

# METODOLOGÍA DE PRESUPUESTACIÓN DE SOFTWARE BASADO EN OBJETIVOS

*Data de aceite: 01/09/2023*

### **Luis Alberto Derico**

Ingeniero Informático. Universidad Católica “Nuestra Señora de la Asunción” Campus Itapúa.

**RESUMEN:** En la zona del sur de Itapúa y áreas circundantes se experimenta un crecimiento acelerado de las MIPYMES, los cuales al ser afectados por los sistemas formales de información y tributación electrónicas, además del creciente volumen de información que estas deben manejar, son un nicho de mercado ideal para el desarrollo de software a medida, donde los equipos de desarrollo por lo general se componen de una o dos personas.

Las características principales de estos desarrollos es que tienen objetivos bien definidos, y son restrictivos en cuanto a tiempo y presupuesto, por lo que una metodología que cubra la presupuestación al mismo tiempo que el desarrollo es necesaria para evitar malos entendidos entre clientes y desarrolladores, así como pérdidas en el presupuesto, infravaloración del trabajo o falta de tiempo de desarrollo e implementación.

En este trabajo propongo una metodología

orientada al objetivo, que se centre en un presupuesto inicial lo más exacto posible y acompañe en cada fase con productos específicos, que permitan la validación de los resultados del desarrollo y el control de calidad del software resultante a lo largo del proceso.

**PALABRAS-CLAVE:** Software-Desarrollo-Presupuesto-Metodología-Ingeniería.

## INTRODUCCIÓN

Al día de hoy, la Micro, Pequeñas y Medianas Empresas constituyen la porción mayoritaria del mercado paraguayo. Según datos del Viceministerio de MYPYMES, para el final del 2017 las MIPYMES conforman el 93% de las unidades productivas del Paraguay.

Con las nuevas disposiciones de la Subsecretaría de Estado de Tributación (SET), el cual está trabajando en un proceso de digitalización total de sus operaciones se hace necesario para las MIPYMES contar con el soporte digital correcto que les brinde las informaciones requeridas por los organismos de control.

Actualmente la SET maneja de forma electrónica los sistemas de información de operaciones gravadas (HECHAUKA), los sistemas de liquidación de operaciones gravadas por el IVA (MARANGATU), los sistemas de retención de IVA (TESAKA) y más recientemente las liquidaciones del Impuesto a la Renta Personal (ARANDUKA) y más recientemente la próxima implementación del sistema de e-facturas.

En este escenario los desarrolladores independientes de software deben adecuarse a las características de esta franja mayoritaria de mercado, para generar un producto que cumpla con las expectativas de los clientes para que el sistema les ayude a manejar organizativamente sus empresas; y por el otro lado que se realice la valuación del trabajo de forma correcta para asegurar la continuidad del prestador del servicio de desarrollo en el mercado, buscando un equilibrio entre un precio competitivo para los clientes y la rentabilidad necesaria para el desarrollador.

El mayor problema para los desarrolladores independientes es que no existe una metodología concreta que se adapte a este contexto y que cumpla con las métricas de aseguramiento de la calidad.

En este trabajo se definen las características principales que se deben tener en cuenta y se explica la propuesta metodológica ideada para acoplar todos estos aspectos.

## DEFINICIÓN DE MIPYMES

Según la ley N° 4457 sancionada por el Congreso de la Nación Paraguaya el 16 de mayo del 2012, las Micro, Pequeñas y Medianas Empresas (MIPYMES) “son todas las unidades económicas, que según la dimensión en que se organicen el trabajo y el capital, se encuentren dentro de las categorías establecidas en el Artículo 5° de esta ley y se ocupen del trabajo artesanal, industrial, agroindustrial, agropecuario, forestal, comercial o de servicio”.

En el Artículo 4° de esta misma ley categoriza a las MIPYMES por dos elementos:

- El número de trabajadores ocupados; y,
- El monto de facturación bruta anual, realizado en el ejercicio fiscal anterior.

En el Artículo 5° de la misma ley clasifica los parámetros de cada categoría y delimita el alcance para cada uno:

**Microempresas:** está formada por un máximo de 10 trabajadores, en la que el propietario o integrantes de su familia trabajan personalmente en el y facture anualmente un máximo de Gs. 500.000.000 (quinientos millones de guaraníes).

