

60521
117



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE QUÍMICA

ENALFANCO Y PÉREZ ESCOBEDO
FAC. DE QUÍMICA

**LA EVALUACIÓN ECONÓMICA DE PROYECTOS CON BENEFICIO.
DESARROLLO DE UNA APLICACIÓN DE CÓMPUTO EN
LENGUAJE VISUAL BASIC.**

TESIS

Que para obtener el título de

INGENIERO QUÍMICO

Presenta

JORGE HERNANDO PÉREZ ESCOBEDO

México D.F.

2003



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Jurado asignado:

Presidente	Prof. José Antonio Ortiz Ramírez
Vocal	Prof. Celestino Montiel Maldonado
Secretario	Prof. Pedro Roquero Tejeda
1^{er}. Suplente	Profa. María Rafaela Gutiérrez Lara
2^o. Suplente	Prof. Eduardo Flores Palomino

Sitio donde se desarrolló el tema:

Laboratorio de simulación y optimización de procesos.
Departamento de Ingeniería Química.
Edificio E, Facultad de Química, UNAM.

Asesor del tema:



I. Q. Celestino Montiel Maldonado

Sustentante:

Jorge Hernando Pérez Escobedo

***A Olivia, Jorge Carlos y Mariana,
porque solo con ellos tiene sentido.
A mis padres.***

INDICE.

Capítulo 1. Introducción

Capítulo 2. Elección de negocios y proyectos con beneficio. La Evaluación económica de Proyectos

- 2.1 Definición de proyecto.
- 2.2 Definición de beneficio.
- 2.3 El valor del dinero en el tiempo.
- 2.4 El proceso de elección. *Rentabilidad de un proyecto.*

Capítulo 3 Rentabilidad y riesgo.

- 3.1 Relación entre la rentabilidad y el riesgo de un negocio.
- 3.2 Vida económica de un proyecto.
- 3.3 Costo de capital (tasa de Interés)
- 3.4 Flujos de efectivo.
 - 3.4.1 Flujos de Efectivo Netos.
 - 3.4.2 Flujos financieros
 - 3.4.3 Flujos operativos

Capítulo 4. Medidas de rentabilidad.

- 4.1 El Periodo de Recuperación Ajustado de la Inversión.
- 4.2 El Valor Presente Neto.
- 4.3 La Tasa Interna de Rendimiento

Capítulo 5. La herramienta de programación.

- 5.1 Fundamentos de Visual Basic.
- 5.2 La programación orientada a objetos.
- 5.3 La programación en Visual Basic.
- 5.4 Estructura de Visual Basic.
- 5.5 Elementos del lenguaje.
- 5.6 Procedimientos
- 5.7 Proyectos.
- 5.8 Estrategia de programación con Visual Basic.

Capítulo 6. Descripción del Programa.

- 6.1 Introducción.
- 6.2 Formularios.
 - 6.2.1. Inicio.
 - 6.2.2. Inversión Neta.
 - 6.2.3. Ingresos.

- 6.2.4. Egresos.
- 6.2.5. Flujo de Efectivo.
- 6.2.6. Rentabilidad.
- 6.2.7. ModuleInv.
- 6.3 Variables
- 6.4 Controles
- 6.5 La captura de datos.

Capítulo 7. Ejemplos

Capítulo 8. Conclusiones

Anexo 1. Glosario.

Anexo 2. Bibliografía.

Capítulo 1. Introducción.

Las dos principales decisiones a las que se enfrenta un inversionista, desde un punto de vista financiero, son la **decisión de inversión** y la **decisión de financiamiento**.

La **decisión de inversión** se refiere a seleccionar, de entre un conjunto de alternativas, la cantidad de recursos que deben comprometerse en activos, así como a determinar la composición de activos que deben adquirirse con dichos recursos.

La **decisión de financiamiento** apunta a la identificación de los individuos o instituciones dispuestas a aportar recursos, así como a la determinación del monto de recursos que debe obtenerse de cada una de las posibles fuentes.

La decisión de inversión no es independiente de la decisión de financiamiento, sin embargo, es conveniente separarlas con objeto de identificar las características y elementos específicos asociados a cada una de ellas.

La decisión de inversión es una de las más retadoras y complejas que deben realizar los empresarios. Su complejidad radica, fundamentalmente, en el hecho de que no existen dos decisiones de inversión idénticas. Cada decisión de inversión conlleva elementos específicos que la caracterizan y separan del resto de las decisiones similares que un inversionista deba enfrentar.

El presente trabajo expone de manera ordenada una metodología rigurosa de análisis para la toma de decisiones de inversión. Esta se compone, principalmente, de un conjunto de herramientas y técnicas de estudio y valuación orientadas a proporcionar al empresario o inversionista criterios generales que normen su toma de decisiones y le permitan optimizar el proceso de asignación de recursos. Así mismo se desarrolla en Visual Basic un programa de cómputo para el cálculo de los principales parámetros asociados al proceso de toma de decisiones.

Debe mencionarse que el cálculo de indicadores de rentabilidad que se incluyen en el programa puede encontrarse en cualquier hoja de cálculo electrónica. En ello no hay nada nuevo. Sin embargo consideramos que el uso de esta herramienta tiene tres ventajas:

1. Conduce de manera ordenada al tomador de decisiones, a través del proceso de captura de datos, brindándole una mejor percepción del proceso.
2. No requiere conocimiento alguno en el manejo de hojas de cálculo. Ello, aunque parezca de poca importancia, en la práctica es una dificultad que enfrentan algunos empresarios.

3. La aplicación puede extenderse para vincularse a datos provenientes de estados financieros.

Capítulo 2. Elección de negocios y proyectos con beneficio. La Evaluación económica de Proyectos.

2.1 Definición de proyecto.

Los empresarios se caracterizan, entre otras cosas, por su constante búsqueda de oportunidades productivas. Por oportunidades productivas entendemos la utilización de los recursos económicos de que dispone un negocio, con objeto de producir un bien o prestar un servicio.

La principal característica de los recursos económicos es que son recursos escasos, es decir, son recursos cuya disponibilidad es limitada. Recursos económicos son, entre otros, el capital (dinero), la mano de obra, la tierra, la maquinaria, el tiempo y los conocimientos. Estos recursos, cuando se utilizan para la producción de un bien o la prestación de un servicio, se convierten en factores o medios de producción.

Al acto de comprometer recursos económicos para desarrollar oportunidades productivas, es decir, para realizar la producción de un bien o la prestación de un servicio, se le denomina inversión y; al proceso de asignar dichos recursos entre las diferentes alternativas de uso de los mismos, se le denomina toma de decisiones de inversión.

Debido a que un negocio dispone únicamente de una cantidad limitada de recursos para desarrollar oportunidades productivas, es muy importante contar con una metodología que permita optimizar el uso de sus recursos.

Un proyecto es un plan de acción para la utilización productiva de los recursos económicos de que dispone un negocio.

Los proyectos son las unidades mínimas susceptibles de análisis financiero. Si bien el propósito de las técnicas de evaluación de negocios y proyectos no es el de ayudar al empresario a identificar oportunidades productivas, si pretenden proporcionarle una metodología para optimizar el uso de los recursos económicos de que dispone un negocio, a través de utilizar criterios específicos para medir el resultado económico asociado con el desarrollo de un determinado proyecto.

Generalmente, un negocio se conforma por un portafolio o conjunto de proyectos de inversión. Sin embargo, a lo largo de esta tesis utilizaremos el término negocio, como sinónimo de proyecto de inversión.

Los proyectos de inversión pueden clasificarse de múltiples maneras. En función de su naturaleza, podemos distinguir, entre otros, los siguientes:

1. Apertura de nuevos negocios.
2. Proyectos de reemplazo de activos fijos.

3. Proyectos de expansión del negocio.
4. Campañas de mercadotecnia.
5. Compra de equipo anticontaminante.
6. Incremento al Capital de trabajo.

De acuerdo a su relación con otros proyectos de inversión susceptibles de ser implementados, estos se clasifican en:

1. Proyectos mutuamente excluyentes (únicamente es posible seleccionar el mejor proyecto.)
2. Proyectos independientes (es posible seleccionar varios proyectos)
3. Proyectos interdependientes (seleccionar uno implica seleccionar otros)

2.2 Definición de Beneficio.

Los empresarios realizan una búsqueda constante de oportunidades de inversión porque persiguen un objetivo. Desde un punto de vista financiero, el único objetivo relevante para el empresario es la realización de un beneficio.

En el ámbito financiero, entendemos por beneficio el grado de satisfacción que un inversionista logra al utilizar productivamente sus recursos económicos. Sin embargo, esta satisfacción es difícil de medir. No es posible decir que un empresario obtiene más satisfacción que otro al realizar una determinada inversión, puesto que el grado de satisfacción que ambos obtienen, es un concepto subjetivo. Aun si diseñáramos una escala de satisfacción y pudiéramos afirmar que un inversionista logra con la realización de un proyecto un grado de satisfacción, por ejemplo, de diez y otro inversionista logra, al realizar la misma inversión, un grado de satisfacción de ocho, esta satisfacción sería extraordinariamente subjetiva.

Este problema de subjetividad se resuelve si suponemos que el grado de satisfacción que la realización del proyecto le genera al empresario, está en función de la riqueza que el proyecto produce. Desde este punto de vista, mayor riqueza significaría mayor satisfacción, es decir, el empresario obtiene un beneficio de la realización de un proyecto, únicamente si dicho proyecto incrementa su riqueza. De acuerdo con estos argumentos, podemos concluir que, desde un punto de vista financiero, el objetivo del empresario es maximizar su riqueza.

Sin embargo, no hemos resuelto todo el problema, puesto que necesitamos definir una medida de riqueza. Así por ejemplo, es posible medir riqueza por el volumen de ventas (en unidades monetarias) que un negocio o proyecto puede generar. Otra forma en que es posible medir riqueza es a través de la utilidad neta. Sin embargo, desde un punto de vista financiero, entendemos por riqueza la cantidad de dinero de que dispone el inversionista. Esta cantidad de dinero estará determinada por la diferencia entre las entradas de dinero y las salidas de dinero que a lo largo de la vida de operación o proyecto se generen. A la diferencia entre

las entradas y salidas de dinero se le denomina *flujo de efectivo*. Por lo tanto, puede decirse que, el objetivo del empresario es maximizar los flujos de efectivo que un proyecto o negocio genera, pues de esta forma maximizará su riqueza o cantidad de dinero disponible. En este punto es necesario hacer algunas precisiones en relación con el valor del dinero en el tiempo, ya que una misma cantidad de dinero, tiene diferente valor, dependiendo del momento en que se reciba.

2.3 El valor del dinero en el tiempo: valor presente y valor futuro.

Podemos afirmar que un peso que se recibe hoy tiene más valor que uno que se reciba en el futuro, debido a la pérdida del valor adquisitivo, producto de la inflación. Por otro lado, un peso disponible hoy, puede invertirse y hacerse crecer, mientras que un peso que se recibirá mañana, puede nunca llegar. Adicionalmente, el dinero es un recurso que tiene un costo. De esta manera podemos hablar de un valor presente y un valor futuro del dinero.

La relación entre ambos valores y el costo del dinero, podemos expresarla con la siguiente ecuación:

$$VP + i \times VP = VF(1) \text{ ----- (1)}$$

agrupando términos:

$$VP(1 + i) = VF(1) \text{ ----- (2)}$$

despejando:

$$VP = VF(1) * Fd \text{ ----- (3)}$$

en donde:

$$Fd = 1/(1 + i)^N \text{ se denomina factor de descuento ----- (4)}$$

VP = Valor presente.

i = Tasa de interés.

VF_N = Valor futuro para N periodos.

N = Número de periodos.

2.4 El proceso de elección. *Rentabilidad de un proyecto*

En ocasiones solemos calificar un negocio en función de que tan bueno ha sido en el pasado y pretendemos usar sólo esta información para decidir sobre inversiones futuras. Ello es un error, el inversionista no está interesado en los

flujos de efectivo que un negocio o proyecto haya generado en el pasado, sino en los flujos de efectivo que puede generar a futuro, puesto que éstos serán los que quedarán disponibles para su consumo o reinversión. De esta manera, podemos afirmar que el objetivo del empresario es maximizar los flujos de efectivo futuros que un negocio o proyecto generará.

Para evaluar un proyecto de inversión es necesario contar con un modelo de toma de decisiones de inversión, que nos permita seleccionar aquellos proyectos que maximicen el valor presente de los flujos de efectivo futuros. El criterio más simple utilizado es el denominado rendimiento o rentabilidad. En su forma más simple, se le denomina rentabilidad o rendimiento de un proyecto al cambio porcentual entre la cantidad de dinero disponible al término de la operación de un proyecto y la cantidad de dinero disponible al inicio del mismo.

$$R = (X(1) - X(0)) * 100 / X(0) \text{ ______}(5)$$

en donde

R = Rentabilidad

X(0) = Cantidad de dinero disponible al inicio del proyecto,

X(1) = Cantidad de dinero disponible al final del proyecto.

Esta es una ecuación muy simple que nos permite fácilmente determinar el rendimiento asociado con la utilización productiva de recursos. No obstante presenta una seria limitación: no nos dice nada respecto de si el rendimiento es adecuado o no.

¿Cómo saber si un determinado rendimiento maximiza la riqueza del inversionista? Una forma simple de contestar esta pregunta es compararlo con el rendimiento que se obtendría al invertir los recursos del negocio o proyecto, en una institución financiera. Si el rendimiento del negocio es superior al de la institución financiera, el negocio es al menos mejor que la inversión financiera. Debemos preguntarnos si esta respuesta es suficiente. La respuesta es, no. Requerimos de una herramienta que nos diga algo más respecto de la inversión. Este modelo se desarrolla en el capítulo 4. Antes expondremos algunos conceptos que es necesario conocer para comprender el modelo.

Capítulo 3 Conceptos básicos.

Todo negocio tiene recursos limitados para poder desarrollar e implantar las nuevas oportunidades productivas disponibles. Esto obliga al inversionista a destinar recursos sólo a aquellos proyectos que ofrecen el mayor beneficio (rendimiento).

Adicionalmente, tiene que asegurarse no únicamente que sus recursos sean asignados a aquellos proyectos que ofrecen el mayor rendimiento, sino también, que sean asignados a aquellos proyectos que ofrecen el mayor rendimiento por arriba del que ofrece el mercado financiero. (El mercado financiero está integrado por: los intermediarios financieros [instituciones financieras públicas y privadas tales como bancos, casas de bolsa, etc.]; los bienes u objetos de intercambio [activos financieros tales como: acciones, certificados, pagarés, bonos, etc.] y, los participantes [personas físicas y morales que compran y venden activos financieros a través de los intermediarios.]) En caso contrario, el empresario obtendría un mayor beneficio si invierte sus recursos en el sistema financiero.

