

INSTRUCTIVO PARA LA CUENTA DE PUNTOS FUNCIÓN

INDICE

Introducción	2
Frontera de la aplicación.....	3
Cuenta de Puntos Función sin ajustar.....	3
Funciones de Datos	4
Funciones Transaccionales.....	4
Mecanismo	5
Ficheros Lógicos Internos (ILF)	5
Ficheros Lógicos Externos (ELF)	5
Determinación de complejidad de los Ficheros Lógicos	6
Entradas (EI)	7
Determinación de la complejidad de las Entradas.....	7
Salidas (EO)	8
Consultas (EQ)	8
Determinación de la complejidad de las Salidas y Consultas	9
Total de Puntos Función	10
Anexo 1 – Identificación de Transacciones de Datos	11
Anexo 2 - Definición de términos utilizados.....	12

Introducción

Sabido es que una forma de resolver problemas es dividiendo el problema en otros más pequeños y fáciles de comprender. Así pues, los problemas que en un contexto global se tornan complejos analizando sus componentes o clases se simplifican en la medida que son más pequeños.

El Análisis de Punto Función es una técnica que mediante la descomposición de un sistema en componentes más pequeños, permite que éstos puedan ser mejor comprendidos y analizados en forma individual. También provee una técnica estructurada para resolver problemas [Longstreet, D., 2001].

El Punto Función es una medida del tamaño de un sistema de software y del proyecto que lo construye. Es una unidad de medida así como la hora lo es para medir el tiempo o el kilómetro para medir la distancia. El Análisis de Punto Función se basa en la teoría de que las funciones de una aplicación son la mejor medida del tamaño de un sistema. El Punto Función mide el software mediante la cuantificación de la funcionalidad que el sistema le brinda al usuario basado fundamentalmente en el diseño lógico. Es independiente del lenguaje de computación, de la metodología de desarrollo, de la tecnología utilizada y de la capacidad del equipo de trabajo para desarrollar la aplicación [IFPUG, CPM., 2000].

El proceso de cuenta que se describe se alinea fuertemente con el manual de práctica de Punto Función desarrollado por IFPUG (conocido como CPM, versión 4.1) y posee a su vez definiciones complementarias a las de IFPUG sobre los elementos del proceso de cuenta y sobre los pasos del proceso de cuenta, brindada por conocidos expertos en el Análisis de Punto Función.

En el mundo del Punto Función los sistemas son divididos en componentes. Los primeros tres componentes son: Entradas (EI = External Input), Salidas (EO = External output)y Consultas (EQ = External Queries). Cada una de estas transacciones actúa sobre ficheros agregando, modificando, eliminando, obteniendo o procesando información. Los otros dos componentes son los ficheros o entidades del sistema, denominados Ficheros Lógicos Internos (ILF = Internal Logical File) y Ficheros Lógicos Externos (ELF = External Logical File).

El Análisis del Punto Función es un método estándar de medición de desarrollo de software desde el punto de vista del usuario. Su objetivo es medir el software basándose en la cuantificación de la funcionalidad brindada al usuario partiendo fundamentalmente de diseños lógicos. La cuenta de Punto Función para proyectos de desarrollo mide las funcionalidades que se le proporcionan al usuario conjuntamente con la primera instalación del software producido cuando el proyecto es terminado.

Si bien el mismo es criticado por no ser aplicable a todos los sistemas, el tamaño de la gran mayoría de los sistemas del mercado puede ser estimado utilizando esta técnica. En general, la mayoría de las soluciones alternativas que se proponen se basan en esta técnica de Albrecht, modificando ligeramente algunos aspectos o incorporándole algún nuevo elemento a estos cinco identificados. Sin embargo, no hay unanimidad de criterio respecto a estas variaciones, no todas son conocidas tan ampliamente como la técnica de los puntos función en la que se basan, y no existen estándares internacionales como los que provee el IFPUG o requieren un grado mucho mayor de información que la disponible en las primeras fases del ciclo de vida. Considerando además, que las estimaciones siempre conllevan un margen de error razonable, se propone la metodología de puntos función para estimar el tamaño de los proyectos que se presentan.

