

Análisis de Puntos Función en la elicitación de requerimientos

Mabel Bertolami¹, Alejandro Oliveros²

¹ Departamento de Informática, Departamento de Matemática, Facultad de Ingeniería,
UNPSJB
mbertolami@gmx.net

² Departamento de Computación, Facultad de Ingeniería, UBA - Magíster de Ingeniería de
Software, Facultad de Informática, UNLP
oliveros@fibertel.com.ar

Resumen. La aplicación del Análisis de Puntos Función a diferentes artefactos producidos en el proceso de desarrollo de software es una práctica establecida en la ingeniería de software, especialmente a los documentos de diseño. Se presenta una investigación desarrollada para medir la funcionalidad de un sistema de información en la etapa de Elicitación de Requerimientos, específicamente de los escenarios producidos en esa etapa. Se desarrolló un enfoque ad-hoc que permite aplicar el Análisis de Puntos Función (FPA) a Escenarios construidos a partir del Léxico Extendido del Lenguaje (LEL). Se analizaron alternativas aplicables de cálculo de los Puntos Función, se seleccionó un método de FPA y se establecieron un conjunto de reglas y procedimientos que soportan el proceso de medición. Se aplicaron en varios casos disponibles produciendo resultados alentadores respecto a la pertinencia de las métricas obtenidas. Se presentan las conclusiones y se indican futuras líneas de investigación.

1 Introducción

El objetivo del desarrollo de sistemas de software es producir productos de calidad adecuada, mediante un proceso eficiente y con un costo razonable. Disponer de parámetros adecuados para el seguimiento del proceso de desarrollo de software, requiere obtener mediciones de los productos resultantes, de los procesos aplicados y de los recursos utilizados.

Una medida fundamental es la de tamaño del artefacto producido. Tradicionalmente se utilizó la cantidad de líneas de código. Esta métrica, de amplia difusión en la disciplina, requiere un conjunto de recaudos para ser aplicada correctamente [26] y ha sido cuestionada severamente como métrica adecuada del tamaño [10]. Desde nuestro punto de interés es una limitación severa que sólo sea aplicable al código.

Un enfoque alternativo a la medición de líneas de código es medir la funcionalidad del producto mediante el Análisis de Puntos Función (FPA). Los métodos de FPA se orientan a medir la funcionalidad del sistema software independientemente de la tecnología que se utilice para implementarlo [11]. Tienen la ventaja de poder aplicarse a los productos obtenidos a lo largo de todo el ciclo de desarrollo. Por lo tanto se pue-

den refinar a medida que se avanza en el proceso, aumentando la precisión de las mediciones.

La bibliografía de FPA propone la aplicación de estos métodos a partir de la especificación de requerimientos [30] y se menciona la importancia de disponer de contabilizaciones de Puntos Función (FP) en las etapas iniciales del desarrollo [6] así como las ventajas de medir FP en los requerimientos y disponer de un marco de referencia adicional para verificar la completitud de los requerimientos funcionales [7]. El objetivo de este trabajo es medir los FP antes de la obtención de la especificación de requerimientos, esto es a partir de L&E (LEL y Escenarios).

El enfoque de L&E soporta la realización de la elicitación de requerimientos y se puede utilizar en el ciclo completo de desarrollo de software [19]. Tiene la ventaja de utilizar el lenguaje natural para las descripciones, lo que favorece la comunicación y validación con el usuario [14].

Los escenarios permiten entender la aplicación y su funcionalidad: cada escenario describe una situación específica de la aplicación centrando la atención en su comportamiento [20]. Esto significa que L&E se puede utilizar como documentación base para el cálculo de FP.

La ventaja decisiva del enfoque desarrollado consiste en la disponibilidad de una estimación del tamaño del producto a construir en una etapa muy temprana del proyecto de desarrollo. Esto permite anticipar una amplia gama de decisiones –decidir entre desarrollar o comprar un producto de software y especialmente cuando se estudia el reemplazo de un sistema en funcionamiento- con las consiguientes ventajas para la gestión del proyecto.

El resto de este paper está organizado de la siguiente manera: en el punto 2 se mencionan las propuestas relacionadas, en el 3 se sintetizan los conceptos de L&E, en el 4 se resumen los conceptos clave de los métodos de Análisis de Puntos Función. En los puntos 5 y 6 se presentan los conceptos del desarrollo realizado y en el 7 se presenta el proceso de cálculo de Punto Función de Escenarios. El punto 8 presenta los casos analizados y el 9 propone las conclusiones de la investigación y enuncia próximos pasos.

2 Propuestas relacionadas

Existen varias propuestas para aplicar el modelo de FP al software orientado a objetos [28] basadas en el modelo de *Casos de uso*. La aplicación de FPA a *Casos de uso* resulta muy directa: FPA es un método para medir software desde una perspectiva de los requerimientos y *Casos de uso* es un método para desarrollar requerimientos. Ambos tratan de ser independientes de la tecnología usada para implementar la solución de software [22] [32].

Fetcke et al. en su trabajo “Mapping the OO-Jacobson Approach into Function Point Analysis” proponen un mapeo del método OOSE basado en *Casos de uso* de Jacobson con el modelo abstracto de FP [11].

Se observa una proximidad entre el enfoque mencionado en los párrafos previos y el propuesto en este trabajo, en tanto los *Casos de uso* y los escenarios presentan características similares y son una representación del comportamiento. La diferencia

sustancial del enfoque propuesto con el FPA aplicado a *Casos de Uso*, es que los *Casos de Uso* y los escenarios acá utilizados se encuentran en dos etapas distintas del proceso de la Ingeniería de Requerimientos. En efecto, considerando el modelo de elicitación-especificación-validación [21], los *Casos de Uso* son una herramienta de especificación de requerimientos, en tanto que los escenarios propuestos en el enfoque de L&E son una herramienta de elicitación de requerimientos.