Finalmente, a pesar de que un inversionista identifique y asigne sus recursos únicamente a aquellos proyectos que le ofrecen el mayor rendimiento posible, por arriba del que le ofrece el mercado financiero, el hecho del que el rendimiento que le ofrece un proyecto es un rendimiento estimado, es decir, que se espera que ocurra a futuro, introduce el elemento de incertidumbre o riesgo, definido como la posibilidad de que rendimiento esperado por la implementación de un proyecto no se materialice, es decir, no se realice o, en caso de que se realice, sea inferior al rendimiento originalmente estimado.

3.1 Relación entre la rentabilidad y el riesgo del negocio

Existe una relación inseparable entre el rendimiento (beneficio) de cualquier negocio o proyecto y su correspondiente riesgo. Si en un negocio se quiere ganar más, también se tiene que arriesgar más.

Ejemplo 1

Supongamos que tenemos dos alternativas para incrementar \$10,000 que tenemos disponibles.

- 1. La primera alternativa es "echar un volado" por \$ 10,000 con el dueño de una tienda de abarrotes, con el cual podríamos ganar \$ 10,000.*
- 2. La segunda alternativa es comprar un billete de La Lotería Nacional, con el cual podríamos ganar 50 millones.*

¿Qué opción debemos escoger?

Desde el punto de vista de rentabilidad (rendimiento), la segunda alternativa ofrece un mayor rendimiento que la primera, pues ganaríamos 10,000 veces lo que invertimos, mientras que en la primera alternativa, sólo duplicaremos nuestra inversión.

¿Debemos entonces escoger la opción de la Lotería Nacional, ya que ésta nos ofrece una mayor rentabilidad? La respuesta es: no necesariamente. Debemos analizar otro elemento que debe tenerse siempre presente en la evaluación de todo negocio o proyecto: el riesgo del negocio o proyecto.

En la primera alternativa ("volado"), tenemos la misma probabilidad (50%) de ganar y de perder. Con la segunda alternativa ("billete de lotería") si consideramos que hay 50,000 números distintos que participan en cada sorteo, tenemos sólo una posibilidad de uno en 50,000 de ganar.

En cuanto a riesgo se refiere, es mucho menos atractiva la segunda alternativa, ya que es mucho más difícil ganar la lotería que el "volado." Por lo tanto, si queremos una mayor rentabilidad, también tendremos que asumir un mayor riesgo.

La decisión final por una de las dos alternativas (o quizás decidamos que, después de todo, es mejor no arriesgar nuestros \$ 10,000 y rechazar ambas alternativas), dependerá de nuestras preferencias personales hacia el "juego" (a esto se le conoce como "aversión al riesgo").

En el caso de un negocio o proyecto, a mayor variación en la rentabilidad (rendimiento esperado) de un negocio o proyecto, mayor es el riesgo de éste.

Ejemplo 2

De nuevo, asumamos que se tienen dos negocios A y B. El negocio A puede dar un rendimiento del 20% si las "cosas salen bien"; o puede dar un rendimiento del 15%, si las "cosas salen mal".

El negocio B puede dar un rendimiento del 30% si las "cosas salen bien", o puede dar un rendimiento del 10% si las "cosas salen mal".

¿Cuál es el negocio con mayor riesgo?

El negocio con más riesgo es el negocio B, ya que tiene una mayor variación en sus dos posibles rentabilidades (tiene una variación de: $30\% - 10\% = 20\%$), mientras que el negocio A (tiene una variación de: $20\% - 15\% = 5\%$).

Si dos negocios ofrecen la misma rentabilidad pero diferente riesgo, el negocio con el menor riesgo obviamente es el más atractivo.

Para que el negocio con mayor riesgo sea seleccionado, éste deberá ofrecer también una mayor rentabilidad que el negocio con menor riesgo. Existen varias

maneras de incorporar explícitamente el factor riesgo en la evaluación de un negocio.

El propósito de este trabajo no es detallar como se lleva a cabo dicha incorporación explícita del riesgo, ya que es un concepto matemáticamente complejo y rebasa los propósitos del presente trabajo. (Cuando un negocio o proyecto tiene un riesgo diferente al de los otros negocios o proyectos, lo que se hace es una de dos cosas: se ajusta la tasa de interés (costo de capital) o se ajustan los flujos de efectivo del negocio o proyecto. Si el riesgo es mayor, lo que se hace es incrementar el valor de la tasa de interés (costo de capital) o alternativamente, disminuir el monto de los flujos de efectivo. La magnitud del ajuste se puede determinar matemáticamente, pero su cálculo es complejo.

Lo que si se enfatiza es la importancia que tiene el incorporar, aunque sea de manera implícita el elemento riesgo a la evaluación del negocio, especialmente cuando el riesgo de dicho negocio es diferente al riesgo de otros que se evalúan y cuyas rentabilidades se desean comparar.

La recomendación práctica es: si los negocios o proyectos tienen un riesgo similar se selecciona él o los proyectos que le den la rentabilidad más alta. Cuando dos o más proyectos tienen un riesgo diferente, los negocios o proyectos que den la rentabilidad más alta no son necesariamente los mejores. En este caso, se deberá ponderar la mejor rentabilidad del negocio o proyecto con su mayor riesgo y decidir si está o no dispuesto a asumir ese riesgo mayor.

Debido a estos factores, el empresario o inversionista requiere de utilizar técnicas y herramientas que le permitan medir el riesgo y medir el rendimiento que un proyecto de inversión le ofrece. Al proceso de medición del riesgo y rendimiento asociados a los proyectos de inversión, le denominaremos valuación.

Si el conjunto de negocios o proyectos susceptibles de implementarse, es valuado siguiendo las mismas técnicas y criterios de valuación, entonces estaremos en posibilidades de comparar el riesgo y el rendimiento asociados a cada negocio o proyecto y seleccionar, únicamente, aquellos negocios o proyectos que permitan maximizar la riqueza con el mínimo riesgo.

Las técnicas y herramientas de análisis y evaluación de proyectos que se expondrán en capítulos posteriores, aplican criterios generalmente aceptados. La utilización consistente y correcta de dichos criterios, permiten la comparación entre proyectos (basadas en unidades de riesgo – rendimiento)

Los criterios generalmente aceptados en que se basan los métodos de análisis y evaluación de proyectos que se expondrán en esta tesis son los siguientes:

1. La riqueza se mide en términos de dinero, es decir, en flujos de efectivo.
2. El dinero tiene un valor en el tiempo que se mide en términos de tasas de interés.

3. La medida del riesgo (unidades de riesgo) debe incorporarse en los flujos de efectivo (dinero) o incorporarse en el valor del dinero en el tiempo (tasas de interés).

3.2 Vida económica de un negocio o proyecto.

La vida económica de un negocio o proyecto es el periodo de tiempo que se considera relevante para fines de evaluación.

Normalmente medimos el periodo que dura un negocio o proyecto, es decir, su vida económica, en años. Sin embargo, dependiendo del tipo de proyecto o negocio que se evalúa, es posible utilizar como unidad de medida de la vida económica cualquier otra unidad de tiempo, tal como: días, semanas o meses.

La vida económica de un negocio o proyecto inicia cuando se comprometen recursos para "arrancar" las operaciones del mismo, es decir, normalmente inicia cuando se realiza la inversión requerida para empezar las operaciones del negocio. Al periodo en que se inicia el negocio o proyecto se le denomina Periodo 0.

La vida económica de un negocio o proyecto termina cuando éste pierde su capacidad de generar flujos de efectivo positivos. Cuando esto ocurre, normalmente el dueño o empresario vende su negocio o cierra sus operaciones. Podemos considerar la vida económica de un negocio o proyecto, como un agregado de las vidas económicas de los bienes o activos que se requieren para implementar y operar el negocio o proyecto.

La vida económica de un activo puede terminar cuando el mercado para el producto o servicio generado con ese activo ha desaparecido, es decir, cuando el producto o servicio ya no puede ser comercializado. La vida económica de un bien o activo también puede terminar cuando éste es vendido o cuando el bien o activo es utilizado en otro negocio o proyecto distinto a aquel para el que fue adquirido. Por supuesto, la vida económica de un bien o activo puede terminar también cuando finaliza su vida física u operativa, si el empresario conserva el activo hasta que esto ocurre.

La Vida Operativa o Física y la Vida Tecnológica de un bien o activo, no son necesariamente iguales a su vida económica. Un bien o activo puede seguir siendo utilizado desde un punto de vista físico o tecnológico, sin embargo el periodo relevante de utilización u operación, desde un punto de vista de evaluación de proyectos, es el relativo a la vida económica.

El periodo a lo largo del cual un bien activo se deprecia, tampoco es necesariamente igual a su vida económica. La Vida o Periodo de Depreciación de

un bien o activo, está basado en principios contables o fiscales que no necesariamente guardan relación con la vida económica del bien.

Puede decirse que la vida económica de un proyecto o negocio siempre será superior o al menos igual a la vida económica de los activos o bienes necesarios para su implementación. Asimismo, la vida física o de operación de los activos o bienes asociados con un proyecto puede ser inferior o superior a la vida económica del proyecto; sin embargo, en este último caso, cuando la vida física es superior a la vida económica del proyecto, únicamente será relevante para la evaluación financiera del mismo, el periodo de tiempo que dure el proyecto, es decir, su vida económica.

Cuando se evalúan dos o más negocios o proyectos mutuamente excluyentes y que tiene vidas económicas diferentes es necesario igualar sus vidas económicas para poderlos comparar desde el punto de vista financiero.

3.3 Costo de capital (tasa de interés).

El costo de capital es la tasa de interés que se utiliza para "descontar" los flujos de efectivo netos del negocio para encontrar su valor presente neto; también es la tasa de interés contra la que se compara la tasa interna de rendimiento para ver si el negocio o proyecto es atractivo.

Toda inversión en un negocio o proyecto requiere ser financiada. Si el empresario recibe un préstamo para financiar un negocio o proyecto, dicho empresario va a tener inicialmente una entrada de dinero producto del préstamo y, posteriormente, va a tener salidas de dinero originadas por el pago de los intereses y del monto que inicialmente le prestaron

Aunque estas entradas y salidas de dinero son flujos de efectivo financieros que se originan sólo si el negocio o proyecto es aprobado, no deben incluirse como parte de los flujos de efectivo netos en la evaluación del negocio o proyecto.

La razón por la cual no se deben incluir los flujos de efectivo financieros como parte de los flujos de efectivo netos del negocio, es que la tasa de interés con la cual se van a "descontar" dichos flujos de efectivo, ya incluye el costo de financiamiento.

Por lo tanto si los flujos de efectivo financieros se incluyen como parte de los flujos de efectivo del negocio o proyecto, se estaría contabilizando "doble" el efecto financiero, ocasionando una evaluación errónea del negocio.

Cualquier ajuste por financiamiento se debe ver reflejado en la tasa de descuento (costo de capital) y no en los flujos de efectivo del negocio.

3.4 Flujos de efectivo.

Los flujos de efectivo de un negocio o proyecto están integrados por dos componentes:

1. La inversión neta.
2. Los flujos de efectivo operativos netos.

Inversión neta.

La inversión neta esta formada por las entradas y salidas de dinero que un negocio o proyecto genera, debido a:

1. La adquisición de activos fijos.
2. Los cambios o variación en el capital de trabajo.
3. Los valores residuales o de rescate que se obtienen por la liquidación de los activos fijos y el capital de trabajo al término del negocio o proyecto.

Flujo de efectivo es la cantidad de dinero que entra menos la del dinero que sale del negocio durante su vida económica.

Flujo de efectivo es la cantidad de dinero que queda disponible a un negocio después de considerar todas sus entradas de dinero y todas sus salidas de dinero, en un periodo determinado. Al empresario le interesa determinar los Flujos de Efectivo que su negocio produce para cada periodo o ciclo completo de operación, es decir, para cada día, semana, mes o año de vida económica.

El concepto de Flujo de Efectivo puede ser expresado también en forma aritmética, de la siguiente forma:

$$\text{FLUJO DE EFECTIVO} = \text{ENTRADAS DE DINERO} - \text{SALIDAS DE DINERO}$$

En virtud de que las operaciones de un negocio conforman un ciclo continuo que, normalmente, inicia con la compra (salida de dinero) de materias primas y productos para la venta y termina con la venta (entrada de dinero) del producto o servicio, el flujo de efectivo, también puede visualizarse como un ciclo continuo de entradas y salidas de dinero.

Dentro de este ciclo podemos identificar, en función de la procedencia de las entradas de dinero y el destino de las salidas del dinero, dos clases de flujos de efectivo: los Flujos de Efectivo del Negocio y los Flujos de Efectivo Financieros.

Como su nombre lo indica, los flujos del negocio son aquellos que se originan por las actividades continuas y cotidianas del negocio o proyecto.

Los componentes principales de Ingresos (entradas de dinero) y Egresos (salidas de dinero) que componen estos dos tipos de Flujos de Efectivo, se muestran en la fig. 3.1

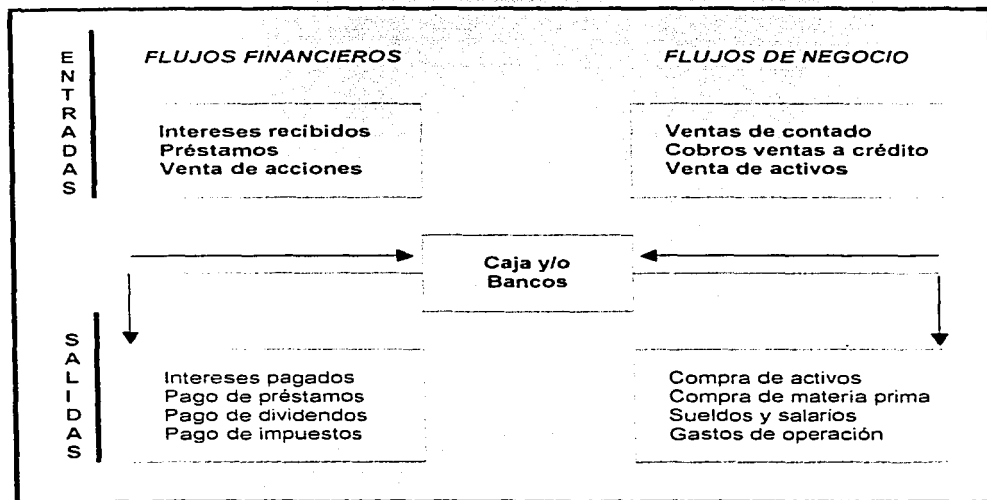


Fig.3.1 Componentes principales de Ingresos y Egresos de un negocio.