Frontera de la aplicación

La frontera de la aplicación indica el límite entre el software que está siendo medido. Frecuentemente los sistemas computacionales interactúan con otros sistemas computacionales y con usuarios o dispositivos. Es crucial tener bien identificada la frontera de la aplicación a ser medida para luego recién enfocarse en la descomposición de sus componentes. La frontera de la aplicación debe ser dibujada en función del punto de vista del usuario.

En resumen, la frontera marca el borde entre el proyecto o aplicación a ser medido y las aplicaciones externas o el dominio del usuario. Una vez establecida la frontera, los componentes pueden ser clasificados y contados. Define qué es lo externo a la aplicación. Actúa como una membrana a través de la cual los datos procesados por transacciones (EI, EO, EQ) cruzan la frontera hacia adentro y/o hacia fuera de la aplicación. Determina los datos lógicos mantenidos por la aplicación (ILF) así como por consiguiente facilita la identificación de cuáles datos lógicos son referenciados por la aplicación pero mantenidos por otro sistema (ELF).

Las siguientes reglas deben aplicarse para el establecimiento de la frontera:

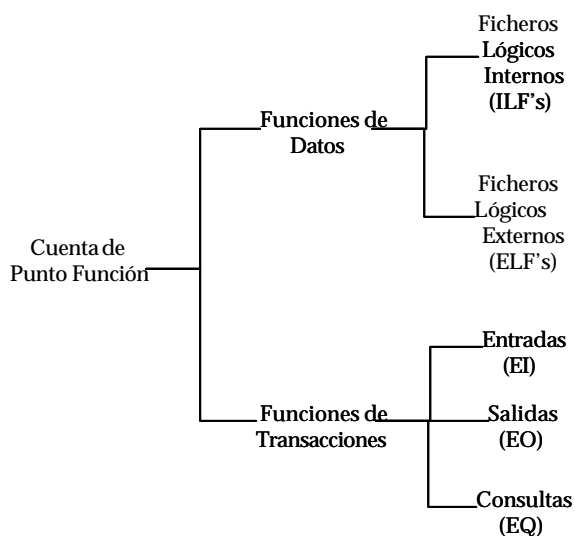
- La frontera es determinada basándose en el punto de vista del usuario. Se focaliza en qué el usuario puede entender o describir.
- La frontera entre aplicaciones relacionadas está basada en áreas funcionales separadas siempre desde el punto de vista del usuario y no en consideraciones técnicas.
- La frontera establecida por una aplicación existente que está siendo modificada no se influye por el alcance de la cuenta.

La localización de la frontera de la aplicación entre el software que está siendo medido y otras aplicaciones de software suele ser algo subjetivo y difícil de determinar dónde una aplicación termina y la otra comienza. Es importante que la frontera de la aplicación sea dibujada con cuidado dado que todo aquel dato que atraviesa la frontera puede potencialmente ser incluido en el alcance de la cuenta.

Cuenta de Puntos Función sin ajustar

La cuenta de Punto Función sin ajustar refleja las específicas funcionalidades contabilizadas que se le proporcionan al usuario por el proyecto o la Aplicación. Las funcionalidades de usuario específicas son evaluadas en términos del *Qué* es lo que proporcionan y no en el *Cómo* es proporcionado.

La cuenta sin ajustar tiene dos tipos de funciones: de datos y de transacciones.



Funciones de Datos

Las funciones de datos representan las funcionalidades brindadas al usuario para reunir los requerimientos de datos internos y externos a la frontera de la aplicación. Las funciones de datos son siempre ficheros lógicos internos (Internal Logical File o ILF) o ficheros lógicos externos (External Logical File o ELF).

