administrado por la aplicación que está siendo contada pero accesado y usado por otra aplicación diferente. Es contado como EIF por la aplicación que lo accesa.
- Un dato formateado y enviado por la aplicación que se está contando a otra aplicación. Debería ser contado como una consulta externa (EQ, en inglés “external inquiry”) o una salida externa (EO, en inglés “external output”).
- Archivos temporales o iteraciones del mismo archivo. Estos no se cuentan en el FPA.
- Archivos de trabajo no se cuentan en el FPA.
- Archivos de ordenamiento no se cuentan en el FPA.
- Archivos de extracciones o vistas que contienen datos extraídos de EIF previamente contados, antes de desplegarlos o imprimirlos. No se cuentan en el FPA pero son reconocidos como parte del procesamiento necesario para producir las salidas o consultas externas.
- Archivos introducidos por la tecnología no se cuentan en el FPA.
- Archivos de índices, uniones, relaciones o conexiones, a menos que ellos contengan mantenidos independientemente, atributos no claves.
- Datos de auditoría o históricos, estos deben ser contados junto con los datos transaccionales de la aplicación.

3. Reglas para determinar la complejidad y contribución de los ILF y de los EIF

El conteo físico de ILF’s y EIF’s junto con la complejidad funcional relativa de cada uno, determina la contribución de los tipos funcionales de datos para el conteo de los puntos de función desajustados. A cada ILF y EIF se le debe asignar una complejidad funcional basada en el número de tipos de elementos de datos (DET) y en los tipos de elementos de registros (RET) asociados con el ILF o el EIF.

Cada término usado en la definición se define a continuación:

Complejidad funcional es el peso asignado a cada función de dato individual. Posibles valores como bajo, medio y alto son asignados usando una matriz de complejidad la cual considera el número de DET y RET.

Tipos de Elementos de Datos (DET) son campos/atributos únicos, reconocibles por el usuario, no repetitivos y pueden incluir claves foráneas.

Tipos de Elementos de Registros (RET) son subgrupos de elementos de datos reconocibles por el usuario (opcionales u obligatorios) que se encuentran dentro de un ILF o un EIF. Los RET son típicamente representados dentro de un diagrama de entidad relación como los subtipos de entidad o entidades de atributo, comúnmente llamados relaciones padre-hijo. Opcionales: el usuario tiene la opción de usar uno o ningún subgrupo opcional durante la ejecución de un proceso elemental que agrega o crea una instancia de datos. Obligatorio: el usuario debe usar al menos uno de los subgrupos obligatorios.

IFPUG ha definido las siguientes reglas para el conteo de elementos de datos, campos o atributos (conocidos como DET por IFPUG) para los ILF y los EIF:

- Contar un DET por cada campo reconocible, no repetido, administrado o recuperado desde un ILF o EIF a través de un proceso elemental. Por ejemplo: Un número de cheque, una cantidad, una fecha, un número de cuenta, almacenados en un registro de cuenta de cheques puede ser contado cada uno como un DET, sin tomar en cuenta el número de cheques escritos con datos exclusivos de cada uno y sin tomar en cuenta cómo se va a almacenar el dato físicamente.
- Cuando dos o más aplicaciones mantienen y/o referencian el mismo ILF o EIF pero con DET independientes, contar solamente los DET que están siendo usados por cada aplicación. Por ejemplo un ILF que es actualizado por dos aplicaciones (A y B) y es referenciado por una tercera (C), podría contar los DET en forma exclusiva, tal como se muestra en la tabla 1. Consecuentemente A puede ser contado con 8 DET, B con 7 y C con 2.
- Contar un DET por cada pieza de dato requerido por el usuario para establecer una relación con otro ILF o EIF. Este tipo de DET se conoce como llave foránea. Por ejemplo: una relación que existe con otro ILF o EIF requiere una clave con el número de parte y el código del fabricante. El número de parte y el código del fabricante pueden ser contados como dos DET a menos, por supuesto, que cualquiera o ambos se hayan contado previamente.

Los campos que aparecen más de una vez en un ILF o un EIF por causa de la tecnología o técnicas de implementación, tales como las llaves ya mencionadas, son contadas solamente una vez en el ILF o EIF. Sin embargo un campo puede ser un DET para muchos ILF o EIF.

Los campos repetitivos que son idénticos en formato y existen para permitir ocurrencias múltiples de valores de datos son contados solo una vez en el ILF o EIF. Por ejemplo: un ILF que contiene campos con 12 cantidades mensuales y un campo con un total anual puede ser contado como 2 DET – uno por cada cantidad mensual repetitiva y uno para identificar el total; un DET adicional debe haber indudablemente para identificar el mes.

Tabla 1. Ejemplo de conteo.

DET	Uso del DET	Conteo del DET
Número de parte	Clave primaria de A, B y C	DET para A, B y C.
Nombre de parte	Mantenido por A.	DET para A
	Referenciado por B y C.	DET para B y C
Uso semanal	Mantenido por B.	DET para B
Uso por departamento	Mantenido por B.	DET para B
Precio de compra	Mantenido por A.	DET para A
	Referenciado por B.	DET para B
Nombre proveedor	Mantenido por A.	DET para A
	Referenciado por B.	DET para B
Dirección Calle proveedor	Mantenido por A.	DET para A

Ciudad proveedor	Mantenido por A.	DET para A
Estado proveedor	Mantenido por A.	DET para A
Código postal proveedor	Mantenido por A.	DET para A
Dirección total proveedor, leído como un bloque de datos	Referenciado por B.	DET para B

Los datos de auditoría, los cuales incluyen imágenes antes y después de 15 campos distintos, se cuentan como 2 DET – uno para la imagen del antes y uno para la imagen del después.

Los cálculos que ocurren internamente durante el procesamiento de una entrada externa (EI) y almacenado en la base de datos se cuenta como un DET del ILF.

IFPUG recomienda las siguientes reglas para contar RETs dentro de un ILF o EIF:

- **Se cuenta un RET por cada subgrupo opcional u obligatorio en un ILF o un EIF.** No se cuentan los RET que existen por causa de la tecnología o por la metodología utilizada (estos es encabezados, archivos de texto independientes, etc.) Dos o más archivos a menudo pertenecen al mismo archivo lógico (ILF o EIF). Estos RET pueden también ser identificados por la existencia de llaves secundarias usadas para el almacenamiento de datos y la creación de relaciones entre los archivos lógicos. Los datos en un archivo lógico típicamente están en tercera forma normal.
- **Si no hay subgrupos contar el ILF o el EIF como un RET.**

4. Determinar los puntos de función sin ajustar (PFSA) para los ILF y los EIF.

Para determinar los puntos de función sin ajustar de los ILF y de los EIF se deben seguir las siguientes instrucciones:

1. Identificar con base en el alcance de la aplicación y en las secciones 3.1 y 3.2 los ILF, los EIF y sus atributos.
2. Aplicar las reglas descritas en las secciones 3.1.1 y 3.2.1 para determinar el **número de ILF** y de **EIF** y obtener una lista clasificada por ILF y EIF.
3. Aplicar las reglas de la sección 3.3 y obtener los **DET** y los **RET** para cada **ILF** y **EIF** de la lista anterior.
4. Con base en la lista con ILF, EIF, con sus DET y RET y la “Matriz de complejidad para los ILF y EIF” (refiérase a la tabla 2), determinar la complejidad de cada ILF y EIF. Por ejemplo: un ILF con 3 RET y 51 DET tiene una complejidad Alta. Un EIF con 2 RET y 22 DET tiene una complejidad Media.

Tabla 2. Matriz de complejidad para los ILF y EIF.

Número de Registros (RET)	Número de Elementos de Datos (DET)		
	1-19	20-50	≥51
1	Baja	Baja	Media
2-5	Baja	Media	Alta
>5	Media	Alta	Alta

5. Una vez determinada la complejidad de cada ILF y EIF se le debe asignar un peso. Con base en la tabla 3 se ubica la fila de los ILF y EIF y la columna correspondiente a la complejidad (baja, media o alta) obtenida en el punto anterior, y se multiplica el número de ILF/EIF por el peso que le corresponde. Siguiendo el ejemplo: con el **ILF** de complejidad **Alta** y el **EIF** de complejidad **Media** obtenidos en el punto anterior, podemos contabilizar **15** puntos de función desajustados para el **ILF** y **7** puntos de función desajustados para el **EIF** obteniendo un total de **22** puntos de función sin ajustar.

Tabla 3. Conteo de los ILF y EIF.

Componentes	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	x7	x10	1 x15
Archivos de Interfase Externa (EIF)	x5	1 x7	x10
Entradas Externas (EI)	x3	x4	x6
Salidas Externas (EO)	x4	x5	x7
Consultas Externas (EQ)	x3	x4	x6
		7	15
Total	22 puntos de función sin ajustar (PFSA)		

Paso 4. Identificar todas las funciones transaccionales y su complejidad y determinar los puntos de función desajustados.

Existen tres tipos de funciones transaccionales que se deben identificar las cuales son:

- Entradas Externas (EI, en inglés “External Inputs”) tales como agregar, cambiar y borrar datos de un archivo
- Consultas Externas (EQ, en inglés “External Inquiries”) tales como recuperar datos de un archivo
- Salidas Externas (EO, en inglés “External Outputs”) tales como la generación de reportes.

A continuación se explica el detalle de cada una de estas funciones transaccionales.

1. Definición de las Entradas Externas (EI, en inglés “External Inputs”)

Una entrada externa (EI) es un proceso elemental de la aplicación que procesa datos o información de control, que ingresa desde afuera de la frontera de la aplicación. Los datos procesados administran uno o más ILF's. La información de control procesada puede o no administrar un ILF. La intención principal de un EI es administrar uno o más ILF y/o alterar la conducta de la aplicación a través de su lógica de procesamiento.

A continuación se describen los siguientes términos:

Un *proceso elemental* es la unidad de actividad más pequeña que tiene significado para el usuario final. El proceso debe estar autocontenido, es decir no debe su existencia a algún otro proceso, dejándolo en un estado consistente una vez que finaliza. Por ejemplo: una transacción de entrada al sistema para ingresar los beneficios de un empleado, puede consistir de tres pantallas, sin embargo la entrada se considera incompleta hasta que las tres pantallas se completen, debido a que el proceso elemental requiere la completitud de las tres pantallas. Si toda la información, reconociendo que algunos campos no son obligatorios se ha ingresado, la transacción podría estar concluida y el negocio puede quedar en un estado consistente.

Los *datos* se refieren a un conjunto de hechos procesados por la transacción de entrada. Por ejemplo: los datos pueden ser los campos de datos incluidos en la transacción de beneficio de un empleado descrita arriba. Podría ser el nombre del empleado, selección del beneficio, acuerdo para participar(S/N), porcentaje de contribución, entre otros.

La *información de control* es un dato que influye en un proceso elemental de la aplicación que está siendo contada; En el caso de los EI, estos datos, reglas o parámetros o parámetros pueden ser almacenados o usados independientemente para administrar/iniciar un proceso sin necesidad de que sea almacenado después de que el proceso esté completo. Por ejemplo un dato de control es usado para administrar la configuración del sistema. La información de control es usada por la aplicación para asegurar el cumplimiento de los requerimientos del negocio especificados por el usuario. Esto puede incluir características tales como: comparar o calcular. En sistemas de tiempo real puede incluir una señal de entrada desde una alarma, un instrumento, un sensor, u otra aplicación.

Administrar se refiere a la capacidad de modificar datos a través de un proceso elemental de otra aplicación. Contar un EI por cada actividad de mantenimiento, por ejemplo: agregar, cambiar, borrar, revisar, actualizar, asignar, guardar y crear. No hay límite en el número de verbos que se pueden seleccionar para mantener datos.

La *lógica de procesamiento* incluye específicamente los requerimientos solicitados por un usuario para completar un proceso elemental. Usualmente se requiere una lógica de procesamiento para completar un proceso elemental. Una lógica de procesamiento no se debe usar para determinar EI, EO o EQ exclusivos. Un proceso elemental de un EI puede incluir múltiples validaciones, filtros, ordenamientos, etc. El ordenamiento por ejemplo no determina la exclusividad de la transacción, es decir, la capacidad de ordenar no es una transacción adicional.

Una lógica de procesamiento para un EI puede incluir:

- Validaciones.
- Fórmulas o cálculos matemáticos.
- Conversiones de valores equivalentes.
- Filtros y selección de datos usando criterios específicos para comparar múltiples grupos de datos.
- Análisis de condiciones para determinar cuál es aplicable.
- Actualización de ILF (es obligatorio para un EI administrar uno o más ILF o alterar la conducta del sistema).
- Referencias a ILF o EIF.
- Recuperación de datos o información de control.
- Creación de datos derivados.
- Alteración de la conducta del sistema (es obligatorio para un EI administrar uno o más ILF o alterar la conducta del sistema).
- Preparación y presentación de información fuera de la frontera.
- Capacidad de aceptar datos o información de control que ingrese a la aplicación (obligatorio para un EI).
- Ordenamiento o clasificación de un conjunto de datos.

1.1 Reglas para contar los EI.

Las siguientes reglas se deben aplicar a los datos que están siendo contados como entradas externas (EI):

- **El dato debe ser recibido desde fuera de la frontera de la aplicación.**
- **El dato debe ser administrado en al menos un ILF a través de un proceso elemental de la aplicación.**
- **El proceso debe ser la actividad más pequeña con significado para el usuario.**
- **El proceso debe dejar el negocio de la aplicación que está siendo contada en un estado consistente.**
- **Para el proceso identificado se debe aplicar una de las tres siguientes reglas:**
 1. **La lógica de procesamiento debe ser única/diferente dentro de la aplicación (no deben haber dos lógicas de procesamiento iguales).**
 2. **El conjunto de datos identificados debe ser diferente del identificado en otras EI en la aplicación (no deben haber dos flujos iguales).**
 3. **Lo ILF y EIF referenciados deben ser diferentes de los referenciados por otras EI en la aplicación.**

Para la información de control que se está contando como EI, IFPUG recomienda las siguientes reglas:

- **La información de control debe ser recibida desde fuera de la frontera de la aplicación.**
- **La información de control debe ser especificada por el usuario para asegurar el cumplimiento de los requerimientos.**

- **El proceso debe ser la actividad más pequeña con significado para el usuario.**
- **El proceso debe dejar el negocio de la aplicación que está siendo contada en un estado consistente.**
- **Para procesos identificados se debe aplicar una de las siguientes reglas:**
 - 1. La lógica de procesamiento debe ser única o diferente de otras lógicas de procesamiento realizadas por EI en la aplicación.**
 - 2. El conjunto de datos identificado es diferente a los conjuntos identificados por otras EI en la aplicación (no deben haber dos flujos iguales).**
 - 3. Los ILF o EIF referenciados deben ser diferentes de aquellos referenciados por otras EI en la aplicación.**

Algunos ejemplos de Entradas Externas (EI) adicionales son:

- Los datos transaccionales que se usan para administrar un ILF tales como: una venta, un recibo, una transferencia, una fórmula de seguro, etc.
- Las entradas que proveen información de control tal como: un reporte de los movimientos de la tierra.
- Los mensajes de otras aplicaciones que requieren procesamiento.
- Los archivos transaccionales de otras aplicaciones, que pueden incluir transacciones múltiples de diferente tipo, que requieren procesamiento separado y único, por ejemplo: transacciones de tarjetas de ventas al contado o de crédito en cuyo caso podrían ser múltiples EI's.
- Las entradas que mantienen un ILF.
- Las funciones de usuario que inician un control o ingresan datos.
- Los archivos de datos administrados en una aplicación anterior que deben ser procesados a través de un esfuerzo de conversión dentro de un ILF nuevamente desarrollado, cuando el dato está migrando como parte de un proyecto en desarrollo o en mantenimiento. Esto puede ser incluido dentro del conteo del proyecto pero no dentro del conteo de la aplicación.
- Los datos físicos que inician el procesamiento, tal como una temperatura.
- El mantenimiento de cualquier ILF incluyendo ayuda, cualquier archivo de mensajes, parámetros, etc..

Los siguientes son ejemplos de EI mal identificados:

- Un dato referenciado que es leído por la aplicación desde datos almacenados en otra aplicación, pero que no se usa para administrar ningún ILF en la aplicación que está siendo contada; esto es típicamente reconocido como EIF.
- La parte de solicitud de entrada ya sea de una consulta o de una salida.
- Las pantallas de menú usada para navegación o selección de opciones y que no administra ningún ILF.
- Las pantallas de ingreso que facilitan al usuario la entrada a la aplicación.

- Métodos múltiples para invocar una misma lógica; por ejemplo, teclas de acción que realizan la misma función o la misma transacción en diferentes pantallas deben ser contadas solo una sola vez.
- Posicionamiento del cursor sobre una pantalla para llenar o mover campos.
- Los datos para refrescar o cancelar una pantalla.
- Respuesta a mensajes que requieren una confirmación del usuario, como por ejemplo borrar u otra transacción.
- Los datos enviados en línea/lote dentro de la misma aplicación, que no cruzan la frontera de la aplicación.
- Los datos enviados entre el cliente/servidor dentro de la misma aplicación, que no cruzan la frontera de la aplicación.

1.2 Reglas para determinar la complejidad y contribución de los EI

El conteo de los EI junto con la *complejidad funcional* relativa de cada uno, determina la contribución de las entradas externas para el conteo de los puntos de función desajustados. A cada EI identificado se le debe asignar una complejidad funcional basada en el número de *tipo de elementos de datos* (DET) y los *tipos de archivos referenciados* (FTR's) asociados con el EI.

A continuación se describe cada término de la definición:

Complejidad funcional: es el porcentaje asignado a cada función transaccional. Se asignan de acuerdo a la matriz de complejidad que considera el número de DET y FTR.

Tipo de elemento de dato (DET): usualmente son campos o atributos únicos, reconocibles por el usuario, no repetitivos, incluyendo atributos de claves foráneas, que cruzan la frontera de la aplicación.

Tipo de archivo referenciado (FTR): se refiere al número total de ILF administrados o leídos y de EIF leídos por la transacción EI.

IFPUG recomienda las siguientes reglas para contar los elementos de datos, campos o atributos (conocido como DET por IFPUG) para un EI:

- **Se cuenta un DET para cada campo/atributo único, reconocible por el usuario, incluyendo atributos de clave foránea** que cruzan (ya sea porque ingresan o existen) la frontera de la aplicación para completar un proceso elemental del EI. Típicamente tales campos o atributos se mantienen en un ILF. Por ejemplo: nombre del empleado, monto de pago, número de ítem, cantidad vendida.
- **No se cuenta como un DET cada campo que no lo ingresa el usuario** (no cruza la frontera de la aplicación) **pero a través de un EI es derivado por la aplicación y administrado en un ILF**. Algunos ejemplos son: una fecha generada por el sistema, un valor recuperado, un número de cuenta y un valor calculado.

- Se cuenta como un DET cada campo lógico almacenado físicamente como múltiples campos, pero requerido por el usuario como una simple pieza de información. Por ejemplo: líneas de una dirección.
- Contar un simple DET para cubrir todos los mensajes del sistema que indican que un error ha ocurrido durante el procesamiento, que confirman que el procesamiento ha terminado, o que verifican que debe continuar. Aunque hayan múltiples posibilidades de mensajes contar solo un DET para todos los procesos elementales. Es decir contar solo un DET para todos los mensajes, aunque hayan mensajes de errores múltiples, de verificación y de confirmación.
- Contar como un simple DET la capacidad de especificar una acción a tomar, aún si hay múltiples métodos para invocar la misma lógica de procesamiento. Contar un DET por línea de comando o tecla de función/acción (PF) que provea la capacidad de especificar la acción a tomar por el EI. Contar esto como un DET por EI. No contar un DET por cada línea de comando y tecla de función/acción (PF).

IFPUG recomienda las siguientes reglas que se aplican para contar tipos de archivos referenciados (FTR):

- Contar un FTR para cada ILF administrado por un proceso elemental del EI.
- Contar un FTR para cada ILF o EIF leído durante el procesamiento del EI.
- Contar solamente un FTR para cada ILF que es administrado y leído por el EI.

1.3 Determinar los puntos de función sin ajustar (PFSA) para los EI.

Para determinar los puntos de función sin ajustar de los EI se deben seguir las siguientes instrucciones:

1. Aplicar las reglas descritas en la sección 4.1.1 para listar los EI de la aplicación.
2. Aplicar las reglas de la sección 4.1.2 y obtener los **DET** y los **FTR** para cada EI identificado en el punto anterior.
3. Con base en la lista de EI con sus DET y FTR y la “Matriz de complejidad para los EI” (refiérase a tabla 4) determinar la complejidad de cada EI. Por ejemplo: supongamos que la aplicación requiere una pantalla que requiere relacionar dos tablas por lo tanto tiene 2 FTR. Además tiene 10 DET con campos de ambas tablas y 2 DET porque va a tener al menos un mensaje de error y una tecla de comando. Con un total de 2 FTR y 12 DET y utilizando la tabla “Matriz de complejidad para los EI” esa pantalla tendrá una complejidad **Media**. Otra pantalla podría requerir la relación de 3 tablas por lo que tienen 3 FTR. Está compuesta de 17 DET incluyendo 2 DET de mensaje de error y de tecla de comando. Esta segunda pantalla tendrá una complejidad de **Alta**.