**Pequeña Empresa:** está formada por un máximo de 30 trabajadores y facture anualmente un máximo de Gs. 2.500.000.000 (dos mil quinientos millones de guaraníes).

**Mediana Empresa:** está formada por un máximo de 50 trabajadores y facture anualmente un máximo de Gs. 6.000.000.000 (seis mil millones de guaraníes).

## IDENTIFICACIÓN DE LAS RESTRICCIONES DE LAS MIPYMES

Las dos principales restricciones en cuanto al desarrollo de un proyecto de software son:

- **Tiempo Limitado:** cuando una microempresa o pequeña empresa decide implementar un software administrativo se debe generalmente a que el sistema administrativo y fiscal actual ya no cubre la demanda actual operativa, ya sea haciendo internamente los controles contables y administrativos o por profesionales tercerizados.

El volumen de operaciones de la empresa creció más de lo que su actual estructura soporta y se ve en la obligación de automatizar procesos para ahorrar tiempo.

Cómo estos cambios están sometidos al régimen fiscal, lo más probable es que el tiempo de desarrollo e implementación esperados sean menos de un año.

- **Presupuesto:** las MIPYMES no ven los proyectos de desarrollo de software como proyectos de inversión o como servicios, sino en forma de productos, lo que hace que esperen un precio fijo por determinadas características.

La empresa no espera ni concuerda en que en determinados casos el presupuesto puede aumentar a razón de cambios de última hora o por características que no habían sido consideradas al inicio del desarrollo del software. Lo que la empresa espera es que se pacte al inicio del proyecto un valor total y que se entreguen las funciones tal y como se establecieron, sin excepciones.

## MÉTRICAS DE SOFTWARE

Las métricas son estándares sistematizados que se aplican al software o a cualquier proceso para verificar que posee alguna propiedad específica, tanto para la medición de la productividad como para para el aseguramiento de la calidad del software.

Existen muchos estándares de métrica, pero no todas nos sirven para la presupuestación, ya que realizan la medición sobre el resultado, y no sobre el esfuerzo futuro de desarrollo e implementación del software.

Algunas de las métricas más conocidas que podemos citar son:

- Métrica ABC
- Grado de cohesión
- Modelo de costo constructivo (COCOMO)
- Puntos de función
- Número de líneas de código
- Tamaño de programa binario

## OBJETO DE CÁLCULO POR MÉTRICA

**Métrica ABC:** mide la complejidad de un software basado en tres variables principales, asignaciones, ramas y condicionales. El objeto de medición de la métrica es el software desarrollado, lo que hace que no sirva para presupuestar, ya que el sistema debe estar terminado para medirlo.

**Grado de cohesión:** sistema de medición ordinal que clasifica el software entre rasgos deseables e indeseables. El objeto de medición es el software desarrollado y el objetivo es medir la calidad del software, no el esfuerzo invertido en su creación.

**Modelo de costo constructivo (COCOMO):** es un modelo matemático estimativo donde se calcula el costo del producto final basado en el tamaño final y cantidad de archivos generados y otras características, pero no toma en cuenta la productividad y proporciona valores irreales al utilizar orientación a objetos, por las propiedades de herencia y reutilización, que afecta la cantidad final de código escrito. Su objeto de medición es incompleto, porque prevé el código generado pero no los recursos ni la productividad necesaria para crearlo.

**Puntos de función:** métrica que mide la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software. La forma de medición puede ser un poco empírica ya que depende de parámetros de ajuste aplicados por el que realiza la medición. Su objeto de medición son las características del software a desarrollar, pero es muy subjetivo en su aplicación y puede dar valores incorrectos en su aplicación.

**Número de líneas de código:** pretende calcular el costo de un software directamente por la cantidad de código generado. El objeto de medición es el software desarrollado impidiendo que se pueda usar para presupuestar antes de desarrollar el producto.

**Tamaño de programa binario:** Como el anterior, pero en vez de calcular por líneas de código mide el peso de los archivos binarios generados por el software. El objeto de medición es el software desarrollado impidiendo que se pueda usar para presupuestar antes de desarrollar el producto.

## SELECCIÓN DE MÉTRICA

Como se ve en la lista anterior, las opciones más recomendables son las que permitan calcular el esfuerzo antes de desarrollar el software, dejando como la opción más viable la métrica por Puntos de función, ajustando las variables de la forma más objetiva posible para adecuar al entorno actual.