3 LEL y Escenarios

EL *Léxico Extendido del Lenguaje* (LEL) tiene como finalidad la comprensión del vocabulario de la aplicación, sin considerar en esta etapa la comprensión del problema. El LEL es una representación hipertextual de los símbolos del lenguaje del cliente en el contexto de la aplicación. Cada símbolo podrá ser una palabra o frase, especialmente las más repetidas por el cliente o de importancia relevante para el sistema [15]. Cada símbolo contiene: *Nombre*, que identifica el símbolo; *Noción*, que denota el significado del símbolo e *Impacto*, que establece cómo repercute en el sistema. Con un conjunto de heurísticas se obtienen los escenarios a partir del LEL definitivo.

Los *escenarios* representan descripciones parciales del comportamiento del sistema en un momento específico de la aplicación, se representan en lenguaje natural, están vinculados naturalmente al LEL, evolucionan durante el proceso de desarrollo del software y se representan de manera estructurada [19]. Un escenario se compone de: *Nombre*, es el título del escenario; *Objetivo*, es la finalidad a alcanzar en el macrosistema; *Contexto*, describe la ubicación geográfica/temporal y/o estado inicial del escenario; *Recursos*, medios de soporte, dispositivos u otros elementos pasivos necesarios para el escenario; *Actores*, personas o estructuras organizacionales que cumplen un rol en el escenario; *Episodios*, una serie de acciones que detallan al escenario y describen su comportamiento; *Excepciones*, usualmente reflejan la falta o mal funcionamiento de un recurso necesario, una excepción impide alcanzar el objetivo de un escenario, el tratamiento de la excepción puede ser expresado a través de otro escenario; *Restricciones*, un requerimiento de alcance o calidad referido a una entidad dada, es un atributo de los recursos, episodios básicos o subcomponentes del contexto.

LEL y Escenarios constituyen el producto obtenido como resultado del proceso de elicitación.

4 Métodos de Análisis de Puntos Función

Los FP miden el tamaño del software cuantificando la funcionalidad provista al usuario basándose solamente en el diseño lógico y las especificaciones funcionales [16].

La medición puede entenderse como una abstracción que captura ciertos atributos de los objetos de interés; la teoría de la medición visualiza esta abstracción como un mapeo que asigna objetos numéricos a objetos empíricos. En la medición de *tamaño funcional*, esos objetos empíricos son las aplicaciones de software [12]. Esos métodos definen una abstracción del software que representa los ítems considerados relevantes

para el *tamaño funcional*. Luego esos items se asocian con números. Los métodos son el de Albrecht y el MarkII.

El primer método basado en métricas de funcionalidad fue propuesto por Albrecht¹. La propuesta consistía en el desarrollo de un índice de la funcionalidad de un sistema como una medida asociada a su tamaño [30]. El proceso de medición se realiza en el contexto de un modelo abstracto del sistema que se compone de *transacciones* y *archivos* identificables a partir de los documentos de requerimientos, técnicas de diseño estructurado u otros modelos.

El método MarkII fue desarrollado por Symons a fines de la década de 1980 y tiene la misma finalidad y estructura que el de Albrecht. Éste es el método seleccionado, por lo que nos detendremos en él. Este método puede usarse para medir el *tamaño funcional* de cualquier aplicación de software que se pueda describir en términos de *transacciones lógicas*. Lo primero es determinar el *límite del sistema*: la aplicación o parte de la aplicación incluida dentro del *límite* debe ser un cuerpo coherente de funcionalidad que comprende una o más *transacciones lógicas* [25].

Una *transacción lógica* es una combinación de tres componentes:

- Un elemento de entrada a través del *límite*.
- Un elemento de procesamiento de los datos dentro del *límite*.
- Un elemento de salida a través del *límite*.

El *tamaño funcional* de una aplicación es la suma de los tamaños de cada *transacción lógica*.

- **Componente de procesamiento en las transacciones lógicas:** su tamaño es proporcional a la cantidad de entidades de datos *referenciadas* en cada *transacción*, entendiéndose como *referencia* a los procesos de creación, lectura, eliminación, actualización, listado [25]. Una *entidad* se define como: alguna cosa en el mundo real (objeto, transacción, período de tiempo, tangible o intangible) acerca de la cual se requiere mantener información [30].
- **Componentes de entrada y salida en las transacciones lógicas:** el tamaño relativo de estos componentes en cada *transacción lógica* es proporcional a la cantidad de *DET* en cada componente respectivo. Un *DET* es un ítem único de información que es indivisible para el propósito de la *transacción lógica* y es parte de un flujo de datos en la *entrada* y *salida* a través del *límite* [25].

El proceso de evaluación del *Índice de Puntos Función (FPI)*, requiere de una serie de pasos que se describen en [30] y concluye con la aplicación de la siguiente fórmula:

$$FPI = W_I * \sum N_I + W_E * \sum N_E + W_O * \sum N_O$$

W_I , W_E y W_O : pesos promedio en la industria (*industry average weights*) para los componentes de entrada (I), entidades referenciadas (E) y de salida (O), en general se acepta: $W_I = 0.58$, $W_E = 1.66$ y $W_O = 0.26$. Además: N_I = cantidad de *DET* de entrada, N_E = cantidad de entidades referenciadas y N_O = cantidad de *DET* de salida.

Opcionalmente si es necesario calcular los FP considerando los requerimientos técnicos, se calcula el *Ajuste de complejidad técnica* $TCA = (TDI * C) + 0.65$. El va-

¹ Algunos de los trabajos principales fueron [1], [2], pero no han podido ser consultados. En cada oportunidad que se haga mención a Albrecht, será a través de referencias indirectas.

lor actual aceptado en la industria de C es 0.005 y TDI es el total de los puntajes para cada una de las 19 características, llamado *grado de influencia total*.

El *Índice de Puntos Función Ajustado* se expresa como: $AFPI = FPI * TCA$

5 Aplicación del Análisis de Puntos Función a L&E y Escenarios

Beneficios de la aplicación del enfoque. El cálculo de FP en la etapa de elicitación de requerimientos FP de un sistema software posee los beneficios de disponer de un cálculo de FP de un sistema software [25] y los resultantes de la medición temprana. En este último grupo, se encuentran:

- Estimación de costos, esfuerzo, duración. Permite tomar decisiones sobre un proyecto sin una especificación completa evitando invertir tiempo y esfuerzo en un proyecto que quizás no realice.
- Tomar la decisión temprana de desarrollar o comprar un producto de software. Evitando desarrollar una especificación sólo para evaluar las opciones. Esto es especialmente ventajoso al evaluar reemplazar un sistema, por el conocimiento que se tiene de su funcionalidad.
- Estimar el esfuerzo requerido por tareas tempranas.