Tabla 4. Matriz de complejidad para los EI.

Número de Archivos Referenciados (FTR)	Número de Elementos de Datos (DET)		
	1-4	5-15	≥16
<2	Baja	Baja	Media
2	Baja	Media	Alta
>2	Media	Alta	Alta

4. Una vez determinada la complejidad de cada EI se le debe asignar un peso. Con base en la tabla 5 se ubica la fila de los EI y columna correspondiente a la complejidad (alta, media o baja) obtenida en el punto anterior y se multiplica el número de EI por el peso que le corresponde. Siguiendo el ejemplo: teníamos 1 EI con complejidad **Media** y 1 EI con complejidad **Alta**, obtenidos en el punto anterior. Podemos contabilizar **10** puntos de función desajustados para los EI.

Tabla 5. Tabla de conteo para los EI.

Componentes	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	x7	x10	x15
Archivos de Interfase Externa (EIF)	x5	x7	x10
Entradas Externas (EI)	x3	1x4	1x6
Salidas Externas (EO)	x4	x5	x7
Consultas Externas (EQ)	x3	x4	x6
		4	6
Total	10 puntos de función sin ajustar (PFSA)		

2. Definición de las Salidas Externas (EO, en inglés “External Output”)

Una salida externa (EO) es un proceso elemental de la aplicación que envía datos o información de control fuera de las fronteras de la aplicación. El objetivo principal de una EO es presentar información al usuario final a través del procesamiento de dicha información, además de recuperar los datos o información de control. La lógica de procesamiento debe contener al menos una fórmula matemática o un cálculo, o crea datos derivados, o mantiene uno o más ILF y/o altera la conducta del sistema.

A continuación se describen los siguientes términos:

Un *proceso elemental* es la unidad de actividad más pequeña que tiene significado para el usuario. El proceso debe estar autocontenido, es decir no debe su existencia a algún otro proceso, y debe dejar el negocio en un estado consistente una vez que finaliza. Por ejemplo:

un reporte puede consistir de muchas páginas pero si las páginas no son requeridas y producidas independientemente, el reporte solo cuenta un EO. Producir alguno de los campos, aún una página, podría no ser autocontenido ni dejar el negocio en un estado consistente. Si toda la información, reconociendo que algunos campos no se llenan y pueden ser nulos o blancos, fueron completados, esta transacción está completa y el negocio podría quedaría en un estado consistente.

Los *datos* se refieren a un conjunto de hechos y/o figuras procesados por una transacción de salida. Por ejemplo: los datos podrían ser campos de datos incluidos en el reporte descrito anteriormente. Podríamos esperar ver campos como: el nombre del departamento, el número del departamento, la dirección, el mes, el total de ventas mensuales, el total de compras y el total por año.

La *información de control* es un dato usado por la aplicación para influir en un proceso elemental de la aplicación que está siendo contada; en el caso de los EO, estos datos, reglas o parámetros pueden ser enviados por la aplicación a un usuario u otra aplicación. Por ejemplo información de control puede ser enviada por la aplicación para asegurar el cumplimiento de los requerimientos especificados por el usuario. Esto podría incluir un aviso al usuario que advierta ciertos controles internos que se han puesto, tal como una alarma.

La *lógica de procesamiento* incluye específicamente los requerimientos solicitados por un usuario para completar un proceso elemental. Usualmente se requiere una combinación de lógica de procesamiento para completar un proceso elemental. Una lógica de procesamiento por sí misma, no se debe usar para determinar la exclusividad de un EI, EO o EQ. Un proceso elemental de un EO puede incluir múltiples validaciones, filtros, ordenamientos, etc. Clasificar, ordenar o formatear un conjunto de datos no se considera una lógica de procesamiento exclusiva, sin embargo podría haber otras lógicas de procesamiento exclusivas durante cualquiera de esas operaciones. Como ejemplo un reporte puede ordenar totales por mes, producto y departamento. Si los campos del reporte fueran los mismos (pero desplegados en formato diferente), los totales fueran por columna y el cálculo no fuera exclusivo, podríamos contar un EO. Múltiples EO podrían ser contados si los totales fueran sumados en forma diferente y los cálculos fueran únicos para cada ordenamiento. Quizás los totales internos podrían representar un total del producto por mes dentro de un departamento, un total mensual de un departamento por división y las ventas diarias del mes.

Lógicas de procesamiento para un EO podrían ser:

- Validaciones.
- Fórmulas o cálculos matemáticos.
- Conversiones de valores equivalentes.
- Filtros y selección de datos usando criterios específicos para comparar múltiples grupos de datos.
- Análisis de condiciones para determinar cuál es aplicable.
- Actualización de ILF.
- Referencias a ILF o EIF.
- Recuperación de datos o información de control.
- Creación de datos derivados.

- Alteración de la conducta del sistema.
- Preparación y presentación de información fuera de la frontera (obligatorio para un EO).
- Capacidad de aceptar datos o información de control que ingrese a la aplicación.
- Ordenamiento o clasificación de un conjunto de datos.

Los *datos derivados* requieren procesamiento, en lugar de recuperación directa, conversión y edición de la información, de uno o más ILF o EIF. El dato derivado es creado por la transformación de datos existentes, para crear un dato adicional. Frecuentemente los EO contienen datos derivados.

Administrar se refiere a la habilidad de modificar datos a través de un proceso elemental. Por ejemplo, automáticamente se podría ingresar un número de cheque en un ILF, durante el proceso de generar los cheques de una planilla.

2.1. Reglas para contar los EO.

Para los datos o información de control que está siendo contada como EO, todas las siguientes reglas del IFPUG se deben aplicar:

- **Los datos o información de control deben salir de la frontera de la aplicación.**
- **La lógica de procesamiento del proceso elemental de un EO debe realizar uno de los siguientes:**
 - 1. Contener al menos una fórmula matemática.**
 - 2. Crear datos derivados.**
 - 3. Mantener al menos un ILF.**
 - 4. Alterar la conducta de la aplicación.**
- **El proceso debe ser la unidad más pequeña de actividad reconocible para el usuario.**
- **El proceso debe estar autocontenido y deja el negocio en un estado consistente.**
- **Para identificar el proceso se debe aplicar una de las tres siguientes reglas:**
 - 1. La lógica del proceso debe ser única/diferente respecto a las otras EO dentro de la aplicación (no hay dos procesos iguales).**
 - 2. El conjunto de elementos de datos identificados es diferente de los conjuntos identificados por otras EO dentro de la aplicación.** Note que hay diferentes elementos de datos y no diferentes datos en los mismo campos. Por ejemplo un reporte de cuenta producido para personas que tienen diferentes datos en los mismos campos son contados como un EO. Dos reportes producidos en forma separada a nivel de detalle y a nivel de resumen, pueden ser contados como dos EO porque tienen una lógica de procesamiento y cálculos únicos.
 - 3. Los ILF o EIF referenciados son diferentes de aquellos referenciados por otros EO dentro de la aplicación.**

Algunos ejemplos adicionales de EO se presentan a continuación. Estos son solamente ejemplos; cualquier dato o información de control listada aquí debe corresponder con la definición indicada y con las reglas para contar EO:

- Reportes que requieren el uso de algoritmos o cálculos. Por ejemplo un reporte de cuentas corrientes por semana o mensual.
- Transferencia de datos, archivos y/o mensajes que se envían a otra aplicación cuando el dato es calculado o derivado y la actualización ocurre como parte del proceso elemental. Por ejemplo un archivo de transacciones se envía desde una aplicación de “Cuentas por cobrar” a una aplicación “Mayor general” (administrada en forma separada) cuando algún total es calculado u ocurre una entrada a la base de datos de la aplicación de “Cuentas por cobrar” para indicar cuál transacción fue enviada (se cuenta como un EO para la aplicación de “Cuentas por cobrar” y como un EI para la “Aplicación General”).
- Un cheque que cuando se creó, simultáneamente actualizó el registro de cheques con el número de cheques (este es un proceso elemental).
- Un reporte de conversión que reporta el esfuerzo de conversión cuando los datos están migrando como parte de un proyecto en desarrollo o mantenimiento; esto debería ser incluido como parte del conteo del proyecto pero no dentro del conteo de la aplicación.
- Información derivada o calculada desplegada sobre una pantalla o o pasada en un archivo.
- Información derivada/calculada desplegada sobre una pantalla.
- Despliegues gráficos tales como gráfico de barras, gráfico pastel cuando ellos requieren cálculos.
- Respuestas calculadas devueltas por el teléfono.
- Una notificación de que una tarjeta de crédito ha sido reportada como perdida.
- Cálculo de un premio propuesto por una política de seguros.

Los siguientes son ejemplos que frecuentemente por error se cuentan como EO:

- Reportes idénticos con valores de datos diferentes (se cuenta solo una vez) tales como reportes de departamentos.
- Reportes que no tienen fórmulas, cálculos o datos derivados y no administran un ILF dentro de la aplicación que está enviando datos (más probable es que se cuente como un EQ).
- Campos reaumenten dentro de un reporte de detalles (el reporte de detalles es el EO).
- Archivos enviados a otra aplicación que no tiene fórmulas, cálculos o datos derivados y no administran un ILF dentro de la aplicación que está enviando datos (más probable es que se cuente como un EQ).
- Múltiples medios cuando la lógica de procesamiento no es diferente.
- Pantallas de refrescamiento o cancelación de datos (no se cuenta).
- Reordenamiento o clasificación de un conjunto de datos sin otra lógica de procesamiento.

- Un dato referenciado leído por otra aplicación desde el almacén de datos en la aplicación que se está contando (el dato no se cuentan como EO por la aplicación contada).
- Ayudas (contadas como EQ)
- Sistema de ingreso.
- Métodos múltiples para invocar el mismo proceso de salida.
- Mensajes de error que resultan de una edición o validación de un EI o la parte de solicitud de un EQ o EO.
- Mensajes de confirmación que reconoce que los datos se han procesado.
- Mensajes que solicita al usuario confirmar un borrado o cualquier otra transacción.
- Datos idénticos que se envían a más de una aplicación (contado solo una vez).
- Reportes ad-hoc que los usuarios dirigen o controlan a través de lenguajes como SQL o FOCUS.
- Datos enviados en línea o en lote dentro de la misma aplicación; no cruzan la frontera de la aplicación.
- Datos enviados entre el cliente y el servidor dentro de la misma aplicación; no cruzan la frontera de la aplicación.

2.2 Reglas para determinar la complejidad y Contribución de los EO

El conteo físico de los EO junto con la complejidad funcional relativa de cada uno, determina la contribución de las EO para el conteo de los puntos de función desajustados. Cada EO identificado debe ser asignado a la complejidad funcional basada en el número del tipo de elementos de datos (DET) y los archivos referenciados (FTR) asociados con el EO.

A continuación se describen los siguientes términos:

Complejidad funcional es un peso asignado a cada función transaccional. De acuerdo a la matriz de complejidad la calificación puede ser alta, media o baja considerando el número de DET y RET.

DET son usualmente campos o atributos únicos, reconocibles por el usuario, no repetitivos que cruzan la frontera de la aplicación.

FTR son simplemente archivos referenciados. Se refieren al número total de ILF leídos o administrados y de EIF leídos por la transacción EO.

Las siguientes reglas del IFPUG se deben aplicar para contar los elementos de datos, campos, o atributos (conocidos como DET) para los EO:

- **Se cuenta un DET para cada campo reconocible por el usuario, no repetitivo que ingresa la frontera de la aplicación y requiere especificar qué, cuándo y/o cómo el dato es recuperado y /o generado por un proceso elemental.** Tales campos a menudo se consideran como información de control, información de selección, o parámetros de procesamiento.

- **Se cuenta un DET para cada campo reconocible por el usuario, no repetitivo o a tributo que sale de la frontera de la aplicación.** Se incluyen atributos de llaves foráneas e información de control.
- **Si un DET ingresa o sale de la frontera se cuenta solo una vez para el proceso elemental.**
- **Se cuenta un DET por la capacidad de enviar un mensaje de respuesta a un sistema que está fuera de la frontera de la aplicación para indicar que ha ocurrido un error durante el procesamiento, confirmar que el procesamiento está completo, o verificar que el proceso debe continuar.** Aunque hayan múltiples posibilidades de mensaje, se cuenta solo un DET para el proceso elemental. Esto es, solamente se cuenta un DET para todos los mensajes, aunque hayan múltiples mensajes de error, verificación y confirmación.
- **Se cuenta un DET por la capacidad de especificar una acción a tomar por el EO aún si hay métodos múltiples o llaves múltiples requeridos para invocar una misma lógica de procesamiento.** Por ejemplo: botones de aceptación, teclas de función, teclas de acción, o un “click” al ratón.
- **No contar variables de paginación generadas por el sistema, incluyendo: el número de página, información de las posiciones de filas y columnas en pantallas o reportes, comandos de paginación (como: previo, próximo, flechas) o campos de hora/fecha.** Se cuentan DET para incluir un campo de fecha recuperado, pero no las fechas generadas por el sistema, tal como cuándo se imprime el reporte.
- **No contar literales como DET, incluyendo: títulos de reportes, identificadores de pantallas, encabezados de columnas, o títulos de campos.**
- **No contar campos que son administrados por el ILF durante el proceso elemental del EO si los campos no cruzan la frontera.**
- **Contar un DET para cada campo lógico que es almacenado físicamente como campo múltiples pero que es requerido por el usuario como una única pieza de información.** Por ejemplo un dato o nombre almacenado en tres campos pero usado como un campo se cuenta como un DET. Una dirección (calle, ciudad y código postal) puede ser contado como un DET sobre una etiqueta de dirección si este es considerado para representar un grupo de datos.
- **Contar un DET para cada tipo de etiqueta y cada tipo de equivalente numérico en un gráfico.** Por ejemplo un gráfico pastel puede tener dos DET, uno para la categoría y uno para el porcentaje aplicado, aún si el porcentaje no tiene etiqueta.
- **Contar un DET para cada información de texto que consista de una simple palabra, una oración, o muchos párrafos.**

Las siguientes reglas de IFPUG se aplican para contar tipos de archivos referenciados (conocidos como FTR por IFPUG) para los EO:

- **Contar un FTR por cada ILF administrado o EIF leído durante el procesamiento del EO.**
- **Contar un FTR por cada ILF administrado por el proceso elemental del EO.**
- **Contar solamente un FTR para un ILF que es tanto leído como administrado por el EO.**

2.3 Determinar los puntos de función sin ajustar (PFSA) para los EO.

Para determinar los puntos de función sin ajustar de los EO se deben seguir las siguientes instrucciones:

1. Aplicar las reglas descritas en la sección 4.2.1 para listar los EO de la aplicación.
2. Aplicar las reglas de la sección 4.2.2 y obtener los **DET** y los **FTR** para cada EO identificado en el punto anterior.
3. Con base en la lista de EO con sus DET y FTR y la “Matriz de complejidad para los EO” (refiérase a la tabla 6) determinar la complejidad de cada EO. Por ejemplo: supongamos que la aplicación requiere un reporte que requiere relacionar dos tablas, realizar un cálculo y actualizar un campo de 1 de las 2 tablas, por lo tanto tiene 2 FTR. Las características de actualizar las tablas y de realizar los cálculos lo ubica como 1 EO y no como EQ. Además presentará más de 20 DET con campos de ambas tablas, incluyendo 1 DET de al menos un mensaje de error y 1 DET de una tecla de comando. Con un total de **2 FTR y más de 20 DET** y utilizando la tabla “Matriz de complejidad para los EO” ese reporte tendrá una complejidad **Alta**. Otro reporte podría requerir la relación de 3 tablas y también actualiza alguna de las tablas de la aplicación. El reporte tendrá 3 FTR y aproximadamente entre 6 y 19 DET, incluyendo 2 DET de mensaje de error y de tecla de comando. Este segundo reporte tendrá una complejidad **Media**.

Tabla 6. Matriz de complejidad para los EO.

Número de Archivos Referenciados (FTR)	Número de Elementos de Datos (DET)		
	1-5	6-19	≥20
<2	Bajo	Bajo	Medio
2-3	Bajo	Medio	Alto
>3	Medio	Alto	Alto

4. Una vez determinada la complejidad de cada EO se le debe asignar un peso. Con base en la siguiente tabla se ubica la fila de los EO y columna correspondiente a la complejidad (alta, media o baja) obtenida en el punto anterior y se multiplica el número de EO por el peso que le corresponde. Siguiendo el ejemplo: teníamos 1 EO con complejidad **Media** y 1 EO con complejidad **Alta**, obtenidos en el punto anterior. Podemos contabilizar **12** puntos de función desajustados para los EO.

Tabla 7. Conteo para los EO.

Componentes	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	x7	x10	X15
Archivos de Interfase Externa (EIF)	x5	x7	X10
Entradas Externas (EI)	x3	x4	X6
Salidas Externas (EO)	x4	1x5	1X7
Consultas Externas (EQ)	x3	x4	X6

		5	7
Total	12 Puntos de función sin ajustar (PFSA)		

3. Definición de las Consultas externas (EQ, en inglés “External Inquiries”)

Una consulta externa (EQ) es un proceso elemental de la aplicación que resulta de recuperar un dato o información de control que se envía fuera de la frontera de la aplicación. El objetivo principal de una EQ es presentar información al usuario a través de la recuperación de datos o información de control desde un ILF o un EIF. La lógica de procesamiento no posee fórmulas matemáticas ni cálculos y no se crea datos derivados. Una EQ no administra un ILF durante su procesamiento, ni altera la conducta del sistema.

A continuación se describen los siguientes términos:

Un *proceso elemental* es la unidad de actividad más pequeña que tiene significado para el usuario. El proceso debe estar autocontenido, y dejar el negocio un estado consistente. Por ejemplo: para realizar una recuperación particular los usuarios han solicitado tener la capacidad de ingresar 5 campos de información de control – por ejemplo: un modelo, color, estilo, año de producción y garantía. Cualquier número de combinaciones puede ser posible, pero cualquier de las 5 puede ser seleccionada como una selección “y/o”. Ingresar uno de los campos no es un proceso elemental ni ingresar los 5. Posibles campos que satisfagan los criterios de la información de control ingresada deben ser extraídos y presentados al solicitante, antes de que la transacción se considere un proceso elemental. Una EQ involucra una solicitud de información (generada internamente desde un proceso de la aplicación o externa a la aplicación desde un proceso, persona, cosa u otra aplicación), un extracto de uno o más ILF o EIF, y la entrega de información al usuario; sin las 3 subactividades la transacción no está completa.

Los *datos* se refieren a campos de información procesados por la EQ. Por ejemplo: podrían haber campos de datos recuperados y desplegados como resultado de la consulta descrita anteriormente - modelo, color, estilo, año de producción y garantía.

La *información de control* es un dato que influye en un proceso elemental de la aplicación que está siendo contada; en el caso de los EQ, estos datos, reglas o parámetros especifican qué, cuándo o cómo el dato va a ser procesado. La información de control no es un proceso elemental por sí mismo.

La *lógica de procesamiento* incluye aquellos requerimientos solicitados específicamente por un usuario para completar un proceso elemental. Usualmente se requiere una combinación de lógica de procesamiento para completar un proceso elemental. Una lógica de procesamiento por sí misma, no se debería usar para determinar la exclusividad de un EI, EO o EQ. Un proceso elemental de un EQ puede incluir muchas validaciones, filtros, ordenamientos, etc. Clasificar, ordenar o formatear un conjunto de datos no se considera

una lógica de procesamiento exclusiva, sin embargo podría haber otras lógicas de procesamiento exclusivas durante cualquiera de esas operaciones.

Los *datos derivados* requieren procesamiento en lugar de recuperar, convertir y editar información, de uno o más ILF o EIF. El dato derivado es creado por la transformación de datos existentes para crear datos adicionales. Los EQ no contienen datos derivados.

Administrar se refiere a la capacidad de modificar datos a través de un proceso elemental. Un EQ no administra datos, un EI o un EO administran datos.

3.1 Reglas para contar los EQ.

Para datos o información de control ser contada como EQ, las siguientes reglas de IFPUG se deben aplicar:

- Los datos o información de control deben salir de la frontera de la aplicación.
- Los datos o información de control deben ser recuperados de uno o más ILF o EIF.
- La lógica de procesamiento del proceso elemental no debe crear datos derivados.
- La lógica de procesamiento del proceso elemental no debe contener fórmulas matemáticas o cálculos.
- La lógica de procesamiento del proceso elemental no debe alterar la conducta de la aplicación.
- El proceso no debe administrar un ILF.
- El proceso debe ser la unidad más pequeña de actividad que tiene significado para el usuario.
- El proceso debe estar autocontenido y dejar la aplicación en un estado consistente.
- Para el proceso identificado se debe aplicar una de las tres siguientes reglas:
 1. La lógica de procesamiento debe ser única/diferente de las lógicas de procesamiento de otras EQ dentro de la aplicación. (No hay dos procesos iguales)
 2. El conjunto de elemento de datos identificado es diferente de los conjuntos identificados en otras EQ dentro de la aplicación (No hay dos flujos iguales).
 3. Los ILF o EIF referenciados son diferentes de aquellos referenciados por otras EQ dentro de la aplicación.

