C a l i d a d d e s o f t w a r e

**Técnicas de Estimación**

Brenda Juárez Santiago

PROFESORA

Quinto Cuatrimestre

# Estimación de proyectos de software por puntos de función: Introducción



Todo proyecto de desarrollo de software comienza con una etapa de estimación y planificación, en la cual debemos determinar, a partir de los requerimientos del software, cuánto esfuerzo, personal, recursos materiales, tiempo y en última instancia dinero tomará construir el sistema o producto.

No solo los gerentes de proyectos estiman software, pues a cada desarrollador, analista de sistemas y software tester se le suele asignar un componente y se le pregunta cuánto tiempo le tomará elaborarlo.

Estas estimaciones suelen elaborarse de forma empírica, en base a la experiencia pasadas y conocimiento del trabajo a realizar, sin embargo, suelen ser imprecisas debido a que los requerimientos y sus implicaciones técnicas no se conocen en su totalidad y además ningún desarrollo de software es exactamente igual a uno anterior con el que se pueda comparar.

Frente a esto, se han desarrollado métodos como el de puntos de función, donde estimamos las funcionalidades usando fórmulas matemáticas basadas en parámetros como el tipo de componente, complejidad, factores del entorno, entre otros aspectos.

En este artículo presentamos una introducción a las estimaciones de proyectos de software usando los puntos de función, incluyendo que es, diferencias respecto a estimar en base a experiencia, un sencillo ejemplo y como puede ser de utilidad a distintos profesionales de ingeniería de software.

**¿Qué es el método de puntos de función?**

Fue desarrollado originalmente por Allan Albrecht en 1979 mientras trabajaba para IBM, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado.

El trabajo de Albrecht fue continuado por el grupo internacional de usuarios de puntos de función, quienes plasmaron sus conceptos en el método [IFPUG-FPA](http://www.ifpug.org/).

IFPUG-FPA realiza las valoraciones a partir de la funcionalidad del sistema, primero clasificándolas, luego asignando una complejidad y ponderación a cada una según unas tablas predefinidas, determinando así el valor de puntos de función.

Sumando los puntos de todas las funcionalidades se obtiene la valoración de todo el proyecto y finalmente se puede aplicar un factor de ajuste, que puede depender de características generales del sistema como por ejemplo [requerimientos no funcionales](http://www.pmoinformatica.com/2013/01/requerimientos-no-funcionales-porque.html) como el rendimiento, reusabilidad, facilidad de instalación y operación entre otros aspectos.

Los puntos de función permiten traducir el tamaño de funcionalidades de software a un número, a través de la suma ponderadas de las características que este tiene.

Una vez que tenemos los puntos de función, podemos traducirlos en horas hombre o días de trabajo, según factor de conversión que dependería de mediciones históricas de nuestra productividad. Con las horas hombre, podemos determinar el costo y presupuesto de los proyectos.

**¿Existen otros métodos de puntos de función distintos de IFPUG-FPA?**

Si, desde que surgió el IFPUG-FPA, los métodos de puntos de función se han diversificado, produciendo las siguientes variantes:

* **NESMA:** Método definido por la Netherlands Software Metrics Association.
* **MkII:** Método definido por el United Kingdom Software Metrics Association.
* **COSMIC:** Denominado Full Function Points o COSMIC-FFP, fue desarrollado por el Common Software Metrics Internacional Consortium.
* **FiSMA:** De la asociación finlandesa de medición de software.

Cada uno de estos métodos tiene aplicaciones en distintos tipos de proyectos de software. Además, cada uno cuenta con su estándar ISO.

**¿Para qué se utiliza el método de puntos de función?**

**Estimación de proyectos de software**

Una de las principales aplicaciones del método es en la determinación de [valoraciones (estimaciones) del producto de software a desarrollar](http://www.pmoinformatica.com/2014/05/pasos-para-elaborar-estimaciones-de.html), que es una parte fundamental de todo proceso de ingeniería de software y de la gerencia de estos proyectos.

