

# Puntos *por* Función. Una métrica estándar para establecer el tamaño del software



No puedo controlar, lo que no  
puedo medir

Sergio Eduardo Durán Rubio\*

## Introducción

La sociedad moderna cada vez se ha tornado más dependiente del software (SW). Los programas de computadora los podemos encontrar en tareas tan variadas como hacer el pago en una tienda departamental, telefonía, medicina, transacciones bancarias, búsqueda de información en internet o trámites gubernamentales. El SW se ha convertido en el bien de nuestra era; un bien al que cada vez se le invierte más en las empresas o instituciones, en su búsqueda por ofrecer más y mejores servicios, o simplemente por reducir costos operativos.

Al ser el SW un bien en el que se invierten grandes cantidades, se hace muy importante poder controlarlo. En los años recientes, el gasto en tecnologías de información ha trasladado su énfasis del hardware al software, provocando que la relación entre el segundo y el primero suba de 32.5% en 1996 a 40% en 1999<sup>8</sup>. Sin embargo, a pesar de su importancia, es un bien intangible que se ha tornado difícil medir. Ya sea que se decida comprar un paquete o que se haga un desarrollo a la medida, interno o subcontratado, las organizaciones necesitan controlar esta inversión. En ese sentido, el medir el software no es un tema académico, sino un tema de valor, de inversión, de negocio.

De hecho es conocido que grandes proyectos han fracasado al no estar a tiempo o dentro de presupuesto por una mala estimación de esfuerzo o duración, o de las capacidades requeridas de los ingenieros y de la empresa.

---

El autor es Director Comercial, CERTUM.

---

<sup>8</sup> Fuente; Digital planet; The Global Information Economy. WITSA. Noviembre de 2000

Este artículo muestra cómo el uso de métricas y en este caso de una métrica de tamaño basada en la funcionalidad, **Puntos Función**, nos puede ayudar a tener mejor control y una mejor evaluación de la inversión en proyectos de tecnología basados en SW.

## Lo relevante es la funcionalidad

Como en cualquier inversión, a las empresas lo que les interesa es la capacidad de hacer algo con lo que compran. Por ejemplo, si se compra un camión, de alguna forma se adquirió la capacidad de transportar mercancías y con un tamaño de metros cúbicos por viaje. En primera instancia lo relevante es qué tanto requiero transportar por viaje. En otro ejemplo, si se compra un edificio, entonces se adquirió la capacidad de disponer de un espacio de X metros cuadrados distribuido en pisos; lo relevante es cuánto espacio necesita la empresa o institución.

En SW no tiene por qué ser distinto; al adquirir un paquete o al desarrollar una aplicación, lo primero que debe evaluarse es qué nuevas capacidades estoy adquiriendo. Estas capacidades o funcionalidad estarán en términos de qué transacciones puedo realizar de forma automatizada y qué grupos de datos puedo guardar.

Parece obvio, pero la mayoría de las empresas cuentan con una vaga idea de la funcionalidad en SW con la que cuentan y mucho menos cuánto de esa funcionalidad realmente se aprovecha. Tampoco cuentan con la productividad de sus programadores en términos de cuánta funcionalidad pueden desarrollar por unidad de tiempo. Esto les impide estimar, con fundamento, la inversión y el tiempo necesario para que su personal pueda desarrollar un proyecto de SW o para evaluar la propuesta de un externo. En consecuencia, los retrasos en los proyectos así como los excesos en presupuesto son normales. De hecho el problema del año 2000 fue un caso muy evidente donde las empresas no sabían lo que gastarían para hacer las adecuaciones necesarias ya que no había datos precisos de con cuánto SW contaban.

Para evaluar una inversión en SW, normalmente se considera el costo de desarrollarlo contra el precio de comprarlo ya hecho. Pero en esta comparación el costo o precio es resultado de un esfuerzo estimado por un costo unitario.

Los dos principales determinantes para estimar el esfuerzo son: el tamaño de lo que se requiere y la productividad de quien lo va a hacer. En este artículo nos concentraremos en el primero, ya que es el primer elemento de la ecuación.