Algunos ejemplos adicionales de EQ se presentan a continuación. Estos son solamente ejemplos; cualquier dato o información de control listada aquí debe corresponder con la definición indicada y con las reglas para contar EQ:

- Un dato transaccional que es recuperado desde uno o más ILF y/o EIF y desplegado -- por ejemplo: un puesto, una descripción de un elemento, un dato de empleado, un dato de pago.

- Las funciones de usuario tales como ver, desplegar, listar, imprimir (recuerde que una impresión y una vista con la misma lógica de procesamiento son contadas como una EQ y no como dos).
- Las consultas implícitas (recuperaciones de datos antes de una función de modificación o borrado) que hacen que la consulta pueda ser usada como un proceso único y no una duplicación de otro previamente contada como EQ.
- Los reportes generados sobre una base periódica que no contiene fórmulas, cálculos o datos derivados y no administran un ILF.
- Los datos del sistema, parámetros y la configuración a menos que sean calculados.
- Las pantallas de ingreso que provean una seguridad específica a la aplicación.
- Cualquier nivel de ayuda; por ejemplo, un campo, o una pantalla cuando se recuperó de un ILF O EIF.
- Recuperaciones de datos administrados vía electrónica o telefónica.
- Archivos Enviados a otra aplicación que no contiene fórmulas, cálculos o datos derivados y no administran un ILF dentro de la aplicación que está enviando los datos (la transacción puede ser un EI para la aplicación receptora).
- Recuperación de correo desde el “mailbox”.
- Cajas de listas o posicionamiento de un dato sobre la pantalla para devolver un dato administrado desde un ILF o EIF.

Los siguientes son ejemplos de aspectos que frecuentemente mal identificados como EQ:

- Métodos múltiples para invocar la misma lógica; por ejemplo dos teclas que realizan la misma acción o la misma transacción sobre múltiples pantallas (contarlas solamente una vez).
- Métodos múltiples cuando la lógica de procesamiento no es diferente.
- Consultas que pueden ser accesadas desde múltiples áreas o pantallas de una aplicación (se cuentan solo una vez).
- Pantallas de menú usadas para navegar y seleccionar, y no recuperan datos administrados (no contarlas).
- Pantallas para ingresar que facilitan la entrada al usuario en la aplicación pero no invocan seguridad (no contarlas).
- Datos derivados o calculados en oposición a recuperación de datos (contarlos como un EO).
- Ordenamiento o clasificación de un grupo de datos sin otra lógica de procesamiento.
- Respuestas a mensajes que requieren una confirmación del usuario.
- Mensajes de error y/o confirmación.
- Documentación del sistema en línea.
- Datos que pasan entre línea y lote dentro de la misma aplicación, no cruzan la frontera de la aplicación.

- Datos que pasan entre el cliente y el servidor dentro de una misma aplicación, no cruzan la frontera de la aplicación.
- Datos que no son recuperados de los datos administrados. Por ejemplo: datos hard-coded (no contarlos).

3.2 Reglas para determinar la complejidad y Contribución de los EQ

El conteo físico de los EQ junto con la complejidad funcional relativa de cada uno, determina la contribución de las consultas externas al conteo de los puntos de función desajustados. A cada EQ identificado se le debe asignar una complejidad funcional basada en el número de DET y FTR asociados con el EQ.

A continuación se describen los siguientes términos:

Complejidad funcional es un peso asignado a cada función transaccional. De acuerdo a la matriz de complejidad la calificación puede ser alta, media o baja considerando el número de DET y FTR.

DET son usualmente campos o atributos únicos, reconocibles por el usuario, no repetitivos o atributos que cruzan la frontera de la aplicación.

FTR son simplemente archivos referenciados. Se refieren al número total de ILF y EIF leídos por la transacción EQ.

Las siguientes reglas IFPUG se deben aplicar para contar los elementos de dato, campos o atributos (conocidos como DET por IFPUG) para los EQ:

- **Se cuenta un DET para cada campo reconocible por el usuario, no repetitivo que ingresa la frontera de la aplicación y requiere especificar qué, cuándo y/o cómo el dato es recuperado y /o generado por el proceso elemental.** Tales campos a menudo se consideran como información de control, información de selección, o parámetros de procesamiento.
- **Se cuenta un DET para cada campo reconocible por el usuario, no repetitivo o a tributo que sale de la frontera de la aplicación.** Se incluyen atributos de llaves foráneas e información de control.
- **Si un DET ingresa o sale de la frontera se cuenta solo una vez para el proceso elemental.**
- **Se cuenta un DET por la capacidad de enviar un mensaje de respuesta fuera de la frontera de la aplicación para indicar que ha ocurrido un error durante el procesamiento, confirmar que el procesamiento está completo, o verificar que el proceso debe continuar.** Aunque hayan múltiples posibilidades de mensaje, se cuenta solo un DET para el proceso elemental. Esto es, solamente se cuenta un DET para todos los mensajes, aunque hayan múltiples mensajes de error, verificación y confirmación.
- **Se cuenta un DET por la capacidad de especificar una acción a tomar por el EQ aún si hay métodos múltiples o llaves múltiples requeridas para invocar una misma lógica de procesamiento.** Por ejemplo: botón de aceptación, teclas de función, teclas de acción, o un “clic” al ratón.

- **No contar variables de paginación generadas por el sistema, incluyendo número de página, información de posiciones de filas y columnas en pantallas o reportes, comandos de paginación (como: previo, próximo, flechas) o campos de hora/fecha.** Se cuenta un DET por incluir un campo de fecha recuperado, pero no las fechas generadas por el sistema, tal como cuándo un reporte es impreso.
- **No contar literales como DET, incluyendo: títulos de reportes, identificadores de pantallas, encabezados de columnas, o títulos de campos.**
- **No contar campos que son administrados por el ILF durante el proceso elemental del EQ si los campos no cruzan la frontera.**
- **Contar un DET para cada campo lógico que es almacenado físicamente como campo múltiples pero que es requerido por el usuario como una única pieza de información.** Por ejemplo un dato o nombre almacenado en tres campos pero usado como un campo se cuenta como un DET. Una dirección (calle, ciudad y código postal) puede ser contado como un DET sobre una etiqueta de dirección si este es considerado para representar un grupo de datos.
- **Contar un DET para cada tipo de etiqueta y cada tipo de equivalente numérico en un gráfico.** Por ejemplo un gráfico pastel puede tener dos DET, uno para la categoría y uno para el porcentaje aplicado, aún si el porcentaje no tiene etiqueta. Un gráfico puede ser un EQ si no hay cálculos cuando el gráfico es generado: esto es el porcentaje es leído de datos almacenados.
- **Contar un DET por cada información de texto que consista de una simple palabra, una oración, o muchos párrafos.**

La siguiente regla de IFPUG se aplica para contar los tipos de archivos referenciados (conocidos como FTR por IFPUG) para los EQ:

- **Contar un FTR por cada ILF administrado o EIF leído durante el procesamiento del EQ.**

3.3 Determinar los puntos de función sin ajustar (PFSA) para los EQ.

Para determinar los puntos de función sin ajustar de los EQ se deben seguir las siguientes instrucciones:

1. Aplicar las reglas descritas en la sección 4.3.1 para listar los EQ de la aplicación.
2. Aplicar las reglas de la sección 4.3.2 y obtener los **DET** y los **FTR** para cada EQ identificado en el punto anterior.
3. Con base en la lista de EQ con sus DET y FTR y la “Matriz de complejidad para los EQ” (refiérase a la tabla 8) determinar la complejidad de cada EQ. Por ejemplo: supongamos que la aplicación requiere una consulta que requiere relacionar dos tablas, por lo tanto tiene 2 FTR. Además presentará más de 20 DET con campos de ambas tablas, incluyendo 1 DET de al menos un mensaje de error y 1 DET de una tecla de comando. Con un total de **2 FTR y más de 20 DET** y utilizando la tabla “Matriz de complejidad para los EQ” ese reporte tendrá una complejidad **Alta**. Otra consulta podría requerir la relación de 3 tabla y aproximadamente entre 6 y 19 DET, incluyendo 2 DET de mensaje de error y de tecla de comando. Esta segunda consulta tendrá una complejidad **Media**.

Tabla 8. Matriz de complejidad para los EQ.

Archivos Referenciados (FTR)	Datos (DET)		
	1-5	6-19	≥20
1	Bajo	Bajo	Medio
2-3	Bajo	Medio	Alto
>3	Medio	Alto	Alto

4. Una vez determinada la complejidad de cada EO se le debe asignar un peso. Con base en la siguiente tabla se ubica la fila de los EO y columna correspondiente a la complejidad obtenida en el punto anterior y se multiplica el número de EO por el peso que le corresponde. Siguiendo el ejemplo: teníamos 1 EO con complejidad **Media** y 1 EO con complejidad **Alta**, obtenidos en el punto anterior. Podemos contabilizar **12** puntos de función desajustados para los EO.

Tabla 9. Conteo para los EQ.

Componentes	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	x7	x10	x15
Archivos de Interfase Externa (EIF)	x5	x7	x10
Entradas Externas (EI)	x3	x4	x6
Salidas Externas (EO)	x4	x5	x7
Consultas Externas (EQ)	x3	1x4	1x6
		4	6
Total	10 Puntos de Función sin Ajustar (PFSA)		

Paso 5. Determinar el Factor de Ajuste de Valor (FAV) a través de las 14 Características Generales

Las funciones propias de los sistemas de información incluyen factores generales que no son suficientemente representadas por las funciones transaccionales y las funciones de datos. Esta sección describe las definiciones, reglas y guías para identificar el Factor de Ajuste de Valor (FAV), el cual es usado como un multiplicador de los Puntos de Función Desajustados para calcular los Puntos de Función Ajustados. El cálculo del FAV se basa en la identificación de 14 Características Generales del Sistema.

Las Características Generales del Sistema deben ser evaluadas en términos del Grado Total de Influencia (GTI) sobre una escala de 0 a 5:

- 0 Factor no presente o ninguna influencia
- 1 Influencia insignificante

- 2 Influencia moderada
- 3 Influencia media
- 4 Influencia significativa
- 5 Influencia fuerte

A continuación se describen las 14 características generales con sus correspondiente Grado Total de Influencia.

1. Comunicación de Datos

Los datos y la información de control usados en la aplicación son enviados o recibidos a través de las facilidades de comunicación. El peso se aplica como sigue:

- 0 La aplicación es procesamiento en lote puro o solo PC. No hay interactividad.
- 1 La aplicación es en lote pero tiene capacidad de ingreso remoto o de impresión remota.
- 2 La aplicación es en lote pero tiene capacidad de ingreso remoto y de impresión remota.
- 3 La aplicación incluye un conjunto de datos en línea o teleprocesamiento (TP) front-end hacia un proceso en lote o sistema de consulta. Son aplicaciones que tienen pantallas de entrada de datos front-end, pero actualizan archivos lógicos internos a través de un procesamiento en lote.
- 4 La aplicación es más que un front-end, pero soporta solo un tipo de protocolo de comunicación TP. Igual que 3 pero la actualización ocurre interactivamente.
- 5 La aplicación es más que un front-end, pero soporta más de un tipo de protocolo de comunicación TP.

Por lo general las aplicaciones en lote reciben un peso de 0 a 3, en línea de 3 a 4 y en tiempo real, telecomunicación o sistemas de control de procesos reciben de 4 a 5.

2. Procesamiento de Datos Distribuido

Es el grado en que una aplicación transfiere datos entre los componentes de la aplicación. El peso se aplica como sigue:

- 0 La aplicación no soporta la transferencia de datos o funciones de procesamiento entre los componentes del sistema. Totalmente monolítica.
- 1 La aplicación prepara los datos para que el usuario los procese en otro componente del sistema, tal como una hoja electrónica o un BBMS en una PC.
- 2 La aplicación prepara los datos para transferirlos, entonces son transferidos y procesados sobre otro componente del sistema (no para procesamiento de usuario final).
- 3 El procesamiento es distribuido y la transferencia de datos es en línea y en una sola dirección.

4 El procesamiento es distribuido y la transferencia de datos es en línea y en ambas direcciones.

5 Las funciones de procesamiento son dinámicamente ejecutadas en muchos componentes del sistema.

Solo las aplicaciones distribuidas o en tiempo real tienen peso. Las distribuidas primitivas pueden ser 1 o 2. Las de cliente-servidor o Web son de 2 a 4. Un peso de 5 puede asignarse cuando son múltiples servidores o procesadores, cada uno de los cuales se ha seleccionado dinámicamente sobre la base de disponibilidad en tiempo real.

3. Rendimiento

Describe el grado en que el tiempo de respuesta y consideraciones de rendimiento influyen en el desarrollo de la aplicación. Los objetivos de rendimiento son indicados o aprobados por el usuario en respuesta al diseño, desarrollo, instalación y soporte de la aplicación. El peso se aplica como sigue:

0 No hay requerimientos de rendimiento especiales indicados por el usuario.

1 Los requerimientos de rendimiento y diseño fueron indicados y revisados por el usuario pero no se requieren acciones especiales. El retraso en el procesamiento se deja para el siguiente día comercial.

2 El tiempo de respuesta es crítico durante las horas pico pero no se requiere un diseño especial para la utilización del CPU.

3 El tiempo de respuesta es crítico durante todas las horas comerciales, pero no se requiere un diseño especial para la utilización del CPU. Los requerimientos en el retraso de procesamiento se consideran como restricciones.

4 Además, los requerimientos de rendimiento indicados por el usuario son lo suficientemente críticos como para realizar análisis del rendimiento durante el diseño.

5 Además, las herramientas de análisis de rendimiento fueron utilizadas durante el diseño, el desarrollo y/o implementación para alcanzar los requerimientos de rendimiento indicados por el usuario.

Esta característica se parece mucho a la 5, pero requiere consideraciones de rendimiento durante las fases del desarrollo de la aplicación. El tiempo de respuesta se relaciona típicamente con el procesamiento interactivo, y el throughput con el procesamiento en lote. Un peso de 4 requiere tareas de análisis de rendimiento durante la fase de diseño. Un peso de 5 requiere el uso de herramientas de análisis de rendimiento.

4. Configuración muy utilizada (procesamiento de equipo muy utilizado)

Una configuración operacional pesada describe el grado en que las restricciones de recursos computacionales influyen en el desarrollo de la aplicación. Una configuración operacional pesada requiere consideraciones especiales de diseño. El peso se aplica como sigue:

0 No hay restricciones operacionales implícitas o explícitas incluidas.

1 Las restricciones operacionales existen pero son menos restrictivas que una aplicación típica. Ningún efecto especial es necesario para alcanzarlas.

2 Existen algunas consideraciones de seguridad o de tiempo.

3 Hay requerimientos de procesador específicos para una parte de la aplicación.

4 Las restricciones de operación indicadas requieren restricciones especiales sobre la aplicación en un procesador central o en un procesador dedicado.

5 Además, hay restricciones especiales sobre la aplicación en los componentes distribuidos del sistema.

5. Tasa Transaccional

La tasa transaccional describe el grado en el que la tasa de transacción comercial influye en el diseño, desarrollo, instalación y soporte de la aplicación. El peso se aplica como sigue:

0 No se anticipa un período pico transaccional.

1 Se anticipa un período transaccional pico mensualmente, trimestralmente, semestralmente o anualmente.

2 Un período transaccional pico se anticipa.

3 Un período transaccional pico se anticipa diariamente.

4 Los requerimientos de porcentaje transaccional indicados por el usuario son lo suficientemente críticos como para requerir análisis del rendimiento durante el diseño.

5 Además, se requiere el uso de herramientas de análisis de rendimiento durante el diseño, el desarrollo y la implementación para alcanzar los requerimientos del usuario.

6. Ingreso de Datos en Línea

El ingreso de datos en línea describe el grado en el que los datos se ingresan a través de transacciones interactivas. La entrada de datos en línea y las funciones de control son provistas en la aplicación. El peso se aplica como sigue:

0 Todas las transacciones son procesadas en lote.

1 De un 1% a un 7% de las transacciones son entrada de datos interactiva.

2 De un 8% a un 15% de las transacciones son entrada de datos interactiva.

3 De un 16% a un 23% de las transacciones son entrada de datos interactiva.

4 De un 24% a un 30% de las transacciones son entrada de datos interactiva.

5 Más de un 30% de las transacciones son entrada de datos interactiva.

Las transacciones se refieren a los EI, EO y EQ de la aplicación. Uno de los problemas de GSCs es que la guía no se ha actualizado en años. Consecuentemente

los pesos no son realísticos, sin embargo, los datos de la industria se han calculado utilizando estas guías. Típicamente las aplicaciones en lote reciben de 0 a 1; y los sistemas en línea, en tiempo real, telecomunicaciones, o control de procesos un peso de 5.

7. Eficiencia del Usuario Final

Describe el grado de consideración de factores humanos y facilidad de uso para el usuario de la aplicación que se está midiendo. Las funciones en línea enfatizan un diseño específico para lograr la eficiencia del usuario final. Esto incluye lo siguiente:

- Ayuda de navegación (por ejemplo: teclas de función, menús generados dinámicamente)
- Menús.
- Documentación de ayuda en línea.
- Movimiento del cursor automático.
- “Scroolling”.
- Impresora remota.
- Teclas de función preasignadas.
- Tareas en lote enviadas desde transacciones en línea.
- Interfases con “mouse”.
- Ventanas pop-up.
- Mínimo de ventanas como sea posible para realizar las funciones comerciales.
- Uso pesado de video en reverso, colores, subrayados y otros indicadores.
- Documentación de usuario “hard-copy” de transacciones en línea.
- Soporte bilingüe (si soporta 2 lenguajes se cuenta como 4 items).
- Soporte multilenguaje (si soporta más de 2 lenguajes se cuenta como 6 items).

El peso se aplica como sigue:

0 Ninguna de las anteriores.

1 Una de las tres anteriores.

2 De 4 a 5 de las anteriores.

3 Seis o más de las anteriores pero no se indican requerimientos de usuario específicos relacionados con la eficiencia.

4 Seis o más de las anteriores y los requerimientos indicados por el usuario para la eficiencia son lo suficientemente fuertes para requerir tareas de diseño de factores humanos (por ejemplo usar plantillas, maximizar casos por omisión).

5 Seis o más de las anteriores y los requerimientos indicados por el usuario para la eficiencia son lo suficientemente fuertes para requerir herramientas especiales y procesos que demuestren que los objetivos se han logrado.

8. Actualización en Línea

Describe el grado en que los archivos lógicos internos (ILF) se actualizan en línea. La aplicación provee actualización en línea para los ILF. El peso se aplica como sigue:

0 Ninguna.

1 Se requiere la actualización en línea de 1 a 3 archivos de control. El volumen de actualización es bajo y la recuperación es fácil.

2 Actualización en línea de 4 o más archivos de control. El volumen de actualización es bajo y la recuperación es fácil.

3 Se requiere la actualización en línea de archivos lógicos internos importantes.

4 Además la protección contra pérdida de datos es esencial y ha sido específicamente diseñada y programada en el sistema.

5 Además volúmenes altos traen consideraciones de costos en el proceso de recuperación. Procedimientos de recuperación altamente automatizados con un mínimo de intervenciones de operación.

9. Procesamiento Complejo

Describe el grado en el que la lógica de procesamiento influye en el desarrollo de la aplicación. Esto incluye lo siguiente:

- Control sensitivo (por ejemplo procesamiento de auditoría especial) y/o procesamiento de seguridad específica en la aplicación.
- Procesamiento lógico extenso.
- Procesamiento matemático extenso.
- Muchos procesamiento de excepción que resultan de transacciones incompletas que deben ser procesadas de nuevo (por ejemplo: transacciones incompletas por interrupciones, datos ausentes o validaciones sin éxito).
- Procesamiento complejo para manejar posibilidades de entrada/salida, por ejemplo multimedia.

El peso se aplica como sigue:

0 Ninguna de las anteriores.

1 Una de las anteriores.

2 Dos de las anteriores.

3 Tres de las anteriores.

4 Cuatro de las anteriores

5 Todas las anteriores.

10. Reusabilidad

Describe el grado en que la aplicación y el código de la aplicación han sido específicamente diseñados, desarrollados y soportados para que otras aplicaciones la utilicen. El peso se aplica como sigue:

- 0 No hay código reutilizable.
- 1 El código reutilizable es usado dentro de la aplicación.
- 2 Menos del 10% de la aplicación se considera para más de una necesidad del usuario.
- 3 Diez por ciento o más de la aplicación se considera para más de una necesidad del usuario.
- 4 La aplicación fue específicamente empaquetada y documentada para facilitar la reutilización y la aplicación es adaptada al usuario en un nivel de código fuente.
- 5 La aplicación fue específicamente empaquetada y documentada para facilitar la reutilización y la aplicación es adaptada al usuario en un nivel de código fuente por medio de parámetros de mantenimiento.