Los flujos de Efectivo Financieros, no son necesarios para evaluar proyectos de inversión.

El empresario está interesado en determinar los flujos de efectivo del negocio después de impuestos, debido a que el pago de impuestos es una salida de dinero que disminuye su riqueza disponible. A los flujos de efectivo del negocio, después de impuestos, les denominaremos flujos de efectivo netos.

Para facilitar nuestra comprensión de los diferentes elementos que integran los flujos de efectivo netos, los dividiremos en dos grandes grupos. Al primero lo denominaremos inversión neta, al segundo flujos de efectivo operativos.

Por lo tanto, los flujos de efectivo netos para cada periodo de la vida económica de un negocio o proyecto serán calculados de acuerdo con la siguiente fórmula:

$$\text{FLUJOS DE EFECTIVO DEL NEGOCIO NETOS} = \text{FLUJOS DE EFECTIVO OPERATIVOS} - \text{INVERSIÓN NETA}$$

Los diferentes elementos que integran los flujos de efectivo netos se muestran en la siguiente tabla 3.1.

CONCEPTO/VIDA ECONÓMICA	FLUJOS DE EFECTIVO NETOS					
	PERIODO 0	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO n
1.- Entradas de efectivo: Ventas de contado						
2.- + Cobros ventas a crédito						
3.- = Total de ingresos en efectivo						
4.- Salidas de efectivo						
5.- Costos variables de producción						
6.- + Costos fijos de producción						
7.- + Gastos de operación						
8.- = Total de egresos en efectivo						
9.- Ingresos en efectivo - egresos en efectivo						
10.- - Impuestos						
11.- = Flujo de efectivo operativo						
12.- Inversión neta						
13.- Compra de activos fijos						
14.- Cambios en capital de trabajo						
15.- Valores de rescate						
16.- = Inversión neta						
FLUJO DE EFECTIVO DEL NEGOCIO						

TESIS CON
FALLA DE ORIGEN

Tabla 3.1 Esquema general de flujos de efectivo de un negocio o proyecto

Esta tabla nos servirá de guía, a lo largo del capítulo, para facilitar nuestra comprensión de los métodos de determinación de los flujos de efectivo del negocio netos que se producen a lo largo de la vida económica de un proyecto o negocio.

El Análisis y Evaluación de Proyectos se lleva a cabo para tomar mejores decisiones de inversión. Ello implica que **las decisiones de inversión deben tomarse antes de desarrollar el negocio o proyecto y no cuando ya se han comprometido recursos para su implementación y desarrollo.** Por esta razón, es muy importante tener siempre presente que pretendemos **determinar flujos de efectivo que ocurrirán a futuro.**

Inversión neta

La Inversión Neta es la cantidad de dinero que se requiere para la adquisición de los activos y bienes necesarios para iniciar y mantener en operación un negocio o proyecto.

Todo negocio o proyecto requiere de comprometer recursos para su implementación y desarrollo. La inversión neta está compuesta de tres tipos principales de flujos de efectivo:

- Las entradas y salidas de dinero asociadas con la adquisición de los activos fijos necesarios para el negocio o proyecto.
- Los recursos que se comprometen para mantener la operación del negocio y que se denominan capital de trabajo.
- Las entradas y salidas de dinero que se reciben por concepto de la venta o disposición de los activos fijos y el capital de trabajo, cuando éstos llegan al término de su vida económica, a las cuales se denominan valores de rescate.

Adquisición de activos fijos

El Activo Fijo de un negocio o proyecto está conformado por todos aquellos bienes y derechos, propiedad del empresario o del negocio, que se compran con objeto de usarlos para realizar las actividades operativas (los principales bienes y derechos que conforman el activo fijo son: terrenos, edificios, maquinaria, mobiliario y equipo de oficina, equipo de entrega o de reparto, etc.).

La adquisición de activos fijos se realiza a lo largo de la vida económica del proyecto, sin embargo, normalmente, es necesario realizar un desembolso de

efectivo (salida de dinero) para comprar los activos fijos mínimos requeridos para que el negocio o proyecto inicie sus operaciones. Esta salida de dinero inicial ocurre en el Periodo 0 de la vida económica del negocio o proyecto.

Hemos mencionado que la inversión neta se integra, parcialmente, por las salidas y desembolsos de dinero asociados con la adquisición de los activos fijos necesarios para realizar las actividades propias de un negocio. Sin embargo, el precio de la compra de estos bienes no es, normalmente, el único desembolso requerido para su adquisición. Todos aquellos desembolsos y pagos (salidas de dinero) necesarios para instalar y poner al activo en condiciones de operación, también son considerados parte del flujo de efectivo derivado de la adquisición del activo fijo. Así pues, todos los pagos realizados por concepto, por ejemplo, de empaque, de transporte y fletes, de instalación, de impuestos y derechos, o aquellos realizados para la compra de pólizas de seguro para garantizar el transporte del bien o activo, son considerados parte del flujo efectivo que se produce por la adquisición.

En muchos negocios y proyectos, la inversión neta que se deriva de la adquisición de activos fijos no involucra únicamente salidas de dinero; también se pueden presentar entradas de dinero. Estas entradas deben considerarse también como parte integrante de la inversión neta.

En ocasiones, debido al tamaño o complejidad del negocio o proyecto o debido a que la adquisición de los activos fijos se realiza a crédito o a plazos, los flujos de efectivo de la inversión neta se realizan a lo largo de más de un periodo (años) de vida económica.

En este caso, debemos considerar las salidas y entradas de dinero asociadas a la adquisición de los activos fijos, en el periodo en que ocurren, es decir, en el periodo en que efectivamente se realizan.

Capital de trabajo

Hasta el momento, hemos considerado únicamente las entradas y salidas de dinero resultantes de la adquisición y puesta en operación de activos fijos como parte integrante de los flujos de efectivo del negocio que hemos denominado inversión neta. Sin embargo, para que un nuevo negocio o proyecto inicie o mantenga sus operaciones, también suele ser necesario realizar otro tipo de desembolsos o salidas de dinero. Así, por ejemplo, antes de poder iniciar sus actividades productivas y de comercialización, normalmente un negocio requiere de un inventario inicial de materia prima y un inventario inicial de producto terminado. Asimismo, un negocio puede requerir, para iniciar sus operaciones, de otorgar crédito a sus clientes o mantener una cantidad de dinero disponible en efectivo o en inversiones a corto plazo para enfrentar gastos inmediatos de

operación, como la compra de papelería, el pago de transportes, etc. Todos estos conceptos requieren el comprometer recursos para mantener la operación del negocio o proyecto, por lo que son considerados como salidas de dinero

Además, no necesariamente la adquisición de bienes indispensables para iniciar o mantener la operación del negocio debe realizarse de contado. Por ejemplo, la adquisición de productos para constituir un inventario inicial de materia prima puede realizarse a crédito. Las compras a crédito originan el reconocimiento de la deuda reconocida en los libros contables del negocio como una cuenta por pagar.

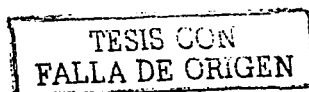
Lo mismo ocurre con otro tipo de desembolsos que pueden ser pospuestos, por cortos periodos, tales como salarios por pagar y/o créditos a corto plazo. Desde un punto de vista financiero, estos conceptos representan entradas de dinero para el negocio, puesto que, al posponer la salida de dinero, el empresario cuenta con recursos adicionales para emplearlos productivamente.

Podemos resumir que para iniciar y mantener sus operaciones, un negocio o requiere comprometer recursos adicionales a la adquisición y puesta en operación de activos fijos. Estos recursos adicionales corresponden a las modificaciones (incrementos y disminuciones) de las cuentas registradas contablemente como activos circulantes y pasivos circulantes (de acuerdo con los principios generalmente aceptados de contabilidad, el activo circulante o flotante se conforma por los bienes y derechos del negocio que pueden ser fácilmente convertidos en efectivo y se encuentran en rotación constante. El pasivo circulante o flotante se conforma con las deudas y obligaciones que tienen un vencimiento menor a un año y también se encuentran en constante rotación del Balance General o Estado de Posición Financiera.

Las principales cuentas de activos y pasivos circulantes que podemos encontrar en el Balance General de un negocio, se presentan en la tabla 3.2.

BALANCE GENERAL	
ACTIVOS CIRCULANTES	PASIVOS CIRCULANTES
Caja y bancos + Documentos por cobrar + Deudores diversos + Pagos anticipados + Inversiones a corto plazo + Inventario de mercancías + Inventario de materia prima + Depósitos de garantía = TOTAL ACTIVO CIRCULANTE	Proveedores + Documentos por pagar + Acreedores diversos + Cobros anticipados + Gastos por pagar + Salarios por pagar + Impuestos por pagar = TOTAL PASIVO CIRCULANTE

Tabla 3.2 Cuentas de activo circulante y pasivo circulante.



Así por ejemplo, incrementos en los niveles de cualquier cuenta correspondiente al activo circulante representan una salida de efectivo (el movimiento contrario, una disminución en cualquier cuenta del activo circulante representaría una entrada de dinero), e incrementos en los niveles de cualquier cuenta de pasivo circulante representaría una entrada de dinero (el movimiento contrario, una disminución del pasivo circulante, representaría una salida de dinero). Para facilitar la comprensión, sinteticemos esta información en el siguiente cuadro:

	<u>INCREMENTO</u>	<u>DISMINUCIÓN</u>
TOTAL ACTIVO CIRCULANTE	Salida de dinero	Entrada de dinero
TOTAL PASIVO CIRCULANTE	Entrada de dinero	Salida de dinero

Tabla 3.3 Relación entre circulante y flujo de efectivo

Ejemplo 3

Un empresario desea fabricar llaveros, por lo que ha comprado una máquina que le costo \$ 12 millones. Él sabe que esta inversión en activo fijo de \$ 12 millones no es la única inversión que va a necesitar para "arrancar" el negocio, puesto que también necesitará mantener un inventario de cadenas y hojas de aluminio (inventario de materia prima) así como un inventario de llaveros (inventario de mercancías) para surtir rápidamente los pedidos de sus clientes.

El empresario estima que la inversión en los dos tipos de inventarios será de \$1.5 millones, la cual pagará de contado. Así mismo, también estima que requiere dar crédito a sus clientes, por lo cual realizará una inversión en cuentas por cobrar de \$ 4 millones. De igual manera, requiere obtener crédito de sus proveedores de materia prima, por lo cual considera que mantendrá un saldo en cuentas por pagar de \$ 3 millones. Esta situación se mantendrá a lo largo de la vida económica del negocio de llaveros, que es de cinco años

Las entradas y salidas de dinero que se producen en el Periodo 0 debido a los niveles de activos y pasivos circulantes que se requiere mantener para iniciar el negocio de llaveros, son las siguientes:

	<u>ENTRADAS DE DINERO</u>	<u>SALIDAS DE DINERO</u>
Cuentas por cobrar		\$ 4.0
Inventarios		\$ 1.5
		<u>\$ 5.5</u>
Total de Activo Circulante	<u>\$ 3.0</u>	
Cuentas por pagar	<u>\$ 3.0</u>	

**TESIS CON
FALLA DE ORIGEN**

El flujo de efectivo originado por la inversión adicional en activos y pasivos circulantes es igual a las entradas de dinero menos las salidas de dinero, es decir, es igual a \$ 2.5 millones (\$ 5.5 - \$ 3.0).

Un concepto útil para determinar la inversión neta que se requiere en activos circulantes y pasivos circulantes es el concepto de capital de trabajo. **Por capital de trabajo entendemos el capital que se dispone para las actividades ordinarias o normales del negocio.** Este capital está compuesto por los activos y pasivos circulantes. En forma aritmética, podemos definir al capital de trabajo, como sigue:

$$\text{CAPITAL DE TRABAJO CIRCULANTE} = \text{TOTAL DE ACTIVOS CIRCULANTES} - \text{TOTAL DE PASIVOS CIRCULANTES}$$

Utilizando este concepto, podemos identificar rápidamente la inversión neta requerida para iniciar o mantener operando un negocio; puesto que, un aumento en el capital de trabajo de un periodo de vida económica a otro (cambio en el capital de trabajo) nos indicará que los activos circulantes se incrementaron en mayor medida que los pasivos circulantes y por tanto, ocurrió una salida de dinero del negocio. Asimismo, si en capital de trabajo disminuye de un periodo de vida económica a otro, esto significará que los pasivos circulantes se incrementaron en mayor medida que los activos circulantes, por lo cual, existe una entrada de dinero al negocio. Esta información puede resumirse en forma tabular:

CAMBIOS EN EL CAPITAL DE TRABAJO

INCREMENTO	DISMINUCIÓN
Salida de dinero	Entrada de dinero

Un factor importante que debemos tener en cuenta, es que no todos los cambios en capital de trabajo deben considerarse para propósitos de evaluación de proyectos. Deben excluirse aquellos cambios que se derivan de activos y pasivos circulantes no operativos, tales como: inversiones a corto plazo, intereses por pagar, préstamos por pagar, etc. Esto se debe a que los cambios que ocurren en los saldos de estas cuentas son movimientos que afectan el flujo de efectivo financiero, por lo cual no forman parte de los flujos de efectivo que se derivan de las operaciones normales del negocio y, como se señaló anteriormente, no son relevantes para la evaluación de un proyecto o negocio. Por esta razón, puede

decirse que el monto de los cambios en capital de trabajo asociados con la inversión neta y por lo tanto con los flujos de efectivo del negocio, deben calcularse sobre la base de un capital de trabajo ajustado, es decir, una cifra de capital de trabajo que únicamente tome en cuenta cambios en el mismo originados por las operaciones normales del negocio o proyecto y no por aquellos relacionados con el financiamiento de las operaciones, o sea, los relacionados con los flujos de efectivo financieros.

Ejemplo 4.

Se requerirá invertir \$ 3 millones, en el Periodo 0, para adquirir el inventario de materia prima con el cual operará un negocio durante el primer año de vida económica. Al término de este primer año, se requerirá entonces invertir \$ 3 millones adicionales (mantenimiento de un inventario de materia prima de \$ 6 millones) para continuar sus operaciones durante el segundo año.