## Medir el tamaño

El tamaño del SW podría medirse en términos de los *bytes* que ocupa en el disco, el número de programas, el número de líneas de código, la funcionalidad que proporciona, o simplemente el número de pantallas o reportes que tiene.

A simple vista podríamos intuir que algunas de estas propuestas son mejores que otras si queremos medir el tamaño de una forma que tenga más correlación con el esfuerzo. Pero antes de seleccionar alguna, hagamos otras consideraciones.

Una aplicación de software es un conjunto de líneas de código que se ejecutan en una computadora. Sin embargo mucho del costo de producir ese SW no está directamente relacionado con la codificación, que es entre el 20 y 25% del costo total. Elementos como la administración del proyecto, el nivel de detalle de la documentación técnica o la documentación de pruebas, y las pruebas por sí mismas también deben considerarse.

Por otro lado, hoy en día hay una amplia gama de lenguajes y herramientas para producir SW, lo que ha provocado que pueda generarse la misma funcionalidad con lenguajes de programación distintos. Y esto con un número de líneas de código distintos y, lo que es más impactante, con un esfuerzo distinto.

Veamos un poco más a detalle las implicaciones del enunciado anterior mediante un ejemplo sencillo. Supongamos que yo requiero de cierta funcionalidad en una aplicación de SW y tengo la opción de hacerlo con dos empresas. La empresa A utiliza un lenguaje que requeriría 8 mil líneas de código. La empresa B utiliza otro lenguaje que requeriría aproximadamente 5 mil líneas de código. A simple vista parecería que con la empresa A tendría “más” trabajo, pero con la empresa B me podría salir más barato, dado que tienen que hacer menos líneas de código. La comparación se complica si los programadores de ambas empresas producen a distintas velocidades. Es decir, los de la compañía A producen casi el doble de código por día que los de la compañía B. Como resultado de lo anterior, tendría que la línea de código de la empresa A podría costar mucho menos que la de la compañía B. Entonces parecería que debería comprarle a la empresa A dado que cuesta menos cada línea de código, pero también podría comprarle a la empresa B dado que tiene que producir menos líneas de código, y por lo tanto el costo total sería menor o podría tener menos posibilidad de errores.

Ciertamente esta secuencia de análisis puede ser engañosa. Si retomamos los conceptos mencionados previamente, entonces una mejor métrica para establecer el tamaño es la basada en los requerimientos del usuario y no en la tecnología que se va a utilizar; una métrica basada en la funcionalidad.

## Características de una métrica funcional

A continuación mencionaré las características principales de una métrica funcional que nos ayudarán a entender mejor los beneficios de este tipo de métrica.

**INDEPENDIENTE DE LA TECNOLOGÍA.**— Como vimos en el ejemplo anterior, si nos basamos en líneas de código (en la tecnología) vamos a obtener resultados no comparables. Pero más que eso, tal vez lo relevante de esta característica es que una

vez establecida la funcionalidad que requiero, debo escoger la tecnología que me haga más productivo para obtener tal funcionalidad.

**SIMPLE.** – Desde su concepción, este tipo de métrica se planteó que no requiriera grandes esfuerzos para obtener una medida. Una ventaja es que puedo establecer el tamaño de un SW que tal vez llegue a tener miles de líneas de código en pocas horas. Sin embargo, el costo de esta simplicidad es que la métrica no es muy sensible a consideraciones muy detalladas, como lo sería por ejemplo, la complejidad de fórmulas matemáticas.

**ENFOQUE EN LA FUNCIONALIDAD PROPORCIONADA.**– Como mencionamos con anterioridad, el primer criterio para evaluar debe ser qué nuevas capacidades voy a obtener con el nuevo SW, antes de cualquier evaluación técnica.