11. Facilidad de Instalación (migración)

Describe el grado en el que la conversión desde un ambiente previo influye en el desarrollo de la aplicación. La facilidad de conversión e instalación de son características de la aplicación. Un plan de conversión e instalación y/o herramientas de conversión serán provistas y probadas durante la fase de pruebas del sistema. El peso se aplica como sigue:

- 0 No hay consideraciones especiales indicadas por el usuario ni un set-up especial es requerido para la instalación.
- 1 No hay consideraciones especiales indicadas por el usuario pero se requiere un set-up especial para la instalación.
- 2 Los requerimientos de conversión e instalación fueron indicados por el usuario y las guías de conversión e instalación serán provistas y probadas. El impacto de conversión del proyecto no se considera importante.
- 3 Los requerimientos de conversión e instalación fueron indicados por el usuario y las guías de conversión e instalación serán provistas y probadas. El impacto de conversión del proyecto se considera importante.
- 4 Además del punto 2 la conversión automatizada y las herramientas de instalación serán provistas y probadas.
- 5 Además del punto 3 la conversión automatizada y las herramientas de instalación serán provistas y probadas.

12. Facilidad de Operación (soporte de respaldo)

Describe el grado en el que la aplicación atiende los aspectos operacionales tales como; inicialización, respaldo y recuperación. La aplicación minimiza las necesidades de actividades manuales, tales como: cargar una cinta, manejo del papel, intervención manual directa. Los pesos se aplican como siguen:

0 No hay consideraciones operacionales especiales más que el procedimiento de respaldo normal indicado por el usuario.

1-4 Seleccionar los siguientes elementos que se aplican. Cada elemento vale un punto.

- Procesos de inicialización, respaldo y recuperación fueron provistos, pero la intervención del operador es necesaria.
- Procesos de inicialización, respaldo y recuperación fueron provistos pero ninguna intervención del operador es necesaria (contar 2 elementos).
- La aplicación minimiza la necesidad de montar cintas.
- La aplicación minimiza la necesidad del manejo del papel.

5 La aplicación es diseñada para que no se requiera la intervención del operador. La recuperación de errores en forma automática es una característica de la aplicación.

13. Múltiples Ubicaciones

Describe el grado en que la aplicación se ha desarrollado para múltiples ubicaciones y organizaciones del usuario. La aplicación ha sido específicamente diseñada, desarrollada y soportada para ser instalada en múltiples lugares. El peso se aplica como sigue:

0 No hay requerimientos del usuario que consideren más de un lugar de instalación.

1 La necesidad de múltiples lugares fue considerada en el diseño, y la aplicación es diseñada para operar solamente bajo un ambiente de hardware y software idéntico.

2 La necesidad de múltiples lugares fue considerada en el diseño, y la aplicación es diseñada para operar solamente bajo un ambiente de hardware y software similar.

3 La necesidad de múltiples lugares fue considerada en el diseño, y la aplicación es diseñada para operar solamente bajo diferentes ambientes de hardware y software.

4 El plan de documentación y soporte es provisto y probado para apoyar la aplicación en múltiples lugares y la aplicación se describe como en el punto 1 o el 2.

5 El plan de documentación y soporte es provisto y probado para apoyar la aplicación en múltiples lugares y la aplicación se describe como en el punto 3.

14. Facilidad de Cambio

Describe el grado en el que la aplicación se ha desarrollado para facilitar la modificación de la lógica de procesamiento o la estructura de datos. Sus características son las siguientes:

- Provee capacidad para reportar/consultar de manera flexible.
- Los datos de control del negocio son agrupados en tablas mantenidas por el usuario.

El peso se aplica como sigue:

0 No hay requerimientos especiales del usuario para facilitar los cambios.

1 –5 Seleccione cualquiera de los siguientes elementos:

- La facilidad de consulta y reporte flexible permite manejar solicitudes simples- por ejemplo: lógica aplicada a un solo ILF (contar como un elemento).
- La facilidad de consulta y reporte flexible permite manejar solicitudes de complejidad media – por ejemplo: lógica aplicada a más de un ILF (contar como dos elementos).
- La facilidad de consulta y reporte flexible permite manejar solicitudes complejas – por ejemplo: lógica combinada sobre uno o más ILF (contar como tres elementos).
- Los datos de control comercial se mantienen en tablas que son administradas por el usuario con un proceso interactivo en línea pero los cambios tienen efecto solamente el próximo día comercial (contar como un elemento).
- Los datos de control se mantienen en tablas que son administradas por el usuario con un proceso interactivo en línea y los cambios tienen efecto inmediatamente (contar como 2 elementos).

Cálculo del Factor de Ajuste de Valor

Una vez evaluadas las 14 Características Generales descritas anteriormente se suman para obtener el Grado Total de Influencia o GTI. Posteriormente el GTI se aplica en la siguiente fórmula y se obtiene el Factor de Ajuste de Valor o FAV:

$$\text{FAV} = (\text{GTI} * 0.01) + 0.65$$

De acuerdo a [1, pág. 159] se espera que para una aplicación en lote simple se obtenga un GTI menor a 15, para una aplicación “front-end” un GTI entre 15 y 30, para una aplicación interactiva un GTI entre 30 y 45 y para un sistema en tiempo real, o de telecomunicaciones o de control de procesos un GTI entre 30 y 60.

Paso 6. Calcular el Punto de Función de Ajuste Final.

Para determinar el cálculo de Puntos de Función finales aplicamos la siguiente fórmula:

$$\text{PF} = \text{FAV} * \text{PFSA}$$

PF: Puntos de Función

FAV: Factor de Ajuste de Valor

PFSA: Puntos de Función sin Ajustar.

El FAV se obtiene a través del paso 5 de este procedimiento y los PFSA a través de los pasos 3 y 4.

Capítulo 2

Casos Prácticos de Estimación de Puntos de Función

Caso 1: Sistema de Planilla

Los requerimientos del usuario son los siguientes:

- Capacidad para consultar, mantener y reportar información del empleado. El reporte podría incluir los datos de ubicación de un determinado empleado obtenido de un archivo al que otra aplicación le da mantenimiento.
- Capacidad para consultar, mantener y reportar información de las tareas. El usuario considera que la descripción de la tarea debe ser un conjunto de 80 caracteres por línea que describen la tarea. Esta información no se mantiene independientemente de la tarea.
- Capacidad para consultar, mantener y reportar información de la asignación de tareas a los empleados.
- Capacidad para consultar y reportar las diferentes ubicaciones dentro de la compañía, incluyendo la lista de empleados en una ubicación particular. La ubicación es leída solamente porque se mantiene en otra aplicación.

Un modelo de procesos podría ser el siguiente:

Mantenimiento de Empleados

- Crear Empleados
- Consultar Empleados
- Actualizar Empleados
- Borrar Empleados
- Reportar Empleados

Mantenimiento de Tareas

- Crear Tareas
- Consultar Tareas
- Actualizar Tareas
- Borrar Tareas
- Reportar Tareas

Mantenimiento del Asignador de tareas

- Asignar Tarea al Empleado
- Consultar Asignación de la Tarea
- Transferir Empleado
- Evaluar Empleado
- Borrar Asignación
- Reportar Asignación de la Tarea

Reportar Ubicación

- Consultar Ubicación
- Reportar Ubicación

Los campos contenidos en cada entidad son los siguientes:

EMPLEADO tipo entidad

- Nombre
- Número Cédula
- Sexo
- Edad
- Número Dependientes
- Tipo (Docente o Administrativo)
- Nombre Ubicación (Clave Externa)

EMPLEADO_DOCENTE subtipo o subgrupo de la entidad

- Régimen Docente
- Número Pasos
- Puntos
- Carga
- Escalafón docente
- Anualidad docente

EMPLEADO_ADMINISTRATIVO subtipo o subgrupo de la entidad

- Horario
- Años servicio
- Escalafón
- Anualidad

TAREA tipo entidad

- Nombre Tarea
- Número Tarea
- Tipo de Pago

DESCRIPCION_TAREA tipo entidad (solamente para el proceso de implementación, no es un subgrupo para los usuarios)

- Número Tarea (Clave Externa)
- Número Línea (solamente para el proceso de implementación, no tiene significado para el usuario)
- Línea Descripción

ASIGNADOR_TAREA tipo entidad

- Fecha ingreso
- Salario
- Rendimiento
- Número Tarea (Clave Externa)
- Cédula Empleado (Clave Externa)

UNIDAD tipo entidad

- Nombre Unidad
- Dirección
- Cédula Empleado (Clave Externa)

Un Diagrama de Entidad Relación se muestra en la figura 2.1.

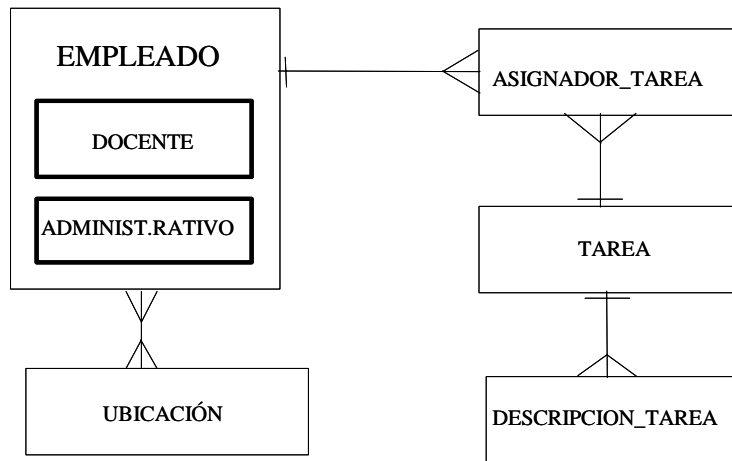


Figura 2.1 Diagrama de Entidad Relación

A continuación se explica la metodología de puntos de función tomando como base el Sistema de Empleados descrito anteriormente.

Paso 1. ¿Cómo determinar el tipo de conteo de Puntos de Función?

El Sistema de Planilla es un proyecto en desarrollo, no está en mantenimiento, ni es una aplicación ya desarrollada.

Paso 2. ¿Cómo identificar el alcance del conteo y las fronteras de la aplicación?

El identificar las fronteras de la aplicación es una tarea muy sencilla. Las tablas que son administradas por otra aplicación están fuera de las fronteras y se convierten en EIF. Las tablas administradas por la aplicación que se está estimando, están dentro de las fronteras y corresponden a los ILF. En el caso del Sistema de Planilla la única tabla que está fuera de las fronteras de la aplicación, que será leída y es administrada por otra aplicación es Ubicación.

Paso 3. ¿Cómo identificar todas las funciones de datos y su complejidad?

De acuerdo a este ejemplo en la tabla 2.1 se muestra la identificación de los ILF, los EIF y sus respectivos DET y RET.

Tabla 2.1 Identificación de los archivos incluidos

EMPLEADO entidad	ILF: No cuenta como RET porque tiene subgrupos
Número Cédula	DET 1
Nombre	DET 2
Sexo	DET 3
Edad	DET 4
Número Dependientes	DET 5
Tipo (docente o administrativo)	DET 6
Nombre Ubicación (clave externa)	DET 7
EMPLEADO_DOCENTE subgrupo	1 RET de FUNCIONARIO
Régimen docente	DET 8
Número Pasos	DET 9
Puntos	DET 10
Carga	DET 11
Escalafón docente	DET 12
Anualidad docente	DET 13
EMPLEADO_ADMINISTRATIVO subgrupo	1 RET de FUNCIONARIO
Horario	DET 14
Años servicio	DET 15
Escalafón administrativo	DET 16
Anualidad administrativo	DET 17
TAREA entidad	ILF: Cuenta como RET porque no tiene subgrupos
Nombre tarea	DET 1
Número tarea	DET 2 *
Tipo de pago	DET 3
DESCRIPCIÓN_TAREA entidad	No cuenta como ILF porque se crea por razones de implementación
Número tarea (clave externa)	Ya fue contada en TAREA*
Número línea (implementación)	No cuenta
Descripción de línea	DET 4
ASIGNADOR_TAREA entidad	ILF: Cuenta como RET porque no tiene subgrupos
Fecha ingreso	DET 1
Salario	DET 2
Tasa de rendimiento	DET 3
Número tarea (clave externa)	DET 4
Cédula (clave externa)	DET 5
UBICACIÓN entidad	EIF: Cuenta como RET porque no tiene subgrupos
Nombre ubicación	DET 1

Dirección	DET 2
Cédula empleado (clave externa)	DET 3

Una vez identificados y contados los ILF y los EIF con sus DET y RET se debe determinar el nivel de complejidad de cada uno de acuerdo a la matriz mostrada en la tabla 2.2.

Tabla 2.2. Matriz de complejidad para ILF y EIF.

Nº Registros (RET)	Nº de Campos (DET)		
	1 - 19	20 - 50	≥ 50
1	Baja	Baja	Media
2 - 5	Baja	Media	Alta
> 5	Media	Alta	Alta

El nivel de complejidad asociado a los ILF y EIF con sus correspondientes DET y RET se muestra a continuación:

Tabla 2.3. Nivel de complejidad para los ILF y EIF.

Entidad	TIPO(ILF o EIF)	DET	RET	Complejidad
EMPLEADO	1 ILF	17 DET	2 RET	Bajo
TAREA	1 ILF	4 DET	1 RET	Bajo
ASIGNADOR_TAREAS	1 ILF	5 DET	1 RET	Bajo
UBICACIÓN	1 EIF	3 DET	1 RET	Bajo
Resultado:	3ILF con complejidad Baja y 1 EIF con complejidad Baja			

La tabla 2.4 muestra el total de PFSA para las funciones de datos y su complejidad. Como se puede observar para el Sistema de Planilla se contaron 3 ILF con complejidad baja y 1 EIF con complejidad baja obteniendo un total de 26 PFSA.

Tabla 2.4. Tabla de PFSA par los ILF y los EIF.

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	3 x7 = 21	x10	x15
Archivos de Interfase Externa (EIF)	1 x5 = 5	x7	x10
Entradas Externas (EI)	x3 =	x4=	x6 =
Salidas Externas (EO)	x4 =	x5 =	x7
Consultas Externas (EQ)	x3=	x4	x6
Total PFSA	26 PFSA		

Paso 4. ¿Cómo identificar todas las funciones transaccionales y su complejidad?

Identificación de los EI

Se deben identificar en el Sistema de Planilla los EI con sus DET y FTR y obtener posteriormente su complejidad. Realmente no hay suficiente información porque está iniciando el proceso de desarrollo pero se puede estimar haciendo unos pocos supuestos. Vamos a asumir que cada transacción de entrada devuelve un mensaje de error (contaremos un DET para cada EI por mensaje de error) y que cada EI tiene al menos una tecla de comando (contaremos otro DET por cada EI). También vamos a asumir que en nuestras funciones de Crear y Actualizar accedamos todos los campos del ILF que se está creando o actualizando, pero en las funciones de Borrar solo tocamos los campo(s) de la clave principal del ILF. Para las funciones de Asignar y Transferir no se toca el campo de Rendimiento y para la función de Evaluar no se toca el campo Salario. Recuerde los 2 DET's extras para cada transacción.

Para nuestros FTR debemos contar el ILF que está siendo administrado y cualquier otro ILF o EIF, que debemos referenciar para propósitos de edición. Por ejemplo: debemos referenciar la ubicación cuando ingresamos al campo Ubicación para crear un empleado, de esta manera tenemos 2 FTR cuando creamos un empleado: EMPLEADO y UBICACIÓN.

En la siguiente tabla podemos observar en la columna izquierda 10 transacciones EI y en la columna derecha los DET y los FTR identificados por cada EI.

Tabla 2.5. Identificación del los EI con sus DET y FTR.

TRANSACCIONES EI	CONTADAS COMO
MANTENIMIENTO EMPLEADOS	
Crear EMPLEADOS	19 DET (*) y 2 FTR, EMPLEADO y UBICACIÓN
Actualizar EMPLEADOS	19 DET (*) y 2 FTR, EMPLEADO y UBICACIÓN
Borrar EMPLEADOS	3 DET (*) y 2 FTR, EMPLEADO y ASIGNADOR TAREA (no borrar un empleado si tiene asignada una tarea)
MANTENIMIENTO TAREAS	
Crear TAREAS	6 DET (*) y 1 FTR, TAREA
Actualizar TAREAS	6 DET (*) y 1 FTR, TAREA
Borrar TAREAS	3 DET (*) y 2 FTR, TAREA y ASIGNADOR TAREA (no borrar una tarea con un empleado asignado)
MANTENIMIENTO ASIGNADOR TAREAS	
Asignar EMPLEADO a TAREA	6 DET (*) y 3 FTR, EMPLEADO, TAREA y ASIGNADOR TAREA
Transferir EMPLEADO	6 DET (*) y 3 FTR, EMPLEADO, TAREA y ASIGNADOR TAREA
Evaluar EMPLEADO	6 DET(*) y 1 FTR, ASIGNADOR TAREA
Borrar ASIGNACIÓN	4 DET(*) y 1 FTR, ASIGNADOR TAREA

(*) Se cuentan dos DET extras, uno para mensaje y otro para tecla de comando.

Una vez identificados y contados los EI con sus DET y RET se debe determinar el nivel de complejidad de cada EI de acuerdo a la matriz mostrada en la siguiente tabla.

Tabla 2.6. Matriz de complejidad para las EI.

Nº de Archivos Referenciados (FTR)	Nº de Campos (DET)		
	1 - 4	5 - 15	≥ 16
< 2	Baja	Baja	Media
2	Baja	Media	Alta
> 2	Media	Alta	Alta

El nivel de complejidad asociado a los EI con sus correspondientes DET y RET se muestra a continuación:

Tabla 2.7. Nivel de complejidad para los EI.

TRANSACCIONES EI	CONTADAS COMO	COMPLEJIDAD
MANTENIMIENTO EMPLEADOS		
Crear Empleados	19 DET (*) y 2 FTR, EMPLEADO y UBICACIÓN	ALTA
Actualizar Empleados	19 DET (*) y 2 FTR, EMPLEADO y UBICACIÓN	ALTA
Borrar Empleados	3 DET (*) y 2 FTR, EMPLEADO y ASIGNADOR TAREA (no borrar un empleado con una tarea)	BAJO
MANTENIMIENTO TAREAS		
Crear Tareas	6 DET (*) y 1 FTR, TAREA	BAJO
Actualizar Tareas	6 DET (*) y 1 FTR, TAREA	BAJO
Borrar Tareas	3 DET (*) y 2 FTR, TAREA y ASIGNADOR TAREA (no borrar una tarea con un empleado asignado)	BAJO
MANTENIMIENTO ASIGNADOR TAREAS		
Asignar Tarea al Empleado	6 DET (*) y 3 FTR, EMPLEADO, TAREA y ASIGNADOR TAREA	ALTO
Transferir Empleado	6 DET (*) y 3 FTR, EMPLEADO, TAREA y ASIGNADOR TAREA	ALTO
Evaluar Empleado	6 DET (*) y 1 FTR, ASIGNADOR TAREA	BAJO
Borrar Asignación	4 DET (*) y 1 FTR, ASIGNADOR TAREA	BAJO
Resultado: 97 PFSA	6 EI con complejidad baja 4 EI con complejidad alta (Crear y Actualizar Empleado, Asignar Tarea y Transferir Empleado)	

(*) Se cuentan dos DET extras, uno para mensaje y otro para tecla de comando.

Identificación de los EO

Se deben identificar los EO con sus DET y FTR y obtener posteriormente su complejidad. No hay suficiente información para contar los DET, ni para distinguir los EO de los EQ. Vamos a asumir que cada reporte tiene datos derivados o calculados y que todas las consultas se contarán como EQ. Vamos a asumir que cada reporte contiene datos derivados o calculados y que todas las consultas serán EQ.

También asumiremos que cada reporte excepto el reporte de Tareas, tiene entre 6 y 19 DET. De hecho cuando se analiza la complejidad de cualquier transacción, solo necesitamos saber los DET dentro de un rango particular basado en la matriz de complejidad para la transacción particular. Observe que los rangos aplicables para los DET sobre los EO pueden ser menores a 6, entre 6 y 19 y mayores que 19. Normalmente tenemos un buen estimado del rango de DETS antes de concluir la fase de diseño. Finalmente asumamos que el reporte de Tareas tiene 5 DET.

La tarea de contar los DET es relativamente fácil, especialmente cuando se tiene conocimiento del sistema que se está estimando. Contar los FTR es una tarea más difícil. Si al menos tuviéramos una copia de los reportes, podríamos encontrar los ILF por los campos contenidos. Sin embargo, en nuestro caso no contamos con tales reportes.