- ❑ Un ILF es un grupo de datos relacionados lógicamente o información de control reconocida por el usuario y mantenida dentro de la frontera de la aplicación. El objetivo fundamental de un ILF es manejar los datos mantenidos a través de uno o más procesos elementales (o acciones) de la aplicación que está siendo contada.
- ❑ Un ELF es un grupo de datos relacionados lógicamente o información de control reconocida por el usuario y referenciada pero mantenida dentro de la frontera de otra aplicación. El objetivo principal de un ELF es manejar los datos referenciados mediante uno o más procesos elementales (o acciones) dentro de la frontera de la aplicación contada.

Cundo se hace la referencia a un archivo lógico interno (ILF) o interfase de archivo externa (ELF), el término *archivo* no tiene el significado tradicional de procesamiento de datos sino que se refiere a un grupo de datos lógicamente relacionado sin tener en cuenta la implementación física de dicho grupo. Se corresponde con las entidades que se identifican que se gestionan mediante el sistema.

Funciones Transaccionales

Las funciones transaccionales representan las funcionalidades que el sistema le brinda al usuario para procesar datos. Las funciones transaccionales son siempre entradas (EI - external inputs), salidas (EO - external output) o consultas (o EQ - external queries).

- ❑ Un EI es un proceso elemental o una acción que procesa datos o información de control que viene desde afuera de la frontera de la aplicación hacia adentro. Los datos pueden venir desde otra aplicación o desde una pantalla de ingreso de datos. El objetivo fundamental es mantener uno o más archivos lógicos internos (o ILF de Internal Logical File) y/o alterar el comportamiento del sistema.
- ❑ Un EO es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo fundamental es presentar información al usuario a través de procesamiento lógico de los datos o de la información de control obtenida. El procesamiento lógico debe contener al menos un fórmula o cálculo matemático o crear datos derivados de los obtenidos. Un EO podría mantener uno o más ILF y/o alterar el comportamiento del sistema.
- ❑ Un EQ es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo principal de un EQ es presentar información al usuario a través de la obtención meramente del dato o de la información de control. A diferencia del EO, el procesamiento lógico no debe contener ninguna fórmula o cálculo matemático, ni tampoco debe crear datos derivados de los obtenidos. Por otra parte ningún ILF es mantenido mientras se procesa la acción de un EQ ni tampoco el comportamiento del sistema se ve alterado.

Mecanismo

Para cada uno de los 5 tipos de elementos, identificar aquellos que surgen de los requerimientos o funcionalidades que proveerá el sistema. Es posible para ello basarse en las reglas que surgen en los capítulos siguientes. Luego analizar la complejidad de cada uno, que puede ser Alta, Media o Baja. Existen reglas detalladas para definir la complejidad de cada uno de los elementos según su tipo y que se explican a continuación. Con estos datos, y utilizando los pesos correspondientes a cada uno de los elementos identificados es posible calcular el total de Puntos Función.

Ficheros Lógicos Internos (ILF)

Un ILF es un grupo de datos relacionados lógicamente o información de control reconocida por el usuario y mantenida dentro de la frontera de la aplicación. El objetivo fundamental de un ILF es manejar los datos mantenidos a través de uno o más procesos elementales (o acciones) de la aplicación que está siendo contada.

Reglas para identificarlos:

- R1:** El grupo de datos o información de control es un grupo de datos lógico identificable por el usuario que cubre de manera completa requisitos específicos de este.
- R2:** El grupo de datos es mantenido dentro de los límites de la aplicación.
- R3:** El grupo de datos es mantenido o modificado por medio de un proceso elemental de la aplicación.
- R4:** El grupo de datos identificado no se ha contado como ELF de la aplicación.

Ficheros Lógicos Externos (ELF)

Un ELF es un grupo de datos relacionados lógicamente o información de control reconocida por el usuario y referenciada pero mantenida dentro de la frontera de otra aplicación. El objetivo principal de un ELF es manejar los datos referenciados mediante uno o más procesos elementales (o acciones) dentro de la frontera de la aplicación contada.