**BASADA EN LOS REQUERIMIENTOS DEL USUARIO.**– Esta característica ayuda a que el usuario pueda entender el significado e implicaciones del tamaño del SW, sin tener que ser un experto en sistemas. Otro punto muy importante con esta característica es que puedo establecer un tamaño desde que tengo los requerimientos y no necesito esperar a terminar el proyecto para saber de qué tamaño va a ser.

**CONSISTENCIA.**– Los resultados obtenidos entre diferentes personas o en proyectos distintos deben ser consistentes.

## Usos de una Métrica Funcional Estándar

Una vez que tengo el tamaño de un SW, entonces puedo utilizar ese dato para distintos propósitos. La siguiente lista es más enunciativa que limitativa.

**Administrar la productividad.**– Si tengo el tamaño y por otro lado el esfuerzo requeridos entonces puedo establecer indicadores de productividad en términos de Horas por Puntos de Función. Este indicador me puede servir para controlar un plan de mejora y para compararme con la industria<sup>9</sup>.

**Administración de calidad.**– De manera similar si tengo el tamaño y el número de defectos que se entregaron en un desarrollo, entonces puedo establecer un indicador de calidad como defectos por Puntos Función, para la administración de la calidad. Con una historia suficiente de proyectos, puedo estimar proyectos futuros.

**Comparabilidad.**– Al utilizarse la misma métrica en todos mis proyectos, entonces puedo compararlos entre ellos, pero mejor aún, si es una métrica estándar en la industria podría compararme contra otros. Veamos esto en el siguiente ejemplo.

---

<sup>9</sup> Datos de productividad se pueden obtener en *The Software Metrics Compendium*, editado por el *International Software Benchmarking Standards Group*. [www.isbbsg.org](http://www.isbbsg.org)

	Pequeño (<300 PF)		Mediano (300-600 PF)		Grande (>600 PF)	
	Empresa	Promedio industria	Empresa	Promedio industria	Empresa	Promedio industria
Productividad (PF/Esfuerzo)	4.69	10.73	7.38	9.27	5.62	6.34
Esfuerzo (Meses Ingeniero)	39.73	18.64	72.42	43.15	196.2	126.18
Duración (Meses Calendario)	9.62	9	8.5	13	20.83	19

En este ejemplo<sup>10</sup>, la cantidad de esfuerzo de la empresa es mucho mayor al promedio de la industria. La productividad por lo tanto es menor. La duración, sin embargo, es muy similar al promedio de la industria.

- Administración de proyectos.– En lugar de medir únicamente horas planeadas vs. horas consumidas puedo administrar el proyecto basándose en Puntos Función planeados vs. diseñados o construidos.
- Administrar los cambios en el alcance. – Con Puntos Función, puedo estimar el tamaño de los cambios en un proyecto, y a partir de ahí estimar el esfuerzo.
- Estimar la adecuación de un paquete.– Si conozco la funcionalidad de un paquete y la funcionalidad que requiero, con Puntos Función podría estimar qué tanto del 100% que tiene el paquete requiero adecuar.
- Valuar el SW de una organización.– Una vez que tengo el tamaño de cada aplicación dentro de la organización, puedo valuar mejor el activo que tengo en este terreno que sólo usar el costo en que incurri cuando lo compré.
- Estimar los recursos para un proyecto.– Con el tamaño estimado de un SW es más fácil estimar el número de ingenieros que se requieren. No es el único dato que se debe considerar, pero sí el que más impacto tiene al estimar esto.
- Presupuestar el mantenimiento.– Una métrica funcional MF puede ayudar en el presupuesto para el mantenimiento de un portafolio de software de una organización. El tamaño de funcionalidad del portafolio, así como el costo o esfuerzo de mantenimiento comparado con el tamaño de funcionalidad, podrían ser monitoreados. Esta información podría ser usada para pronosticar presupuestos de mantenimiento.
- Administrar contratos.– El contrato con un proveedor puede ser basado en los requerimientos funcionales, tamaño de funcionalidad, una productividad esperada y un costo por unidad de tamaño de funcionalidad.

<sup>10</sup> Dave Garmus. “*Benchmarking Application Development and Maintenance*”. IFPUG event: *Bigger and Better Metrics*. September 2002.

