

### 5 ANÁLISIS COSTO BENEFICIO

#### 5.1 INTRODUCCIÓN

El objetivo del presente capítulo es demostrar que el uso del sistema desarrollado se obtendrá beneficios para el Ministerio de Obras Públicas, Servicios y Vivienda en los aspectos económicos, ecológicos y social.

Para demostrar los beneficios que se obtendrán se hará uso de COCOMO II para calcular el VAN (Valor Actual Neto) que nos ayudara a estimar el costo del sistema desarrollado, basado en el tamaño del mismo. De esta forma con los resultados obtenidos podemos afirmar que el proyecto es viable y comprobar que la mejor opción es invertir en el proyecto.

Además de calcular el VAN, calcularemos también el C/B (Costo Beneficio) y el TIR (Taza Interna de Retorno).

#### 5.2 COCOMO II (Constructive Cost Model)

Es un modelo que surge como respuesta a las falencias de COCOMO original (COCOMO 81) y a la necesidades planteadas por un nuevo escenario para el desarrollo de software (cambios tecnológicos, nuevos modelos del ciclo de vida, nuevas herramientas, reingeniería, generadores de aplicaciones, enfoque orientado a objetos, etc.) proponiendo principalmente el uso de diferentes modelos de tamaño según el avance del proyecto y el nivel de conocimiento del sistema en desarrollo.

En concreto COCOMO 2.0 provee tres modelos de estimación secuenciales que corresponden a las siguientes etapas:

- Etapa 1 o Application Composition Model orientada a apoyar la estimación en las etapas tempranas del desarrollo de software, donde básicamente se cuenta con escasa información sobre los requerimientos del sistema. Este modelo además apoya la estimación para el desarrollo de prototipos rápidos en cualquier etapa del ciclo de vida del software.
- Etapa 2 o Early Design Model enfocada a apoyar la estimación tras la definición de los requerimientos y así respaldar la toma de decisiones sobre la evaluación de arquitecturas y conceptos de operación alternativos para el desarrollo de software.
- Etapa 3 o Post-Architecture Model orientada a apoyar la estimación durante el desarrollo del sistema.

Cada modelo se subdivide en los siguientes modos:

- **Modo orgánico:** En este modo, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía de unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles de líneas (medio), mientras que en los otros dos modos el tamaño varía de pequeño a muy grandes (varios cientos de miles de líneas). En este modo, al igual que en los otros, el coste se incrementa a medida que el tamaño lo hace, y el tiempo de desarrollo se alarga.
- **Modo empotrado:** En este modo, el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con el procesador y el interface hardware. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.
- **Modo semi encajado:** Es un modo intermedio entre los dos anteriores, dependiendo del problema, el grupo puede incluir una mezcla de personas experimentadas y no experimentadas.

Cada uno de estos modos ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software (básico, intermedio y detallado). Para obtener una primera aproximación rápida del esfuerzo del personal en el presente proyecto se hace uso del modelo básico, utilizando su tabla de constantes (Tabla 5.1).

<b>Proyecto de Software</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Orgánico	2.40	1.05	2.50	0.38
Semi encajado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

**Tabla 5.1** Constantes para desarrollo básico COCOMO II

**Fuente:** [Pressman, 2002]

### **5.3 COSTO DEL SISTEMA**

Para determinar el costo total del proyecto se tomara en cuenta los siguientes costos:

- Costo de desarrollo del software
- Costo de elaboración del proyecto

#### **5.3.1 Costo de desarrollo del software**

Para lograr determinar el costo del desarrollo del software, se realizará el cálculo de puntos

de función no ajustados (Tabla 5.2), los cuales nos servirán para medir el tamaño del sistema en términos de la cantidad de funcionalidad del sistema.

<b>Categoría</b>	<b>Cantidad</b>
Entradas	3
Salidas	3
Interacciones o consultas	2
Interfaces externas	5
Archivos internos	2

**Tabla 5.2** Categorías para puntos de función

**Fuente:** Elaboración propia

Donde:

- Entradas de Interfaz de usuario de archivos, base de datos y otros sistemas.
- Salidas de la interfaz de usuario como ser archivos, base de datos y otros sistemas.
- Interacciones o consultas de la interfaz de usuario a archivos o base de datos y viceversa.
- Interfaces externas, integración con otras aplicaciones externas al sistema, como: bases de datos y otros.
- Archivos internos, integración con fuente de datos internas.

Para obtener el PF (Function Points) los datos obtenidos anteriormente (Tabla 5.2), se vuelven a clasificar según su complejidad, asignando valores que van de 3 a 15 respectivamente (Tabla 5.3).