La principal diferencia entre un archivo lógico interno ILF y una interfase de archivo externa ELF es que el ILF es mantenido dentro de la frontera de la aplicación mientras que el ELF es mantenido por otra aplicación. El ELF es referenciado por la aplicación que se está contando pero no mantenido por ella.

Reglas:

- R1:** El grupo de datos o información de control es un grupo de datos lógico identificable por el usuario que cubre de manera completa requisitos específicos de este.
- R2:** El grupo de datos es referenciado y es externo a la aplicación que está siendo contada.
- R3:** El grupo de datos no es mantenido por la aplicación que está siendo contada.
- R4:** El grupo de datos se ha contado como ILF en al menos otra aplicación.
- R5:** El grupo de datos identificado no ha sido contado como un ILF para la aplicación.

No relacionar un ILF o ELF con un archivo físico, sino como una entidad. Si por ejemplo se identifica la entidad factura, la misma probablemente se registre en dos archivos o tablas: cabecal y líneas, sin embargo, estos dos archivos corresponden a una misma entidad lógica por lo que se cuenta únicamente un ILF o ELF.

Determinación de complejidad de los Ficheros Lógicos

La complejidad de los ficheros lógicos, tanto internos como externos se basa en la cuenta de DETs y RETs:

DET: Data Element Type. Contar un DET por cada uno de los campos reconocibles por el usuario que componen el fichero.

Se aplican además las siguientes reglas:

- Los campos que existen como claves foráneas para referenciar otro ILF según los requerimientos, también se consideran como DET.
- UN campo lógico único, almacenado en varios campos físicos, se cuenta como un único DET.
- Los campos repetitivos se cuentan como un único DET cada uno.
- Los campos que aparecen más de una vez por técnicas de implementación se cuentan como un único DET.
- Los campos que aparecen debido a técnicas de implementación, que no son reconocibles por el usuario, no se cuentan como DET.

RET: Record Element Type. Contar un RET por cada subgrupo de datos elementales reconocibles por el usuario.

Existen dos tipos de subgrupos:

- *Opcional:* son aquellos que el usuario tiene la opción de usar uno o ningún subgrupo durante un proceso elemental que agrega o crea una instancia de datos.
- *Obligatorio:* son subgrupos donde el usuario debe usar al menos uno.

A la hora de contar RET's, una de las siguientes reglas debe aplicar:

- Cuente como un RET para cada subgrupo opcional u obligatorio de un ILF/ELF
- De no haber subgrupos, cuente el ILF o ELF como un RET.

Ejemplo: Si para una entidad Ventas, la misma puede ser al contado o con tarjeta de crédito y para el caso de pago con tarjeta se agregan a la venta los datos de la tarjeta, este segundo grupo opcional se cuenta como un RET extra para el ILF Ventas.

En caso de identificar grupos repetitivos en una entidad, sume 1 RET por cada grupo repetitivo. Por ejemplo, para una entidad factura donde se identifican los datos de cada línea que se repiten, súmele un RET a la entidad factura.

Una vez identificados los DETs y RETs de un fichero lógico, utilice la siguiente tabla para determinar su complejidad.

	1 a 19 DETs	20 a 50 DETs	51 o más DETs
1 RET	Baja	Baja	Media
2 a 5 RETs	Baja	Media	Alta
6 o más RETs	Media	Alta	Alta

Entradas (EI)

Un EI es un proceso elemental o una acción que procesa datos o información de control que vienen desde afuera de la frontera de la aplicación hacia adentro. Los datos pueden venir desde otra aplicación o desde una pantalla de ingreso de datos. El objetivo fundamental es mantener uno o más archivos lógicos internos (o ILF de Internal Logical File) y/o alterar el comportamiento del sistema.