## La Métrica de Puntos Función

Esta métrica se define como una métrica funcional, dado que se enfoca a la funcionalidad que el SW proporciona al usuario.

*“Es una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la cantidad de funcionalidad requerida y entregada a los usuarios”,*

O

*“Los Puntos Función miden el tamaño lógico o funcional de los proyectos o aplicaciones de software basados en los requerimientos funcionales del usuario”<sup>11</sup>.*

Partamos de la primera definición para entender las características de la métrica:

- **TAMAÑO** – es una métrica de tamaño, no de la calidad con la que se hizo ese SW, o del valor de ese producto, o del esfuerzo requerido para desarrollarlo, etc.
- **APLICACIONES** – mide las aplicaciones de SW, no considera el HW que utilizará, ni la administración del proyecto, ni la documentación, etc.
- **FUNCIONALIDAD** – se refiere a la capacidad del SW para que un usuario pueda realizar transacciones (lectura, escritura, etc.) y el guardar datos. Si analizamos a detalle, con estos elementos podemos describir cualquier sistema.
- **USUARIO** – quien lo va a usar y no quien lo desarrolló o quien lo diseñó

Así como existe el metro lineal para medir longitudes, Puntos Función es “el metro” para medir tamaño de una aplicación de software.

## Estándar Internacional ISO

La medición de la funcionalidad con la que cuenta un sistema informático ha sido desde hace años una preocupación de la industria. No es suficiente contar con una métrica, sino que sea estándar para así poderla usar entre empresas o para tener indicadores a nivel industria que todos puedan entender y operar. El poder comparar la productividad (Puntos Función por Persona Mes) de una empresa con los datos de la industria, es fundamental en los planes de mejora.

Es por eso que a través de la *International Organization for Standardization* (ISO<sup>12</sup>) se ha desarrollado un estándar internacional:

---

<sup>11</sup> Carol Dekkers. “*Measuring the logical or functional size of software projects and software application*”. *ISO Bulletin May 2003*.

<sup>12</sup> [www.iso.org](http://www.iso.org)

**ISO/IEC 14143** – *Information Technology – Software Measurement – Functional Size Measurement*. Este estándar define los conceptos de una métrica de tamaño basada en la funcionalidad y las características que debe cumplir un método para poder estar homologado al estándar y ser considerado una medida del tamaño de la funcionalidad.

Para poder establecer la cantidad de puntos función que tiene una aplicación, existen varios métodos de conteo. En general todos estos métodos establecen un conteo basado en la identificación del tipo de funciones que tiene la aplicación, y a cada una le asocia un número de puntos función tomando en cuenta su complejidad. Las variantes surgen al buscar conteos más precisos en puntos función conforme al tipo de aplicación. Por ejemplo, un sistema en tiempo real tiene una complejidad muy distinta a un sistema tradicional de negocio, o a un sistema operativo o a una aplicación científica que realiza muchos cálculos, pero el resultado puede ser un solo dato. Los métodos que están homologados con el ISO 14143, aunque no todos son públicos, son:

- ISO/IEC 20926:2003 *Software engineering -- IFPUG 4.1 Unadjusted functional size measurement method*. Este método ha sido definido por el *International Function Point Users Group* (IFPUG<sup>13</sup>) y evolucionado, a partir de la propuesta original de Allan Albrecht en IBM, por lo que es el más conocido y más utilizado, sobre todo en Estados Unidos que es el mercado más grande de SW. Esto es muy importante porque se está convirtiendo *de facto* en el estándar de la industria.
- ISO/IEC 24570. *Software engineering -- NESMA -- Function Point Analysis*. Estándar definido por la *NEtherlands Software Metrics Users Association*. Esta es una pequeña variante del método del IFPUG.
- ISO/IEC 20968:2002 *Software engineering -- Mk II Function Point Analysis*. Este método ha sido desarrollado por la *United Kingdom Software Metrics Association*, simplificando el método y haciéndolo compatible con ideas de análisis y diseño estructurado.
- ISO/IEC 19761:2003 *Software engineering -- COSMIC-FFP -- A functional size measurement method*. Este método ha sido definido por el *COMmon Software Measurement International Consortium*, integrado por expertos de Australia, Canadá, Finlandia, Alemania, Irlanda, Italia Japón, Holanda y el Reino Unido. La idea principal es adecuarse mejor a la medición de sistemas en tiempo real.