Para poder operar durante el tercer año, se necesitará incrementar el inventario en materia prima en \$ 0.6 millones, es decir, mantendrá un nivel de inventarios de \$ 6.6 millones. Finalmente para poder operar durante el cuarto y quinto años se estima necesitar un inventario de materia prima de \$ 7.2 millones y \$ 8.0 millones respectivamente. Esto significa que se requerirá invertir cantidades adicionales de recursos para poder mantener dichos niveles de inventarios de materia prima.

Los flujos de efectivo asociados con la inversión requerida en capital de trabajo se determinan directamente de los cambios en capital de trabajo, para cada periodo.

Como el inventario de materia prima es el único componente del capital de trabajo relevante para determinar la inversión desde el punto de vista de los flujos de efectivo del negocio netos, estos flujos de efectivo derivados de la inversión neta serían:

AÑO	VALOR DEL INVENTARIO (MM)	INVERSIÓN NETA EN CAPITAL DE TRABAJO (MM)
0	\$ 3.0	\$ 3.0
1	\$ 6.0	\$ 3.0
2	\$ 6.6	\$ 0.6
3	\$ 7.2	\$ 0.6
4	\$ 8.0	\$ 0.8
5	\$ 0.0	\$ 0.0
TOTAL		\$ 8.0

MM= Millones de pesos.

Es muy importante recordar siempre que los flujos de efectivo asociados con la inversión neta en capital de trabajo, únicamente son los cambios (incrementos y disminuciones) en el capital de trabajo de un periodo a otro de vida económica (como se ha mostrado en los ejemplos presentados) y no el total de capital de trabajo que se mantiene cada periodo.

Valores de rescate

Un componente importante que debe ser considerado como parte de la inversión neta es su valor de rescate, es decir, al término de la vida económica del proyecto o negocio es razonable esperar que la cantidad de dinero invertida en activos fijos y en capital de trabajo quede nuevamente disponible al empresario.

El Valor de Rescate de la Inversión Neta es el flujo de efectivo que se produce al final de la vida económica del negocio o proyecto, debido a la venta de los activos fijos y a la recuperación del capital de trabajo.

Para determinar tanto el valor de rescate de los activos fijos como el valor de rescate del capital de trabajo, debemos considerar su valor de mercado estimado y no su valor en libros (El Valor en Libros de un activo fijo se determina, de acuerdo con la teoría contable, restando al valor original de adquisición del activo, la depreciación acumulada calculada a lo largo de la vida de utilización del activo).

Por valor de mercado de un bien activo entendemos el precio al que se puede vender ese bien o activo.

Para determinar el valor de rescate del capital de trabajo, si no se cuenta con una estimación sobre su valor de liquidación (valor de mercado), se considera que su valor de rescate es igual al flujo de efectivo que se comprometió en capital de trabajo, es decir, es igual a la suma de los cambios que se presentaron en capital de trabajo a lo largo de toda la vida económica del proyecto.

El valor de rescate de la inversión neta está conformado por el valor de rescate de la adquisición de activos fijos (la cual no sufrió cambios de su estimación original) más el valor de rescate del capital de trabajo.

Determinación de los flujos de efectivo operativos

Los flujos de Efectivo Operativos son las entradas de dinero menos las salidas de dinero que un negocio o proyecto genera por sus operaciones, durante su vida económica.

Ventas y costos

Las entradas de dinero o efectivo se componen, fundamentalmente, por los ingresos generados por la venta de los productos y servicios comercializados por el negocio.

Las salidas de dinero o efectivo se componen, fundamentalmente, por los pagos realizados para cubrir los costos variables y los costos fijos necesarios para operar el negocio. **Costos Variables son aquellos costos que cambian (incrementan o disminuyen) en función del volumen de operaciones** (medido, por ejemplo, en unidades producidas o en horas de operación) **costos fijos son aquellos costos que permanecen constantes, para un intervalo dado de volumen de operaciones.** Únicamente aquellos costos que se pagan efectivamente en cada periodo de vida económica, son los que deben considerarse para la determinación de los flujos de efectivo operativos. Las ecuaciones 5 a 8 expresan la relación de los diferentes ingresos y egresos operativos con el flujo de efectivo

$$Feo = Vn - Co \text{ ----- (6)}$$

$$Vn = Vb - (Des + Dev) \text{ ----- (7)}$$

$$Vb = Vc + Cvc \text{ ----- (8)}$$

$$Co = Cv + Cf \text{ ----- (9)}$$

Donde:

Vn = Ventas netas.

Co = Costos de operación.

Vb = Ventas brutas.

Des = Importe de descuentos.

Dev = Devoluciones.

Cvc = Cobranza de ventas a crédito.

Cv = Costos variables.

Cf = Costos fijos.

Es importante tener en cuenta que existen dos requisitos para incorporar a los flujos de efectivo operativos las ventas y los costos de operación.

El primer requisito es que las entradas y salidas de dinero por estos conceptos se realicen, es decir, que los ingresos y desembolsos de dinero efectivamente ocurran. El segundo es que se reconozcan como parte de los flujos de efectivo

operativos en el periodo de operación o vida económica en que efectivamente se realicen.

Existen dos tipos de costos que deben ser correctamente analizados, ya que pueden llegar a afectar los flujos de efectivo netos de un negocio o proyecto, estos son:

1. Costos de oportunidad
2. Costos "hundidos"

Costos de oportunidad

Los costos de oportunidad son relevantes en la decisión de invertir en un negocio o proyecto, ya que afectan el flujo de efectivo neto del negocio.

En la evaluación financiera un costo no es únicamente el dinero que sale para pagar algún producto o servicio, sino también es el dinero que se deja de recibir como consecuencia de la decisión de aceptar el negocio o proyecto que se evalúa.

Al flujo de efectivo que se deja de recibir como consecuencia de la aceptación del negocio o proyecto, se le conoce como costo de oportunidad.

Los costos de oportunidad representan una salida de dinero y, por lo tanto deben ser incluidos en los flujos de efectivo que se utilizan para evaluar un proyecto o negocio.

Un empresario, puede poseer un activo que esta considerando vender, rentar o utilizar en algún otro negocio o proyecto. Si dicho activo es utilizado en el negocio o proyecto que se esta evaluando, los ingresos potenciales de otras alternativas de utilización del activo se pierden. Estos ingresos perdidos pueden ser vistos como costos (costos de oportunidad), ya que si el empresario acepta el negocio o proyecto, va con ello a perder otras oportunidades de ingresos por el uso de dicho activo.

Ejemplo 5.

Supóngase que un empresario posee una bodega que actualmente no utiliza y que se encuentra ubicada a menos de un kilómetro de la fabrica donde esta considerando fabricar balones de fútbol.

Dicho empresario había estado pensando en rentar dicha bodega. Sin embargo se le acaba de ocurrir que podría aprovechar esta bodega para almacenar su inventario de balones, ya que siendo suya, está "le sale gratis".

¿Está en lo correcto?

La respuesta es no, ya que la utilización de la bodega implica un costo de oportunidad.

Dicho costo de oportunidad es igual al flujo de efectivo neto de la renta que el empresario dejará de recibir si utiliza la bodega para almacenar balones. Si decide que es mejor vender la bodega que rentarla, en caso de no utilizar la bodega para el negocio de balones, el costo de oportunidad sería entonces igual al flujo de efectivo neto, producto de la venta de la bodega. Dicho costo de oportunidad tendría que ser incluido como un componente del flujo de efectivo neto en la evaluación del negocio de balones.

Costos "hundidos"

A diferencia de los costos de oportunidad, los costos "hundidos" son irrelevantes en la decisión de invertir en un negocio o proyecto, pues aunque éstos representan un flujo de efectivo, dicho flujo de efectivo ya ocurrió, es del pasado, y por lo tanto, ya no puede ser modificado o alterado por la decisión de aceptar o rechazar un negocio o proyecto.

La evaluación de un nuevo negocio o proyecto debe ver hacia delante, es decir, proyectar el futuro. Por lo tanto:

Los costos "hundidos" no pueden ser recuperados y, por lo tanto, deben ser ignorados en la evaluación de un negocio o proyecto.

Ejemplo 6.

Del ejemplo anterior, supongamos que el dueño de la fábrica de balones construyó hace un mes una pequeña oficina dentro de la fábrica donde piensa producir los balones de futbol.

La construcción costo \$ 50,000 mas \$ 20,000 en muebles, para una inversión total de \$ 70,000.

Lamentablemente se tendrá que demoler esta pequeña oficina para dejar el único espacio disponible para colocar la máquina de balones.

El empresario piensa que si decide aceptar el proyecto de fabricación de balones, va a perder esta inversión; y no será así, si decide rechazar el proyecto; por lo tanto, insiste en incluir el costo de la construcción (\$ 50,000) como parte del flujo efectivo del negocio de los balones de futbol.

¿Está en lo correcto?

La respuesta correcta es nuevamente no, ya que costo de la construcción de la oficina (\$ 50,000) es una salida de dinero que ya ocurrió y que ya no va a poder ser cambiada, independientemente de que se tome la decisión de poner la

máquina de balones en dicho lugar, si se acepta el proyecto. o mantener la oficina, si se rechaza el proyecto.

Depreciación

Los flujos de efectivo derivados de la adquisición de activos fijos (inversión neta) deben considerarse en el periodo de vida económica del proyecto o negocio en que efectivamente se realizan las entradas o salidas de dinero. Asimismo, dichos flujos de efectivo se integran por el precio o valor de mercado del bien, por todas las salidas de dinero adicionales que se requieran para ponerlo en condiciones de operación y, por todas las entradas de dinero que se producen por la venta de activos fijos antiguos, si es que éstos son reemplazados.

Sin embargo, desde el punto de vista contable y fiscal, las adquisiciones de activos fijos, no son consideradas en la mismas forma.

Desde la perspectiva contable y fiscal, se considera, que, dado que los activos fijos que se adquieren serán utilizados a lo largo de varios periodos de operación o vida contable, el costo o valor de adquisición de estos bienes debe distribuirse a lo largo de su vida contable o fiscal. ***Por vida contable entendemos el periodo de tiempo que el contador estima útil o utilizable el bien o activo fijo con objeto de calcular su depreciación. Por vida fiscal entendemos el periodo de tiempo que las autoridades fiscales consideran útil o utilizable el bien o activo fijo con objeto de calcular su depreciación.***

Existen varios métodos para distribuir el valor de adquisición de los activos fijos a lo largo de la vida contable de un negocio o proyecto. Sin embargo, el método más utilizado es el denominado depreciación en línea recta.

La depreciación no constituye una entrada o salida de dinero, por lo tanto, no es un flujo de efectivo.

Los flujos de efectivo asociados con la adquisición de activos fijos ocurren en el momento en que se efectúa el pago (salida de dinero) por concepto del precio de venta de dicho bien y/o los gastos adicionales para dejarlo en condiciones operativas, y no cuando se registra la depreciación correspondiente en los libros contables de la empresa.

Por lo tanto, la compra o venta de activos fijos generan flujos de efectivo en el momento en que se realizan las entradas o salidas de dinero y la depreciación no forman parte del flujo de efectivo operativo.

La depreciación es sólo uno de los múltiples conceptos contables que, desde el punto de vista financiero, son irrelevantes para evaluar proyectos o negocios, puesto que no reflejan los movimientos reales de riqueza de la empresa o negocio,

los cuales únicamente pueden ser visualizados usando el concepto de flujos de efectivo del negocio netos.

Utilizaremos el término depreciación para referirnos a todos los movimientos contables, tales como: amortizaciones y provisiones de activo y pasivo, que no producen efectos directos en los flujos de efectivo

Si bien la depreciación no produce efectos directos en los flujos de efectivo que un negocio o proyecto genera, en virtud de que la legislación fiscal permite que los montos calculados de depreciación son deducibles de impuestos, esto genera un efecto indirecto sobre los flujos de efectivo operativos. Es decir, dado que la depreciación es un monto deducible, al restarla de las ventas realizadas por el negocio, reduce la utilidad gravable (utilidad gravable es la utilidad del negocio que sirve de base para determinar el monto a pagar de impuestos) del mismo y, por lo tanto, reduce el monto de impuestos que el negocio debe cubrir a las autoridades fiscales.

Impuestos

Los impuestos representan salidas de dinero que disminuyen la riqueza o cantidad de dinero disponible para el empresario. Por este motivo, es necesario determinar los flujos de efectivo del negocio después de impuestos, es decir, los flujos de efectivo del negocio netos.

La incorporación de los efectos fiscales a la determinación de los flujos de efectivo del negocio no es una tarea fácil. La dinámica de cambio de la legislación fiscal, la diversidad de regímenes de contribución, la realización de pagos provisionales, el establecimiento de incentivos fiscales especiales, etc.; son algunos de los factores, entre muchos otros, que dificultan el cálculo de los flujos de efectivo del negocio netos.

Se han hecho algunas generalizaciones con objeto de entender, de manera sencilla y directa, cómo deben incorporarse los aspectos impositivos en la determinación de los flujos de efectivo necesarios para evaluar proyectos de inversión. Asimismo, debido a que el impuesto sobre la renta tiene un impacto muy importante sobre los flujos de efectivo operativos, concentremos nuestra atención fundamentalmente en este tipo de impuesto

El estado financiero que muestra el resultado (medido en términos de utilidad neta) de las operaciones de un negocio o proyecto, desde el punto de vista contable, es el denominado Estado de Resultados o Estado de Pérdidas y Ganancias. A continuación se presenta un Estado de Resultados típico de un negocio:

ESTADO DE RESULTADOS

Ventas brutas

- Devoluciones y rebajas sobre ventas

= Ventas netas

- Costo de ventas

= Utilidad neta

- Gastos de venta

- Gastos generales y de administración

= Utilidad de operación

ESTADO DE RESULTADOS

- Gastos y productos financieros

- Otros gastos y productos

= Utilidad antes de impuestos

- Impuestos

= Utilidad neta

A partir de la utilidad antes de impuestos determinada en el Estado de Resultados, es posible calcular el monto de impuestos (por concepto de impuesto sobre la renta) por cubrir a las autoridades fiscales en un determinado ejercicio fiscal (de acuerdo con las actuales disposiciones fiscales, los ejercicios fiscales se miden en años calendario, es decir de enero a diciembre, con excepción del ejercicio inicial y el ejercicio de liquidación o terminación del negocio). Para ello, se sigue el siguiente procedimiento:

Impuesto sobre la Renta = Utilidad gravable x Tasa de impuestos _ _ _ (10)

Utilidad gravable = Ingresos acumulables – (Deducciones + Resultado fiscal + Pérdidas ejercicios anteriores)_ _ (11)

Determinación de los flujos de efectivo operativos a partir del estado de resultados

La utilidad neta que aparece en el último renglón del Estado de Resultados proporciona una medida contable del beneficio neto que genera el negocio o proyecto. Sin embargo, la utilidad neta no es igual al flujo de efectivo del negocio. Asimismo, la utilidad neta no es igual al flujo de efectivo operativo neto.