Se aplican las siguientes reglas:

- R1:** Los datos se reciben desde fuera de los límites de la aplicación.
- R2:** Los datos mantienen un ILF a través de un proceso elemental de la aplicación.
- R3:** El proceso es la unidad más pequeña de actividad que es significativa para el negocio del usuario final.
- R4:** El proceso es autocontenido y deja la aplicación que está siendo contada en un estado consistente.
- R5:** El proceso identificado debe verificar alguna de estas reglas:
 - Su lógica de proceso es única respecto de otras entradas externas de la aplicación.
 - Los elementos de datos identificados son distintos a los de las otras EI de la aplicación.
 - Los ficheros lógicos referenciados son diferentes

Determinación de la complejidad de las Entradas

La complejidad de las entradas se basa en la cuenta de DETs y FTRs:

DET: Data Element Type. Contar un DET por cada uno de los campos que cruzan las fronteras de la aplicación

Se aplican además las siguientes reglas:

- Contar 1 DET por cada campo reconocible por el usuario, no repetido que cruza los límites de la aplicación.
- No contar los campos que son recuperados o derivados por el sistema y almacenados en un ILF si estos no cruzan la frontera de la aplicación.
- Contar los siguientes elementos como un DET extra cada uno:
 - Definición del comienzo de la ejecución del proceso que procesa la entrada (botón de Aceptar o Enter). Si hay varias formas, contar una sola.
 - Mensajes de error que pueden desplegarse durante la ejecución del proceso (un DET para todos).

FTR: File Type Referenced. Contar un FTR por cada fichero (interno o externo) leído/mantenido por el proceso de la entrada.

Una vez identificados los DETs y FTRs de la salida, utilice la siguiente tabla para determinar su complejidad.

	1 a 4 DETs	5 a 15 DETs	16 o más DETs
1 FTR	Baja	Baja	Media
2 FTRs	Baja	Media	Alta
3 o más FTRs	Media	Alta	Alta

Salidas (EO)

Un EO es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo fundamental es presentar información al usuario a través del procesamiento lógico de los datos o de la información de control obtenida. El procesamiento lógico debe contener al menos una fórmula o cálculo matemático o crear datos derivados de los obtenidos. Un EO podría mantener uno o más ILF y/o alterar el comportamiento del sistema.

Se aplican las siguientes reglas:

- R1:** El proceso envía datos información de control.
- R2:** Los datos o información de control se envían a través de un proceso elemental de la aplicación.
- R3:** El proceso es la unidad más pequeña de actividad que es significativa para el negocio del usuario final.
- R4:** El proceso es autocontenido y deja a la aplicación en un estado consistente.
- R5:** El proceso identificado debe verificar alguna de estas reglas:
 - Su lógica de proceso es única respecto de otras salidas externas de la aplicación.
 - Los elementos de datos identificados son distintos a los de otras EO's de la aplicación.
 - Los ficheros lógicos referenciados son distintos.
- R6:** Debe cumplirse al menos una de las siguientes condiciones:
 - El proceso elemental contiene al menos una fórmula matemática o cálculo.
 - El proceso crea datos derivados
 - El proceso que genera la salida mantiene algún ILF
 - El proceso que genera la salida altera el comportamiento del sistema.
- R7:** La transferencia de datos a otras aplicaciones se cuenta como salidas
- R8:** Los informes escritos y online se cuentan como salidas independientes.
- R9:** Los gráficos se cuentan como una salida cada uno.

Consultas (EQ)

El EQ es un proceso elemental o una acción que envía datos o información de control hacia fuera de la frontera de la aplicación. El objetivo principal de un EQ es presentar información al usuario a través de la mera obtención del dato o de la información de control. A diferencia del EO, el procesamiento lógico no debe contener ninguna fórmula o cálculo matemático, ni tampoco debe crear datos derivados de los obtenidos.

Por otra parte ningún ILF es mantenido mientras se procesa la acción de un EQ ni tampoco el comportamiento del sistema se ve alterado.