Alternativas para el cálculo de FP desde L&E. El FPA considera al sistema como un conjunto de *transacciones lógicas* que interactúan con *entidades* y/o archivos lógicos. Para obtener la funcionalidad aplica un conjunto de reglas a las *entidades* y *transacciones lógicas* obtenidas a partir del diseño del sistema [25].

Para establecer el camino a seguir se analizaron varias alternativas para obtener los FP desde L&E. En la Figura 1 se representaron las alternativas consideradas que se identificaron como enfoque clásico y Alternativa 1, 2 y 3.

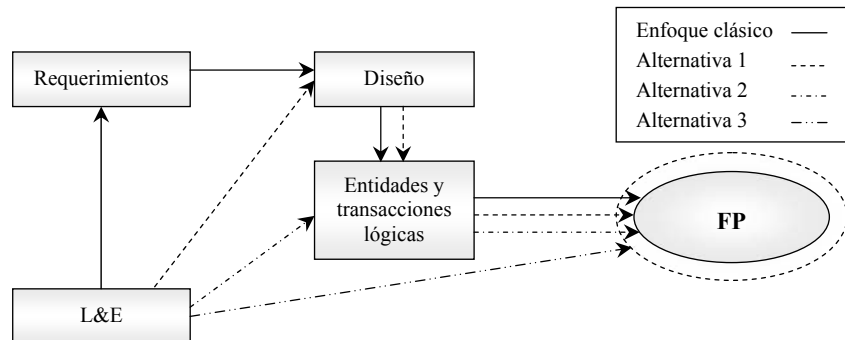


Figura 1. Posibles enfoques de aplicación de FPA a L&E

El *enfoque clásico* no resulta un aporte novedoso relacionado con el objetivo de este trabajo, pues una vez obtenido el diseño se deben seguir los pasos propuestos por los métodos de FPA.

La *Alternativa 1* equivale a construir un sistema sin considerar los requerimientos y la *Alternativa 2* equivale a omitir el diseño. Sin embargo, la determinación de los re-

querimientos es un subproceso dentro del proceso. Los diferentes modelos de desarrollo de software reconocen la existencia de los componentes especificación, diseño e implementación [21]. En este contexto ambas alternativas parecen poco posibles.

En nuestra propuesta se eligió la *Alternativa 3*. Vale decir que se calcularon los FP basándose exclusivamente en el L&E. Este enfoque para medir la funcionalidad requirió establecer un modelo de cálculo de FP, analizar las relaciones entre los elementos del L&E y los conceptos del FPA, enunciar las reglas y fórmulas para calcular los FP.

Debe advertirse que tomando como base del análisis a L&E se está considerando todo el sistema, no exclusivamente las funciones asignadas al software sino también se miden funciones que no ejecutará el software y por lo tanto esto llevará a obtener una cantidad de FP presumiblemente mayor que la que se obtendría a partir del enfoque tradicional. Es que justamente en la etapa de elicitación aún no se han definido las funciones que corresponden al software del sistema total. Un escenario comienza en el macrosistema y evoluciona a medida que el “artefacto” de software se va construyendo [14]. La cantidad de FP obtenida inicialmente se puede refinar a medida que evolucionan los escenarios. Al examinar algunos enfoques para medir la funcionalidad, se ha indicado [10] que éstos miden la funcionalidad desde los documentos de especificación pero pueden aplicarse a productos posteriores en el ciclo de vida para refinar las estimaciones de tamaño y en consecuencia las estimaciones de costo o productividad.

Modelo de medición del tamaño funcional desde L&E. Partiendo del concepto que un método de medición del *tamaño funcional* define una medida que asigna un número de FP a las aplicaciones de software y considerando que el objeto de la medición es L&E, se definió un enfoque que permitiera identificar aquellas partes que representan funciones y asociarlas con números para obtener el *tamaño funcional* de la aplicación. Este proceso requiere dos pasos:

1. Identificar los componentes de L&E que contribuyen a la funcionalidad.
2. Asociar con números a cada uno de los componentes identificados en L&E.

El método de FPA elegido y el Ajuste de complejidad técnica. Una cuestión importante consistió en determinar cuál método de los dos principales, Albrecht o MKII, presentaba un marco conceptual más adecuado para facilitar la identificación de los componentes que aportan funcionalidad.

Se estableció como criterio dar prioridad al enfoque que facilite más el trabajo con L&E. Del análisis comparativo y las consideraciones de algunas publicaciones (por ej. [13]) se concluyó que MKII presentaba un marco conceptual más sencillo para abordar el cálculo de FP a partir de L&E. Parece “más natural” descomponer el modelo L&E en “partes” que se asemejan a las *entidades* y *transacciones lógicas* de MKII y definir un conjunto de reglas para su tratamiento y cálculo de la funcionalidad. La propuesta consistió desarrollar un enfoque similar a MKII, basado exclusivamente en la información disponible en L&E y que brinde una serie de reglas para identificar, clasificar y cuantificar los tipos de componentes lógicos del L&E para así obtener una medida del *tamaño funcional* del sistema.

El **Ajuste de complejidad técnica** también significó un tema crucial. El *tamaño funcional* debía basarse exclusivamente en los requerimientos funcionales del usuario. Los requerimientos técnicos y de calidad no deberían combinarse con los requeri-

mientos funcionales del usuario para producir un tamaño compuesto. El resultado es muy difícil de interpretar y los factores inevitablemente dependen de suposiciones acerca de la tecnología a usar [31].