UTILIDAD NETA

≠ FLUJO DE EFECTIVO DEL NEGOCIO

UTILIDAD NETA

≠ FLUJO DE EFECTIVO OPERATIVO NETO

La diferencia fundamental consiste en que, en tanto los flujos de efectivo del negocio consideran al total de la salida de dinero efectiva asociada a la adquisición de activos fijos (inversión neta), en el momento en que ocurre; la Utilidad Neta que se muestra en el Estado de Resultados considera únicamente a la depreciación del periodo como un egreso (estamos utilizando el término depreciación para referirnos a todos los movimientos contables que no producen efectos directos en los flujos de efectivo).

Asimismo, la depreciación no es un flujo de efectivo operativo (entrada o salida de dinero), por lo tanto no esta incluida en la determinación de los flujos de efectivo operativos. Sin embargo, dado que la depreciación es deducible de impuestos, esto ocasiona un ahorro fiscal que afecta indirectamente al flujo de efectivo operativo, por lo cual únicamente el ahorro fiscal, que se origina por la depreciación, se considera una entrada de dinero, que se incluye en la determinación de los flujos de efectivo operativos netos.

Por lo anteriormente expuesto, para determinar el flujo el flujo de efectivo operativo neto, a partir del Estado de Resultados, bastará con sumar la depreciación correspondiente a cada periodo de la vida económica del negocio o proyecto, a la utilidad neta de dicho periodo.

Ejemplo 6:

Determinemos el flujo de efectivo operativo neto que un negocio generará, a partir del Estado de Resultados de una empresa "X" que se presenta a continuación, así como el flujo de efectivo del negocio neto.

EMPRESA "X", ESTADO DE RESULTADOS COMPARATIVO

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5
Ventas netas	\$ 162.0	\$ 279.0	\$ 318.0	\$ 366.0	\$ 390.0
Ganancia en venta de activos fijos	\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0	\$ 40.0
Total de ingresos	\$ 162.0	\$ 279.0	\$ 318.0	\$ 366.0	\$ 430.0
Costos excluyendo depreciación	\$ 122.7	\$ 186.0	\$ 195.5	\$ 206.1	\$ 217.7
Depreciación	\$ 22.0	\$ 22.0	\$ 22.0	\$ 22.0	\$ 22.0
Total de costos	\$ 144.7	\$ 208.0	\$ 217.5	\$ 228.1	\$ 239.7
Utilidad antes de impuestos	\$ 17.3	\$ 71.0	\$ 100.5	\$ 137.9	\$ 190.3
Impuestos	\$ 6.1	\$ 24.9	\$ 35.2	\$ 48.3	\$ 66.6
Utilidad neta	\$ 11.2	\$ 46.1	\$ 65.3	\$ 89.6	\$ 123.7

Los flujos de efectivo del negocio netos se calcularían de la siguiente forma:

	PERIODO 0	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5
Utilidad neta		\$ 112.0	\$ 46.1	\$ 65.3	\$ 89.6	\$ 123.7
+ Depreciación		\$ 22.0	\$ 22.0	\$ 22.0	\$ 22.0	\$ 22.0
- Ganancia por venta de activos fijos		\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0	\$ -40.0
Flujo de efectivo operativo neto		\$ 33.2	\$ 68.1	\$ 87.3	\$ 111.6	\$ 105.7

	PERIODO 0	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5
Inversión neta						
Compra de activos fijos	(\$ 220.0)	\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0
+ Cambios en capital de trabajo	(\$ 3.0)	(\$ 3.0)	(\$ 0.6)	(\$ 0.6)	(\$ 0.8)	\$ 0.0
- valor de rescate	\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0	\$ 0.0	\$ 158.0
Inversión neta	(\$ 223.0)	(\$ 3.0)	(\$ 0.6)	(\$ 0.6)	(\$ 0.8)	\$ 158.0
Flujo de efectivo operativo neto	(\$ 223.0)	\$ 30.2	\$ 67.5	\$ 86.7	\$ 110.8	\$ 263.7

Como puede observarse en el cuadro, tanto la ganancia en la venta de activo fijo como la depreciación, no producen un efecto directo en los flujos de efectivo. La ganancia en la venta incrementa la utilidad neta calculada en el Estado de Resultados, pero no representa una entrada de dinero, ya que dicha entrada de dinero está considerada en los \$ 150 millones que calculamos como valor de rescate del activo fijo. Por lo tanto, este monto (\$ 150 millones) debe ser restado de la utilidad neta, dado que lo que interesa, para efectos de valuación, es únicamente su impacto en el flujo de efectivo debido al ahorro fiscal producto de las disposiciones fiscales.

Como puede apreciarse, la obtención del flujo de efectivo operativo neto, a partir del Estado de Resultados, produce el mismo resultado que su obtención a través de identificar las entradas y salidas de dinero que se producen en cada periodo de vida económica.

Capítulo 4. Medidas de rentabilidad

Los tres métodos más importantes de medición de rentabilidad son:

- El periodo de recuperación ajustado de la inversión.
- El valor presente neto.
- La tasa interna de rendimiento.

Cada uno de ellos proporciona un criterio de selección de proyectos distinto, puesto que proporciona una medida de rentabilidad diferente. Para el caso de proyectos independientes, estos criterios son:

CRITERIO DE RENTABILIDAD

MÉTODO	UN PROYECTO ES RENTABLE SI:
Periodo de recuperación ajustado de la inversión	El periodo de recuperación del negocio ó proyecto es menor a la vida económica del negocio o proyecto
Valor presente neto	El valor presente neto del negocio o proyecto es positivo
Tasa de rendimiento	La tasa de rendimiento es mayor al costo del capital

A pesar de que los tres métodos utilizan el valor del dinero en el tiempo, no siempre proporcionan la misma decisión de aceptación o rechazo. Cuando ocurre esta discrepancia, el método correcto es el del valor presente neto.

4.1. Periodo de recuperación de la inversión.

El periodo de recuperación de la inversión es el tiempo que necesita el empresario para poder recuperar el dinero que invirtió inicialmente en un negocio o proyecto.

Podemos también pensar en el tiempo de recuperación de la inversión como el tiempo que le toma a la operación del negocio o proyecto generar el suficiente flujo de efectivo para compensar o cubrir la inversión realizada por el empresario.

Al periodo de recuperación se conoce también con el nombre de tiempo de recuperación. Existen dos formas de calcularlo. La primera forma se conoce como periodo de recuperación simple y la segunda como periodo de recuperación ajustado.

Periodo de recuperación simple:

El tiempo o periodo de recuperación simple es el tiempo que se necesita para poder recuperar el dinero que se invirtió inicialmente en un negocio o proyecto.

El tiempo de recuperación simple compara directamente los flujos de efectivo operativos netos generados por un negocio o proyecto, con la inversión neta, para determinar el periodo (número de años, meses, semanas o días) que se requiere para que el dinero que genera el proyecto, sea igual al dinero que se invirtió para iniciar y mantener operando el negocio o proyecto.

El tiempo de recuperación simple considera los flujos de efectivo sin tomar en cuenta el valor del dinero en el tiempo. Por lo tanto, considera, erróneamente, que un peso hoy vale igual que un peso el día de mañana.

Ejemplo 7.

Supóngase que un negocio requiere para iniciar, una inversión neta en maquinaria y capital de trabajo de \$ 9 millones de pesos. La vida económica de la maquinaria del negocio es de seis años. El negocio generará un flujo de efectivo operativo neto de \$ 2 millones el primer año de operación. Se estima que en el segundo y tercer año el negocio tendrá un volumen mayor de ventas. Por este motivo, se espera que el flujo de efectivo operativo neto para el segundo año sea de \$ 3 millones, y para el tercer año de \$ 4 millones. De acuerdo con las proyecciones para los años siguientes se confía que el nivel de ventas se mantendrá constante y lo mismo ocurrirá con los flujos de efectivo operativos netos, es decir, se sostendrán en \$ 4 millones por año.

¿En cuánto tiempo se recuperará la inversión?

Para responder a esta pregunta los que debemos hacer es calcular los flujos de efectivo del negocio e irlos acumulando año con año. De acuerdo con el criterio del periodo de recuperación simple, en el momento que el flujo de efectivo acumulado sea igual a cero habremos recuperado la inversión. En este caso el periodo de recuperación de la inversión es de tres años, tal como se indica en la siguiente tabla:

AÑO	FLUJO ANUAL DE EFECTIVO (MM)¹	FLUJO DE EFECTIVO CUMULADO (MM)
0	\$ (9)	\$ (9)
1	\$ 2	\$ (7)
2	\$ 3	\$ (4)
3	\$ 4	\$ 0
4	\$ 4	\$ 4
5	\$ 4	\$ 8
6	\$ 4	\$ 12

No debemos perder de vista que el periodo de recuperación simple se utiliza para tomar decisiones de inversión, es decir, para decidir si debo aceptar o rechazar un negocio o proyecto. ¿Cuáles son estos criterios de decisión?

Criterios de decisión

Situación 1

Si el empresario se encuentra analizando un negocio o proyecto en forma aislada o un proyecto independiente, es decir, sin considerar otros posibles negocios o proyectos simultáneamente, el criterio de decisión que debe seguirse es el siguiente:

Se acepta el negocio o proyecto si el tiempo de recuperación simple es menor que la vida económica del proyecto.

Es fácil entender este criterio. Si el tiempo de recuperación simple es menor que la vida económica del proyecto, esto significa que el dinero invertido en el negocio va a ser completamente recuperado antes que termine la vida económica del mismo, por lo que, "sobrarán" dinero. Este "sobrante" o exceso de dinero (en el ejemplo anterior es igual a \$ 12 millones) se producirá a partir de que se recupere completamente la inversión y hasta que termine la vida económica del proyecto o negocio.

Situación 2

Si el empresario se encuentra evaluando varias opciones de negocio o proyectos que son mutuamente excluyentes entonces, de acuerdo con el método del periodo de recuperación simple:

Se acepta el negocio o proyecto que tenga el menor tiempo de recuperación de la inversión.

Es fácil entender la lógica de este criterio. En la medida en que un proyecto o negocio genere rápidamente los flujos de efectivo operativos netos requeridos para recuperar la inversión neta, en esa misma medida el empresario dispondrá de flujos de efectivo libres utilizables para la reinversión o consumo.

Obviamente, si los negocios o proyectos que son analizados tienen tiempos de recuperación superiores a la vida económica de los mismos, lo más conveniente será rechazarlos, de acuerdo al primer criterio presentado.

Así pues, el tiempo de recuperación de la inversión es muy importante por dos razones:

1. Siempre es mejor recibir un beneficio hoy que mañana.
A mayor velocidad de recuperación (menor tiempo de recuperación), se dispondrá más rápidamente de fondos para su utilización productiva o su consumo.

2. Mientras más alejado en el tiempo está el beneficio, existe una mayor incertidumbre (riesgo) de recibirlo. Por lo tanto, un menor tiempo de recuperación de un negocio o proyecto implicaría un menor riesgo para el empresario. Desde esta perspectiva, el tiempo de recuperación de la inversión es una medida del tiempo que el empresario estaría expuesto al riesgo de no recuperar (perder parcial o totalmente) su inversión neta.

Ventajas y desventajas del método simple.

Las principales ventajas del criterio del tiempo de recuperación simple son:

1. Es fácil de entender el concepto y transmitirlo a otros.
2. Es de fácil cálculo.
3. Proporciona una medida del riesgo del negocio o proyecto.

La principal desventaja del método simple para calcular el tiempo o periodo de recuperación de una inversión es que no considera el valor del dinero en el tiempo. Es decir, este método no diferencia entre el valor de un flujo de efectivo de \$ 1 durante el primer año, y el valor del mismo flujo de \$ 1 en un año posterior. Ello puede conducir a aceptar proyectos erróneamente.

Adicionalmente, este método no considera los flujos de efectivo operativos netos que se presentan después de que la inversión neta ha sido recuperada, puesto que únicamente indica cuando se recupera la inversión.

Los flujos de efectivo operativos netos que ocurren después de que se recupera la inversión pueden ser muy importantes en la determinación de la rentabilidad de un negocio, por lo que el negocio o proyecto con el menor tiempo de recuperación no necesariamente es el negocio o proyecto más rentable o el que debiera seleccionarse.

Periodo de recuperación ajustado

El periodo de recuperación ajustado se define como el tiempo que se requiere para recuperar el valor presente del dinero que se invirtió inicialmente en un negocio o proyecto.

El tiempo de recuperación ajustado utiliza los flujos de efectivo tomando en cuenta el valor del dinero en el tiempo.

El método del periodo de recuperación ajustado consiste en comparar el valor presente de cada uno de los flujos de efectivo operativos netos con el valor presente de la inversión neta requerida. Es decir, este método calcula el tiempo

que se necesita para que el dinero que entra al negocio, ajustado por su valor en el tiempo, sea igual al monto originalmente invertido, también ajustado por su valor en el tiempo.

Ejemplo 8.

Determinemos el periodo de recuperación ajustado para el ejemplo 7. A continuación presentamos en el cuadro que habíamos preparado para presentar los flujos de efectivo que estima podría generar el proyecto.

FLUJO ANUAL DE EFECTIVO	
AÑO	(MM)
0	\$ (9)
1	\$ 2
2	\$ 3
3	\$ 4
4	\$ 4
5	\$ 4
6	\$ 4
Total	
	\$ 21

Supongamos que el costo de capital del negocio es del 10%.

Para determinar el valor presente de cada uno de los flujos de efectivo, se obtiene primero el factor de descuento aplicando la ecuación (4).