## PROPUESTA DE PRESUPUESTACIÓN EN LA ETAPA DE ANÁLISIS DE REQUISITOS

Las fases en las que se divide el proceso de análisis de requisitos y presupuestación son las siguientes:

1. Libro de Requisitos
  - a. Definición de los objetivos principales y secundarios de un proyecto
  - b. Definición de requisitos funcionales y no funcionales necesarios para que el sistema cumpla con los objetivos
2. Valoración por puntos de función
  - a. Cálculo de punto de función por cada requisito funcional y no funcional
  - b. Determinación del peso en horas de los puntos de función
  - c. Determinación del valor por hora en relación al equipo de trabajo
  - d. Cálculo de costo financiero y de tiempo de trabajo del proyecto
3. Presupuesto
  - a. Determinación de la inversión necesaria
  - b. Cláusulas de servicios incluidos y no incluidos en los presupuestos
  - c. Alcance
  - d. Tiempo de desarrollo e implementación
  - e. Garantías
  - f. Cláusula de protección del desarrollador y cliente
  - g. Acuerdo de inicio de actividades

### LIBRO DE REQUISITOS

El primer paso para determinar el esfuerzo necesario para desarrollar un software es identificar cuáles son las funciones que el mismo debe cumplir. Este paso se debe ejecutar después de una recolección inicial de datos y las entrevistas necesarias con los futuros usuarios y responsables del software para aclarar los resultados esperados del mismo.

#### Definición de los objetivos principales y secundarios de un proyecto

Se deben definir exactamente cuál será la función principal de un sistema, y una vez logrado, enumerar las actividades necesarias para que se pueda cumplir ese objetivo central.

## **Definición de requisitos funcionales y no funcionales necesarios para que el sistema cumpla con los objetivos**

Una vez definidos los objetivos, deben identificarse cuáles son las funciones que necesitan ser desarrolladas para cumplir con estos, y se separan en dos grupos dependiendo de su impacto en los objetivos.

Los requisitos funcionales son las funciones que intervienen directamente en el cumplimiento de los objetivos propuestos.

Los requisitos no funcionales son los que no intervienen para el cumplimiento de los objetivos, pero que son necesarios para el funcionamiento del software.

## **VALORACIÓN POR PUNTOS DE FUNCIÓN**

Cuando termina la identificación de los requisitos, podemos pasar a la métrica de valoración de los mismos. Utilizando los puntos de función, usando el método de análisis de punto de función, asignamos a cada requisito una cantidad de “puntos” que luego convertiremos a una escala más cuantificable.

El primer paso consiste en contar las funciones de datos, desde el punto de vista del usuario. Se identifican los índices de complejidad y posteriormente su contribución en puntos. La forma más sencilla es identificar una entidad del sistema y asignar un punto por cada operación sobre los datos, a saber; alta, baja, modificación y consulta.

Luego se cuentan las funciones transaccionales, como cálculos adicionales, aplicación de fórmulas o informes.

Seguidamente se recuentan los puntos de función y se les aplica un factor de ajuste, que depende de 14 características del software:

1. Comunicación de datos
2. Procesamiento distribuido
3. Performance (desempeño)
4. Configuración del equipamiento
5. Volumen de transacciones
6. Entrada de datos on-line
7. Interfase con el usuario
8. Actualización on-line
9. Procesamiento complejo
10. Reusabilidad
11. Facilidad de implementación
12. Facilidad de operación

13. Múltiples locales

14. Facilidad de cambios

Para esta propuesta se considera que estos factores de ajuste no son relevantes en este proceso de presupuestación y se tomará un ajuste de 1, ya que el cumplimiento de los objetivos debe ser total y se trata de equipos de trabajo pequeños o unipersonales.

### **Cálculo de punto de función por cada requisito funcional y no funcional**

Se confecciona una lista de requisitos funcionales y no funcionales, se les asigna los puntos de función correspondiente y se realiza el recuento final.

### **Determinación del peso en horas de los puntos de función**

En este punto, se realiza una valoración directa de un requisito funcional, realizando en la práctica la ejecución del requisito cronometrando el tiempo invertido, dividiéndolo por los puntos de función asignados para obtener el tiempo que se invierte en un punto de función.