Otra consideración importante que se hizo tiene relación particular con el enfoque propuesto: L&E contiene información relativa al dominio de aplicación, no es posible identificar allí características técnicas vinculadas a la implementación. En [5] se propone una extensión al L&E para favorecer el proceso de elicitación de requerimientos no funcionales. La utilización de ese enfoque permitiría contar con alguna información acerca de las características generales del sistema, pero habrá otras características que será muy difícil o aún imposible ponderarlas a partir de L&E.

Las razones enunciadas sirvieron como soporte para la decisión de basar la medición exclusivamente en las funciones del sistema obtenidas desde L&E.

6 Medición de Puntos Función de L&E

Reducción a episodios. Teniendo en cuenta que los escenarios representan las funciones requeridas por el usuario, se utilizó un enfoque similar al proceso de “descomposición funcional” en que las funciones del sistema son descompuestas o refinadas reiteradamente hasta obtener –en el menor nivel- combinaciones entrada-proceso-salida únicas e independientes [30].

La propuesta consistió en descomponer cada escenario en un conjunto de *episodios* que representen acciones únicas, determinar el conjunto unión de todos los *episodios* asegurando que no haya *episodios* repetidos, calcular los FP de cada uno de ellos y obtener el total de FP del sistema como sumatoria de los FP de cada escenario.

En rasgos generales los pasos son:

- Sea $E = \{E_1, \dots, E_M\}$ una colección de M escenarios. Cada escenario tiene un conjunto de *episodios*, de los que sólo se consideran aquellos que no son escenarios.
- Se eliminan las repeticiones de *episodios* y se obtiene el conjunto $P = \{P_1, \dots, P_N\}$ de *episodios*.
- Se indica con FP_i los FP del *episodio* P_i .
- Se define la cantidad total de FP del sistema como la suma de los FP_i .

En el punto c) se calculan los FP de cada episodio, los FP se obtienen sin considerar el esfuerzo y se calculan mediante la siguiente fórmula: $FP = N_I + N_R + N_O$; donde N_I es la cantidad de *DET* de entrada, N_R la cantidad de *recursos* referenciados y N_O la cantidad de *DET* de salida.

El *Índice de Puntos Función* se calcula mediante:

$$FP = \sum_{i=1}^N N_{I_i} + \sum_{i=1}^N N_{R_i} + \sum_{i=1}^N N_{O_i}$$

siendo N el total de episodios.

Tamaño funcional y episodios. En las primeras etapas de análisis del problema se confecciona una lista de *episodios candidatos*. Un *episodio candidato* es aquél que puede estar incluido en la lista de *episodios* elaborada inicialmente, pero aún debe establecerse su inclusión en la lista definitiva.

El cálculo de la FP de la aplicación se hizo calculando los FP de los *episodios candidatos*. Para calcular los FP de los *episodios candidatos*, se formularon un conjunto de reglas² que establecen los criterios que debe satisfacer un *episodio candidato*. A continuación se presentarán algunas reglas, la numeración que se utilizará es la original [4].

La **Regla 1** excluye de la *lista de episodios candidatos* a los que representan un escenario.

Clasificación de los episodios según su tipo y estructura. Los episodios son el centro de este análisis debido a que cada episodio representa una acción realizada por un actor, donde participan otros actores y se utilizan recursos [18].

Se buscó una analogía de conceptos entre *episodios* y *transacciones lógicas*. Una vez descartados aquellos *episodios* que son escenarios, repetidos o que no incluyen funcionalidad, los restantes *episodios* se clasifican según su tipo y estructura.

Los *episodios* se clasifican según su estructura en única acción o escenario.

Los *episodios* son de tipo *simple*, *condicional* u *opcional*. Independientemente de su tipo, un *episodio* puede ser expresado como una única acción o puede ser un escenario. El modelo de escenarios provee la descripción de comportamientos con diferentes órdenes temporales. Una secuencia de *episodios* implica un orden de precedencia, pero un orden no secuencial requiere el agrupamiento de dos o más *episodios*. Esto se usa para expresar un orden paralelo o secuencial indistinto [20].

En lo que sigue se analizarán algunos de los *episodios* mencionados considerando las definiciones del Modelo de escenarios propuesto en [20] y si corresponde se enunciará una regla. En cada caso el análisis es similar al desarrollado a continuación.

Episodios simples son aquéllos que son necesarios para completar el escenario.

La **Regla 3** establece que un episodio simple que representa una única acción debe ser considerado *episodio candidato*. Una *Única acción* representa una acción atómica disparada por algún *evento* del mundo externo o por una *consulta* para obtener información, que se realiza en forma completa e independiente de otras [25].

Restricciones. Las restricciones provienen del contexto del problema y del contexto del escenario. Es un atributo de los *recursos*, *episodios* básicos o subcomponentes del contexto [20]. Son aquellos aspectos que al presentarse impiden la acción. Las restricciones reflejan requerimientos no funcionales [15].

La **Regla 7** establece que no se considerarán pues no representan funcionalidad.

Desde el punto de vista del FPA, aunque MKII ofrece un factor para considerar la complejidad técnica y ciertos requerimientos de calidad, actualmente se recomienda no incluirlos [25].

Relación entre transacción lógica y episodio. En MKII una *transacción lógica* es: “una combinación de uno o más tipos de entrada, algún procesamiento y uno o más tipos de salida, correspondientes a un proceso único e independiente”. Es disparada por un *único evento del mundo externo significativo para el usuario* o es el resultado de una *consulta para retornar o extraer información* sin modificar los datos almacenados [30].

² Para la definición de las reglas se utiliza el modelo descrito en [29].

En un *episodio* para que un actor pueda realizar una acción sobre el sistema debe transmitir algún mensaje que provoque la realización del procesamiento correspondiente a esa acción. Durante el procesamiento la aplicación utiliza *recursos*. Para que la acción realizada por la aplicación tenga sentido, es necesario que produzca algún resultado como salida. El análisis de los diferentes tipos de *episodios* sugirió establecer como concepto que el *episodio simple que representa una única acción es equivalente al concepto de transacción lógica*.

Clasificación de episodios según el procesamiento requerido. Para clasificar los *episodios* según este criterio, se debe determinar qué necesidades de procesamiento satisface la realización del *episodio*, considerando que las interacciones de un actor con el sistema son provocadas por:

- La ocurrencia de algún suceso en el mundo externo que requiere la realización de algún procesamiento.
- La necesidad de recuperar información.