## El Método Estándar Análisis de Puntos Función

Como se mencionó previamente el método que se está convirtiendo en el estándar de la industria es el definido por el IFPUG, que se llama *Function Point Analysis* (FPA) y sus autores los definen así:

*“Método estándar para medir el desarrollo de software desde el punto de vista del usuario”.*

<sup>13</sup>



Este método utiliza como unidad de medida puntos función y su versión actual es la 4.1.1, cuyo manual de prácticas de conteo puede encontrarse tanto en inglés como en español. A continuación describiremos brevemente en qué consiste este método.

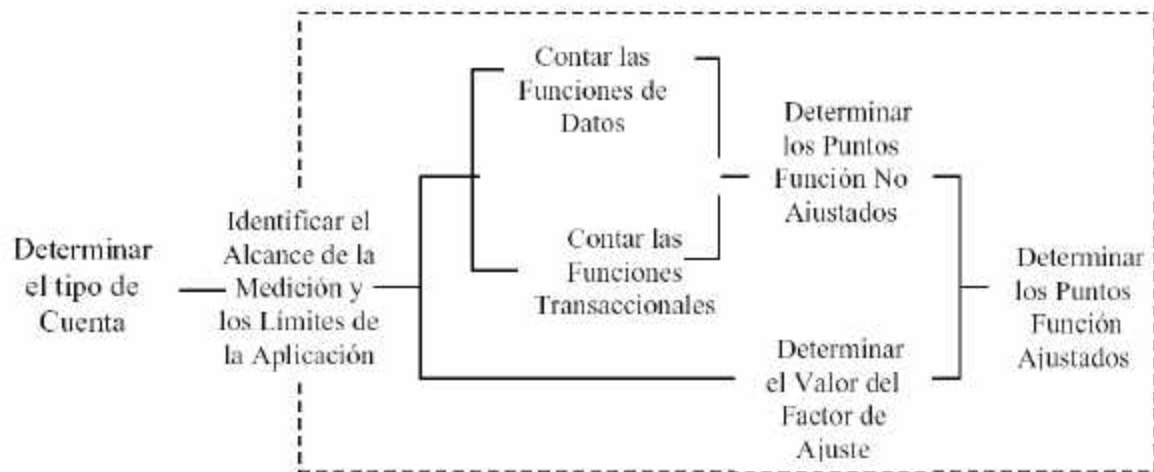
El método se basa principalmente en la identificación de los componentes del sistema informático en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes les asigna un número de puntos por función basándose en el tipo de componente y su complejidad; y la sumatoria de esto nos da los puntos de función sin ajustar. El ajuste es un paso final basándose en las características generales de todo el sistema informático que se está contando.

Veamos un poco más a detalle el procedimiento, para entender mejor los conceptos mencionados y ver su simplicidad.

## Procedimiento<sup>14</sup>

### Paso 1. Determinar el tipo de conteo

Este paso consiste en definir el tipo de conteo entre desarrollo, mantenimiento o de una aplicación ya instalada. Esta es una forma de determinar el objetivo del conteo.