Una vez determinados estos factores, es posible calcular el valor presente de los flujos de efectivo netos, multiplicando cada flujo de efectivo por el factor correspondiente, como se observa a continuación:

Año	Flujo de efectivo neto (mm)¹	Factor de descuento	Valor presente del flujo de efectivo neto	Valor presente del flujo de efectivo neto acumulado
0	\$ (9)	1.0000	\$ (9)	\$ (9)
1	\$ 2	0.9091	\$ 1.818	\$ (7.182)
2	\$ 3	0.8264	\$ 2.479	\$ (4.703)
3	\$ 4	0.7513	\$ 3.005	\$ (1.698)
4	\$ 4	0.6830	\$ 2.732	\$ 1.034
5	\$ 4	0.6209	\$ 2.484	\$ 3.518
6	\$ 4	0.5645	\$ 2.258	\$ 5.776

¹MM = Millones de pesos

Como la inversión neta ocurre en el periodo 0, el cual es le momento de tomar la decisión de implementar este proyecto, este monto ya se encuentra expresado en términos de valor presente, por lo cual el factor de descuento es igual a 1.000.

En el cuadro podemos observar que el periodo de recuperación ajustado debe situarse entre el tercer y cuarto año, ya que al término del tercer año faltan por ser recuperados \$ 1.6988 millones de la inversión original de \$ 9 millones, y al final del cuarto año, ya se recuperó el total de la inversión neta y se ha generado un excedente de dinero de \$ 1.034 millones.

Para determinar con mayor precisión el tiempo de recuperación de la inversión ajustado dividimos el valor presente del monto que falta por recuperar en el tercer año entre el valor presente del flujo efectivo del cuarto año, ya que en este año cuando se termina de recuperar la inversión.

$$\frac{1.698}{2.732} = 0.62 \text{ años}$$

Dado que un año tiene 52 semanas, 0.62 años equivale aproximadamente a 32 semanas ($0.62 \times 52 = 32.3$), o sea aproximadamente a ocho meses ($32.3 / 4 = 8$). Por lo tanto, de acuerdo al método del tiempo de recuperación ajustado, se recuperará su inversión en tres años y ocho meses.

Si comparamos este resultado con el obtenido a través del método del tiempo de recuperación simple, que fue de tres años (calculado en el ejemplo 1), observamos que al considerar el valor del dinero en el tiempo, el tiempo de recuperación de la inversión se incrementó en ocho meses.

El método del tiempo de recuperación ajustado de la inversión proporciona una medida más precisa que el método simple puesto que incorpora el valor del dinero en el tiempo.

El criterio de decisión utilizando este método es similar al anteriormente expresado:

Se acepta el negocio o proyecto que tenga el menor tiempo ajustado de recuperación de la inversión.

El tiempo de recuperación ajustado de la inversión también puede entenderse como una medida de la liquidez de un negocio o proyecto, si entendemos por liquidez la capacidad de generación de recursos líquidos (dinero) con los que cuenta el empresario para hacer frente a sus obligaciones de corto plazo (pasivos circulantes). En la medida que el empresario recupere rápidamente el dinero invertido en un negocio o proyecto, en esa misma medida dispondrá de dinero para pagar sus obligaciones (a proveedores, deudores diversos, etc.), o para reinvertirlo en otras oportunidades de negocio. Por lo tanto, el negocio o proyecto con un menor tiempo de recuperación de la inversión, podría decirse que es un proyecto o negocio más líquido.

Ventajas y desventajas del método ajustado

Las principales ventajas que ofrece el método del tiempo de recuperación ajustado de la inversión como criterio de evaluación de proyectos son:

1. Es fácil de entender el concepto y transmitirlo a otros.
2. Es de fácil cálculo.
3. Da una medida (en tiempo) del riesgo del negocio o proyecto.
4. Considera el valor del dinero en el tiempo.
5. Proporciona una medida de liquidez.

A pesar de estas ventajas, el tiempo de recuperación ajustado, tampoco toma en consideración todos los flujos de efectivo operativos netos que se generan después de recuperar la inversión. Por lo tanto, la utilización del criterio del tiempo de recuperación ajustado para tomar decisiones de inversión, no necesariamente conduce a seleccionar los proyectos o negocios más rentables. Para ello es necesario determinar la tasa interna de rendimiento y el valor presente neto de la inversión.

4.2. Valor presente neto (VPN)

El método de Valor Presente Neto consiste en determinar el valor presente de los flujos de efectivo netos que un proyecto genera en el tiempo, con el fin de comparar correctamente los flujos de efectivo en diferentes periodos a lo largo de la vida económica de un proyecto.

El valor presente neto de un flujo de efectivo se define como la suma de los valores presentes netos de los flujos de efectivo de cada periodo de la vida económica del proyecto, tal como se expresa en la siguiente ecuación:

$$VPN = S_0 + \sum_{t=1}^n \frac{S_t}{(1+i)^t} \quad \text{-----} \quad (12)$$

Donde:

VPN = Valor presente neto de los flujos de efectivo;

S_0 = Inversión inicial;

S_t = Flujo de efectivo neto del periodo t ;

n = número de periodos de la vida económica del proyecto;

i = Tasa de recuperación mínima atractiva (TREMA) o tasa de descuento (T_d).

La tasa de descuento o TREMA es la tasa mínima a la que un inversionista decide arriesgar su dinero. La selección de la misma depende finalmente, del inversionista y se basa, en la mayoría de los casos, en la tolerancia al riesgo por parte de los inversionistas. Como ya mencionamos, un parámetro para seleccionar una tasa de recuperación adecuada suele ser el rendimiento de los mercados de capital.

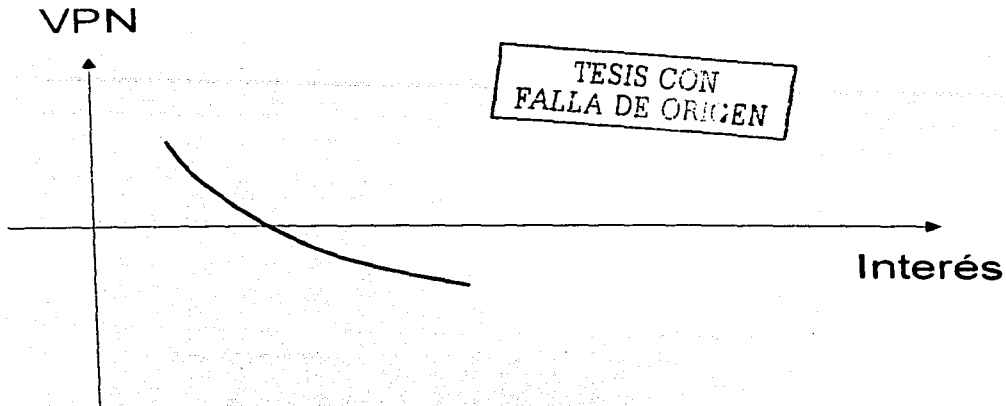
Ejemplo 9.

Supóngase un proyecto con una tasa de rendimiento mínimo aceptable de 15% tiene el flujo de efectivo que se muestra en la tabla siguiente:

Año (t)	0	1	2	3	4	5
Flujo de efectivo, S_t	-6,000	4,000	2,000	-3,000	-2,000	-3,000
Valor presente	-6,000	3,478	1,512	-1,973	-1,144	1,492
Valor presente acumulado	-6,000	-2,522	-1,010	-2,983	-4,127	-2,635

El valor presente neto acumulado del flujo de efectivo es igual -2,635. ¿Debe aceptarse el proyecto? Para resolver a esta pregunta analicemos la relación entre el valor presente neto y la TREMA.

Como podemos observar de la ecuación (12), la relación entre el valor presente neto y la tasa de rendimiento mínima aceptable (tasa de descuento), es una relación inversa. Ello quiere decir que para un flujo de efectivo dado, a menor VPN, mayor debe ser la tasa de rendimiento. Y a la inversa, a mayor valor presente neto, menor tasa de interés. (Esta relación puede apreciarse en la gráfica 1).



Gráfica 1. Relación entre el VPN y la tasa de descuento.

Si el valor presente neto de una inversión es igual a cero, ello implica que el rendimiento de la inversión es igual a la tasa mínima de recuperación aceptable (TREMA).

Análogamente:

Si $VPN > 0$ entonces $\text{rendimiento} < \text{TREMA}$, por lo tanto el proyecto debe aceptarse

Si $VPN < 0$ entonces $\text{rendimiento} > \text{TREMA}$, por lo tanto el proyecto debe rechazarse.

Si el Valor Presente de las entradas de dinero es mayor que el valor presente de las salidas de dinero, de un negocio o proyecto, dicho negocio o proyecto es rentable. Si el valor presente de las entradas de dinero es menor que el valor presente de las salidas de dinero, dicho negocio o proyecto no es rentable

Por rentabilidad entendemos: el cambio porcentual entre la riqueza inicial (cantidad de dinero disponible al empresario al inicio del proyecto o negocio) y la riqueza final (cantidad de dinero disponible al término de la vida económica del negocio o proyecto).

Si el valor presente neto del flujo de efectivo del negocio es positivo, el negocio es rentable; si es negativo, el negocio no es rentable.

Ventajas y desventajas del Valor Presente Neto

Las principales ventajas del método del Valor Presente Neto son:

- Utiliza el concepto del valor del dinero en el tiempo
- Siempre proporciona la decisión correcta de aceptación o rechazo de proyecto.

La principal desventaja del método del Valor Presente Neto es que es muy sensible al valor de la tasa de interés utilizado. Por lo tanto, requiere de una buena estimación de la tasa de interés esperada con la cual se van a "descontar" los flujos de efectivo futuros.

4.3. Tasa Interna de Rendimiento (TIR)

La Tasa Interna de Rendimiento (TIR) proporciona otra medida de la rentabilidad de un negocio o proyecto.

La Tasa Interna de Rendimiento de un negocio o proyecto equivale a la tasa de interés que dicho negocio o proyecto le van a dar a la persona que invirtió ahí su dinero.

Así como la tasa de interés de un banco proporciona una medida de la rentabilidad de la inversión en dicho banco, así también la Tasa Interna de Rendimiento de un negocio o proyecto proporciona otra medida de la rentabilidad de la inversión en dicho negocio.

La Tasa Interna de Rendimiento es la tasa de interés que hace que el valor presente del flujo de efectivo operativo neto sea igual al valor presente de la inversión neta. Expresado en otros términos:

La tasa Interna de Rendimiento (TIR) es a tasa de interés que hace que el Valor Presente Neto de los flujos de efectivo del negocio o proyecto sea igual a cero.

Metodología de cálculo

Dado que la relación entre el valor presente neto y la tasa de interés (tasa de descuento) es implícita, el cálculo de la TIR es iterativo. Ello quiere decir que primero es necesario suponer una tasa de interés y con ella calcular el VPN. Si este es negativo, deberemos escoger una tasa de interés menor (debido a la relación inversa entre VPN y i). Si el VPN es positivo, deberemos seleccionar una tasa de interés mayor.

Este proceso se repite cuantas veces sea necesario (a este proceso de cálculos repetitivos se le conoce como "Aproximaciones sucesivas"), hasta que se obtenga un valor presente neto igual a cero.

La tasa de interés con la cual se obtiene este Valor Presente Neto de los flujos de efectivo del negocio o proyecto igual a cero, será la tasa interna de rendimiento del negocio o proyecto.

Con el fin de reducir el número de aproximaciones sucesivas, se sugiere que una vez que se hayan obtenido Valores Presentes Netos muy cercanos a cero, mediante interpolación, se encuentre la tasa de interés que da un Valor Presente Neto igual a cero.

En el mercado existen actualmente un gran número de calculadoras de bolsillo que realizan estas aproximaciones sucesivas necesarias para encontrar la TIR. El programa que desarrolla esta tesis requiere solamente de proporcionar los flujos de efectivo del proyecto para obtener la TIR.

Ejemplo 10:

Un nuevo proyecto tiene flujos de efectivo netos de \$ 25 millones al iniciar el proyecto y de \$ 10 millones anuales, a lo largo de sus cuatro años de vida económica.

Determine la tasa interna de rendimiento (TIR) del proyecto.

Seleccionemos arbitrariamente una tasa de interés del 10% y calculemos el Valor Presente Neto de los flujos de efectivo del proyecto:

AÑO	FLUJO DE EFECTIVO DEL PROYECTO (MM)	FACTOR (10%)^t	VALOR PRESENTE NETO (MM)
0	\$ (25.00)	1.000	\$ (25.00)
1	\$ 10.00	0.909	\$ 9.10
2	\$ 10.00	0.826	\$ 8.30
3	\$ 10.00	0.751	\$ 7.50
4	\$ 10.00	0.683	\$ 6.80
Total			\$ 6.70

Dado que el valor presente neto, para una tasa del 10%, es positivo (\$ 6.7 millones), tendremos que incrementar el valor de la tasa de interés.

Seleccionemos una tasa de interés del 20 por ciento:

AÑO	FLUJO DE EFECTIVO DEL PROYECTO (MM)	FACTOR (20%)	VALOR PRESENTE NETO (MM)
0	\$ (25.00)	1.000	\$ (25.00)
1	\$ 10.00	0.833	\$ 8.30
2	\$ 10.00	0.694	\$ 6.90
3	\$ 10.00	0.579	\$ 5.80
4	\$ 10.00	0.482	\$ 4.80
Total			\$ 0.80

Dado que el valor presente neto sigue siendo positivo (\$ 0.8 millones), tendremos que incrementar aún más la tasa de interés. Sin embargo, dado que el Valor Presente Neto está muy cercano a cero, dicho incremento a la tasa de interés deberá ser pequeño.

Seleccionemos una tasa de interés del 22 por ciento.

AÑO	FLUJO DE EFECTIVO DEL PROYECTO (MM)	FACTOR (20%)	VALOR PRESENTE NETO (MM)
0	\$ (25.00)	1.000	\$ (25.00)
1	\$ 10.00	0.820	\$ 8.20
2	\$ 10.00	0.672	\$ 6.70
3	\$ 10.00	0.551	\$ 5.50
4	\$ 10.00	0.451	\$ 4.50
Total			\$ 0.10

Dado que le Valor Presente Neto es ligeramente negativo (-\$ 0.10 millones), sabemos entonces que la Tasa Interna de Rendimiento del proyecto está entre 20% y 22%. Dado que los Valores Presentes Netos de ambas tasas son bastante cercanos a cero, para encontrar la Tasa Interna de Rendimiento bastará con hacer una interpolación entre ambos valores:

	TASA DE INTERÉS (%)	VALOR PRESENTE NETO (MM)
	20	\$ 0.80
	22	(\$ 0.10)
DIFERENCIA	2	\$ 0.90

TESIS CON
FALLA DE ORIGEN

$$TIR = \text{tasa interna inicial} = \frac{\text{Valor Presente Neto inicial}}{\text{Diferencia de valores presente netos} \times [\text{diferencias netas}]}$$

$$TIR = 20\% + \frac{\$ 0.80}{\$ 0.90} \times [2\%] = 21.8\%$$

Supongamos que el nuevo proyecto del ejemplo anterior no involucra riesgo para el inversionista.