Teniendo en cuenta las definiciones de *evento* y *consulta* de MKII y la equivalencia entre *transacción lógica* y *episodio simple* y considerando que las únicas interacciones posibles entre un sistema y un actor son las mencionadas previamente, se puede establecer la correspondencia de la primera con el concepto de *evento* y de la segunda con el de *consulta*. Este análisis concluye que todo *episodio* definido según la Regla 3 debe pertenecer a la clasificación *evento* o *consulta*.

Componentes de los episodios. Para calcular los FP de un episodio se debe determinar cuáles son sus componentes que contribuyen al cálculo.

Para modelizar la interacción usuario-sistema se utilizó el modelo *entrada-proceso-salida (E-P-S)* formado por tres componentes que representan: el ingreso de datos proveniente del usuario que atraviesa el *límite del sistema (entrada)*, el procesamiento sobre los datos ingresados dentro del *límite (proceso)* y el retorno de datos hacia el usuario que cruza el *límite (salida)*.

De lo anterior se desprende la similitud del modelo propuesto con la definición de MKII: Cualquier *transacción lógica* debe tener alguna entrada del usuario que atraviesa el *límite*, debe hacer algún procesamiento sobre los *tipos de entidades* dentro del *límite* y retornar la salida a través del *límite* [25].

Según el modelo se deben descomponer los *episodios* en componentes de *entrada*, *proceso* y *salida*.

- *Entrada*: consiste en la adquisición de los datos provistos por el usuario que llegan al sistema, sea para describir un *evento* de interés en el mundo externo o una *consulta* para recuperar información del sistema.
- *Proceso*: consiste en las acciones de almacenamiento y/o recuperación de información que describe el estado de los elementos de interés para el mundo exterior.
- *Salida*: son los ítems provistos al usuario, sea cual fuere la categoría de la acción. Esta parte comprende las tareas de presentación de la información hacia el mundo externo [25].

Los componentes *E-P-S* de un *episodio* contienen una serie de objetos³ que para contabilizar se clasifican en dos grupos: símbolos del LEL y palabras o frases que no

³ En este trabajo se considera la definición: Un objeto es cualquier entidad del mundo real, importante para la discusión de los requerimientos, con un borde claramente definido. [8]

han sido definidas dentro de las entradas del LEL [14]. Estos objetos pueden ser *Definido en el LEL* o *No definido en el LEL* y en el primer caso los objetos pueden ser *Simples* o *Complejos*.

Las **Reglas 8 y 9** establecen que si el *componente de entrada/salida* incluye una entrada del LEL de la clasificación *Simple* o un objeto *No definido en el LEL*, se debe clasificar como DET.

La similitud de las definiciones *Definido en el LEL/Simple* y objeto atómico *No definido en el LEL* con la definición de *DET* de MKII soporta establecer la equivalencia entre los conceptos y sugiere denominar *tipo de elemento dato (DET)* a toda entrada del LEL de la clasificación *Simple* y a los objetos atómicos *No definido en el LEL* que integran los componentes de *entrada y salida* del *episodio*.

Durante la realización de un *episodio* se utilizan *recursos*. Los *recursos* son entradas del LEL que pueden pertenecer a la clasificación *Simple*, *Complejo* o *No definido en el LEL* y que forman parte del componente de *proceso* del *episodio*. Si bien hay alguna similitud entre los conceptos de *entidad* de MKII y *recurso* no se puede establecer una relación uno a uno entre ambos, pues el *recurso* tiene un significado más amplio. Se consideró que las *entidades* de MKII que componen la parte de *proceso* son los *recursos*. (**Regla 12**)

Una vez identificados los *DET* en los componentes de *entrada y salida* y los *recursos* en el componente de *proceso* del *episodio* se establecieron reglas para valorar su contribución a la cuenta de FP del *episodio*.

La equivalencia conceptual entre *transacción* y *episodio* sugiere establecer que el tamaño de los componentes de *entrada y salida* de un *episodio* es proporcional a la cantidad de *DET* en los componentes de *entrada y salida*. La **Regla 13** establece que se debe contabilizar cada *DET* distinto perteneciente a los componentes de *entrada y salida*.

La equivalencia conceptual entre *entidad* y *recurso* permite establecer que el tamaño del componente de *proceso* de un *episodio* es proporcional a la cantidad de *recursos referenciados* durante la realización de las acciones.

En una aplicación que almacena información acerca de un conjunto de *entidades*, la presencia de cada *entidad* implica la existencia de un conjunto mínimo de *transacciones lógicas*, conocido con el acrónimo CRUDL (Create, Read, Update, Delete, List). Vale analizar si existe una relación uno a uno entre ellas y las operaciones posibles en el entorno de este enfoque. Si bien se consideran válidas, deben ampliarse para incluir el concepto de “uso” del *recurso*, cuestión que deriva de la definición de *recurso* y que simplemente puede ser manipularlo para realizar alguna actividad. Las operaciones de creación, lectura, actualización, borrado, listado y uso de un *recurso* se denominan genéricamente *referenciar un recurso*.

La **Regla 14** determina que se debe contabilizar una única *referencia* por cada *recurso* en el componente de *proceso* del *episodio*.

El límite del sistema. MKII FPA se utiliza para medir la funcionalidad requerida por los usuarios de la aplicación dentro de un *límite* definido por el *propósito de la medición* y el *punto de vista* [25]. El *Propósito de la medición* está determinado por el objetivo de la medición. Por otra parte se distinguen tres puntos de vista, *Project*, *Manager Application* y *Business Enterprise*.

El *Límite de la aplicación* define un borde conceptual entre el software y el usuario. Dentro de ese *límite* la aplicación debe conformar un cuerpo consistente de funcionalidad, vale decir que debe incluir procesos completos.