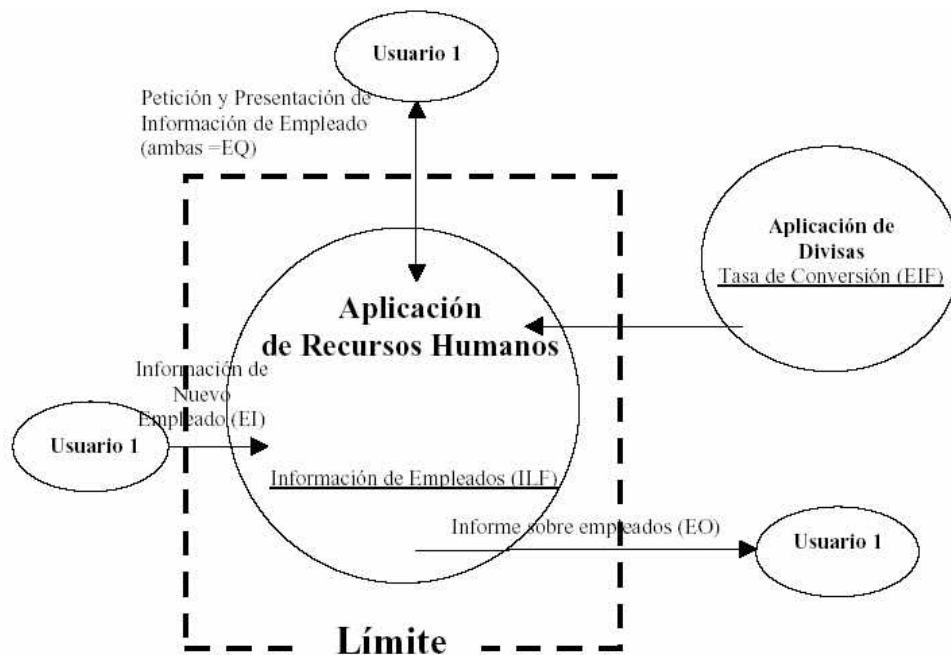


### Paso 2. Identificar los alcances de la medición y los límites de la aplicación.

El propósito de una medición consiste en dar una respuesta a un problema de negocio. El alcance de la medición define la funcionalidad que va a ser incluida en una medición específica y puede abarcar más de una aplicación.

<sup>14</sup> IFPUG. “Manual para la Medición de Puntos Función”. Versión 4.1.1. 1999





### Paso 3. Contar las funciones de datos

Este paso consiste en identificar y contar la capacidad de almacenamiento de los datos. Se distinguen dos tipos de funciones de datos:

*Archivo Lógico Interno* – es un grupo de datos relacionados que el usuario identifica, cuyo propósito principal es almacenar datos mantenidos a través de alguna transacción que se está considerando en el conteo.

*Archivo de Interfaz Externo* - es un grupo de datos relacionados y referenciados pero no mantenidos por alguna transacción dentro del conteo.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando principalmente el número de datos.

### Paso 4. Contar las funciones transaccionales.

Este paso consiste en identificar y contar la capacidad de realizar operaciones.

Se distinguen tres tipos de funciones transaccionales:

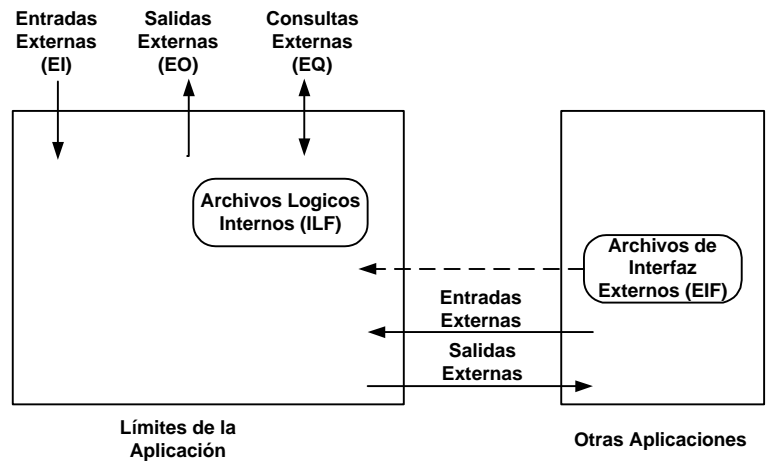
*Entrada Externa* – es un proceso cuyo propósito principal es mantener uno más archivos lógicos internos.