<b>Categoría</b>	<b>Cantidad total</b>	<b>Simples</b>	<b>Promedio</b>	<b>Complejos</b>	<b>Puntos de función</b>
Entradas	9	3x3	4x7	2x15	67
Salidas	3	0x3	2x7	1x15	29
Interacciones o consultas	5	0x3	2x7	3x15	59
Interfaces externas	7	5x3	0x7	2x15	45
Archivos internos	3	0x3	2x7	1x15	29

<b>Total FP</b>	<b>** Expresión incorrecta **</b>
-----------------	---

**Tabla 5.3** Clasificación de complejidad de categorías

**Fuente:** Elaboración propia

Se realiza el ajuste de la complejidad (Tabla 5.4) usando la escala desde 0 (no importante o aplicable) a 5 (absolutamente esencial).

<b>Factor</b>	<b>Valor</b>
Copia de seguridad y recuperación	5
Comunicación de datos	3
Proceso distribuido	3
Rendimiento crítico	3
Entorno operativo existente	5
Entrada de datos en línea	0
Transacciones de entradas en múltiples pantallas	2
Archivos maestro actualizados en línea	3
Complejidad de valores del dominio de información	2
Complejidad del procesamiento interno	5
Código re-diseñado para la re-utilización	3
Conversión – instalación en diseño	2
Instalaciones múltiples	5
Aplicación diseñada para el cambio	3
<b>Total (N)</b>	<b>** Expresión incorrecta **</b>

**Tabla 5.4** Determinación de complejidad

**Fuente:** Elaboración propia

Se calcula el ajuste de complejidad:

$$CAF = 0.65 + 0.01 * N$$

$$CAF = 0.65 + 0.01 * 44$$

$$CAF = 1.09$$

obteniendo los puntos de función:

$$PF = \text{Cuenta Total} * \text{Factor de ajuste}$$

$$PF = 229 * 1.09$$

$$PF = 249.61$$

Ahora se convierte los puntos de función a miles de líneas de código (KLDC) tomando en cuenta la siguiente tabla .

Lenguaje	Nivel	Factor LDC/PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Visual Basic	7	46
ASP	9	36

**Tabla 5.5** Conversión de punto función a KLDC

**Fuente:** [Pressman, 2002]

Reemplazamos los datos en la ecuación:

$$LDC = PF * \frac{\text{Factor LDC}}{PF}$$

$$LDC = 249.61 * 53$$

$$LDC = 13229.33$$

convertimos a KLDC:

$$KLDC = \frac{13229.33}{1000}$$

$$KLDC = 13.23$$

Realizamos el calculo del esfuerzo necesario para la programación del sistema, utilizando el método orgánico:

$$E = a * KLDC^b$$

$$E = 2.4 * 13.23^{1.05}$$

$$E = 36.13$$

reemplazando los datos d y c para encontrar D en la siguiente ecuación:

$$D = c * E^d$$

$$D = 1.05 * 36.13^{0.38}$$

$$D = 4.10$$

Se concluye que el desarrollo del proyecto es de 4 meses aproximadamente.

Para calcular la cantidad de programadores utilizamos la siguiente fórmula:

$$P = \frac{E}{D}$$

$$P = \frac{36.13}{4.10}$$

$$P = 8.81$$

Para el desarrollo del proyecto se necesitan 9 programadores.

El salario promedio de un programador es entre 200 a 300 \$us, tomando como referencia estos datos, obtenemos el promedio de 250 \$us, de esta forma calculamos el costo total del desarrollo del software:

$$\text{Consto mensual de desarrollo} = P * \text{saladio promedio}$$

$$\text{Consto mensual de desarrollo} = 9 * 250$$

$$\text{Consto mensual de desarrollo} = 2250 \$us$$

Como el tiempo de desarrollo del software es 4 meses, tenemos:

$$\text{Costo total de desarrollo} = \text{Costo del software por mes} * \text{Número de meses}$$

$$\text{Costo total de desarrollo} = 2250 * 4$$

$$\text{Costo total de desarrollo} = 9000 \$us$$

### 5.3.2 Costo de elaboración del proyecto

Los costos de la elaboración de proyecto se refieren a los costos del estudio del sistema, en la etapa de análisis y recopilación (Tabla 5.6).