### **Determinación del valor por hora en relación al equipo de trabajo**

Para valorizar las horas de trabajo estimadas para completar los requisitos del software, debe calcularse el costo de trabajo de cada hora invertida. para realizar esto se propone realizar un estimado del valor basado en un año laboral normal, compensando los días libres, tiempo muerto y gastos operativos (calculado sobre la premisa de que el desarrollo se realiza por un equipo que no trabaja en relación de dependencia a una entidad).

### **Cálculo de costo financiero y de tiempo de trabajo del proyecto**

En este punto se realiza la multiplicación de las horas de trabajo por el valor por hora calculado, y se tendrá el costo aproximado del esfuerzo necesario para desarrollar el sistema y la cantidad de tiempo necesario para llevarlo a cabo.

## **PRESUPUESTO**

Con los datos anteriores ya calculados, se puede redactar el presupuesto y presentarlo al responsable del software. Las secciones que debe tener un presupuesto son las siguientes:

## **Funcionalidad y determinación de la inversión necesaria**

Un listado detallado de los requisitos funcionales y no funcionales del sistema, más su valorización global o detallada.

## **Cláusulas de servicios incluidos y no incluidos en los presupuestos**

Delimitación de responsabilidades del equipo de desarrollo y enumeración de actividades previstas y actividades que no son responsabilidad del proyecto de desarrollo.

## **Alcance**

Cual es el marco de trabajo del proyecto, y hasta donde van a llegar las actividades presupuestadas.

## **Tiempo de desarrollo e implementación**

Estimación de tiempo de trabajo total. Cuando se incluye la implementación, los requisitos funcionales deben estar agregados a los del desarrollo del presupuesto.

## **Garantías**

Descripción de garantías sobre las actividades de desarrollo previstas.

## **Cláusula de protección del desarrollador y cliente**

Cláusulas generales de previsión de terminación de actividades, tanto del lado del equipo de desarrollo como el de los responsables del software.

## **Acuerdo de inicio de actividades**

Una sección para marcar el inicio oficial de los trabajos, necesario además para la evaluación del cumplimiento y aplicación de garantías.

## **EJEMPLO DE APLICACIÓN DE LA METODOLOGÍA PROPUESTA**

### **Caso de estudio**

Evaluar el desarrollo de un sistema para un estudio jurídico:

**Objetivo Principal:** gestionar las cuentas corrientes y operaciones administrativas de clientes de un estudio jurídico asignadas a los asociados.

### Objetivos Secundarios:

- Mantener un registro de los clientes
- Listar los servicios prestados por los asociados
- Registrar los costos directos
- Registrar los costos indirectos
- Mantener la cuenta corriente de los clientes
- Facturar los servicios

### Identificación de requisitos funcionales y no funcionales

De los objetivos propuestos, podemos identificar la lista de requisitos funcionales y no funcionales del proyecto:

Nro.	Requisitos	Tipo
1	ABMC asociados	funcional
2	ABMC clientes	funcional
3	ABMC servicios	funcional
4	costos directos	funcional
5	costos indirectos	funcional
6	informe - cta cte cliente	funcional
7	ABMC factura	funcional
8	ABMC usuarios	no funcional
9	funciones	no funcional
10	ABMC permisos	no funcional
11	instalación BD	no funcional
12	instalación servidor aplicación	no funcional

## Cálculo de puntos de función

Nro.	Requisitos	PF de Datos	PF transaccionales
1	ABMC asociados	5	1
2	ABMC clientes	5	1
3	ABMC servicios	5	1
4	costos directos	8	2
5	costos indirectos	6	2
6	informe - cta cte cliente	4	2
7	ABMC factura	7	2
8	ABMC usuarios	5	1
9	funciones	2	0
10	ABMC permisos	6	1
11	instalación BD	0	1
12	instalación servidor aplicación	0	3

Los ABMC pueden considerarse con 6 puntos totales, 4 corresponden a procesos de datos de alta, baja, modificación y consulta; un punto para la interfaz y un punto por la creación de la entidad.

Funciones como la de costos directos implican los mismos 4 puntos de ABMC, pero adicionalmente necesita consultar a las entidades de asociados, clientes, y servicios; en la categoría transaccional se agregan a la interfaz un posible cálculo de subtotales.

Los informes agregan puntos de datos por consulta, más los transaccionales de filtrado y formato de impresión.

Los requisitos no funcionales se evalúan de la misma manera que los funcionales, aunque no impacten directamente en los objetivos.