Se estableció (**Regla 15**) considerar dentro del sistema exclusivamente a los escenarios. La visión del modelo de escenarios se corresponde con el concepto de *límite* de MKII, en tanto los *actores* están fuera de la aplicación y el conjunto de todos los escenarios representa la funcionalidad de la aplicación. Se denomina *límite del sistema* al borde conceptual que encierra a todos los escenarios y todo aquello que no es un escenario se considera fuera del *límite del sistema*.

El *límite del sistema* elicitado utilizando L&E y el del software a desarrollar no necesariamente serán coincidentes debido a que puede ser que no todas las funciones del sistema sean automatizadas.

Una vez definido el *límite del sistema*, se analizaron los conceptos de *usuario* en FPA y *actor* en los escenarios para estudiar la equivalencia entre ambos.

7 Proceso de medición de funcionalidad de L&E

El proceso de medición de funcionalidad de L&E se describe como una serie de etapas en secuencia cuyo esquema se muestra en la Figura 2. La columna principal indica las 6 etapas y para cada etapa se representan los subprocesos en que se descompone. Las flechas indican que el producto resultante de cada etapa es utilizado como entrada por la etapa siguiente. La Etapa 4 se reitera para cada *episodio* obtenido de la Etapa 3. La Etapa 5 se debe reiterar para cada componente de cada *episodio* obtenido de la Etapa 4. El subproceso 6.1 de la Etapa 6 se ejecuta para cada *episodio*.

Durante las diferentes etapas del proceso se aplican las 16 reglas definidas previamente.

El proceso de medición se documenta con tres formularios [4] cuyos objetivos son: catalogar los episodios, registrar el detalle del análisis por episodio y resumir el cálculo de FP.

8 Aplicación a casos de estudio

8.1 Caso de estudio

Se aplicó el enfoque propuesto para calcular los FP sobre el L&E desarrollado para el caso de estudio de la Recepción del Hotel. El objetivo de ese trabajo fue desarrollar la elicitación de requerimientos para el subsistema correspondiente a la Recepción de un Hotel de la ciudad de Comodoro Rivadavia⁴. Se realizaron entrevistas con los

⁴ Este caso de aplicación de L&E se desarrolló en el marco del Magister en Ingeniería de Software de la Universidad Nacional de La Plata en forma conjunta con Elena Centeno.

usuarios del subsistema y además se tuvo acceso a las planillas de uso habitual para registrar las diferentes actividades. En [3] se reproduce el resultado obtenido.

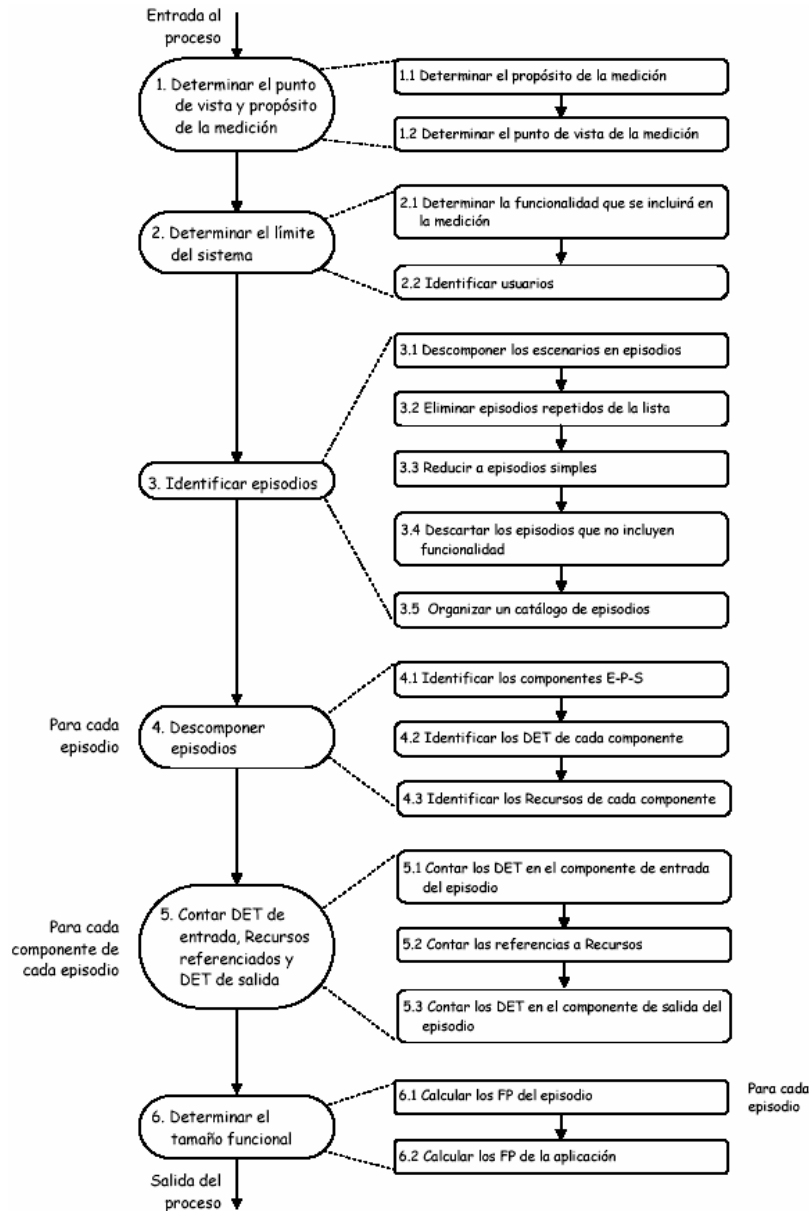


Figura 2. Etapas del proceso de medición

Etapa 1 Determinar el punto de vista y propósito de la medición. El propósito es medir *la funcionalidad requerida por el usuario* a partir de la documentación de

L&E. Se adopta el punto de vista del usuario. Según la Regla 16 se considera usuario a cada uno de los actores presentes en los escenarios.