Dentro del ciclo de ingeniería de software, lo normal es que la estimación del proyecto ocurra después que la ingeniería de requerimientos produzca una primera versión de la especificación de requisitos con suficiente información para elaborar la estimación.

**Validar la calidad de las especificaciones funcionales**

Si asignamos el mismo requisito a dos Analistas de sistemas y la estimación producida por los puntos de función difiere, esto nos puede servir para identificar ambigüedades en las especificaciones funcionales.

**Seguimiento y control de proyectos**

Otras aplicaciones de los puntos de función incluyen el seguimiento y control de proyectos, donde se puede usar para estimar el impacto de cambios en la funcionalidad solicitados durante el desarrollo y también para medir en puntos de función el avance del proyecto e inclusive la [facturación en contratos de servicios](http://www.pmoinformatica.com/2014/06/contratacion-de-servicios-de-software.html).

**Medir la productividad y calidad de nuestro proyecto o servicio de ingeniería de software**

Por otra parte, los puntos de función tienen aplicaciones en mediciones de productividad, por ejemplo si dividimos las horas hombre empleadas para producir cierta cantidad de puntos de función, obtenemos un indicador de productividad de nuestro equipo y de cómo esta puede mejorar en el tiempo.

Con esto adicionalmente podemos calibrar el factor de conversión entre puntos función y horas hombre que utilicemos para nuestras estimaciones.

Otros indicadores interesantes son número de incidencias por puntos de función producidos como medición de la calidad del proceso de desarrollo.

**Formación en Puntos de función**



Una de las actividades fundamentales en los proyectos de ingeniería de software, es el poder estimar el esfuerzo, medido en horas o jornadas que tomara el proyecto.

La técnica del análisis de Puntos de Función (FPA) es considerada la principal herramienta para la medición funcional de productos de software y de los procesos involucrados en su desarrollo.

Este curso esta dirigido a profesionales involucrados en proyectos y servicios de ingeniería de software, tales como [Arquitectos de software](http://www.pmoinformatica.com/2013/10/el-rol-del-arquitecto-de-software.html), Líderes técnicos, Gerentes de proyectos, Analistas de requerimientos, Desarrolladores, Coordinadores y Analistas de pruebas.

**¿En qué se diferencia estimar por puntos de función de estimar en base a la experiencia?**

Los puntos de función permiten estandarizar las mediciones del tamaño del software, obteniendo estimaciones de mayor exactitud, frente a las que obtendríamos al basarnos solamente en nuestra experiencia y aproximación inexacta.

Diferentes Analistas de sistemas que apliquen el método utilizando los mismos parámetros deberían llegar a conclusiones similares (siempre y cuando los requerimientos estén definidos sin ambigüedades), por lo cual la medición del tamaño del software es más objetiva y auditable.

De esta forma, es posible por ejemplo comparar la complejidad de dos funcionalidades, módulos o proyectos enteros de forma más confiable, permitiendo tomar decisiones informadas sobre como priorizarlos.

A la hora de desarrollar software empresarial, el éxito o fracaso del proyecto suele medirse en función del desempeño de este frente a las expectativas establecidas inicialmente, de allí la importancia de la exactitud de las mediciones de tamaño del software que realizamos inicialmente.

**¿Cómo se estima software con el método de puntos de función (Ejemplo)?**

Ahora pongamos un ejemplo sencillo de estimación de proyectos de software por puntos de función:

**Componentes de software a desarrollar**

Supongamos que como Analista de sistemas, has realizado la ingeniería de requisitos de una nueva funcionalidad para un sistema determinado. Estas nuevas funcionalidades son:

* Ingresar nuevos clientes en la aplicación.
* Modificar clientes existentes.
* Producir un listado de clientes.
* Producir un reporte con el número de clientes por país de origen.

Si quieres profundizar más sobre los métodos para identificar los componentes de software de una aplicación, te recomendamos el siguiente artículo de esta serie.