*Salida Externa* – es un proceso cuyo propósito principal es presentar información al usuario mediante un proceso lógico diferente al de sólo recuperar los datos.

*Consulta Externa* – es un proceso cuyo propósito principal es presentar información al usuario leída de uno o más grupos de datos.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando el número de datos utilizado en el proceso y los archivos referenciados.

Estos 5 componentes lógicos básicos son con los que se describe la funcionalidad de una aplicación y los podemos representar gráficamente de la siguiente forma:



**Paso 5. Determinar los puntos de función no ajustados.**

Este paso consiste en sumar el número de componentes de cada tipo conforme a la complejidad asignada y utilizar la siguiente tabla para obtener el total.

	Bajo	Medio	Alto	Total
EI	__ x 3= __	__ x 4= __	__ x 6= __	__
EO	__ x 4= __	__ x 5= __	__ x 7= __	__
EQ	__ x 3= __	__ x 4= __	__ x 6= __	__
ILF	__ x 7= __	__ x 10= __	__ x 15= __	__
EIF	__ x 5= __	__ x 7= __	__ x 10= __	__
				__

**Paso 6. Determinar el valor del factor de ajuste.**

El factor de ajuste se obtiene sumando 0.65 a la sumatoria de los grados de influencia de las 14 características generales del sistema, multiplicado por 0.01. Dentro de las características hay criterios como: complejidad del proceso, facilidad de instalación, entrada de datos en línea, etc.

**Paso 7. Determinar los puntos función ajustados.**

Para determinar los puntos función ajustados se consideran los puntos función no ajustados por el factor de ajuste.

**Cualquier métrica tiene un alcance definido**

Cualquier métrica tiene un ámbito de acción y alcance definido que hay que entender para usarla correctamente. Así por ejemplo, el metro lineal no es lo mejor para medir grandes distancias en el mar.

En nuestro caso, Puntos Función, está enfocado a medir sistemas informáticos completos, no programas. En este sentido no tiene un nivel de precisión suficiente para medir el tamaño de programas individuales. El nivel de granularidad que puede medir la

métrica no es muy pequeño. Adicionalmente el término programa depende de la tecnología y eso va en contra del criterio de que esta métrica es independiente de la tecnología que se use.

Otro punto que se le ha criticado a las métricas funcionales es que requieren que alguien “identifique” la funcionalidad y “evalúe” la complejidad basándose en los criterios y reglas establecidas; no puedo hacer un programa que cuente automáticamente. Debido a esto, distintas personas podrían llegar a un conteo diferente. Para resolver esto, se han venido depurando las reglas de conteo para eliminar posibles ambigüedades y cada vez hay más material de apoyo con ejemplos. Aunado a lo anterior, hay esquemas de certificación como el del IFPUG, donde hay una evaluación formal de la teoría y casos prácticos. Estos elementos buscan reducir las posibles variaciones en un conteo hecho por diferentes personas. Si tomamos en cuenta que la métrica está pensada para contar proyectos o aplicaciones completas, entonces las pequeñas variaciones en un conteo, no van a ser significativas para los indicadores o datos relacionados que obtengamos.

## Estrategia en México

México tiene un nivel de gasto en tecnologías de la información y comunicaciones (TIC) de 3.2 del PIB, ubicándose en el lugar 50 a nivel mundial. Este rezago es aún mayor en términos de gasto en software, que es 6 veces inferior al promedio mundial y 9 veces menos que el de EUA. Con estos niveles no es de extrañarnos que a nivel industria tengamos poco avance en temas como las métricas para SW.

A nivel industria, en México se requiere mejorar y es la intención del Plan Nacional de Desarrollo 2001 - 2006 que plantea el fomento a la industria y el mercado de Tecnologías de la Información (TI) como estrategia para aumentar la competitividad del país. Las TI tienen un efecto transversal en toda la economía, razón por la cual impactan positivamente la competitividad de todos los sectores.