Podríamos asumir que el reporte de Empleados podría referenciar los archivos de Empleados y Ubicación, porque hay una relación entre ambos. Con 2 FTR y un rango entre 6 y 19 DET, tendríamos un EO con complejidad media. El reporte de Tarea tiene 5 DET (que es un número dado) y probablemente solo un FTR, que es el archivo de Tarea. Consecuentemente este EO tiene una complejidad baja. El reporte de Asignador de Tareas tiene entre 6 y 19 DET, y podría referenciar 3 archivos (Asignador de Tarea, Empleado y Tarea). Con 3 FTR este reporte tiene una complejidad media. Finalmente el reporte de Ubicación referencia 2 FTR y contiene al menos 6 DET, obteniéndose una complejidad media.

En la tabla 2.8 podemos observar una columna con los reportes solicitados por el usuario, otra columna con los FTR identificados y otra con el número de DET identificados para cada EO.

Tabla 2.8. Identificación del los EO con sus DET y FTR..

EO	FTR	DET
Reporte de Empleados	2 (EMPLEADOS y UBICACIÓN)	Más de 20
Reporte de Tareas	1 (TAREAS)	5
Reporte de Asignación de Tareas	3 (ASIGNADOR DE TAREAS, EMPLEADO y TAREA)	6 – 19
Reporte de Ubicación	2 (UBICACIÓN y EMPLEADO)	6

Para obtener la complejidad utilizamos la tabla 2.9 y tomamos como base los FTR y los DET de cada reporte.

Tabla 2.9. Matriz de complejidad para EO.

Nº de Archivos Referenciados (FTR)	Nº de Campos (DET)		
	1 - 5	6 - 19	≥20
< 2	Baja	Baja	Media
2 - 3	Baja	Media	Alta
> 3	Media	Alta	Alta

En la siguiente tabla podemos observar la complejidad de cada uno de los EO identificados para el Sistema de Planilla.

Tabla 2.10. Nivel de complejidad para los EO.

EO	FTR	DET	Complejidad
Reportar EMPLEADOS	2 (EMPLEADOS y UBICACIÓN)	Más de 20	Alta
Reportar TAREAS	1 (TAREAS)	5 (número fue dado)	Bajo
Reportar ASIGNADOR TAREAS	3 (ASIGNADOR DE TAREAS, EMPLEADO y TAREA)	6 – 19	Medio
Reportar UBICACIÓN	2 (UBICACIÓN y EMPLEADO)	6	Medio
Resultado:	1 EO con complejidad alta 2 EO con complejidad media 1 EO con complejidad baja		

Identificación de los EQ

Se debe identificar en el Sistema de Planillas los EQ con sus DET y FTR y obtener posteriormente su complejidad. Un EQ así como un EO puede tener campos o información de control que ingresan a la aplicación para recuperar datos. Vamos a asumir en este ejemplo que la información de control para cada EO o EQ es también desplegada sobre la parte de salida. Además se podría esperar un mensaje de error, verificación o confirmación (contaremos un DET por cada EO o EQ para los mensajes de error, verificación o confirmación) y podríamos tener al menos una tecla de comando (contaremos un DET para cada EO o EQ para las teclas de comando cuando aplique).

En la tabla 21 podemos observar una columna con las consultas solicitadas por el usuario, otra columna con los FTR y otra con el número de DET para cada EQ.

Tabla 2.11. Identificación del los EQ con sus DET y FTR.

EQ	FTR	DET
Consultar EMPLEADOS	1 (EMPLEADOS)	19 (*)
Consultar TAREAS	1 (TAREAS)	6 (*)
Consultar ASIGNADOR TAREAS	1 (ASIGNADOR TAREAS)	7(*)
Consultar UBICACIÓN	1 (UBICACIÓN)	5(*)

(*) Se cuentan dos DET extras, uno para mensaje y otro para tecla de comando.

Podríamos preguntarnos por qué solo hay un archivo referenciado, pero es que no necesitamos referenciar ningún otro archivo para validación y ningún campo fue recuperado de otro archivo, sino que solo del archivo principal.

Tabla 2.12. Matriz de complejidad para los EQ.

Archivos Referenciados (FTR)	Datos (DET)		
	1-5	6-19	20 +
1	Bajo (L)	Bajo (L)	Medio (A)
2-3	Bajo (L)	Medio (A)	Alto (H)
>3	Medio (A)	Alto (H)	Alto (H)

En la tabla 2.13 podemos observar la complejidad de cada uno de los EQ identificados para el Sistema de Planilla. Para obtener la complejidad de cada EQ se utilizó la matriz de complejidad de la tabla 2.12.

Tabla 2.13. Nivel de complejidad para los EQ.

EQ	FTR	DET	Complejidad
Consultar EMPLEADOS	1 (EMPLEADOS)	19	Baja
Consultar TAREAS	1 (TAREAS)	6	Baja
Consultar ASIGNADOR TAREAS	1 (ASIGNADOR TAREAS)	7	Baja
Consultar UBICACIÓN	1 (UBICACIÓN)	5	Baja
Resultado:	4 EQ con complejidad baja		

La tabla 2.14 muestra el total de PFSA, tanto para las funciones transaccionales y su complejidad, como para las funciones de datos y su complejidad. Como se puede observar del cálculo de funciones transaccionales y su complejidad se obtuvo un total de 6 EI de complejidad baja, 2 EI de complejidad media y 2 EI de complejidad alta. Además 1 EO de complejidad baja, 3 EO de complejidad media y 1 EQ de complejidad baja.

Tabla 2.14. Puntos de Función Desajustados

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	3 x 7 = 21	x10	x15
Archivos de Interfase Externa (EIF)	1 x 5 = 5	x7	x10
Entradas Externas (EI)	6 x 3 = 18	x4	4 x 6 = 24

Salidas Externas (EO)	1 x4 = 4	2 x5 = 10	1 x7= 7
Consultas Externas (EQ)	4 x3= 12	x4	x6
Total PFSA	101		

Paso 5. Cómo determinar el valor del Factor de Ajuste – basado en las 14 características generales del sistema?

Una vez descritas las 14 Características Generales procederemos a continuar con el ejemplo que venimos trabajando obteniendo un puntaje de 31 puntos de Total de Grados de Influencia (TI):

Característica	Peso
1. Comunicación de Datos	4
2. Procesamiento distribuido.	0
3. Objetivos de rendimiento	3
4. Configuración muy utilizada	2
5. Tasa transaccional.	3
6. Ingreso de datos en línea.	5
7. Eficiencia ara usuario final.	4
8. Actualización en línea.	3
9. Procesamiento complejo.	1
10. Reusabilidad.	0
11. Facilidad de instalación (migración).	0
12. Facilidad operacional.	3
13. Múltiples ubicaciones.	1
14. Facilidad de cambio.	2
Grado Total de Influencia (GTI)	31

Usando la fórmula de Factor de Ajuste de Valor (FAV) sustituimos el GTI:

$$FAV = (GTI * 0.01) + 0.65$$

$$FAV = (31 * 0.01) + 0.65$$

$$FAV = 0.96$$

Paso 6. Calcular los puntos de función ajustados.

Sustituyendo tenemos:

$$PF = 0.96 * PFSA$$

$$PF = 0.96 * 101$$

$$PF = 96.96$$

Caso 2: Sistema de Administración de Contrataciones

Objetivo principal:

Implementar un Sistema de Administración de Contrataciones para una empresa desarrolladora de software que permita optimizar el manejo de las contrataciones y sus requerimientos.

Consideraciones especiales:

- Un cliente puede tener asociadas una o más contrataciones pero una contratación solo puede asociarse a un cliente.
- Una contratación puede tener asociados varios requerimientos, pero un requerimiento solo puede asociarse a una contratación.
- Los requerimientos de interfaz son: permitir hacer todas las transacciones del ILF en una ventana, la cual puede llamar a otra ventana que hace las transacciones del RET. Cada ventana debe tener los campos de los atributos y un datagrid para los registros correspondientes de la tabla, debidamente actualizados.
- Para agilizar el acceso y obtener un mejor tiempo de respuesta se requiere un identificador numérico consecutivo durante la inclusión de cada registro en las tablas.

Definición de requerimientos

Req. 1. Administración de Clientes

1.1 Incluir Clientes: El objetivo de este requerimiento es agregar nuevos clientes a la base de datos del sistema. Los datos de entrada para un cliente son:

- Nombre de cliente
- Cédula jurídica
- Dirección
- Teléfono

El procesamiento consiste en insertar el registro del cliente. El nombre del cliente junto con cualquier posible mensaje de error serán las salidas del sistema.

1.2 Eliminar Clientes: El propósito de este requerimiento es eliminar los registros de un cliente que ya no interesen a la empresa o que fue incluido erróneamente. Al eliminarse un cliente no se eliminan las contrataciones (No borra un cliente si tiene contrataciones asociadas). Como dato de entrada se debe permitir seleccionar de una lista el cliente a eliminar (consulta implícita). Como dato de salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error.

1.3 Modificar Clientes: El objetivo es actualizar la información de un cliente existente. Como dato de entrada se debe permitir seleccionar de una lista el cliente a modificar. Se pueden modificar cualquiera de los campos a excepción de la cédula jurídica. Si se incluyó erróneamente se debe borrar el registro. Como resultado se debe desplegar una confirmación de que el registro fue modificado, o bien, algún mensaje de error.

Req. 2. Administración de Historial de Contrataciones

2.1 Incluir Contrataciones: El objetivo de este requerimiento es agregar la información de nuevas contrataciones y administrar sus requerimientos. Los datos de entrada para una contratación son los siguientes:

- Cliente (obtenido a partir de una lista de selección con los clientes existentes).
- Nombre contratación
- Descripción
- Requerimientos asociados a una contratación. Para cada requerimiento en la contratación se debe administrar la siguiente información:
 - Descripción del requerimiento
 - Fecha inicio
 - Fecha Fin Real
 - Estado

El procesamiento consiste en insertar el registro de una contratación y los posibles requerimientos asociados a ésta. El nombre de la contratación junto con cualquier posible mensaje de error será la salida del sistema.

2.2 Modificar Contrataciones: El objetivo es actualizar la información de una contratación y administrar los requerimientos asociados o los nuevos requerimientos. Es posible modificar cualquiera de los datos de una contratación. Como dato de entrada se seleccionará de una lista la contratación a modificar (consulta implícita). Como resultado se indicará una confirmación de que el registro fue modificado, o bien, algún mensaje de error.

2.3 Eliminar Contrataciones: El propósito de este requerimiento es eliminar el registro de una contratación que ya no le interese a la empresa o que fue incluida erróneamente. Al eliminarse una contratación se eliminan sus requerimientos. Como dato de entrada se seleccionará de una lista la contratación a eliminar y como salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error.

Req. 3. Consulta Compleja

- 3.1** Reportar para un determinado cliente todas sus contrataciones y los requerimientos asociados a cada contratación.

A continuación se detallarán cada uno de los pasos para estimar el tamaño del software en Puntos de Función para el Sistema de Administración de Contrataciones.

Paso 1: ¿Cómo determinar el tipo de Puntos de Función?

Pertenece al Tipo 1 por se un proyecto en desarrollo. No está en mantenimiento no es un proyecto ya desarrollado.

Paso 2 ¿Cómo identifico el alcance del conteo y las fronteras de la aplicación?

Para este proyecto todas las tablas que se requiere administrar se encuentran dentro de sus fronteras. Además el sistema a desarrollar no interactúa con aplicaciones externas.

Paso 3: ¿Cómo estimo los PFSA de los ILF y de los EIF?

De acuerdo al Sistema de Administración de Contrataciones en la tabla 2.15 se muestra la identificación de los ILF, los EIF y sus respectivos DET y RET, en la tabla 2.16 el nivel de complejidad para dichas funciones de datos y en la tabla 2.17 el total de los PFSA correspondiente.

Tabla 2.15 Identificación de los archivos incluidos

CAMPOS DEL ARCHIVO	CONTADOS COMO
CLIENTES	ILF administrado con 1 RET
ID Cliente PK	No se cuenta, se creó por implementación para mejorar tiempo de respuesta.
Nombre cliente	DET 1
Cédula jurídica	DET 2
Dirección	DET 3
Teléfono	DET 4
CONTRATACIONES	ILF administrado con 1 RET
ID Contratación PK	No se cuenta, se creó por implementación. Aunque es la llave, el usuario no la pidió entre los atributos.
Cliente FK	DET 1
Nombre Contratación	DET 2
Descripción	DET 3
REQUERIMIENTOS	ILF administrado con 1 RET
ID Requerimiento PK FK	No se cuenta por implementación
ID Contratación FK	No se cuenta, se contó en el padre.
Descripción requerimiento	DET 1
Fecha inicio	DET 2
Fecha fin	DET 3
Estado	DET 4

La razón por la que Requerimientos es un RET de Contrataciones es por ser débil, no hay contratación sin requerimientos, es decir los requerimientos existen para apoyar la existencia de una contratación.

Tabla 2.16. Nivel de complejidad para los ILF y EIF.

Entidad	ILF	DET	RET	Complejidad
CLIENTES	1 ILF	4 DET	1 RET	Bajo
CONTRATACIONES	1 ILF	3 DET	1 RET	Bajo
REQUERIMIENTOS	1 ILF	4 DET	1 RET	Bajo
Resultado	3 ILF con complejidad Baja			

Archivos Interfaces Externas (EIF)

No se presentan archivos de este tipo

Paso 4: ¿Cómo estimo los PFSA para los EI?

En la tabla 27 podemos observar en la columna izquierda 9 transacciones EI para el Sistema de Administración de Contrataciones, en la columna del medio los DET y los FTR de cada EI y en la columna derecha el nivel de complejidad correspondiente.

Tabla 2.17. Identificación del los EI y su complejidad.

TRANSACCIONES EI	CONTADAS COMO	COMPLEJIDA D
MANTENIMIENTO CLIENTES		
Incluir CLIENTES	6 DET(*) y 1 FTR, CLIENTES	BAJA
Modificar CLIENTES	5 DET(*) y 1 FTR, CLIENTES	BAJA
Eliminar CLIENTES	3 DET(*) y 2 FTR, CLIENTES y CONTRATACIONES (**)	BAJA
MANTENIMIENTO CONTRATACIONES		
Incluir CONTRATACIONES	5 DET(*) y 2 FTR, CONTRATACIONES Y CLIENTES	MEDIA
Modificar CONTRATACIONES	4 DET(*) y 2 FTR's, CONTRATACIONES Y CLIENTES	MEDIA
Eliminar CONTRATACIONES	3 DET(*) y 1 FTR CONTRATACIONES	BAJA
MANTENIMIENTO REQUERIMIENTOS		
Incluir REQUERIMIENTOS	6 DET(*) y 2 FTR, REQUERIMIENTOS, CONTRATACIONES	MEDIA
Modificar REQUERIMIENTOS	5 DET(*) y 2 FTR, REQUERIMIENTOS, CONTRATACIONES	MEDIA
Eliminar REQUERIMIENTOS	4 DET(*) (llave compuesta) y 1 FTR, REQUERIMIENTOS	BAJA
Resultado: 5 EI de complejidad BAJA Y 4 EI complejidad MEDIA		

(*) Incluye 2 de supuestos.

(**) Si se borra un cliente no debe haber contrataciones asociadas

En la función Modificar vamos a asumir que no se puede modificar el campo de la llave principal.

Paso 5: ¿Cómo estimo los PFSA para los EQ?

No se solicitaron reportes que requieran cálculos complejos o actualización de las tablas del sistema.

Tabla 2.18. Identificación del los EQ y su complejidad.

EQ	FTR	DET	Complejidad
Consultar CLIENTES	1 (CLIENTES)	6 (*)	BAJA
Consultar CONTRATACIONES	2(CONTRATACIONES, CLIENTES)	5 (*)	BAJA
Consultar REQUERIMIENTOS	2 (REQUERIMIENTOS, CONTRATACIONES)	6 (*)	MEDIA
Resultado: 2 EQ con complejidad Baja y 1 Media			

Paso 6: ¿Cómo estimo los PFSA para los EO?

Tabla 2.19. Identificación del los EO y su complejidad.

EO	FTR	DET	Complejidad
Reportar un Cliente específico con sus contrataciones y sus respectivos requerimientos	3 (CLIENTE, CONTRATACIONES y REQUERIMIENTOS)	13 máximo (incluye 2 de supuestos)	Media

Resultado: 3 EO con complejidad Media
--

(*) Incluye 2 de supuestos.

La tabla 2.20 muestra el total de PFSA, tanto para las funciones de datos y su complejidad, como para las funciones de transaccionales y su complejidad.

Tabla 2.20. Puntos de Función Desajustados

Componentes	Niveles de Función			
	Baja	Media	Alta	
Archivos Lógicos Internos (ILF)	3 x7 = 21	X10	x15	21
Archivos de Interfase Externa (EIF)	X5 =	X7	x10	
Entradas Externas (EI)	5 x3 = 15	4 x4= 16	x6	31
Salidas Externas (EO)	x4	x5	x7	
Consultas Externas (EQ)	3 x3= 9	x4	x6	9
Total PFSA	61			

Paso 7: ¿Cómo calculo el Factor de Ajuste - 14 Características Generales de Sistemas?

1. Comunicación de datos	4	
2. Procesamiento de datos distribuido	0	
3. Rendimiento	3	
4. Configuración	2	
5. Tasa transaccional	3	
6. Entrada de datos en línea	5	
7. Eficiencia del usuario final	4	
8. Actualización en línea		3
9. Procesamiento complejo	1	
10. Reutilización	0	
11. Facilidad de instalación	0	
12. Facilidad operacional	3	
13. Lugares múltiples	1	
14. Facilidades de cambio		1

Grado Total de Influencia (GTI) 30

Usando la fórmula de Factor de Ajuste de Valor (FAV) sustituimos el GTI:

$$\text{FAV} = (\text{GTI} * 0.01) + 0.65$$

$$\text{FAV} = (30 * 0.01) + 0.65$$

$$\text{FAV} = 0.95$$

Paso 8: ¿Cómo calculo los Puntos de función de Ajuste Final?

Aplicando el Factor de Ajuste de Valor (FAV) a los Puntos de Función sin Ajustar obtenemos los Puntos de Función Ajustados conocidos como PF:

$$PF = FAV * PFSA$$

$$PF = 0.95 * 61$$

$$PF = 57,95$$

Caso 3: Sistema de Administración de Requerimientos

A continuación se muestra un Diagrama Jerárquico de requerimientos con el fin de presentar los requerimientos de este segundo caso correspondiente a un “Sistema de Administración de Requerimientos”.

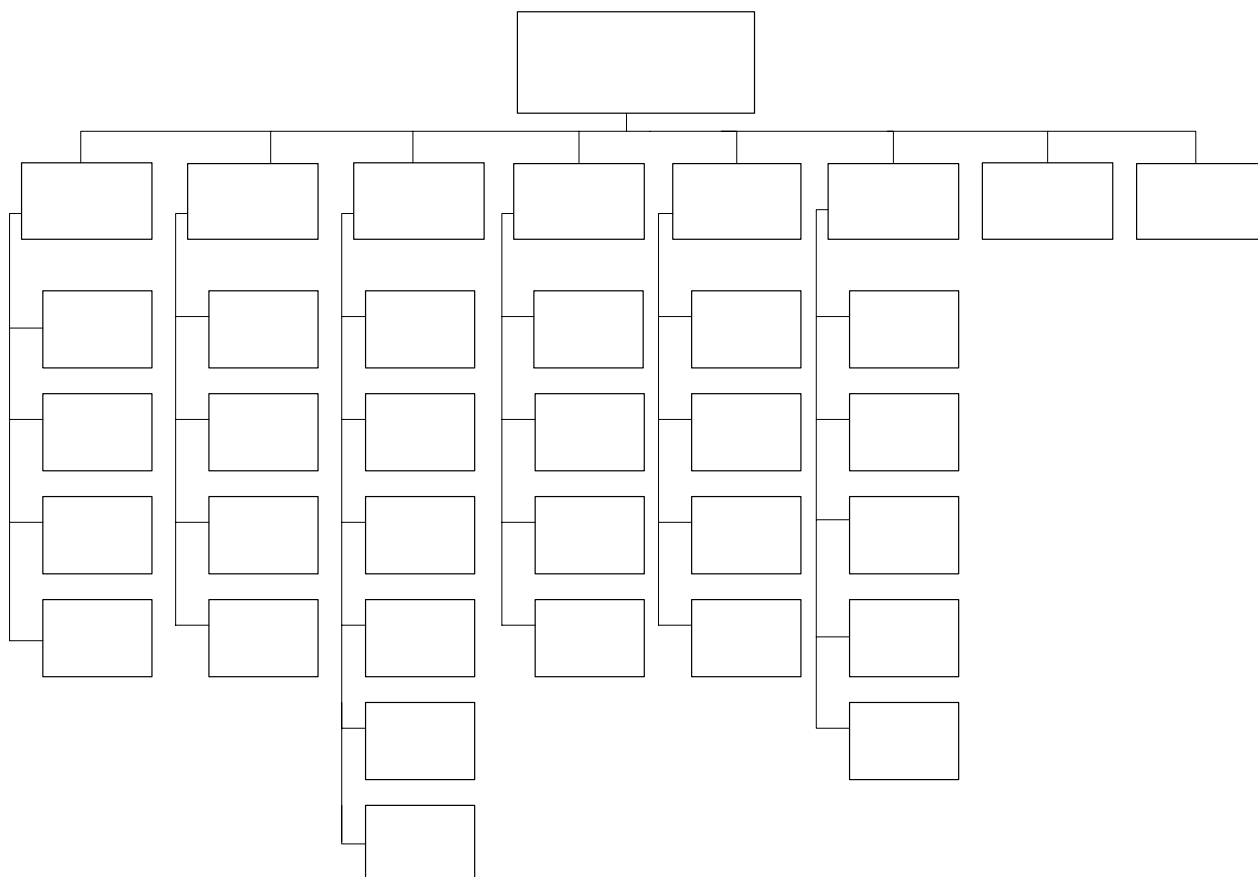


Figura 4. Diagrama Jerárquico de Requerimientos

A continuación se muestra el Diagrama de Entidad Relación para el “Sistema de administración de Requerimientos”.