Trabajos adicionales como instalación de servicios o desarrollo del sistema base deben ser evaluados desde la perspectiva de las actividades necesarias para su puesta a punto, por ejemplo, la instalación de la base de datos en este caso solo un punto, por ser un solo servicio. Por otro lado, el servidor de aplicación estaría compuesto por un servidor HTTP, y un gestor de modelo, más el template básico del sistema.

Para simplificar, no se aplicará el valor de ajuste de los puntos de función, dejando el índice de ajuste en 1.

En total estas funciones acumulan 70 puntos.

## DETERMINACIÓN DEL PESO EN HORAS POR PF

La determinación del peso en horas dependerá de la cantidad de personas del equipo y de la experiencia de las mismas con las herramientas tecnológicas a utilizar.

Como ejemplo, partiremos de la premisa que el desarrollador tarda 12 horas en desarrollar un ABMC completo y funcional desde la creación de la base de datos hasta la implementación de la interfaz, lo que nos daría un valor de base de 72 minutos de trabajo por punto de función. (1 punto de instalación de la base de datos, 3 puntos de instalación de software necesario, 6 puntos para el ABMC completo).

### Determinación del valor por hora

Para la segunda parte de la aplicación de la metodología, necesitamos el valor monetario aplicable por cada hora de trabajo. La determinación de este valor se puede determinar con el siguiente procedimiento:

1. Se debe determinar la cantidad de horas máximas que puedan trabajarse en un periodo de un año: Horas por día x Días laborales x 4 semanas x 12 meses =  $8 \times 4 \times 12 = 2.304$  horas anuales.
2. Se calcula el valor de cada hora sobre un salario mínimo vigente: salario mínimo x 12 meses / Horas totales anuales =  $2.550.307 \times 12 / 2.304 = 13.283$  G. por hora.
3. Se calculan los días libres por año en una ponderación: feriados anuales + días libres + emergencias =  $11 + 5 + 5 = 21$  días por año.
4. También debe hacerse un ajuste de “tiempo muerto”, que corresponden a horas de trabajo que no se efectivizan, ya sea por movilidad, preparaciones, o cualquier tiempo laboral que no se trabaja en realidad. Este índice puede variar pero inicialmente se puede aplicar un 10% de las horas totales:  $2.304 \times 10\% = 230$  horas.
5. Es necesaria una suma de los costos operativos básicos del equipo de desarrollo, que deben costearse durante el año, que varían de acuerdo a cada caso. Un ejemplo básico puede ser: (alquiler + celular + servicio de internet + insumos) \* 12 meses =  $(800.000 + 120.000 + 130.000 + 100.000) \times 12 = 13.800.000$  G. por año.
6. Se debe de evaluar el costo por inactividad y el costo del tiempo muerto más los gastos operativos anuales, y ponderarlo al costo por hora por año:  $((\text{Días libres por año} \times 8 \times \text{Valor por hora}) + (\text{tiempo muerto} \times \text{Valor por hora}) + \text{gastos operativos anuales}) / \text{horas máximas anuales} = ((21 \times 8 \times 13.283) + (230 \times 13.283) + 13.800.000) / 2.304 = 8.284$  G. adicionales por hora.
7. Para el cálculo final de la hora de trabajo, se suman el costo base de hora + la ponderación de los costos anuales + un porcentaje de rentabilidad + impuestos =  $13.283 + 8.284 + 50\% + 10\% = 35.585$  G. por hora de trabajo.

## **Cálculo de costo financiero y tiempo de trabajo**

Con las variables ya calculadas puede determinarse que este proyecto tendrá un costo de  $((70 \text{ PF} * 72 \text{ min}) / 60 \text{ min}) * 35.585 \text{ G.}$ , que da un valor de 2.989.140 G.

El tiempo de trabajo total será de 84 horas u 11 días hábiles.

## **CONCLUSIONES**

El modelo propuesto busca equilibrar la complejidad de evaluar el esfuerzo de desarrollo con la valoración más justa del esfuerzo de trabajo en desarrollo de software, donde se pague un valor realista por los servicios recibidos y que sea factible para que el equipo de desarrollo pueda recibir una digna compensación por su trabajo.

Se toma como parámetro el mercado paraguayo en general y la zona sur de paraguay en particular, orientando hacia el mercado de PYMES y MIPYMES.

Para el cálculo de esfuerzo, se asume que el desarrollo será realizado por un equipo de profesionales independientes o un solo profesional que cumplirá con todos los trabajos del proyecto.