IFPUG-FPA define 5 tipos de componentes de software, que son: Archivo lógico interno, Archivo externo de interfaz, Entrada externa, Salida externa y Consulta externa.

A partir de la ingeniería de requisitos, procedes a definir los componentes de las funcionalidades a desarrollar, que son:



Las salidas externas y consultas externas se diferencian en que las salidas externas producen valor agregado (por ejemplo agrupan datos) mientras que las consultas externas solo toman el contenido de archivos internos y lo presentan (por ejemplo un listado).

**Nivel de complejidad de cada componente**

A continuación, debes asignar un nivel de complejidad para cada componente, estos niveles dependen de factores como por ejemplo número de campos no repetidos, número de archivos a ser leídos, creados o actualizados, número de sub grupos de datos o formatos de registros. A mayor número de factores mayor número de complejidad.

Los umbrales para pasar de un grado a otro son particulares a cada organizacion o empresa y deben ser analizados individualmente.

Supongamos que luego de evaluar estos factores, los niveles de complejidad fueron los siguientes:



**Asignar puntos función según nivel de complejidad**

Con los niveles de complejidad, puedo proceder a asignar los puntos de función no ajustados a cada uno, según la tabla de parámetros que estemos usando.

Por ejemplo podemos usar la siguiente tabla de parametros:



Según esta tabla, los puntos de función a asignar a cada funcionalidad son:



Entonces el número de puntos de función no ajustado es de **28**.

**Factor de ajuste según particularidades de la organización o empresa**

Por último, podemos aplicar un factor de ajuste, basado en 14 características generales de sistema definidas por el IFPUG-FPA.

Estas se especifican en una tabla de parámetros y para cada una estableceremos en qué medida contribuye al factor de ajuste total. Estos parámetros suelen depender de la complejidad de una aplicación, entorno técnico, exigencias no funcionales (rendimiento, mantenibilidad, etc.)

Volviendo a nuestro ejemplo, supongamos que para este proyecto y aplicación en particular hemos calculado un 10% de ajuste, esto significa que el resultado final en puntos función es:

28 +- 10%x28 es decir 28 +- 2,8 puntos de función.

El máximo de puntos de función que tendrá este desarrollo de software es de 30,8 y el mínimo es 25,2.

**Estimar horas hombre (o días hombre) a partir de los puntos de función**

Con los puntos de función puedes calcular las horas hombre aplicando un factor de conversión, pues no necesariamente un punto función equivale a una hora hombre.

Para determinar este factor de conversión necesitas disponer de datos históricos de productividad del equipo para producir cada funcionalidad.

Por ejemplo supongamos que hemos determinado que nuestra organización toma 3 horas en producir 1 punto de función, entonces:

28 puntos de función multiplicados por 3 horas resulta en 84 horas, o 11 días si consideramos 8 horas por día.

Tomando el factor de ajuste mencionado en el punto anterior, por medio del cual sabemos que según las particularidades de nuestra organización tenemos un margen de mas o menos 2,8 puntos de función,  entonces sabemos que nuestro desarrollo de software puede tomarnos 84 horas mas o menos 8,4 horas, u 11 días mas o menos 1 día.

Los puntos de función y su conversión a horas hombre los podemos tomar globales o individuales a cada funcionalidad, esto puede ser útil para elaborar las estimaciones de actividades, costes y presupuesto en los proyectos.

**¿En que me puede ayudar el método de puntos de función?**

Los puntos de función pueden ser de utilidad para todos los profesionales involucrados en un proyecto de [desarrollo de software](http://oficinaproyectosinformatica.blogspot.com/2012/08/algunas-practicas-de-desarrollo-de.html), por ejemplo:

Como **Analista de sistemas (o Ingeniero en sistemas)** pueden solicitarte estimaciones de esfuerzo, recursos y tiempo para desarrollar los requerimientos que estas analizando en tu ingeniería de requisitos. Los puntos de función pueden ayudarte a medir el tamaño del software de forma rápida y confiable, basadas en un modelo paramétrico y no subjetividades.