Etapa 2 Determinar el límite del sistema. En esta etapa se estableció el límite del sistema que incluye 10 escenarios y se confeccionó una lista que incluye 8 usuarios. Escenarios obtenidos: cancelación de reserva, check in, check out, pago, pedido de alojamiento, pedido de servicio extra, recepción del Hotel, reposición de insumos, servicio de despertador, solicitud de reserva. Usuarios obtenidos: Recepcionista, Agencia, otro hotel, pasajero, Departamento Mantenimiento, Departamento Muca-mas, Departamento Administración, Departamento Compras.

Etapa 3 Identificar episodios - Subprocesos 3.1 – 3.4. El primer paso fue organizar una planilla que incluía los 53 episodios pertenecientes a los 10 escenarios, se descartaron: 4 episodios Excepción, 7 Restricción, 15 que no representan funcionalidad de interés, 7 Repetido y 2 Escenario.

Etapa 3 Subproceso 3.5 Catálogo de episodios. En el Formulario “Catálogo de episodios” se incluyeron los 17 episodios obtenidos en la etapa previa pertenecientes a 8 escenarios. Se descartaron dos escenarios, uno cuyos episodios eran escenarios y en el otro los episodios se repetían en otros escenarios.

Etapas 4 y 5 Descomponer episodios y Contar DET de entrada, recursos referenciados y DET de salida. Se descompuso cada episodio del Formulario “Catálogo de episodios” en sus componentes de *entrada*, *proceso* y *salida*. Se identificaron los *DET* en los componentes de *entrada* y *salida* y los *recursos* en el de *proceso*. En un Formulario “Detalle del análisis por episodio” por cada episodio se registró el detalle de los *DET* de entrada, *recursos* y *DET* de salida. Durante la Etapa 5 se calculó la suma de cada uno de esos ítems en cada Formulario “Detalle del análisis por episodio”.

Como resultado de estas etapas se obtuvieron 17 ejemplares del Formulario “Detalle del análisis por episodio”.

Etapa 6 Determinar el tamaño funcional. Se ingresaron en el Formulario “Resumen del cálculo de los FP” los datos de identificación de cada episodio y los totales para los *DET* de entrada y salida y de los *recursos* obtenidos en cada Formulario “Detalle del análisis por episodio”. Esos totales se sumaron para obtener los FP de cada *episodio*. Finalmente se calculó la suma de los FP de cada episodio y se determinó el total de FP de la aplicación.

Resumiendo, se analizaron 8 escenarios y 17 *episodios*, a partir de los cuales se obtuvieron 103 FP.

8.2 Aplicación a otros casos

El proceso se aplicó a los siguientes casos: Sistema nacional de obtención de pasaportes [18], Administradora de Planes de Ahorro para automóviles [24], Estación de servicio [9]. En la Tabla 1 se resume la información.

Tabla 1. Cuadro resumen de los casos de estudio analizados

Caso	Escenarios	Episodios	FP	FP/Escenario	FP/Episodio
A. de Planes de ahorro para automóviles	12	23	79	6,58	3,43
Recepción del Hotel	8	17	103	12,88	6,06
S. N. de obtención de pasaportes	14	24	128	9,14	5,33
Estación de servicio	17	45	219	12,88	4,87

Se observa una tendencia creciente en el número de FP a medida que aumentan los escenarios y los *episodios*. Es importante destacar que los FP, particularmente para los casos *Recepción del Hotel*, *S. N. de obtención de pasaportes* y *Estación de servicio*, se ordenan según lo que sugiere la intuición sobre el tamaño de los sistemas.

Los valores de la relación FP/Episodio están entre un mínimo de 3,43 FP y un máximo de 6,06 FP. Estos valores permitirían hacer estimaciones iniciales rápidas a partir de obtener la cantidad de *episodios* que cumplen las imposiciones del modelo propuesto.

La relación FP/Escenario parece tener menor valor para las estimaciones, pues su rango de variación es bastante más amplio y por otro lado, puede tener más vinculación con el área de aplicación del sistema.

Los resultados obtenidos para el caso de la *Administradora de Planes de ahorro* para automóviles son notablemente más bajos en relación a los otros casos. Posiblemente la diferencia proviene del nivel de detalle de la documentación disponible.

9 Conclusiones y trabajos futuros

La investigación realizada se propuso obtener mediciones de tamaño de los artefactos producidos en las etapas iniciales del proceso de desarrollo. Se desarrolló un enfoque para medir la funcionalidad de los Escenarios producidos en el contexto de L&E, basado en el Método MarkII FPA y que aprovecha fuertemente las similitudes conceptuales entre el modelo de *transacciones* de MKII y los *episodios* de los escenarios.

La primera y más importante conclusión de este trabajo es que se pueden medir los FP a partir de L&E. El enfoque desarrollado se aplicó en varios casos de estudio obteniéndose estimaciones del tamaño de los Escenarios en términos de FP. Este enfoque puede favorecer la estimación del tamaño de una aplicación muy al principio del desarrollo y podrá refinarse en etapas posteriores.

La utilización de un juego de formularios preestablecidos como soporte documental resultó una guía útil para el trabajo de medición y para documentar los estados intermedios. La experiencia realizada, aunque corta, avala plenamente el juego de formularios propuesto.

Un tema importante y delicado de manejar en el proceso de contabilización de FP es la descomposición de episodios. La experiencia realizada sugiere que la descomposición se facilitaría considerablemente si los episodios estuvieran expresados en términos de entrada-proceso-salida, pero ello alude a aspectos clave del modelo de L&E.

Si bien se carece de mediciones del esfuerzo de estimación ejecutado, se observó que cuando se adquirió destreza en el proceso, se redujo notablemente el esfuerzo requerido, pero estimar el esfuerzo requiere mayor experimentación. Este se reduciría aún más con una herramienta automatizada que soporte el cálculo. El FPA propuesto no puede considerarse una carga significativa al proceso de elicitación.

Debe destacarse que esta propuesta no considera el esfuerzo, lo que equivale a considerar equivalentes los esfuerzos para las diferentes transacciones lógicas. Esto puede resultar un impedimento si se pretende comparar los resultados de su aplicación con las medidas obtenidas por los métodos convencionales.