Es una combinación de entrada/salida que se obtiene de una búsqueda de datos, no actualiza ficheros lógicos y no contiene datos derivados (aquellos que requieren un proceso distinto a búsqueda, edición o clasificación).

Se aplican las siguientes reglas:

- R1:** Una petición entra dentro del límite de la aplicación.
- R2:** Un resultado sale del límite de la aplicación
- R3:** Hay recuperación de datos
- R4:** Los datos recuperados no contienen datos derivados.
- R5:** El proceso lógico no contiene fórmulas matemáticas o cálculos
- R6:** El proceso que genera la consulta no mantiene ningún ILF ni altera el comportamiento del sistema
- R7:** La petición de entrada y el resultado de salida juntos, hacen del proceso la unidad de actividad más pequeña que es significativa para el negocio del usuario final.
- R8:** El proceso es autocontenido y deja a la aplicación que está siendo contada en un estado consistente.

R9: El proceso no actualiza ILF's.

R10: El proceso verifica alguna de estas dos reglas:

La lógica del proceso sobre la entrada y la salida es única respecto a otras consultas de la aplicación

Los elementos de datos que forman la entrada y la salida son distintos a los de las otras consultas de la aplicación.

Los ficheros lógicos referenciados son distintos.

R11: El proceso no actualiza ILF's.

R12: El proceso verifica alguna de estas dos reglas:

La lógica del proceso sobre la entrada y la salida es única respecto a otras consultas de la aplicación

Los elementos de datos que forman la entrada y la salida son distintos a los de las otras consultas de la aplicación.

Los ficheros lógicos referenciados son distintos.

Determinación de la complejidad de las Salidas y Consultas

La complejidad de las entradas se basa en la cuenta de DETs y FTRs:

DET: Data Element Type. Contar un DET por cada uno de los campos que cruzan las fronteras de la aplicación (tanto en la entrada como en la salida para el caso de las consultas)

Se aplican además las siguientes reglas:

Contar un DET por cada campo reconocible por el usuario, no repetido, que entra a la aplicación y que se requiere para especificar cuando, que o como deben ser recuperados los datos de la salida.

Contar un DET por cada campo reconocible por el usuario no repetido que sale de los límites de la aplicación.

Si un dato entra y sale de la aplicación, contarlos una sola vez.

Sumar un DET por todos los mensajes de error del proceso.

Sumar un DET por la iniciación del proceso, solo una vez aunque haya varias formas de hacerlo.

No contar los datos recuperados o derivados por el sistema si no cruzaron los límites de la aplicación.

FTR: File Type Referenced. Contar un FTR por cada fichero (interno o externo) leído por el proceso de la salida o consulta

Una vez identificados los DETs y FTRs de la salida, utilice la siguiente tabla para determinar su complejidad.

	1 a 5 DETs	6 a 19 DETs	20 o más DETs
1 FTR	Baja	Baja	Media
2 a 3 FTRs	Baja	Media	Alta
4 o más FTRs	Media	Alta	Alta

Total de Puntos Función

En base a la cantidad de elementos identificados de cada tipo y de cada complejidad, se llega a la cantidad total de puntos función sin ajustar.

Para ello, a cada uno de los elementos se le asigna un peso según su complejidad, de forma tal que multiplicando la cantidad de elementos por su peso correspondiente, y sumando todos estos resultados se obtiene el resultado final.

En la tabla que se adjunta a continuación se especifican los pesos de cada uno de los elementos identificados durante la cuenta.