Como **desarrollador de software (o programador)** con frecuencia te preguntarán cuánto tiempo tomará realizar el desarrollo de uno o varios requisitos. Puedes usar el método de puntos de función para medir el tamaño del desarrollo a partir de sus componentes y funcionalidades. Conociendo el tamaño puedes decirle a tu supervisor, líder técnico o gerente cuanto tiempo tomará el desarrollo, y el número que proporciones será más confiable que el que hubieras producido con una apreciación general de la funcionalidad.

En la **Gestión de proyectos**, el método de puntos de función puede ayudar en la estimación de tiempos y costos. Los puntos función proporcionan una forma fiable de determinar el tamaño del software a partir de la definición de su alcance. Luego con estas mediciones puedes establecer la relación de puntos de función con los días que tomará desarrollar cada componente, produciendo así estimado de tiempo, costos y presupuesto.

Como software tester puedes usar los puntos de función para medir la complejidad de los casos de prueba, y contestar la pregunta de cuánto tiempo me tomará [ejecutar el software Testing](http://www.pmoinformatica.com/2014/02/pruebas-de-calidad-de-software-10-pasos.html) de unos requerimientos de terminados.

# Método de Estimación Puntos Casos de Uso (Use Case Points)

El método de Puntos Casos de Uso (Use Case Points) fue desarrollado en 1993 por *Gustav Kamer*, bajo la supervisión de *Ivar Jacobson* (creador de los casos de uso y gran promovedor del desarrollo de UML y el Proceso Unificado).

Este método ha sido ampliamente utilizado por la empresa Rational. Su principal ventaja es su rápida adaptación a empresas que ya estén utilizando la técnica de Casos de Uso.

Para el cálculo se procede de forma similar a Puntos de Función: se calcula una **cuenta no ajustada Puntos Casos de Uso (UAUCP)**, asignando una complejidad a los actores y a los casos de uso.

Esta complejidad será ponderada con un *Factor de Ajuste técnico* y por un *Factor de Ajuste relativo al entorno de implantación*, obteniendo tras ello una **cuenta de Puntos Casos de Uso Ajustados**.

Veamos a continuación en detalle los pasos del método:

* **1) Clasificar cada interacción entre actor y caso de uso según su complejidad y asignar un peso en función de ésta.** Para poder clasificar la complejidad de los actores debemos analizar la interacción de éste con el sistema que se va a desarrollar.
La complejidad de los actores puede corresponderse con una de las tres categorías posibles:
	+ *a. Simple.* Representa a otro sistema con una API definida. Se le asigna un peso de valor 1.
	+ *b. Medio.* Representa a otro sistema que interactúa a través de un protocolo de comunicaciones. Por ejemplo TCP/IP o a través de un interfaz por línea de comandos. Se le asigna un peso de valor 2.
	+ *c. Complejo.* La interacción se realiza a través de una interfaz gráfica. Se le asigna un peso de valor 3.

| **Tipo de Interacción** | **Peso Asignado** |
| --- | --- |
| **Simple** (a través de API) | 1 |
| **Media** (a través de protocolo) | 2 |
| **Compleja**(a través de interfaz gráfica) | 3 |

Tabla con los pesos en función de la complejidad de la interacción con los actores

* **2) Calcular la complejidad de cada caso de uso según el número de transacciones o pasos del mismo.** Para calcular la complejidad de un caso de uso debemos determinar el número de transacciones, incluyendo los caminos alternativos.