Si el banco le ofrece al inversionista una tasa de interés del 15%, ¿debería el inversionista invertir su dinero en el negocio o depositarlo en el banco?

Si el inversionista percibe que la opción de invertir en el nuevo proyecto tiene un riesgo similar a la opción de invertir en el banco, el inversionista seleccionará la opción que le ofrezca la mayor tasa de interés (el mayor rendimiento).

El nuevo negocio le ofrece una tasa de interés del 21.8%; el banco ofrece una tasa de interés del 15 por ciento.

Por lo tanto, el inversionista deberá invertir su dinero en un nuevo negocio si el inversionista hubiera percibido que el nuevo proyecto es mucho más riesgoso que tener su dinero en el banco, éste podría decidirse por depositar su dinero en el banco, aunque este último le ofrezca un menor rendimiento.

Un proyecto es aceptado, si la Tasa Interna de Rendimiento es superior al costo de capital.

Ventajas y desventajas de la Tasa Interna de Rendimiento TIR

Las principales ventajas del método de la Tasa Interna de Rendimiento son:

- Es fácil para la mayoría de las personas entender la TIR (ya que su resultado está dado por una tasa de interés), que el resultado del VPN (ya que su resultado está dado por un monto en pesos).

Por ejemplo, es más fácil explicarle a una persona que determinado negocio o proyecto da una tasa de interés (TIR) del 12%, que explicarle a la misma persona que dicho proyecto tiene un Valor Presente Neto de \$ 27.2 millones.

- Utiliza el concepto del Valor de Dinero en el Tiempo.

- La principal desventaja del método de la Tasa Interna de Rendimiento es que puede dar una decisión errónea cuando se tienen proyectos mutuamente excluyentes.

Capítulo 5. La herramienta de programación.

5.1 - Fundamentos de Visual Basic.

El diseño de aplicaciones ha cambiado drásticamente en los últimos años y consecuentemente los estilos de programación. De una programación secuencial se pasó a una programación modular, para llegar a una programación estructurada con muy buenos resultados. A partir de la programación estructurada los esfuerzos se centraron en potenciar la modalidad y la reutilización del código. Ello condujo a la programación orientada a objetos que se complementa con una programación basada en componentes de software. Pero que los estilos hayan cambiado, no significa que el último elimine los anteriores; al contrario, se apoya en ellos.

El lenguaje de programación BASIC (*Beginner's All purpose Symbolic Instruction Code*) nació en el año 1964 como una herramienta destinada a principiantes, buscando una forma sencilla de realizar programas, empleando un lenguaje casi igual al usado en la vida ordinaria (en inglés), y con instrucciones muy sencillas y escasas. Teniendo en cuenta el año de su nacimiento, este lenguaje cubría casi todas las necesidades para la ejecución de programas. Téngase en cuenta que las máquinas existentes en aquella época estaban estrenando los transistores como elementos de conmutación, los ciclos de trabajo llegaban a la impensable cifra de 10.000 por segundo y la memoria no pasaba de unos pocos k's, en toroides de ferrita.

La evolución del BASIC por los años 70 fue escasa dado el auge que tomaron en aquella época lenguajes de alto nivel como el FORTRAN y el COBOL. En 1978 se definió una norma para unificar los Basics existentes creándose la normativa BASIC STANDARD

Con la aparición de las primeras computadoras personales dedicadas comercialmente al usuario particular allá por la primera mitad de los ochenta, el BASIC resurgió como lenguaje de programación para principiantes, y muchos de estas pequeñas computadoras domésticas lo usaban como único sistema operativo.

Con la popularización de la PC, surgieron varias versiones del BASIC que funcionaban en este tipo de computadoras (Versiones BASICA, GW-BASIC), pero todas estas versiones del BASIC no hicieron otra cosa que terminar de liquidar BASIC. Los programadores profesionales no llegaron a utilizarlo, habida cuenta de las desventajas de éste respecto de otras herramientas (PASCAL, C, CLIPPER). El BASIC con estas versiones para PC llegó incluso a perder crédito entre los profesionales de la informática.

Las razones para ello eran obvias:

- No era un lenguaje estructurado. (**Un lenguaje es estructurado cuando todo programa puede ser diseñado utilizando únicamente las estructuras básicas: secuencial, alternativa, repetitiva.** Un programa, por complejo que sea, se puede diseñar anidando convenientemente estas tres estructuras básicas.)
- No existían herramientas de compilación fiables.
- No disponía de herramientas de intercambio de información.
- No tenía librerías.
- No se podía acceder al interior de la máquina.

Tal fue el abandono por parte de los usuarios, que la aparición del Quick-BASIC de Microsoft, una versión ya potente del BASIC, que corregía casi todos los defectos de las versiones anteriores pasó prácticamente inadvertida, a no ser porque las últimas versiones del sistema operativo MS-DOS incluían una versión de Quick-BASIC algo recortada (Q-Basic) como un producto mas dentro de la amplia gama de ficheros ejecutables que acompañan al sistema operativo, y aprovecha el editor de textos (Cada vez que se llama al EDIT estamos corriendo el editor del Q-Basic).

Esta versión del popular BASIC es ya un lenguaje estructurado, lo que permite crear programas modularmente mediante subrutinas y módulos, capaz de crear programas ya competitivos con otros lenguajes de alto nivel. Sin embargo llegaba tarde, pues los entornos MS-DOS estaban ya superados por el entorno gráfico Windows.

Sin embargo algo había en el BASIC que tentaba a superarse: su gran sencillez de manejo. Si a esto se le añade el entorno gráfico Windows, el aprovechamiento al máximo de las posibilidades de Windows en cuanto a intercambio de información, de sus librerías, de sus *drivers* y controladores, manejo de bases de datos, etc.; el producto resultante puede ser algo que satisfaga todas las necesidades de programación en el entorno Windows. La suma de todas estas características es VISUAL - BASIC. Esta herramienta conserva del BASIC de los años 80 únicamente su nombre y su sencillez, y tras su lanzamiento al mercado, la aceptación en un nivel profesional hizo borrar por fin el "mal nombre" asociado a la palabra BASIC.

Actualmente se comercializa la versión 6.0 de este producto, que es la que se emplea en el presente trabajo. Desde su salida al mercado, cada versión supera y mejora la anterior. Dados los buenos resultados en el nivel profesional de este producto, y el apoyo prestado por el fabricante para la formación de programadores, Visual-Basic se ha convertido en la primera herramienta de desarrollo de aplicaciones en entorno Windows.

Es obligado decir sin embargo, que sigue siendo BASIC. No se pueden comparar sus prestaciones con otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros. No es ese el fin perseguido con VB y si es necesario llegar a esas precisiones, será necesario utilizar otro lenguaje que permita bajar el nivel de programación (Visual-C), o realizar librerías (DLLs) que lo hagan. No obstante, en la mayor parte de las aplicaciones, las herramientas aportadas por VB son mas que suficiente para lograr un programa fácil de realizar y de altas prestaciones.

Visual-Basic es una herramienta de diseño de aplicaciones para Windows, en la que éstas se desarrollan en una gran medida a partir del diseño de una interfaz gráfica.

5.2 La programación orientada a objetos.

El tema de la orientación a objetos no es nuevo. Sin embargo, en los últimos años ha tenido un impacto tan grande en diferentes áreas de la computación que se hace obligatorio, para cualquier persona relacionada con esta disciplina, el manejo de los conceptos involucrados. De la misma forma ha surgido una proliferación de definiciones e interpretaciones que conducen, primero a confusión, y luego a malas interpretaciones y abusos de los términos..

Cuando Fortran apareció en 1957 como una forma de automatizar la labor de aquellos que programaban en lenguaje máquina y en ensamblador, hubo mucho escepticismo. Había que probar que un proceso automático de traducción podía competir con la eficiencia y el ingenio que aplicaban aquellos programadores para poder ahorrar un microsegundo o una unidad de almacenamiento; y así fue.

Los lenguajes de la primera generación como Fortran I y Algol 58 estaban orientados a facilitar la manipulación de expresiones matemáticas y liberar a los programadores de los incómodos detalles que involucra (y más en ese entonces) el lidiar con la máquina en forma directa. Aunque manejaban el concepto de subprograma, este era aplicado más que todo para reutilización. Además, los programas eran relativamente planos, en el sentido de que consistían de una serie de subprogramas que referenciaban datos globales. Es fácil deducir que en programas de no necesariamente gran complejidad, empezarán a darse problemas serios si un sólo subprograma no funcionaba correctamente.

Para 1960, la popularización del transistor y la tecnología de circuitos integrados hicieron posible que los costos de las computadoras disminuyeran y al mismo tiempo fuera posible aumentar su capacidad. Surgieron entonces lenguajes como Cobol y Algol 60, enmarcados dentro de la segunda generación. A diferencia de sus antecesores, éstos empezaron a dar mayor importancia al concepto de subprograma como un medio para realizar funciones abstractas. Es decir, al afrontar un problema se idealizaba un conjunto de acciones macro para

solucionarlo, y estas a su vez serían atacadas como nuevos problemas, hasta llegar a completar la solución final. Lo anterior condujo al anidamiento de subprogramas, y al desarrollo de teorías referentes a estructuras de control, y al alcance y visibilidad de variables. Es decir, se construyeron las bases de la programación estructurada, aplicando una abstracción de los procedimientos requeridos.

Posteriormente, en la *tercera generación* se incluyó el concepto de módulo, como una forma de satisfacer la necesidad de manejar partes independientes de la solución que eran atacadas por diferentes personas, dado que los programas eran cada vez más complejos y más voluminosos.

Sin embargo, aunque ya se había trabajado bastante en la abstracción al nivel de módulos y de procedimientos, no se había hecho mucho en lo que se refería a la abstracción al nivel de datos. Aunque lenguajes como Cobol daban mucha importancia a los datos dentro de las aplicaciones, existía una separación clara entre datos y procedimientos. La solución era trazada con base en funciones, no en los datos mismos.

Incluso, al aplicar una metodología clásica de análisis y diseño estructurado, es posible tomar dos enfoques: 1) *realizar primero el estudio y la especificación de las funciones requeridas* y 2) *luego formular el diagrama de datos*, o al contrario, iniciar primero con el diseño del diagrama de los datos que se requerirán y las relaciones entre ellos, para luego deducir cuáles son los procesos necesarios que permitirán que los datos se mantengan actualizados y evolucionen para brindar la información última que se desee en el sistema.

Estas ideas se vieron plasmadas por primera vez en el lenguaje *Simula*, un lenguaje para programar simulaciones en el computador en donde los objetos de una simulación eran modelados en forma directa como objetos de software. *Simula* es el antecesor de lenguajes como Smalltalk, Object Pascal, C++, Cios, y Ada, que se puede decir que son de desarrollo reciente. Estos lenguajes son llamados **orientados a objetos**. En ellos, un módulo es una colección de objetos, que son cápsulas que contemplan tanto un comportamiento (función) como un estado (datos), conformando una unidad donde los datos son el fundamento y estos pueden realizar "por sí mismos" un conjunto de acciones determinadas, tratando de simular el comportamiento de los objetos del mundo real.

La programación orientada a objetos (POO) es una forma de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización del código.

Elementos básicos de la programación orientada a objetos.

Los elementos básicos de la programación orientada a objetos son: objetos, mensajes, métodos y clases.

Objetos

Un programa tradicional se compone de procedimientos y de datos. Un programa orientado a objetos se compone solamente de objetos. ***Un objeto es una encapsulación genérica de datos y de los procedimientos para manipularlos. Dicho de otra forma, un objeto es una entidad que tiene unos atributos particulares: las propiedades, y unas formas de operar sobre ellos: los métodos. Por lo tanto, un objeto contiene, por una parte, operaciones que definen su comportamiento, y por otra, variables manipuladas por esas operaciones que definen su estado.***

Por ejemplo, una ventana del sistema operativo Windows es un objeto. El color de fondo de la ventana, el ancho, el alto, etc. son propiedades. Las rutinas, lógicamente transparentes al usuario, que permiten maximizar la ventana, minimizarla, etc. son métodos

Mensajes

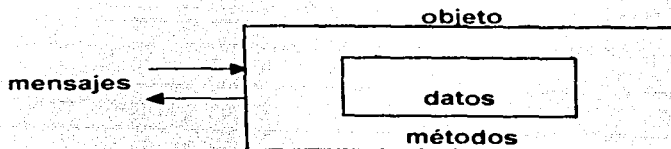
Cuando se ejecuta un programa orientado a objetos, los objetos están recibiendo, interpretando y respondiendo a mensajes de otros objetos. Esto marca una clara diferencia con respecto a los elementos de datos pasivos de los sistemas tradicionales. Por ejemplo, en Visual Basic un mensaje está asociado con un procedimiento, de tal forma que cuando un objeto recibe un mensaje la respuesta a ese mensaje es ejecutar el procedimiento asociado. Este procedimiento recibe el nombre de método.

Otro ejemplo; cuando un usuario quiere maximizar una ventana Windows, lo que hace simplemente es pulsar el botón de la misma que realiza esa acción. Eso, provoca que Windows envíe un mensaje a la ventana para indicar que tiene que maximizarse. Como respuesta a este mensaje se ejecutará el método programado para ese fin.

Métodos

Un método se implementa en una clase de objetos y determina cómo tiene que actuar el objeto cuando recibe un mensaje. En adición, las propiedades permitirán almacenar información para dicho objeto. Un método puede también enviar mensajes a otros objetos solicitando una acción o información.

La estructura más interna de un objeto está oculta para otros usuarios y la única conexión que tiene con el exterior son los mensajes. Los datos que están dentro de un objeto solamente pueden ser manipulados por los métodos asociados al propio objeto.



La ejecución de un programa orientado a objetos realiza fundamentalmente tres cosas:

1. Crea los objetos necesarios.
2. Los mensajes enviados a unos y a otros objetos dan lugar a que se procese internamente la información.
3. Finalmente, cuando los objetos no son necesarios, son borrados, liberándose la memoria ocupada por los mismos.

Clases

Una clase es un tipo genérico de objetos definido por el usuario. Una clase equivale a la generalización de un tipo específico de objetos.