Parámetro	Complejidad	Peso	Cantidad	Total = cantidad * peso
Ficheros Lógicos Internos	Alta	15		
	Media	10		
	Baja	7		
Ficheros Lógicos Externos	Alta	10		
	Media	7		
	Baja	5		
Entradas	Alta	6		
	Media	4		
	Baja	3		
Salidas	Alta	7		
	Media	5		
	Baja	4		
Consultas	Alta	6		
	Media	4		
	Baja	3		
			Total	Puntos Función = Suma de Totales

Anexo 1 – Identificación de Transacciones de Datos

El procesamiento lógico se define como un requerimiento específicamente solicitado por el usuario para completar un proceso elemental. Tales requerimientos pueden incluir:

1. Ejecución de validaciones
2. Ejecución de cálculos y formas matemáticas
3. Conversión de valores equivalentes
4. Filtrado y selección de datos usando un criterio específico para comparar conjuntos de datos
5. Análisis de condiciones para determinar cuál de las condiciones se debe aplicar
6. Actualización de uno o más ILFs
7. Referencia a uno o más ELF's
8. Obtención de datos o información de control
9. Generación de datos derivados mediante la transformación de datos o la creación de datos adicionales
10. Alteración del comportamiento del sistema
11. Preparación y presentación de la información hacia fuera de la frontera de la aplicación
12. Capacidad para aceptar datos o información de control que entre a la frontera de la aplicación
13. Reordenamiento o reorganización de datos

La siguiente tabla resume qué formas de procesamiento lógico deben ser ejecutadas por un EI, EO o un EQ. Para cada tipo de función transaccional, cierto tipo de procesamiento lógico debe ser ejecutado a los efectos de acompañar el objetivo principal de la función transaccional.

Formas de procesamiento lógico:	Tipo de Función Transaccional		
	EI	EO	EQ
Ejecución de validaciones	P	P	P
Ejecución de cálculo y formas matemáticas	P	O ²	N
Conversión de valores equivalentes	P	P	P
Filtrado y selección de datos usando un criterio específico para comparar conjuntos de datos	P	P	P
Análisis de condiciones para determinar cual de las condiciones se debe aplicar	P	P	P
Actualización de uno o más ILF's	O ²	O ²	N
Referencia a uno o más ELF's	P	P	O ¹
Obtención de datos o información de control	P	P	O ¹
Generación de datos derivados mediante la transformación de datos o la creación de datos adicionales	P	O ²	N
Alteración del comportamiento del sistema	O ²	O ²	N
Preparación y presentación de la información hacia fuera de la frontera de la aplicación	P	O ¹	O ¹
Capacidad para aceptar datos o información de control que entre a la frontera de la aplicación	O ¹	P	P
Reordenamiento o reorganización de datos	P	P	P

Leyendas:

O¹ = es obligatorio que la función transaccional ejecute la forma de procesamiento lógico

O² = es obligatorio que la función transaccional ejecute la forma de procesamiento lógico

P = la función transaccional puede ejecutar la forma de procesamiento lógico pero no es obligatorio

N = la función transaccional NO puede ejecutar la forma de procesamiento lógico

Anexo 2 - Definición de términos utilizados

Información de Control: Son datos que influyen un proceso elemental o una acción de la aplicación que está siendo contada. Especifica *qué*, *cuándo* y *cómo* el dato será procesado.

Datos Reconocidos por el Usuario: El término *reconocido por el usuario* se refiere a requerimientos definidos por procesos y/o grupos de datos que fueron acordados y entendidos entre el usuario y el desarrollador de software.

Mantenidos: El término mantenidos es la capacidad de modificar datos mediante un proceso elemental como ser: agregar, cambiar, eliminar, publicar, crear y asignar.

Proceso Elemental: Un proceso elemental es la menor unidad de actividad que es significativa para el usuario.

Un ejemplo de un proceso elemental sería: El usuario requiere poder agregar una nueva persona a la aplicación. La definición del usuario de una persona incluye información dependiente de la persona como ser domicilio y/o teléfono. Desde el punto de vista del usuario, la menor unidad de actividad es agregar una nueva persona. Agregar una pieza de información como ser el domicilio o el teléfono no es una actividad que calificaría como un proceso elemental. Un proceso elemental debe ser auto controlado y dejar la información del negocio de la aplicación en un estado consistente.