Las similitudes conceptuales que se identificaron entre ciertas categorías que considera el modelo MKII y del enfoque de L&E, permitieron establecer una conceptualización de contabilización de FP, construir un proceso para llevarla a cabo y aplicar el desarrollo a un conjunto de casos, obteniéndose resultados prometedores. A partir de este punto las futuras investigaciones se orientarán en la dirección de estabilizar y profundizar el resultado.

En párrafos previos se hicieron referencias a temas inmediatos de investigación: construir una herramienta CASE que soporte el proceso, establecer mediciones del esfuerzo requerido por el proceso de contabilización de FP en L&E, estimar el tamaño a lo largo del ciclo de vida completo.

Las próximas investigaciones deberán validar la propuesta, lo que requerirá generar un conjunto de datos de casos y aplicar el cálculo de FP a un espectro más amplio de casos. En la dirección de ampliar la propuesta se investigará la posibilidad de reemplazar el enfoque MKII por IFPUG así como extender su alcance a otros tipos de escenarios de elicitación [23].

Referencias

- [1] Albrecht, A., "Measuring Application Development Productivity", *IBM Applications Development Symposium*, Monterey, CA, 1979.
- [2] Albrecht, A., Gaffney, J., Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Transactions on Software Engineering*, Vol. SE-9, no. 6, Nov. 1983.
- [3] Bertolami, M., Centeno, E., *LEL y Escenarios de la Recepción del Hotel*, Caso de estudio desarrollado en el marco del Magister en Ingeniería del Software, U. N. L. P., 2001.
- [4] Bertolami, M., *Una propuesta de Análisis de Puntos Función aplicado a LEL y Escenarios*, Tesis del Magister en Ingeniería de Software presentada a la Facultad de Informática de la U. N. L. P., 2003.
- [5] Cysneiros, L., Leite, J., "Using the Language Extended Lexicon to Support Non-Functional Requirements Elicitation", *Workshop on Requirements Engineering*, Buenos Aires, Argentina, 2001.
- [6] Dekkers, C.A., "Managing (the Size of) Your Projects A Project Management Look at Function Points", *CROSSTALK The Journal of Defense Software Engineering*, February 1999
- [7] Dekkers, C., Aguiar, M. "Applying Function Point Analysis to Requirements Completeness", *CROSSTALK The Journal of Defense Software Engineering*, February 2001
- [8] Davis, A., *Software Requirements: Objects, Functions, and States*, Englewood Cliffs, Prentice Hall, 1993.

- [9] Esteban, N., Heidanowski, A., *LEL y Escenarios de una Estación de Servicio*, Trabajo Final de Tópicos I, U. N. L. P., 1998.
- [10] Fenton, N., Pfleeger, S., *Software metrics A rigorous and Practical approach*, Second Edition, PWS Publishing Company, 1996.
- [11] Fetcke, T., Abran, A., Nguyen, T., *Mapping the OO-Jacobson Approach into Function Point Analysis*, Université du Québec à Montreal, Software Engineering Management Research Laboratory, 1998.
- [12] Fetcke, T., *The Warehouse Software Portfolio: A Case Study in Functional Size Measurement*, Report No. 1999-20, Technische Universität Berlin, Fachbereich Informatik, 1999.
- [13] Grant Rule, P., *Scenarios, Use Cases & MKII FPA*, GIFPA (Guild of Independent Function Points Analysts), 1998.
- [14] Hadad, G., Kaplan, G., Oliveros, A., Leite, J., Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos. Aplicación a un caso real, Universidad de Belgrano, 1996.
- [15] Hadad, G., Kaplan, G., Oliveros, A., Leite, J., “Construcción de Escenarios a partir del Léxico Extendido del Lenguaje”, 26 *JAIIO*, Buenos Aires, Argentina, 1997.
- [16] International Function Point Users Group, *Frequently Asked Questions*, 2000.
- [17] International Function Point Users Group, *About Function Point Analysis*, 2000.
- [18] Leite, J., Oliveros, A., Rossi, G., Balaguer, F., Hadad, G., Kaplan, G., Maiorana, V., *Léxico extendido del lenguaje y escenarios del sistema nacional para la obtención de pasaportes*, Universidad de Belgrano, 1996.
- [19] Leite J., Rossi G., et al, “Enhancing a Requirements Baseline with Scenarios”, *Proceedings of RE 97' International Symposium on Requirements Engineering*, IEEE, 1997.
- [20] Leite, J., Hadad, G., Doorn, J., Kaplan, G., *A Scenario Construction Process*, Requirements Engineering Journal, 2000.
- [21] Loucopoulos, P., Karakostas, V., *System Requirements Engineering*, McGraw-Hill, 1995.
- [22] Longstreet, D., *Use cases and function points*, Longstreet Consulting Inc, 2000.
- [23] Maiden, N., “CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements”, *Automated Software Engineering*, vol 5, nro4, 1998
- [24] Mauco, V., Ridao, M., del Fresno, M., Rivero, L., Doorn, J., *Proyecto Sistema de Planes de Ahorro*, Facultad de Ciencias Exactas, U. N. C. P. B. A., Argentina, 1997.
- [25] *MKII Function Points Analysis Counting Practices Manual*, Versión 1.3.1, UKSMA (United Kingdom Software Metrics Association), 1998.
- [26] Park, R.E., *Software Size Measurement: A Framework for Counting Source Statements*, CMU/SEI-92-TR-20, 1992
- [27] Pressman, R., *Ingeniería del Software. Un enfoque práctico*, Tercera edición, 1993.
- [28] Probasco, L., “Dear Dr. Use Case: What About Function Points and Use Cases?”, *RationalEdge*, August 2002
- [29] Rolland, C., *Modeling the Requirements Engineering Process*, Université de Paris I Pantheon-Sorbonne, 1993.
- [30] Symons, C., *Software sizing and estimating MKII FPA*, John Wiley and Sons, 1991.
- [31] Symons, C., Come back function point analysis (modernised) – All is forgiven!, Software Measurement Services Ltd, 2001.
- [32] Tavares, H., Carvalho, A., Castro, J., “Medição de pontos por função a partir da especificação de requisitos”, *Workshop on Requirements Engineering*, Valencia, España, 2002.