Detalle	Importe (\$us)
Análisis y diseño del proyecto	400
Material de escritorio	100
Conexión a Internet	100
Otros	50
<b>Total</b>	<b>650</b>

**Tabla 5.6** Costo de elaboración de proyecto

*Fuente: Elaboración propia*

### 5.3.3 Costo total del software

Se obtiene el costo total de la suma de los costos vistos anteriormente (Tabla 5.7):

Detalle	Importe (\$us)
Costo de desarrollo	9000
Costo de elaboración de proyecto	650
<b>Total</b>	<b>9650</b>

*Tabla 5.7 Costo total de software*

*Fuente: Elaboración propia*

## 5.4 VALOR ACTUAL NETO

El VAN (Valor Actual Neto) es el balance neto de todos los ingresos y gastos del proyecto ponderados en el tiempo.

La formula que se utilizara para hallar el VAN es la siguiente:

$$VAN = \sum \frac{Ganancias}{(1+k)^n} - \sum \frac{Costos}{(1+k)^n}$$

Donde:

**Ganancias:** Ingreso de flujo anual

**Costos:** Salidas de flujo anual

**n:** Numero de periodo

**k:** Tasa de descuento o tasa de interés al préstamo

Para verificar si un proyecto es rentable utilizaremos la siguiente tabla:

Valor VAN	Interpretación
$VAN > 0$	Proyecto rentable
$VAN = 0$	Proyecto rentable
$VAN < 0$	Proyecto no rentable

*Tabla 5.8 Interpretación VAN*

*Fuente: Elaboración propia*

Para estimar los gastos y ganancias se usara la tasa de descuento del 12%, esta tasa de

interés es la que actualmente usan las entidades bancarias. La estimación que se realizara para el lapso de 4 años (Tabla 5.9). Para las entradas de costos anuales se tomaron en cuenta los gastos relacionados con el uso de papel en la institución.

<b>Año</b>	<b>Costos</b>	<b>Ganancias</b>	<b><math>\frac{Costos}{(1+i)^n}</math></b>	<b><math>\frac{Ganancias}{(1+i)^n}</math></b>	<b>Resultado</b>
1	9650	0	8616	0	-8616
2	800	4500	637	3587	2950
3	500	4800	355	3416	3061
4	200	5000	127	3177	3050
$\Sigma$	** Expresión incorrecta **	** Expresión incorrecta **	** Expresión incorrecta **	** Expresión incorrecta **	
<b>VAN</b>					<b>445</b>

**Tabla 5.9** *Calculo de VAN*

**Fuente:** *Elaboración propia*

Por los resultados obtenidos de VAN (Tabla 5.9), se puede afirmar que el desarrollo del **proyecto es rentable**.

## 5.5 TASA INTERNA DE RETORNO

La tasa de retorno se obtiene cuando el valor del VAN toma un valor igual a 0, la TIR es la rentabilidad que nos proporciona el proyecto.

La ecuación de la TIR es:

$$TIR = \sum \frac{Ganancias - Costos}{(1-i)^n}$$

Donde:

**Ganancias:** Flujo de entrada de un periodo

**Costos:** Flujo de salida de un periodo

**i:** Tasa de interés de ahorro

**n:** Número de periodo



Reemplazando los datos obtenemos la TIR (Tabla 5.10):

Año	Costos	Ganancias	$\frac{\text{Ganancias} - \text{Costos}}{(1-i)^n}$
1	9650	0	-9368
2	800	4500	3487
3	500	4800	3935
4	200	5000	4264
<b>TIR</b>			<b>2318</b>

**Tabla 5.10** Tasa Interna de Retorno

**Fuente:** Elaboración propia

De esta forma podemos concluir que el proyecto nos dará una rentabilidad de 2318 \$us y en el caso de haber ahorrado la inversión inicial de 9650 \$us en una entidad financiera (interés de 3%), al cabo de 4 años se contaría con la suma de 10808 \$us, con lo cual se obtendría una ganancia de 1158 \$us. Por los datos obtenidos, se puede apreciar que la rentabilidad es superior a la inversión realizada en ahorro, afirmando que es conveniente invertir en el proyecto.

## 5.6 COSTO BENEFICIO

El análisis costo-beneficio es una herramienta financiera que mide la relación entre los costos y beneficios asociados a un proyecto de inversión con el fin de evaluar su rentabilidad. Para hallar el costo – beneficio aplicamos la siguiente ecuación:

$$C/B = \frac{\sum \text{Ganancias}}{\sum \text{Costos}}$$

Reemplazando los datos obtenemos:

$$C/B = \frac{14300}{11150}$$

$$C/B = 1.28$$

El resultado se lo representa de la siguiente forma: Por cada dólar invertido en el proyecto, la unidad tiene una ganancia de 1.28 \$us.