Dado el gran potencial con que cuenta México para desarrollar esta industria, la Secretaría de Economía, en coordinación con organismos empresariales y empresas del sector, diseñó el Programa para el Desarrollo de la industria del Software (PROSOFT<sup>15</sup>). Este programa contempla entre sus estrategias el alcanzar niveles internacionales en Capacidad de Procesos de Desarrollo de SW y esto incluye el uso de métricas.

## Conclusiones

Como hemos visto, la industria del SW tanto desarrolladores como compradores, requiere mejores prácticas. Las métricas son un elemento fundamental de control en cualquier ingeniería y aquí no son la excepción. El tamaño del SW es un determinante fundamental en el esfuerzo de un proyecto de desarrollo de SW al igual que para la adquisición de un paquete. El tamaño basado en la funcionalidad que se obtiene centra las decisiones en

<sup>15</sup> Secretaría de Economía. “Programa para el Desarrollo de la Industria del Software”. [www.economia.gob.mx](http://www.economia.gob.mx)

obtener más funcionalidad por la inversión, una decisión de negocio. Entonces las decisiones tecnológicas se convierten en elegir la tecnología que me haga más productivo para el tipo de solución que requiero, principalmente.

Las compras de SW por parte de entidades gubernamentales son en muchos casos desarrollos a la medida. Pero ¿cómo comprar si todavía no tengo el diseño?. Una opción que habilita una métrica como Puntos Función es comprar por “metro” de SW, a una empresa que tenga una productividad mínima establecida, conforme al tamaño de las aplicaciones que requiero. Ya durante el proyecto puedo ir solicitando esos Puntos Función que compré, en las aplicaciones que requiero. Adicionalmente, la ley de adquisiciones permite hacer compras basadas en estándares internacionales, que como vimos, ya existen en este terreno. Adicionalmente en el Programa para el Desarrollo de la Industria del Software, de la Secretaría de Economía, hay un punto específico que ayudará en este tema: “Promover que se elaboren las normas necesarias para que se cumpla el Reglamento de la Ley de Adquisiciones, Arrendamientos y Servicios del Sector Público en lo que se refiere a la existencia de normas nacionales o internacionales en las adquisiciones de software”<sup>16</sup>.

En mi opinión, el uso de métricas en los proyectos de SW va a empezar a cambiar en la medida que los compradores lo exijan. Es por eso muy importante que los compradores analicen que sólo comprar por precio sin distinguir tamaño de las aplicaciones para escoger a un proveedor que tenga experiencia con esa complejidad o indicadores como productividad o calidad, nos va a seguir llevando a las historias que hemos oído de grandes proyectos que no terminan a tiempo o que exceden por mucho el presupuesto, o en el peor de los casos nunca llegan a operación. Se requiere un modelo que permita identificar capacidades de los proveedores, para entonces poder comparar propuestas de proveedores de capacidad similar. “Los proyectos nacionales en informática requieren de empresas con capacidades reales y funcionarios/ejecutivos competentes y exigentes”<sup>17</sup>.

Este artículo es una forma de ir entendiendo el tema para que los compradores puedan empezar a solicitar nuevas prácticas que den mayor certeza a las inversiones en tecnología. También hemos tenido acercamientos con la Secretaría de Economía con el objetivo de que a nivel país podamos tener prácticas comunes que mejoren las inversiones del gobierno y desarrollen una industria que a pesar de estar tan cerca del mercado más grande, Estado Unidos, todavía no figura en el mercado de SW.

Puedo mencionar también que ya empieza a haber proyectos que como parte de sus bases para invitar a proveedores, solicitan contar con métricas y con registros históricos que fundamenten los estimados en esfuerzo y tiempo, para tener mejor certeza en que el proyecto que están comprando va a llegar a buen término, a tiempo y en presupuesto. Próximamente empezaremos a ver los primeros resultados y sea motivo de un siguiente artículo alguno de estos casos de éxito.

<sup>16</sup> idem.

<sup>17</sup> Alberto Balderas, Arnoldo Díaz. “Fábrica de software. Un modelo de negocio certificable basado en Estructura y Capacidades”. Soluciones Avanzadas. No. 59