Requerimiento 1
Mantenimiento
Personas

Requerimiento 2
Mantenimiento
Proyectos

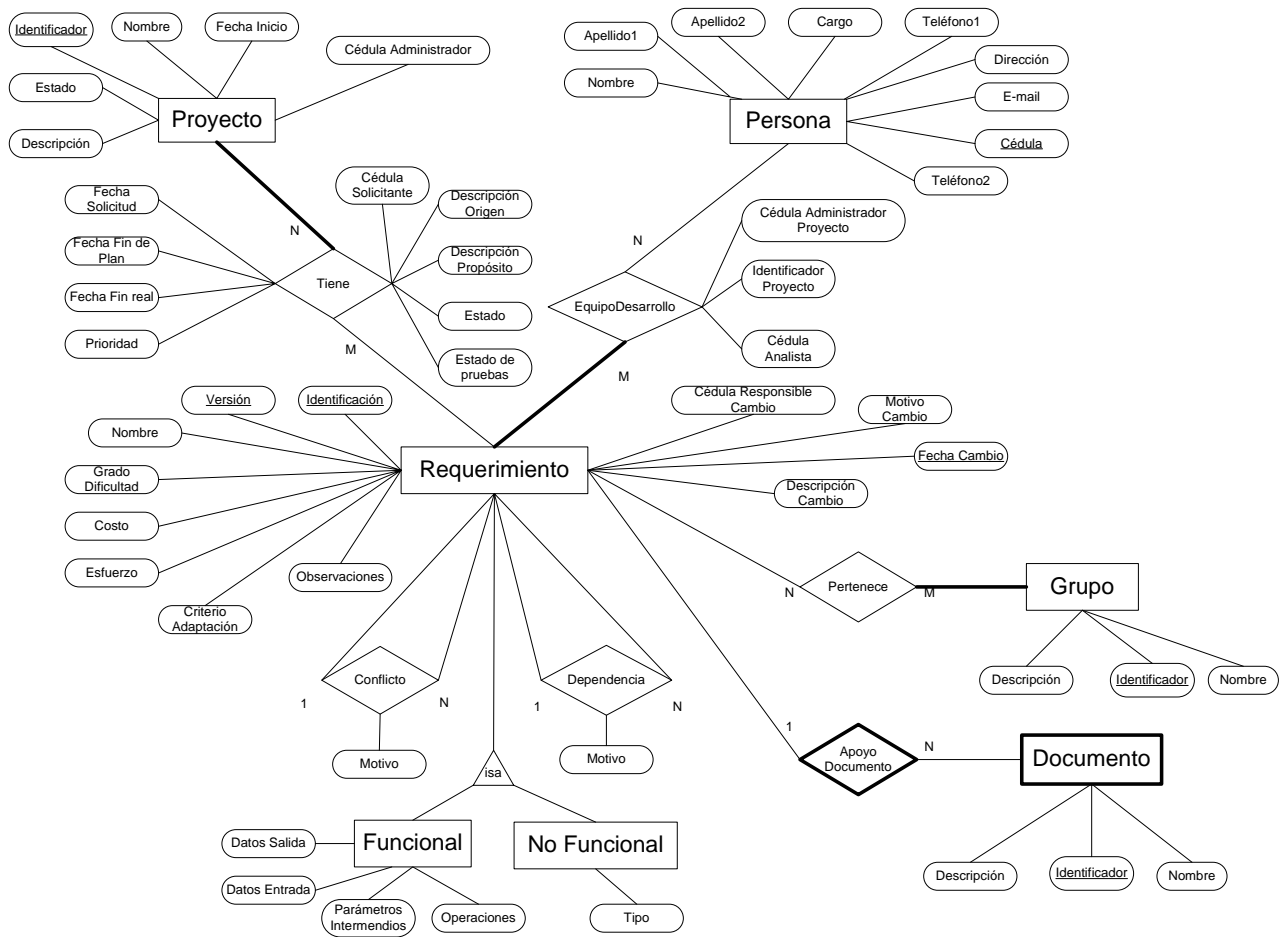


Figura 5. Modelo Entidad – Relación

A continuación se detallarán cada uno de los pasos para estimar el tamaño del software en Puntos de Función para el Sistema de Administración de Requerimientos.

Paso 1: ¿Cómo determinar el tipo de Puntos de Función?

El proyecto pertenece al Tipo 1 por ser una aplicación en desarrollo, no está en mantenimiento, no ya está desarrollada.

Paso 2 ¿Cómo identificar el alcance del conteo y las fronteras de la aplicación?

La aplicación a desarrollar no accesa ninguna aplicación externa, todas las tablas están dentro de las fronteras de la aplicación.

Paso 3: ¿Cómo identificar y estimar puntos de función para los ILF y los EIF?

En la tabla 2.21 se muestran los ILF identificados con sus respectivos DET y RET, en la tabla 2.22 el nivel de complejidad para dichas funciones de datos y en la tabla 2.23 el total de los PFSA correspondiente a funciones de datos y su complejidad.

Tabla 2.21 Identificación de los archivos incluidos

CAMPOS DEL ARCHIVO	CONTADOS COMO
PERSONA	ILF mantenido con un RET
Cédula	DET1
Nombre	DET2
Apellido1	DET3
Apellido2	DET4
Cargo	DET5
Dirección	DET6
Teléfono1	DET7
Teléfono2	DET8
E-mail	DET9
PROYECTO	ILF mantenido con un RET
Identificador	DET1
Nombre	DET2
Fecha de inicio	DET3
Estado	DET4
Descripción	DET5
Cédula del administrador (Clave externa)	DET6
EQUIPO DE DESARROLLO	ILF mantenido con un RET
Cédula miembro equipo	DET1
Identificador Requerimiento	DET2
Versión Requerimiento	DET3
Identificador Proyecto (Clave Externa)	DET4
Cédula Analista (Clave Externa)	DET5
Cédula Administrador (Clave Externa)	DET6
REQUERIMIENTO	ILF no se cuenta como un RET ya que hay subgrupos
Identificador	DET1
Versión	DET2
Nombre	DET3
Grado de dificultad	DET4
Estado	DET5
Costo	DET6
Esfuerzo	DET7
Criterio de adaptación	DET8
Observaciones	DET9
Funcional	DET10
Fecha Cambio	DET11
Motivo	DET12
Descripción	DET13
Cédula Responsable (Clave externa)	DET14
REQ. FUNCIONAL	RET1 bajo REQUERIMIENTO
ID Requerimiento (Clave externa)	Previamente contado como DET1
Versión de Requerimiento (Clave externa)	Previamente contado como DET2
Datos de Salida	DET15
Datos de Entrada	DET16
Parámetros Intermedios	DET17
Operaciones	DET18
REQ. NO FUNCIONAL	RET2 bajo REQUERIMIENTO
ID Requerimiento (Clave externa)	Previamente contado como DET1

Versión Requerimiento (Clave externa)	Previamente contado como DET2
Tipo	DET19
DEPENDENCIA	RET3 bajo REQUERIMIENTO
ID Requerimiento Padre	Previamente contado como DET1
Versión de Requerimiento Padre	Previamente contado como DET2
Identificador de Requerimiento Hijo	Previamente contado como DET1
Versión de Requerimiento Hijo	Previamente contado como DET2
Motivo	DET20
CONFLICTO	RET4 bajo REQUERIMIENTO
ID Requerimiento	Previamente contado como DET1
Versión de Requerimiento	Previamente contado como DET2
ID Requerimiento en conflicto	Previamente contado como DET1
Versión Requerimiento en conflicto	Previamente contado como DET2
Motivo	DET21
DOCUMENTO DE APOYO	RET5 bajo REQUERIMIENTO
Identificador	DET22
Nombre	DET23
Descripción	DET24
ID Requerimiento (Clave externa)	Previamente contado como DET1
Versión Requerimiento (Clave externa)	Previamente contado como DET2
GRUPO	ILF no se cuenta como un RET ya que hay subgrupos
Identificador	DET1
Nombre	DET2
Descripción	DET3
REQUERIMIENTO GRUPO	No se cuenta por implementación
Identificador Grupo	No se cuenta por implementación
Identificador de Requerimiento	No se cuenta por implementación
Versión de Requerimiento	No se cuenta por implementación
REQUERIMIENTO PROYECTO	ILF Mantenido con un RET
Identificador Proyecto (Clave Externa)	DET1
Identificador Requerimiento (Clave Externa)	DET2
Versión de Requerimiento	DET3
Cédula Solicitante (Clave Externa)	DET4
Fecha Solicitud	DET5
Fecha Fin Plan	DET6
Fecha Fin Real	DET7
Prioridad	DET8
Descripción Origen	DET9
Descripción Propósito	DET10
Estado	DET11
Estado Pruebas	DET12

Tabla 2.22. Nivel de complejidad para los ILF y EIF

ENTIDAD	ILF	DET	RET	Complejidad
PERSONA	1 ILF	9 DET	1 RET	Baja
PROYECTO	1 ILF	6 DET	1 RET	Baja
EQUIPO DESARROLLO	1 ILF	6 DET	1 RET	Baja
REQUERIMIENTO	1 ILF	24 DET	5 RET	Media
GRUPO	1 ILF	3 DET	1 RET	Baja

PROYECTO REQUERIMIENTO	1 ILF	12 DET	1 RET	Baja
TOTAL	5 ILF con complejidad Baja y 1 ILF con complejidad Media			

Archivos Interfaces Externas (EIF)

No se presentan archivos de este tipo

Tabla 2.23. Puntos de Función Sin Ajustar

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	5 x7 = 35	1x10 = 10	x15
Archivos de Interfase Externa (EIF)	X5	x7	x10
Entradas Externas (EI)	X3	x4=	x6 =
Salidas Externas (EO)	X4	x5	X7
Consultas Externas (EQ)	X3=	x4 =	X6=
Total de PFSA	45		

Paso 4: ¿Cómo identificar y estimar los puntos de función sin ajustar para los EI?

En la tabla 2.24 podemos observar en la columna izquierda 10 transacciones EI para el sistema de Administración de Requerimientos, en la columna del medio los DET y los FTR correspondientes a cada EI y en la columna derecha el nivel de complejidad correspondiente.

Tabla 2.24. Identificación del los EI y su complejidad.

TRANSACCIONES EI	CONTADAS COMO	COMPLEJIDAD
MANTENIMIENTO PERSONA		
CREAR PERSONA	11 DET Y 1 FTR PERSONA	BAJA
ACTUALIZAR PERSONA	11 DET Y 1 FTR PERSONA	BAJA
BORRAR PERSONA	3 DET Y 1 FTR PERSONA	BAJA
MANTENIMIENTO PROYECTO		
CREAR PROYECTO	8 DET Y 2 FTR PROYECTO Y PERSONA	MEDIA
ACTUALIZAR PROYECTO	8 DET Y 2 FTR PROYECTO Y PERSONA	MEDIA
BORRAR PROYECTO	5 DET Y 1 FTR PROYECTO	BAJA
MANTENIMIENTO REQUERIMIENTO		
CREAR REQUERIMIENTO	26 DET Y 1 FTR REQUERIMIENTO	MEDIA

ACTUALIZAR REQUERIMIENTO	26 DET Y 1 FTR REQUERIMIENTO	MEDIA
BORRAR REQUERIMIENTO	3 DET Y 1 FTR REQUERIMIENTO	BAJA
MANTENIMIENTO EQUIPO		
CREAR EQUIPO	8 DET Y 3 FTR, EQUIPO, PROYECTO Y PERSONA	ALTA
ACTUALIZAR EQUIPO	8 DET Y 3 FTR, EQUIPO, PROYECTO Y PERSONA	ALTA
BORRAR EQUIPO	5 DET Y 1 FTR, EQUIPO	ALTA
MANTENIMIENTO GRUPO		
CREAR GRUPO	5 DET Y 1 FTR GRUPO	BAJA
ACTUALIZAR GRUPO	5 DET Y 1 FTR GRUPO	BAJA
BORRAR GRUPO	3 DET Y 1 FTR GRUPO	BAJA
MANTENIMIENTO PROYECTO REQUERIMIENTO		
CREAR PROYECTO REQUERIMIENTO	14 DET Y 4 FTR PERSONA, PROYECTO, REQUERIMIENTO Y PROYECTO REQUERIMIENTO	ALTA
ACTUALIZAR PROYECTO REQUERIMIENTO	14 DET Y 4 FTR PERSONA, PROYECTO, REQUERIMIENTO Y PROYECTO REQUERIMIENTO	ALTA
BORRAR PROYECTO REQUERIMIENTO	5 DET Y 1 FTR PROYECTO REQUERIMIENTO	BAJA
TOTALES	5 EI con complejidad Alta 4 EI con complejidad Media 9 EI con complejidad Baja	

Paso 5: ¿Cómo identifico y estimo los puntos de función sin ajustar para los EO?

Los siguientes reportes tienen filtros y selección de datos de acuerdo a criterios específicos. Además se referencian varios ILF.

Tabla 2.25. Identificación de los EO y su complejidad.

EO	FTR	DET	Salida / Entrada	Complejidad
CONSULTAR UN REQUERIMIENTO DE UN PROYECTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	20	Parte de Salida	ALTA
CONSULTAR REQUERIMIENTOS SOLICITADOS POR UN USUARIO O ASIGNADOS A UN ANALISTA	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	6	Parte de Salida	MEDIA
CONSULTAR REQUERIMIENTOS INCONCLUSOS	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	8	Parte de Salida	MEDIA
CONSULTAR	4 (REQUERIMIENTO,	19	Parte de Salida	MEDIA

REQUERIMIENTOS DE UN PROYECTO	REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)			
CONSULTAR HISTORIAL DE REQUERIMIENTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO Y PERSONA)	25	Parte de Salida	ALTA
CONSULTAR DOCUMENTOS DE APOYO DE UN REQUERIMIENTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO, DOCUMENTO DE APOYO)	4	Parte de Salida	BAJA
CONSULTAR EQUIPO DE DESARROLLO DE UN REQUERIMIENTO	4 (PERSONA, REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	4	Parte de Salida	BAJA
CONSULTAR EQUIPOS DE PROYECTO	3 (PERSONA Y REQUERIMIENTO, PROYECTO REQUERIMIENTO)	10	Parte de Salida	MEDIA
CONSULTAR DEPENDENCIAS DE UN REQUERIMIENTO	3 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	5	Parte de Salida	BAJA
CONSULTAR CONFLICTOS DE UN REQUERIMIENTO	3 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	5	Parte de Salida	BAJA
CONSULTAR PRIORIDAD DE LOS REQUERIMIENTOS DE UN PROYECTO	3 (PROYECTO, REQUERIMIENTO, REQUERIMIENTO DE PROYECTO)	4	Parte de Salida	BAJA
CONSULTAR GRUPO DE REQUERIMIENTOS	2 (GRUPO, REQUERIMIENTO)	5	Parte de Salida	BAJA
Total	8 EO de complejidad Baja 5 EO de complejidad Media 2 EO de complejidad Alta			

Paso 6: ¿Cómo identifico y estimo puntos de función sin ajustar de los EQ?

La lógica de procesamiento de las siguientes consultas no posee fórmulas matemáticas, ni se hacen cálculos, ni se crean datos derivados. Tampoco se administra ningún ILF durante su procesamiento que altere la conducta del sistema.

Tabla 2.26. Identificación del los EQ con su complejidad.

EQ	FTR	DET	Salida / Entrada	Complejidad
CONSULTAR PERSONA	1 (PERSONA)	10	Parte de Salida	BAJA
CONSULTAR PROYECTO	2 (PROYECTO, PERSONA)	8	Parte de Salida	MEDIA
CONSULTAR UN REQUERIMIENTO DE UN PROYECTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	20	Parte de Salida	ALTA
CONSULTAR REQUERIMIENTOS SOLICITADOS POR UN USUARIO O ASIGNADOS A UN ANALISTA	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	6	Parte de Salida	MEDIA
CONSULTAR REQUERIMIENTOS INCONCLUSOS	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	8	Parte de Salida	MEDIA
CONSULTAR RELACION DE UN REQUERIMIENTO CON CASOS DE USO	3 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	4	Parte de Salida	BAJA
CONSULTAR REQUERIMIENTOS DE UN PROYECTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PERSONA, PROYECTO)	19	Parte de Salida	MEDIA
CONSULTAR HISTORIAL DE REQUERIMIENTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO Y PERSONA)	25	Parte de Salida	ALTA
CONSULTAR DOCUMENTOS DE APOYO DE UN REQUERIMIENTO	4 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO, DOCUMENTO DE APOYO)	4	Parte de Salida	BAJA
CONSULTAR EQUIPO DE DESARROLLO DE UN REQUERIMIENTO	4 (PERSONA, REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	4	Parte de Salida	BAJA
CONSULTAR EQUIPOS DE PROYECTO	3 (PERSONA Y REQUERIMIENTO,	10	Parte de Salida	MEDIA

	PROYECTO REQUERIMIENTO)			
CONSULTAR DEPENDENCIAS DE UN REQUERIMIENTO	3 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	5	Parte de Salida	BAJA
CONSULTAR CONFLICTOS DE UN REQUERIMIENTO	3 (REQUERIMIENTO, REQUERIMIENTO DE PROYECTO, PROYECTO)	5	Parte de Salida	BAJA
CONSULTAR PRIORIDAD DE LOS REQUERIMIENTOS DE UN PROYECTO	3 (PROYECTO, REQUERIMIENTO, REQUERIMIENTO DE PROYECTO)	4	Parte de Salida	BAJA
CONSULTAR GRUPO DE REQUERIMIENTOS	2 (GRUPO, REQUERIMIENTO)	5	Parte de Salida	BAJA
Total	8 EQ's de complejidad Baja 5 EQ's de complejidad Media 2 EQ's de complejidad Alta			

Tabla 2.27. Puntos de Función Desajustados

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	5 x7 = 35	1x10= 10	x15
Archivos de Interfase Externa (EIF)	x5	x7	x10
Entradas Externas (EI)	5 x3 = 15	4 x4= 16	9 x6 = 54
Salidas Externas (EO)	x4	x5	X7
Consultas Externas (EQ)	8 x3= 24	5x4 =20	2X6=12
Total PFSA	187 PFSA		

Paso 7: ¿Cómo calculo el Factor de Ajuste - 14 Características Generales de Sistemas?

- | | | |
|---------------------------------------|---|---|
| 1. Comunicación de datos | 4 | |
| 2. Procesamiento de datos distribuido | 0 | |
| 3. Rendimiento | 3 | |
| 4. Configuración | 2 | |
| 5. Tasa transaccional | 3 | |
| 6. Entrada de datos en línea | 5 | |
| 7. Eficiencia del usuario final | 4 | |
| 8. Actualización en línea | | 3 |

9. Procesamiento complejo	1	
10. Reutilización	0	
11. Facilidad de instalación	0	
12. Facilidad operacional	3	
13. Lugares múltiples	1	
14. Facilidades de cambio		1

Grado Total de Influencia (GTI) 30

Usando la fórmula de Factor de Ajuste de Valor (FAV) sustituimos el GTI:

$$\text{FAV} = (\text{GTI} * 0.01) + 0.65$$

$$\text{FAV} = (30 * 0.01) + 0.65$$

$$\text{FAV} = 0.95$$

Paso 8: ¿Cómo calculo los Puntos de función de Ajuste Final?

Aplicando el Factor de Ajuste (FVA) a los Puntos de Función sin Ajustar:

$$\text{PF} = \text{FAV} * \text{PFSA}$$

$$\text{PF} = 0.95 * 187$$

$$\text{PF} = 177.65$$

Caso 4: Sistema de Control de Horas

Objetivo:

Llevar a cabo el levantamiento de requerimientos del Sistema de Registro de Horas.

Objetivos específicos:

Durante el desarrollo de este proyecto el estudiante tendrá que::

1. Documentar y revisar los requerimientos del software.
2. Utilizar UML y Rational Rose para la especificación de requerimientos del software.
3. Incorporar técnicas básicas de aseguramiento de la calidad para la revisión de los requerimientos del software.