Se entiende por transacción a un conjunto de actividades atómicas, donde se ejecutan todas ellas o ninguna.
En función del número de transacciones que posee un caso de uso se clasifica el caso de uso como simple, medio o complejo, siendo la asignación de pesos la que se muestra en la tabla siguiente:

| **Nº de Transacciones del Caso de Uso** | **Tipo** | **Peso** |
| --- | --- | --- |
| menor o igual que 3 | **Simple** | 5 |
| mayor o igual que 4 y menor que 7 | **Medio** | 10 |
| mayor o igual que 7 | **Complejo** | 15 |

* **3) Calcular los Puntos Casos de Uso No Ajustados (UUCP) del sistema.** Se obtienen sumando los Puntos Casos de Uso de todos y cada uno de los actores y casos de uso que se han identificado y catalogado en función de su complejidad.
* **4) Cálculo de los Factores Técnicos (TCF).** A cada uno de los Factores Técnicos de la tabla siguiente se le asigna un valor de influencia en el proyecto entre 0 (no tiene influencia) a 5 (esencial), 3 se considera de influencia media.
Obtenidos los grados de influencia se multiplican por el peso de cada factor y con la siguiente fórmula se calcula el Factor Técnico que aplica:



| **Factor** | **Descripción** | **Peso** | **Influencia** |
| --- | --- | --- | --- |
| R1 | Sistema Distribuido | 2 | n |
| R2 | Objetivos de rendimiento | 1 | n |
| R3 | Eficiencia respecto al usuario final | 1 | p |
| R4 | Procesamiento complejo | 1 | q |
| R5 | Código reutilizable | 1 | r |
| R6 | Instalación sencilla | 0,5 | s |
| R7 | Fácil utilización | 0,5 | t |
| R8 | Portabilidad | 2 | u |
| R9 | Fácil de cambiar | 1 | v |
| R10 | Uso Concurrente | 1 | w |
| R11 | Características de seguridad | 1 | x |
| R12 | Accesible por terceros | 1 | y |
| R13 | Se requiere formación especial | 1 | z |

Tabla con los Factores Técnicos para el cálculo del TCF

* **5) Cálculo de los Factores de Entorno.** A cada uno de los Factores de Entorno de la tabla siguiente se le asigna un valor de influencia en el proyecto entre 0 (no tiene influencia) a 5 (esencial), 3 se considera de influencia media.
Obtenidos los grados de influencia se multiplican por el peso de cada factor y con la siguiente fórmula se calcula el Factor de Entorno que aplica:



| **Factor** | **Descripción** | **Peso** | **Influencia** |
| --- | --- | --- | --- |
| R1 | Familiar con RUP | 1,5 | n1 |
| R2 | Experiencia en la aplicación | 0,5 | n2 |
| R3 | Experiencia con orientación a objetos | 1,0 | n3 |
| R4 | Capacidades de análisis | 0,5 | n4 |
| R5 | Motivación | 1,0 | n5 |
| R6 | Requisitos estables | 2,0 | n6 |
| R7 | Trabajadores a tiempo parcial | -1,0 | n7 |
| R8 | Lenguaje complejo | -1,0 | n8 |

Tabla con los Factores de Entorno para el cálculo del EF

* **6) Obtención de los Puntos Casos de Uso Ajustados.** Una vez calculados los dos factores calculamos el valor ajustado de Puntos Casos de Uso con la siguiente fórmula:



Una vez obtenido el número de Puntos Casos de Uso, si se quiere obtener el esfuerzo necesario para llevarlos a cabo en el método se provee de un factor de productividad.
El autor propone un valor de 20 horas/persona aunque existen distintas propuestas sobre este valor.

Este esfuerzo calculado no abarcaría a todas las fases del proyecto sino únicamente a la codificación de los Casos de Uso no estando contemplada otras fases del desarrollo.

Por tanto, para calcular el esfuerzo total del proyecto habría que estimar el esfuerzo en realizar el resto de actividades del proyecto y sumarlas a las obtenidas por el método de Puntos Casos de Uso.

# Referencias

<http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>

<http://www.laboratorioti.com/2013/02/14/metodo-de-estimacion-puntos-casos-de-uso-use-case-points/>

<http://www.utim.edu.mx/~svalero/docs/id45.pdf>