Definición de requerimientos

Con la ayuda de los encargados de cada área funcional de la empresa se han definido las siguientes necesidades.

Con la ayuda de los encargados de cada área funcional de la empresa se han definido las siguientes necesidades.

Req. 1. Administración de Clientes

1.1 Incluir Clientes: El objetivo de este requerimiento es agregar nuevos clientes a la base de datos del sistema. Los datos de entrada para un cliente son:

- ◆ Nombre de cliente
- ◆ Dirección
- ◆ Información de personas contacto. Pueden existir varias personas contacto para un cliente. Para cada persona contacto se indica la siguiente información:
 - Tipo de Contacto (Técnico, Administrativo)
 - Cédula
 - Nombre
 - Teléfono
 - Email
 - Dirección

El procesamiento consiste en insertar el registro del cliente. El nombre del cliente junto con cualquier posible mensaje de error serán las salidas del sistema.

1.2 Eliminar Clientes: El propósito de este requerimiento es eliminar los registros de un cliente que ya no interesen a la empresa o que fue incluido erróneamente. Como dato de entrada se debe permitir seleccionar de una lista el cliente a eliminar, y como dato de salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error. Junto con el cliente debe eliminarse la información de contactos del mismo.

1.3 Modificar Clientes: El objetivo es actualizar la información de un cliente existente o de sus contactos. Como dato de entrada se debe permitir seleccionar de una lista el cliente y sus contactos a modificar. Como resultado se debe desplegar una confirmación de que el

registro fue modificado, o bien, algún mensaje de error.

1.4 Consultar Clientes: El objetivo es mostrar la información de un cliente y sus contactos. Como dato de entrada se debe permitir seleccionar de una lista el cliente a consultar. Como salida se presentan todos los datos del cliente y sus contactos.

Req. 2. Administración de Contrataciones

2.1 Incluir Contrataciones: El objetivo de este requerimiento es agregar la información de nuevas contrataciones a la base de datos del sistema. Los datos de entrada para una contratación son los siguientes:

- Cliente (obtenido a partir de una lista de selección con los clientes existentes).
- Nombre de la contratación
- Fecha Inicio
- Fecha Final Estimada
- Fecha Final Real
- Horas de esfuerzo estimadas
- Estado (En Preventa, Iniciada, Cancelada o Finalizada)
- Métrica de productividad
- Puntos de Función estimados al inicio del proyecto (Albrecht)
- Puntos de Función estimados al final de análisis (Garnus)
- Puntos de Función Reales, actualizados al final de la contratación
- Costo Estimado
- Costo Real
- Observaciones
- Entregables, donde para cada entregable se incluye:
 - Fecha Entregable Estimada
 - Fecha Entregable Real
 - Descripción Entregable
 - Estado (Pendiente, Entregado, Aprobado, Cancelado)
- Riesgos Asociados, donde para cada riesgo se incluye
 - Descripción de Riesgo
 - Estrategia (Evitar, Mitigar, Transferir)
 - Descripción de Estrategia
 - Responsable

El procesamiento consiste en insertar el registro de una contratación. El nombre de la contratación junto con cualquier posible mensaje de error serán las salidas del sistema.

2.2 Eliminar Contrataciones: El propósito de este requerimiento es eliminar el registro de una contratación que ya no le interese a la empresa o que fue incluida erróneamente. Como dato de entrada se seleccionará de una lista la contratación a eliminar y como salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error.

2.3 Modificar Contrataciones: El objetivo es el de actualizar la información de una contratación. Es posible modificar cualquiera de los datos de una contratación, con la excepción del identificador consecutivo manejado por el sistema y del cliente asociado a la

contratación. Como dato de entrada se seleccionará de una lista la contratación a modificar. Como resultado se indicará una confirmación de que el registro fue modificado, o bien, algún mensaje de error.

2.4 Consultar Contrataciones: El objetivo es mostrar la información de una contratación. Como dato de entrada se seleccionará de una lista la contratación a consultar y como salida se presentan todos los datos de la misma.

Req. 3. Administración de Registros de Horas de Labor

3.1 Incluir registros de horas de labor: El objetivo de este requerimiento es registrar en el sistema el evento de un número de horas laboradas por un recurso en un tipo de actividad asociada a una contratación. Los datos de entrada para este proceso son los siguientes

- ◆ Contratación (obtenido a partir de una lista de selección, con los nombres de las contrataciones en estado: preventiva o iniciada)
- ◆ Tipo de actividad (obtenido a partir de una lista de selección con los nombres de las actividades)
- ◆ Recurso (el sistema lo determina a partir del usuario con la sesión actual en el sistema)
- ◆ Fecha de labor
- ◆ Cantidad de horas
- ◆ Descripción: detalle de las labores realizadas

El procesamiento consiste en insertar un Registro de Horas de Labor. El nombre de la contratación y el de la actividad junto con cualquier posible mensaje de error serán las salidas del sistema.

3.2 Eliminar Registros de labor: El propósito de este requerimiento es eliminar un registro de horas de labor. Como dato de entrada se seleccionará de una lista el registro de horas de labor y como salida una confirmación de que el registro fue eliminado, o bien algún mensaje de error.

3.3 Modificar Registros de labor: El objetivo es actualizar la información de un registro de horas de labor. Como resultado se indicará una confirmación de que el registro fue modificado, o bien, algún mensaje de error.

4.4 Consultar Registros de labor: El objetivo es mostrar la información de un registro de horas de labor.

Req. 4. Administración de Tipos de Actividad

4.1 Incluir tipo de actividad: El objetivo de este requerimiento es agregar la información de un tipo de actividad a la base de datos del sistema. Una actividad consiste de un tipo de clasificación de las labores posibles de un recurso de la empresa. Ejemplos de tipos de actividad son: Venta, Recolección de requerimientos, Análisis, Diseño de la arquitectura, Programación, Pruebas, Montaje HTML, etc.

Los datos de entrada para un tipo de actividad son los siguientes:

- ◆ Descripción del tipo de actividad

La identificación del tipo de actividad junto con cualquier posible mensaje de error serán

las salidas del sistema.

4.2 Eliminar Tipo de Actividad: El propósito de este requerimiento es eliminar el registro de un tipo de actividad que ya no le interese a la empresa o que fue incluida erróneamente. Como dato de entrada se seleccionará de una lista el tipo de actividad a eliminar y como salida una confirmación de que el registro fue eliminado o bien, un mensaje de error.

4.3 Modificar Tipo de Actividad: El objetivo es actualizar la información de un tipo de actividad. La información que es posible modificar es la descripción del tipo de actividad. Como dato de entrada se seleccionará de una lista la actividad que se está modificando. Como resultado se mostrará una confirmación de que el registro fue modificado o bien, algún mensaje de error.

4.4 Consultar Tipos de Actividad: El objetivo es mostrar la descripción de un tipo de actividad seleccionada de una lista. Como salida se presenta la descripción del mismo,

Req. 5. Administración de Áreas Funcionales de la Empresa

5.1 Incluir área: El objetivo de este requerimiento es agregar nuevas áreas funcionales a la base de datos del sistema. Ejemplos de áreas son: Desarrollo, Diseño Gráfico y Data Center. Los datos de entrada para un área son:

- Nombre del área
- Nombre del encargado del área

El procesamiento consiste en insertar el registro del área. Cualquier posible mensaje de error o de éxito serán las salidas del sistema.

5.2 Eliminar áreas: El propósito de este requerimiento es eliminar el registro de una área que ya no interese a la empresa o que fue incluida erróneamente. Como dato de entrada se seleccionará de una lista el área a eliminar y como salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error o éxito.

5.3 Modificar áreas: El objetivo es actualizar la información de un área existente. Como resultado se indicará una confirmación o bien, algún mensaje de error.

5.4 Consultar áreas: El objetivo es mostrar información de un área seleccionada de una lista. Como salida se presentan todos los datos de la misma.

Req. 6. Administración del equipo de trabajo

6.1 Incluir recurso: El objetivo es permitir agregar la información de nuevos recursos (personal de la empresa) a la base de datos del sistema. Los datos de un recurso son los siguientes:

- Área (obtenido a partir de una lista de selección con las áreas existentes)
- Nombre
- # de cédula
- Teléfono de la casa.
- Teléfono celular.
- Email.
- Dirección
- Datos de uso del sistema

- Login
- Password
- Tipo Usuario

Durante la inclusión debe insertarse el registro del recurso. El nombre, junto con cualquier posible mensaje de error serán las salidas del proceso.

6.2 Eliminar recursos: El propósito de este requerimiento es eliminar el registro de un recurso que ya no labora para la empresa o que fue incluido erróneamente. Como dato de entrada se seleccionará de una lista el recurso que se desea eliminar y como salida una confirmación de que el registro fue eliminado, o bien, algún mensaje de error.

6.3 Modificar recursos: El objetivo es permitir actualizar la información de un recurso. Como dato de entrada se seleccionará de una lista el recurso que se está modificando. Como resultado se indicará una confirmación de que el registro fue modificado, o bien, algún mensaje de error.

6.4 Consultar recursos: El objetivo es mostrar la información de un recurso. El dato de entrada se selecciona de una lista el recurso a consultar y como salida se presentan todos los datos asociados al mismo.

Req. 7. Consultas Complejas

7.1 Consultar horas de labor para una contratación: El objetivo es mostrar todos los registros de horas de labor asociados a una contratación, opcionalmente por fechas. Como datos de entrada se seleccionará de una lista la contratación a consultar y opcionalmente un rango de fechas en los cuales se registraron horas de labor. Como salida se mostrará una lista de registros de horas de labor, con toda la Información especificada en el requerimiento 3.1 (Contratación, Tipo de actividad, Recurso, Fecha de labor, Cantidad de horas y Descripción); así como el total de registros encontrados y la suma de horas de dichos registros.

7.2 Consultar horas de labor para un tipo de actividad: El objetivo es mostrar todos los registros de horas de labor asociados a un tipo de actividad, opcionalmente por fechas. Como datos de entrada se seleccionará de una lista el tipo de actividad y opcionalmente un rango de fechas en los cuales se registraron horas de labor. Como salida se mostrará una lista de registro de horas de labor, con toda la información especificada en el requerimiento 3.1; así como el total de registros encontrados y la suma de horas de dichos registros.

7.3 Consultar horas de labor para un recurso: El objetivo es mostrar todos los registros de horas de labor registrados por un recurso, opcionalmente por fechas. Como datos de entrada se seleccionará de una lista el recurso y opcionalmente un rango de fechas en los cuales se registraron horas de labor. Como salida se mostrará una lista de registro de horas de labor, con toda la información especificada en el requerimiento 3.1; así como el total de registros encontrados y la suma de horas de dichos registros.

7.4 Cualquier otra consulta que el analista recomiende después de hacer un análisis de las necesidades de métricas para la empresa.

Req. 8. Requisitos de seguridad

8.1 Ingreso al sistema: Solamente puede hacer uso del sistema personal de la empresa

registrado como recurso en la base de datos del mismo. Para el ingreso al sistema se utilizará una pantalla de autenticación en donde debe especificarse el login y password del recurso que hará uso del sistema y en las restantes funcionalidades del sistema debe validarse que exista un usuario con una sesión activa. Adicionalmente se deben considerar tres tipos de usuario (Supervisor, Operativo y Administrativo).

8.2 Salida del sistema: debe existir un mecanismo dentro del menú de opciones del sistema para salir del mismo, el cual invalidará la sesión activa y dirigirá nuevamente a la pantalla de inicio.

Req. 9. Requisitos técnicos

9.1 Tiempos de respuesta: 2 segundos.

9.2 Plataformas de software: La aplicación debe ser en "Web services" de .NET. La base de datos MSQlServer. Se puede utilizar cualquier lenguaje de programación que incluya la plataforma .NET.

A continuación se detallarán cada uno de los pasos para estimar el tamaño del software en Puntos de Función para el Sistema de Control de Horas.

Paso 1: ¿Cómo determinar el tipo de Puntos de Función?

Estos pertenecen al Tipo 1, Puntos de Función para Proyectos en Desarrollo

Paso 2 ¿Cómo identificar el alcance del conteo y las fronteras de la aplicación?

Aunque es una aplicación cliente/servidor se considera que toda la funcionalidad está dentro de las fronteras de la aplicación porque no interactúa con ninguna otra aplicación.

Paso 3: ¿Cómo calculo los ILF y los EIF?

De acuerdo al Sistema de Control de Horas en la tabla 37 se muestra la identificación de los ILF, los EIF y sus respectivos DET y RET, en la tabla 38 el nivel de complejidad para dichas funciones de datos y en la tabla 2.28 el total de los PFSA correspondiente.

Tabla 2.28. Identificación de funciones de datos.

CAMPOS DEL ARCHIVO	CONTADOS COMO
CLIENTE	ILF. No se cuenta como RET ya que hay subgrupos
NombreCliente	DET1
DirecciónCliente	DET2
CódigoCliente	No se cuenta por ser de implementación
CONTACTO	RET1 bajo Cliente
NombreCliente(Llave foránea)	Previamente contada como DET1
TipoContacto	DET3
CédulaContacto	DET4
NombreContacto	DET5
TeléfonoContacto	DET6
EmailContacto	DET7
DirecciónContacto	DET8
CONTRATACION	ILF. No se cuenta como RET ya que hay subgrupos
CódigoCliente(Llave foránea)	DET1
NombreContratación	DET2

CódigoContratación	No se cuenta por ser de implementación
FechaInicioContratación	DET3
FechaFinalEstimadaContratación	DET4
FechaFinalRealContratación	DET5
HorasEsfuerzoEstimadasContratación	DET6
EstadoContratación	DET7
MétricaProductividadContratación	DET8
PuntosFunciónInicialesContratación	DET9
PuntosFunciónFinalAnálisisContratación	DET10
PuntosFunciónRealesContratación	DET11
CostoEstimadoContratación	DET12
CostoRealContratación	DET13
ObservacionesContratación	DET14
EncargadoContratación (Llave foránea)	DET15
ENTREGABLE	RET1 bajo Contratación
CódigoContratación (Llave foránea)	No se cuenta por ser de implementación
CódigoEntregable	No se cuenta por ser de implementación
DescripciónEntregable	DET16
FechaEntregaEstimadaEntregable	DET17
FechaEntregaRealEntregable	DET18
EstadoEntregable	DET19
RIESGO	RET2 bajo Contratación
CódigoContratación (Llave foránea)	No se cuenta por ser de implementación
DescripciónRiesgo	DET20
CódigoRiesgo	No se cuenta por ser de implementación
EstrategiaRiesgo	DET21
DescripciónEstrategiaRiesgo	DET22
CédulaResponsable (Llave foránea)	DET23
ÁREA FUNCIONAL	ILF mantenido con un RET
NombreÁreaFuncional	DET1
CódigoÁreaFuncional	No se cuenta por ser de implementación
ResponsableÁreaFuncional (Llave foránea)	DET2
RECURSO HUMANO	ILF mantenido con un RET
CódigoÁreaFuncional (Llave Foránea)	DET1
CédulaRecursoHumano	DET2
NombreRecursoHumano	DET3
TeléfonoCasaRecursoHumano	DET4
TeléfonoCelRecursoHumano	DET5
EmailRecursoHumano	DET6
DirecciónRecursoHumano	DET7
NombreUsuarioRecursoHumano	DET8
TipoUsuarioRecursoHumano	DET9
ACTIVIDAD	ILF mantenido con un RET
DescripciónActividad	DET1
CódigoActividad	No se cuenta por ser de implementación
HORAS LABOR	ILF mantenido con un RET
CédulaRecursoHumano (Llave Foránea)	DET1
CódigoContratación (Llave Foránea)	DET2
CódigoActividad (Llave Foránea)	DET3
FechaHorasLabor	DET4
CantidadHorasLabor	DET5
DescripciónHorasLabor	DET6

Tabla 2.29. Nivel de complejidad para los ILF y EIF

<i>Entidad</i>	<i>ILF</i>	<i>DET</i>	<i>RET</i>	<i>Complejidad</i>
Cliente	1 ILF	8 DET	1 RET	Bajo
Contratación	1 ILF	23DET	2 RET	Medio
Área Funcional	1 ILF	2 DET	1 RET	Bajo
Recurso Humano	1 ILF	9 DET	1 RET	Bajo
Actividad	1 ILF	1 DET	1 RET	Bajo
Horas de Labor	1 ILF	6 DET	1 RET	Bajo
TOTAL	5 ILF con complejidad Baja y 1 ILF con complejidad Media			

Nota: como solicitud técnica todas las tablas a excepción de los RET deben manejarse con identificadores numéricos para agilizar el procesamiento interno. Estos identificadores numéricos se manejan en forma automática, no deberían mostrarse en las interfaces porque al usuario no le interesa, son campos de implementación por lo que no deberían contarse como DET. En las tablas de Conteo de los ILF debe aparecer como un campo pero indicarse “que no se cuenta por ser de implementación”.

- Ejemplo: Cliente, Contrataciones, Actividades, Áreas deberían tener este identificador numérico para no manejar hileras de texto como llaves.
- La tabla de Recursos Humanos tiene el campo cédula por lo que no es necesario agregar otro identificador numérico, más bien es desperdicio.
- Para las tablas de los RET es suficiente con la llave foránea del Padre.
- En la tabla de Horas Labor no es necesario un identificador numérico porque con la llave compuesta que se crea de la relación con las tres tablas Contratación, Actividad y Recurso es suficiente.

Las llaves foráneas cuando están en un RET no se cuentan porque esto se hizo por motivo de implementación. Se cuentan solo en el ILF padre.

- Ejemplo: la llave foránea Código Contratación que se encuentra en los RET Entregables y Riesgos solo se cuenta en la tabla Padre, es decir en la tabla Contratación).

Cuando la llave foránea está en un ILF si se cuenta dentro de los DET de ese ILF porque en este caso esa tabla está necesitando conocer información de otra tabla y esto no es por motivo de implementación sino por solicitud del usuario.

- Ejemplo 1: en la tabla Horas de Labor hay tres llaves foráneas que se deben contar como DET.
- Ejemplo 2: De acuerdo a los requerimientos se necesita conocer el responsable de cada área funcional, entonces, en la tabla de Área Funcional debería estar el campo clave (Cédula Rec. Humano Responsable) que permita relacionar a esta tabla con Recurso Humano. Este campo se cuenta porque es el usuario el que lo está pidiendo.

Tabla 2.30. PFSA para las funciones de datos.

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	$5 \times 7 = 35$	$1 \times 10 = 10$	$\times 15$
Archivos de Interfase Externa (EIF)	$\times 5$	$\times 7$	$\times 10$
Entradas Externas (EI)	$\times 3$	$\times 4$	$\times 6$
Salidas Externas (EO)	$\times 4$	$\times 5$	$\times 7$
Consultas Externas (EQ)	$\times 3$	$\times 4$	$\times 6$
Total PFSA	45 PFSA		

Archivos Interfaces Externas

No se presentan archivos de este tipo

Paso 4: ¿Cómo calculo los EI's?

En la tabla 2.31 podemos observar en la columna izquierda 18 transacciones EI para el Sistema de Control de Horas, en la columna del medio los DET y los FTR de cada EI y en la columna derecha el nivel de complejidad correspondiente

Tabla 2.31. Identificación del los EI y su complejidad.

TRANSACCIONES	CONTADAS COMO	COMPLEJIDAD
CLIENTE		
Insertar Clientes	$8 + 2 = 10$ DET, 1 FTR(Cliente)	Bajo
Modificar Clientes	$8 + 2 = 10$ DET, 1 FTR(Cliente)	Bajo
Eliminar Cliente	$1 + 2 = 3$ DET, 1 FTR(Cliente)	Bajo
CONTRATACIÓN		
Insertar Contrataciones	$23 + 2 = 25$ DET, 3 FTR(Contratación/Recurso Humano /Cliente)	Alto
Modificar Contrataciones	$23 + 2 = 25$ DET, 3 FTR(Contratación/Recurso Humano /Cliente)	Alto
Eliminar Contrataciones	$1 + 2 = 3$ DET, 1 FTR(Contratación)	Bajo
ACTIVIDAD		
Insertar Actividades	$1 + 2 = 3$ DET, 1 FTR(Actividad)	Bajo
Modificar Actividades	$1 + 2 = 3$ DET, 1 FTR(Actividad)	Bajo
Eliminar Actividades	$1 + 2 = 3$ DET, 1 FTR(Actividad)	Bajo
HORAS LABOR		
Insertar Horas Labor	$6 + 2 = 8$ DET, 4 FTR(Horas Labor / Actividad / Recurso Humano / Contratación)	Alto
Modificar Horas Labor	$6 + 2 = 8$ DET, 4 FTR(Horas Labor / Actividad / Recurso Humano / Contratación)	Alto
Eliminar Horas Labor	$1 + 2 = 3$ DET, 1 FTR(Horas Labor)	Bajo

RECURSO HUMANO		
Insertar Recurso Humano	9 + 2 = 11 DET, 2 FTR(Recurso Humano / Área Funcional)	Medio
Modificar Recurso Humano	9 + 2 = 11 DET, 2 FTR(Recurso Humano / Área Funcional)	Medio
Eliminar Recurso Humano	1 + 2 = 3 DET, 1 FTR(Recurso Humano)	Bajo
ÁREA FUNCIONAL		
Insertar Área Funcional	2 + 2 = 4 DET, 2 FTR(Área Funcional / Recurso Humano)	Bajo
Modificar Área Funcional	2 + 2 = 4 DET, 2 FTR(Área Funcional / Recurso Humano)	Bajo
Eliminar Área Funcional	1 + 2 = 3 DET, 1 FTR(Área Funcional)	Bajo
TOTAL	12 EI con complejidad baja, 2 EI Media y 4 EI Alta	

Los RET de un ILF no se cuentan como FTR.

- Ejemplo: Contacto, Entregables y Riesgos no son FTR

Los DET de las transacciones **INCLUIR** y **MODIFICAR** se cuentan de acuerdo a los campos que el usuario desea que se presenten en la pantalla. En nuestro caso me interesan todos los campos de las tablas menos los identificadores numéricos que les pedí agregar a algunas tablas por motivos implementación (para mejorar el rendimiento del sistema).

La forma de contar los FTR de las transacciones **INCLUIR** y **MODIFICAR** se deduce de las llaves foráneas y de la decisión del usuario. Se cuenta el ILF al que se le está haciendo el cálculo de EI como un FTR y se agrega las tablas correspondientes a llaves foráneas como otros FTR (si el usuario requiere que se incluya o modifique algún campo de esos FTR).

- Ejemplo: Contrataciones tiene una llave foránea a Recursos Humanos entonces se contarían dos FTR (Contrataciones y si el usuario requiere información de Recursos humanos este se agrega como FTR). En nuestro caso si lo requiere
- En la Tabla “Horas labor” se deben contar 4 FTR que corresponde a 1 de Horas Labor más 3 de llaves foráneas porque el usuario si requiere información de esas tablas.

La forma de contar los DET y los FTR de la transacción **BORRAR** cuando **NO** hay llave compuesta es la siguiente:

- Se cuenta como DET solo la llave principal más 2 de supuestos. La justificación de esto es que para el usuario esa llave compuesta tiene sentido esa llave como una pieza de información.
- Respecto a los FTR solo se cuenta la tabla a la que se les está calculando los EI como un FTR aunque hayan llaves foráneas.

La forma de contar los DET y los FTR de la transacción **BORRAR** cuando hay llave compuesta es la siguiente:

- Una llave compuesta se cuenta como un DET. Confirmar esto en el ejemplo visto en clase: la tabla Asignador de Tareas tiene una llave compuesta por el Número de Tarea y el SSN del Empleado. En el ejemplo cuando se cuantifican los DET de “Borrar una Asignación”, cuentan esa llave compuesta como un DET más 2 DET de supuestos dando un total de 3 DET. Respecto a los FTR cuentan solo la tabla Asignador de Tareas como un FTR.
- En nuestro caso en la tabla “Horas de Labor” la llave compuesta de Contrataciones, Actividad y Recursos es la llave primaria por lo que se cuenta como un DET. Se deben agregar 2 de supuestos. Respecto a los FTR a pesar de que hay dos llaves foráneas solo se cuenta un FTR que corresponde a la tabla “Horas de Labor”.

La forma de contar los DET y los FTR de la transacción BORRAR cuando **NO** hay llave compuesta es la siguiente:

- Se cuenta como DET solo la llave principal más 2 de supuestos.
- Respecto a los FTR solo se cuenta la tabla a la que se les está calculando los EI como un FTR aunque hayan llaves foráneas.

Paso 5: ¿Cómo calculo los EQ's?

En la tabla que presentan deben indicar si es por parte de salida o es por parte de entrada.

La parte de entrada se refiere a los criterios de búsqueda y para el cálculo se debe seguir exactamente las mismas reglas de un EI (se agregan DET por supuestos y se usan las mismas tablas de complejidad).

La parte de salida es el resultado de la búsqueda y para el cálculo se debe seguir exactamente las mismas reglas de un EO (NO se agregan DET por supuestos y se usan las mismas tablas de complejidad).

Para esta aplicación se tomaron las consultas (que son parte del IMEC) como un EQ. Aunque se deben definir varios criterios de búsqueda, por lo general la salida va a ser siempre mayor porque se supone que deberían desplegar una mayoría de datos del ILF. Recuerde que el número de DET es un rango aproximado. Si es la parte de salida no deben incluir supuestos dentro de los DET porque como acabo de indicar siguen las mismas reglas de los EO.

Tabla 2.32. Identificación del los EQ y su complejidad.

EQ	FTR	DET Entrada	DET Salida	Complejidad
Cliente				
Consultar Cliente	1 (Cliente)	3	8	Bajo
Contratación				
Consultar Contratación	3 (Contratación / Cliente/ Recurso Humano)	3	26	Alto
Actividad				
Consultar Actividad	1 (Actividad)	3	1	Bajo
Horas Labor				
Consultar Horas Labor	4 (Horas Labor / Contratación / Actividad / Recurso Humano)	5	6	Medio
Recursos Humanos				
Consultar Recursos Humanos	2 (Recurso Humano / Área Funcional)	3	9	Medio
Área Funcional				
Consultar Área Funcional	2 (Área Funcional / Recurso Humano)	3	2	Bajo
TOTAL	3 EQ de complejidad Baja, 2 EQ de complejidad Media y 1 Alta			

Paso 6: ¿Cómo calculo los EO's?

En la tabla 2.33 se muestran 3 consultas o reportes lo cuales fueron considerados EO porque cuando se generan se realizan cálculos con los datos.

El número de DET de los EO depende de la cantidad de campos que el usuario quiere que se presente en el reporte. El número de FTR depende de las tablas que se están accedendo para presentar el reporte que el usuario quiere.

Tabla 2.33. Nivel de complejidad para los EO.

EO	FTR	DET	Complejidad
Consultas Complejas			
Consultar Horas de Labor para una Contratación	5 (Horas de Labor /Contratación / Cliente / Recurso Humano / Actividad)	7	Alto
Consultar Horas de Labor para una Actividad	4 (Horas de Labor / Actividad / Recurso / Contratación)	5	Medio
Consultar Horas de Labor para un Recurso	5 (Horas de Labor/Recurso Humano / Área Funcional / Actividad / Contratación)	6	Alto
TOTAL	2 EO de complejidad Alta y 1 Media		

La tabla 2.34 muestra el total de PFSA, tanto para las funciones transaccionales y su complejidad, como para las funciones de datos y su complejidad. Como se puede observar del cálculo de las funciones transaccionales y su complejidad se obtuvo un total de 12 EI de complejidad baja, 2 EI de complejidad media, 4 EI de complejidad alta, 3 EQ de complejidad baja, 2 EQ de complejidad Media, y 1 EQ de complejidad Alta, 1 EO de complejidad Media y 2 EO de complejidad Alta.

Tabla 2.34. Puntos de Función Desajustados

Componentes	Niveles de Función		
	Baja	Media	Alta
Archivos Lógicos Internos (ILF)	$5 \times 7 = 35$	$1 \times 10 = 10$	$x15$
Archivos de Interfase Externa (EIF)	$x5$	$x7$	$x10$
Entradas Externas (EI)	$12 \times 3 = 36$	$2 \times 4 = 8$	$4 \times 6 = 24$
Salidas Externas (EO)	$x4$	$1 \times 5 = 5$	$2 \times 7 = 14$
Consultas Externas (EQ)	$3 \times 3 = 9$	$2 \times 4 = 8$	$1 \times 6 = 6$
Total PFSA	155 PFSA		

Paso 7: ¿Cómo calculo el Factor de Ajuste de Valor- 14 Características Generales de Sistemas?

Puntaje	
1. Comunicación de datos	4
2. Procesamiento distribuido de datos	4
3. Rendimiento	4
4. Configuración operacional	2
5. Porcentaje de transacción	0
6. Entrada de datos en línea	5
7. Eficiencia del Usuario Final	2
8. Actualización en Línea	4
9. Procesamiento Complejo	2
10. Reutilización	1
11. Facilidad de instalación	0
12. Facilidad operacional	0
13. Lugares múltiples	0
14. Facilidad de cambio	0
Grado Total de Influencia (GTI)	28

Usando la fórmula de Factor de Ajuste de Valor (FAV) sustituimos el Grado Total de Influencia (GTI) con el resultado de la suma de las 14 características:

$$FAV = (GTI * 0.01) + 0.65$$

$$FAV = (28 * 0.01) + 0.65$$

$$FAV = 0.93$$

Paso 8: ¿Cómo calculo los Puntos de función de Ajuste Final?

Aplicando el Factor de Ajuste de Valor (FAV) a los Puntos de Función sin Ajustar obtenemos los Puntos de Función Ajustados conocidos como PF:

$$PF = FAV * PFSA$$

$$PF = 0.93 * 155$$

$$PF = 144.15$$

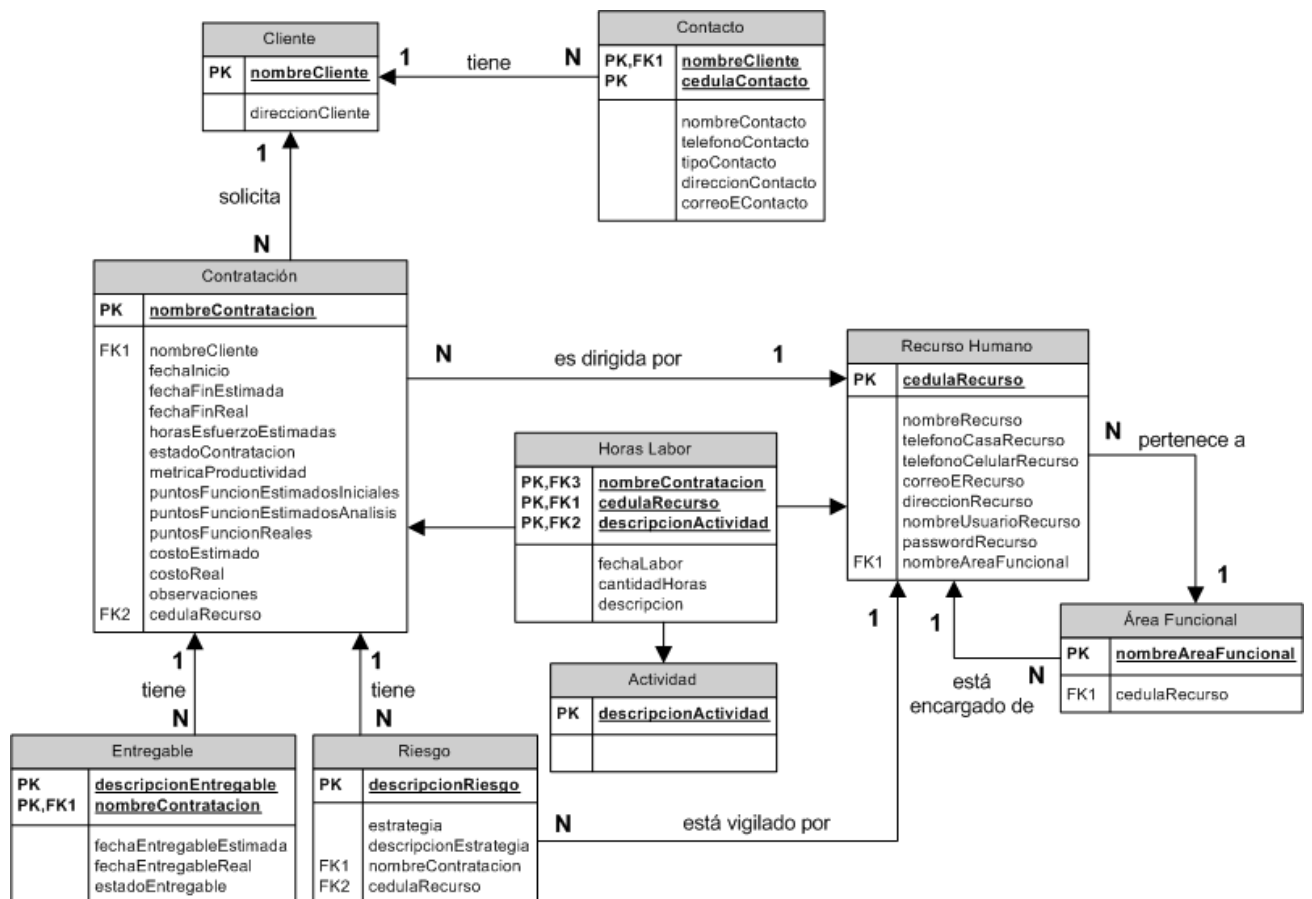


Figura 6. Diagrama de Entidad Relación

Capítulo 3

Estimación del costo y el esfuerzo con base en los Puntos de Función estimados

Procedimiento para estimar costo y esfuerzo cuando no contamos con datos históricos.

Para estimar el costo y el esfuerzo de un nuevo proyecto necesito:

- PF del nuevo proyecto. (Supongamos que el nuevo proyecto se estimó en 65 PF.)
- Datos históricos de Productividad y del Costo por PF.

Aunque el ejemplo que se muestra a continuación presenta los datos históricos de un único proyecto es conveniente disponer de datos históricos de diferentes proyectos para obtener un promedio.

¿Cómo genero los datos históricos? Supongamos que la organización cuenta con los siguientes datos obtenidos de un proyecto anterior ya desarrollado:

PF : 49
Duración: 3 meses
Personal: 2 programadores a 400 000 el mes
1 líder a 600 000 el mes
30 horas extras a 100 000 en total
Local y equipo: 3000 000

Con base en los datos anteriores se va a calcular la Productividad y luego el Costo x PF.

1. Cálculo de la productividad:

$$\text{Productividad} = \text{PF} / \text{Esfuerzo}$$

Según esta fórmula para calcular la productividad necesito el esfuerzo generado para crear un proyecto (ya concluido) y los PF de éste.

¿Cómo calculo el esfuerzo?

- a. Se requiere conocer el total de horas gastadas en un determinado proyecto durante el tiempo que duró.
- b. Una vez que se obtiene ese dato se debe convertir a Personas Mes (P/M) para conocer el esfuerzo necesario por persona en un mes.

Suponiendo que el proyecto duró 3 meses el total de horas gastadas se obtiene de la siguiente manera:

Total de horas en un mes: 6 horas * 5 días * 4 semanas * 1 mes = 120 horas

Total horas en 3 meses con 3 personas: 3 personas * 120 horas * 3 meses = 1080 + 30 horas extras = 1110 horas (equivale a las horas gastadas por las tres personas en los 3 meses incluyendo las 30 horas extras)

Esfuerzo = Total de horas en 3 meses / Total de horas en 1 mes = 1110/ 120= 9.25 PM (convertido a persona/mes)

Teniendo el esfuerzo y los PF se puede calcular la productividad.

Productividad = PF / Esfuerzo = 49 PF / 9.25 PM = **5.29 PF/PM** (En un mes una persona puede crear 5.29 PF.)

2. Cálculo del Costo por PF.

$$\text{Costo x PF} = \text{Costo total de proyecto} / \text{PF}$$

El Costo total del proyecto incluye:

- gastos operativos (como: salarios de desarrolladores y de personal de soporte con sus correspondientes cargas sociales y horas extras)
- herramientas de software
- equipo de cómputo
- espacio de oficina, instalaciones, etc.

Costo Total = 4,200.000 de salarios + 30.000 de horas extras + 3,000.000 de local y equipo = 7,300.000

Costo x PF = Costo total / PF = 7,300.000 / 49 = **148,979 colones x PF**

3. Estimación del esfuerzo y el costo del nuevo proyecto

Con base en los datos históricos calculados en los puntos 2 y 3 tenemos:

- Productividad media= 5.29 PF/ PM
- Costo x PF = 148.979 colones x PF

Nota: La productividad media es el promedio de la productividad de varios proyectos, sin embargo en este ejemplo para simplificar el problema solo se utilizó un proyecto.

Para estimar el esfuerzo y el costo del nuevo proyecto se utilizan las siguientes fórmulas:

1. Esfuerzo estimado = PF / Productividad media

2. Costo estimado = PF * Costo PF

Utilizando las 2 fórmulas anteriores realizo el siguiente cálculo:

1. Esfuerzo estimado = PF / Productividad media = 65 PF / 5.29 PF /PM = **12.28 PM** (12 PF por persona en un mes aproximadamente)

2. Costo estimado = PF * Costo x PF = 65 * 148.979 = 9,683.635 colones

Recuerde que el costo por PF era 148,979 y que la aplicación tenía un tamaño de 65 PF.

Según el estándar de productividad del IEEE:

$$\text{Productividad} = \frac{\text{Unidades de salida}}{\text{Unidades de entrada}} \quad \frac{(\text{Cantidad del producto})}{(\text{Esfuerzo gastado en crearlo})}$$

Unidades de salida: Líneas de código, Puntos de Función, Documentos.

Unidades de entrada: Las entradas representan el esfuerzo aplicado para desarrollar el producto de software. Una de las principales medidas se expresa en unidades de horas-persona. Esto corresponde a una hora de tiempo gastada por un miembro del personal. Esta medida debe incluir las horas extras, las compensadas y las no compensadas.

Los atributos de esta medida son los siguientes:

Horas directas: Es el esfuerzo realizado por los miembros para definir o crear el producto. Incluye análisis, diseño, programación, pruebas y documentación.

Horas soporte: Es el esfuerzo realizado por los miembros que no definen o crean el producto pero asisten a las personas que lo hacen. Incluye al personal que realiza tareas de soporte, de administración de la configuración y del aseguramiento de la calidad.

Ejemplo:

Si se tiene un programa de 500 LDC y el esfuerzo para crearlo fue de 40 horas-persona

Productividad= 500 LDC / 40 hora-persona= 12.5 LDC por hora persona

En una hora una persona puede crear 12.5 LDC.

De acuerdo a estadísticas en USA la productividad promedio es de 4 y 20 PF/PM.

En un mes una persona puede crear entre 4 y 20 PF.

Procedimiento para estimar costo y esfuerzo cuando contamos con datos históricos.

Datos históricos de Productividad y del Costo por PF del mercado.

1. Estimar los PF del proyecto a planificar.

Seguir la metodología descrita en el capítulo 1. Supongamos que tenemos un proyecto con 130 PF.

2. Cálculo del tiempo total del proyecto en horas

De acuerdo a históricos obtenidos durante 3 años de estudiantes del curso Ingeniería de Software, en donde los estudiantes trabajan durante dos semestres proyectos con un tamaño entre 150 y 200 Puntos de Función tenemos que la productividad promedio es de 5.91 horas por Punto de Función (aproximadamente 6 horas por PF). Hay que tomar en cuenta que los estudiantes desconocen la metodología de análisis y diseño orientada a objetos y las herramientas (Rational, Rose, ASP.NET), ya que esto es parte de lo que aprenden en dicho curso para desarrollar el proyecto.

Tiempo total (en horas) = $PF * 6 \text{ horas} = 130 * 6 \text{ horas} = 780 \text{ horas}$

3. Cálculo de la duración en meses

Para calcular el esfuerzo que se requiere en meses se dividen las 780 horas entre 120 (equivalente a 6 horas efectivas de trabajo por día x 20 días hábiles al mes). Si se quiere en días /hombre se divide solamente entre 6.

Duración con 1 persona = $780 / 120 = 6,5 \text{ mes/hombre}$ (o PM persona/mes). Esto significa que con una persona el proyecto va a durar 6,5 meses.

4. Cálculo de la duración total

Para obtener una duración total aproximada del proyecto algunos estudios recomiendan dividir los meses/hombre (obtenidos en el punto anterior) entre los miembros del equipo que trabajarán en el proyecto. Sin embargo esto no es tan cierto, por lo tanto lo más conveniente, especialmente si se trabaja con una herramienta como Project, dejar que la herramienta estime la duración total. Para lograrlo se debe definir una fecha inicial en el calendario del proyecto y distribuir la duración (en meses u horas) entre las actividades y tareas del WBS de acuerdo a las recomendaciones de la siguiente tabla.

FASE DE DESARROLLO	PORCENTAJE DEL TOTAL DEL PROYECTO
Conceptualización y Planificación	12%
Análisis y diseño	43%
Programación y pruebas	43%
Instalación (***)	2%

Los estimados de la tabla anterior se basa en observaciones realizadas sobre desarrollo de estudiantes utilizando un paradigma incremental, la metodología de análisis y diseño orientada a objetos, una plataformas de arquitectura N capas y una aplicación WEB en ASP.NET y como lenguaje de programación C#.

(***) En la fase de instalación lo único que se hace es preparar y probar el software de instalación y los manuales del usuario y el de instalación.

Referencias bibliográficas

[1] Stephen H. Kan. “Metrics and Models in Software Quality Engineering. Second Edition. Addison Wesley, 2003

[2] Garmus y Herron David. “Measuring the Software Process. A Practical Guide to Functional Measurements”. Addison Wesley, 2001.