Un objeto de una determinada clase se crea en el momento en que se declare una variable de dicha clase. Por ejemplo:

```
Private MiTexto As TextBox 'declara MiTexto de la clase TextBox  
Set Mitexto = Text2       'asigna a MiTexto el objeto Text2
```

Para enfocar la caja de texto MiTexto de una determinada ventana, esto es, poner el cursor sobre ella, le enviaremos el mensaje correspondiente; en Visual Basic, esto se hace invocando en método SetFocus así:

```
MiTexto.SetFocus          'enfocar el objeto referenciado por MiTexto
```

Características de la programación orientada a objetos (POO).

Las características fundamentales de la POO son: **abstracción, encapsulamiento, herencia y polimorfismo.**

TESIS CON
FALLA DE ORIGEN

Abstracción

Existen muchas definiciones de este término aquí emplearemos la de Shaw que define el concepto como **"una descripción simplificada de un sistema, que enfatiza algunos de los detalles o propiedades del mismo, pero que también omite otros"**. **Una buena abstracción es una que enfatiza detalles que son significativos al lector y que suprime otros detalles que son, por el momento, irrelevantes"**.

Una abstracción es un medio que permite un mejor manejo de sistemas complejos, al permitir suprimir o posponer detalles irrelevantes en cierto momento, y concentrarse más bien, en detalles verdaderamente esenciales. El reconocer similitudes entre diferentes objetos también permite simplificar el dominio en estudio, pues hace posible enunciar comportamientos o características generales que los describen, sin importar detalles que podrían no ser importantes.

Por medio de la abstracción conseguimos no detenernos en los detalles concretos de las cosas que no interesen en cada momento, sino generalizar y centrarse en los aspectos que permitan tener una visión global del tema. Por ejemplo, el estudio de un ordenador podemos realizarlo a nivel de funcionamiento de sus circuitos electrónicos, en términos de corriente, tensión etc., o a nivel de transferencia entre registros, centrándose así el estudio en el flujo de información entre las unidades que lo componen (memoria, unidad aritmética, unidad de control, registros, etc.), sin importarnos el comportamiento de los circuitos electrónicos que componen estas unidades.

Encapsulamiento

El término encapsulamiento tiende a confundirse con mucha frecuencia debido a que se le asocian diferentes significados. Formalmente, **encapsulamiento se refiere al fortalecimiento de las barreras de abstracción mencionadas anteriormente**. Una abstracción debe poderse manipular como una sola unidad independientemente que cumple con ciertas funciones. Cada abstracción debe ser claramente distinguible de las demás, así como la forma en que interactuarán entre ellas. **Encapsular significa agrupar y manejar el grupo resultante como tal, y no cada parte a su vez**.

Esta característica permite ver un objeto como una caja negra en la que se ha metido de alguna manera toda la información relacionada con dicho objeto. Esto nos permitirá manipular los objetos como unidades básicas, permaneciendo oculta su estructura interna.

La abstracción y la encapsulación están representadas por la clase. La clase es una abstracción, porque en ella se definen las propiedades de un determinado conjunto de objetos con características comunes, y es una encapsulación, porque constituye una caja negra que encierra tanto los datos de que almacenan los objetos como los métodos que permiten manipularlos.

Herencia

La herencia es el mecanismo para compartir automáticamente métodos y datos entre clases y subclases de objetos. El usuario de Visual Basic no dispone de esta característica.

Polimorfismo

Esta característica permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la que se realice la implementación. Esto hace que se pueda acceder a una variedad de métodos distintos (todos con el mismo nombre) utilizando exactamente el mismo medio de acceso.

5.3 La Programación en Visual Basic

Visual Basic 6.0 soporta la abstracción, la encapsulación, el polimorfismo y la reutilización del código. Los objetos de Visual Basic están encapsulados; es decir, contienen su propio código y sus propios datos. Así mismo tienen propiedades, métodos y eventos. Las propiedades son los datos que describen un objeto. Los eventos son hechos que pueden ocurrir sobre un objeto (un clic sobre un botón es un evento que produce un mensaje) Un método agrupa el código que se ejecuta en respuesta a un evento.

Al conjunto de propiedades y métodos se le llama interfaz. Además de su interfaz predeterminada, los objetos pueden implementar interfaces adicionales para proporcionar polimorfismo. El polimorfismo le permite manipular muchos tipos deferentes de objetos sin preocuparse de su tipo.

En una aplicación Visual - Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfaz gráfica.

Es por tanto un termino medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos. Combina ambas tendencias. Ya que no podemos decir que VB pertenezca por completo a uno de esos dos tipos de programación, debemos inventar una palabra que la defina: PROGRAMACION VISUAL.

Visual Basic es hoy el lenguaje de programación más popular del mundo. Es un producto con una interfaz gráfica de usuario para crear aplicaciones para Windows basado en el lenguaje Basic y en la programación orientada a objetos.

La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario. En lugar de escribir numerosas líneas de código para implementar una interfaz, se utiliza el ratón para arrastrar y colocar los objetos prefabricados al lugar deseado dentro de un formulario.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

5.4 Estructura de Visual Basic.

No es el propósito del presente trabajo proporcionar una descripción detallada de todos los elementos utilizados para el desarrollo de aplicaciones en Visual Basic. Para ello existen cursos y libros excelentes. Describiremos aquí solamente aquellos elementos que consideramos fundamentales para comprender como se desarrolla una aplicación en Visual Basic, y por supuesto aquellos que se utilizan en el Programa que aquí se desarrolla.

5.4.1 Formularios y controles.

Los formularios y los controles constituyen la base para construir programas en Visual Basic. Desde el punto de vista del Programador, un formulario y un control son objetos con propiedades, métodos y eventos. Desde la perspectiva del usuario, un formulario es el lugar en el que se colocan todas las cosas con las que interactúa cuando utiliza el programa. Estas cosas que se colocan en el formulario son los controles, que permiten que el usuario que utiliza su programa realice acciones, como introducir texto y pulsar botones.

Los formularios son tratados como controles con sus propiedades, eventos y métodos. Aunque se trata en realidad de un control especial que utilizamos para contener los controles más usuales con los que construimos nuestras aplicaciones: *textbox*, (cajas de texto) *label* (etiquetas), *commandbutton* (botones), etc.

Formulario de Arranque

Es el primer formulario que se despliega en la Aplicación. Por lo general es el primer formulario que se crea en el Ambiente de desarrollo. No necesariamente una Aplicación tiene que arrancar con el primer formulario del Proyecto ya que esto se puede modificar.

Controles Personalizados y Objetos Insertables

Un Control personalizado es un archivo con una extensión *.vb*x u *.oc*x que cuando se añade a un Proyecto extiende la caja de herramientas de Visual Basic, pueden añadirse dentro de la opción de Menú **Proyecto** en la opción **Componentes**.

Un Objeto Insertable es un tipo de control Personalizado que es un Objeto y una Aplicación servidora OLE, como una hoja de trabajo en Excell o un Documento de Word, esto lo puede acceder dentro de la opción **Proyecto, Componentes** en la pestaña **Objetos Insertables**.

La ventaja de usar los Controles Personalizados es obvia puesto que no es necesario Programar algo que ya esta hecho.

En Visual Basic el uso de Formularios es una Actividad realmente flexible. Puede hacer que todos los Formularios de un programa sean visibles de forma simultanea o podrá cargar y descargar formularios a medida que el Programa los valla necesitando.

Un formulario que tenga que ser utilizado cuando se muestre por pantalla se denomina formulario modal (el Formulario acaparara la atención del Programa hasta que el usuario pulse Aceptar, Cancelar o sea eliminado por otros medios).

Un formulario que el usuario puede seleccionar o ignorar en pantalla recibe el nombre de **Formulario No Modal**. La mayoría de aplicaciones utilizan formularios no Modales para mostrar Información, ya que proporcionan más flexibilidad de empleo del usuario.

Propiedades de los Formularios.

Las propiedades más comunes de los formularios y que son similares al resto de los controles son las siguientes:

Name: Nombre del formulario, necesario para llamar al formulario desde el código.

Caption: Texto que aparece en el título del formulario

BackColor: Color de fondo del formulario.

ForeColor: color del texto del formulario.

Una aplicación suele contener más de un formulario. Para utilizar varios formularios hay que añadir cada uno de ellos al Proyecto mediante la opción de menú **Insert / Form** o pulsando sobre el botón.

Propiedades, métodos y eventos

Dicho de manera sencilla, las propiedades describen objetos. Los métodos hacen que el objeto haga algo. Los eventos son lo que ocurre cuando un objeto hace algo.

Todo objeto, como un formulario o como un control, posee un conjunto de propiedades que lo describen. Aunque este conjunto no es idéntico para todos los objetos, algunas propiedades son comunes a la mayoría de los controles. Puede ver todas las propiedades de un control determinado en la Ventana de Propiedades del IDE.

Tabla 5.1 Propiedades comunes de los controles de Visual Basic.

Propiedad	Descripción
Left	La posición de la parte izquierda de un control con respecto a su contenedor
Top	La posición de la parte superior de un control con respecto a su contenedor
Height	La altura de un control

Tabla 5.1 Propiedades comunes de los controles de Visual Basic.

Propiedad	Descripción
Width	El ancho de un control
Name	El valor de cadena que se emplea para hacer referencia a un control
Enabled	El valor booleano(true/false) que determina si los usuarios pueden manipular el control
Visible	El valor booleano(true/false) que determina si los usuarios pueden ver el control

Los métodos son bloques de código diseñados en un control que le indican al control cómo hacer las cosas, como moverse a otra ubicación de un formulario. Al igual que sucede con las propiedades, no todos los controles tienen los mismos métodos, aunque algunos métodos comunes existen.

Tabla 5.2 Métodos comunes de los controles de Visual Basic.

Método	Uso
Move	Cambia la posición de un objeto en respuesta a la solicitud de código
Drag	Maneja la ejecución de una operación de arrastrar y soltar hecha por el usuario
SetFocus	Da el enfoque al objeto especificado en la llamada del método
ZOrder	Determina el orden en que aparecen múltiples objetos en la pantalla.

Los eventos son lo que ocurre en torno a un programa. Por ejemplo, cuando un usuario pulsa un botón, se producen muchos eventos: el botón del ratón está presionado, el CommandButton del programa está pulsado y el botón del ratón se suelta. Estas tres cosas se corresponden con el evento MouseDown, con el evento Click y con el evento MouseUp. Durante este proceso, también tienen lugar el evento GotFocus del CommandButton y el evento LostFocus del objeto que tuviera antes el enfoque.

De igual manera, no todos los controles tienen los mismos eventos, pero algunos eventos los comparten muchos controles. Estos eventos se producen como resultado de ciertas acciones específicas del usuario, como mover el ratón, pulsar

una tecla del teclado o pulsar un cuadro de texto. Estos tipos de eventos los inicia el usuario y normalmente será en ellos donde escriba código.

Tabla 5.3 Eventos comunes de los controles de Visual Basic

Evento	Ocurrencia
Change	El usuario modifica texto en un cuadro combinado o cuadro de texto.
Click	El usuario pulsa un objeto con el botón izquierdo del ratón.
DbClick	El usuario pulsa dos veces un objeto con el botón izquierdo del ratón.
DragDrop	El usuario arrastra un objeto a otra ubicación.
DragOver	El usuario arrastra un objeto sobre otro control.
GotFocus	El objeto recibe el enfoque.
KeyDown	El usuario pulsa una tecla del teclado mientras un objeto tiene el enfoque.

Tabla 5.3 Eventos comunes de los controles de Visual Basic

KeyPress	El usuario pulsa y suelta la tecla del teclado mientras un objeto tiene el enfoque.
KeyUp	El usuario suelta una tecla del teclado mientras un objeto tiene el enfoque.
Event	Ocurrencia.
LostFocus	Un objeto pierde el enfoque.
MouseDown	El usuario pulsa cualquier botón del ratón mientras el puntero se encuentra sobre el objeto.
MouseMove	El usuario mueve el puntero del ratón sobre el objeto.
MouseUp	El usuario suelta cualquiera de los botones del ratón mientras el puntero se encuentra sobre el objeto.

Relación existente entre propiedades, métodos y eventos.

Aunque las propiedades, métodos y eventos hacen diversas cosas, es importante saber que a menudo están interrelacionados. Por ejemplo, si mueve un control por medio del método Move (con toda probabilidad de respuesta a un evento), una o más propiedades de la posición del control (Top, Height, Left y Width) cambiará como consecuencia de ello. Dado que el tamaño del control ha cambiado, se produce el evento Resize.

La interdependencia implica que a veces se pueden realizar las mismas tareas en el código si se manipulan las propiedades y métodos de objeto.

La importancia de la programación conducida por eventos

Cuando se crea un programa en Visual Basic, lo que se está haciendo generalmente es programación conducida por eventos. Este concepto significa que la mayor parte del código que escriba se ejecutará a medida que los usuarios vayan haciendo cosas en el programa o incluso cuando se produzcan ciertas cosas en Windows —cuando se produzcan eventos—. Evidentemente, la programación, concebida de esta forma, implica que tendrá que saber cuándo se

producen los eventos y cuándo debe escribir código que haga que el programa se comporte de una manera determinada en respuesta a un evento.

Afortunadamente, Windows y Visual Basic hacen la mayor parte del trabajo. Siempre que se produce un evento, Windows envía un mensaje al programa. Su programa lee este mensaje y luego ejecuta el código que el usuario haya agregado al evento. Si no especifica código para un evento, el programa lo ignorará.

Generalmente, este código se conoce como procedimiento, que se define como un bloque de código que se puede llamar desde su aplicación. Este código puede ser utilizado para mover objetos por un formulario, calcular un valor a partir de una fórmula o escribir datos en una base de datos. Independientemente de cuál sea la finalidad, un procedimiento siempre usa este formato:

```
[Public : Private] [Static] Sub : Function : Property _  
    function_name (arguments) [As Type]  
{ . . . Código del procedimiento . . . }  
End Sub : Function : Property
```

Un procedimiento de evento es el lugar del proyecto en el que se coloca el código, que se ejecutará cuando se produzca un evento. Para escribir un procedimiento de evento, deberá acceder a la Ventana Código del objeto haciendo una de estas cosas:

- Pulsar dos veces el objeto.
- Seleccionar el objeto con el ratón y pulsar F7.
- Seleccionar el objeto y elegir Código en el menú Ver.
- Seleccionar el formulario del objeto en el Explorador de proyectos, pulsando el botón Ver Código y eligiendo el objeto en la Ventana Código.

5.5 Elementos del Lenguaje

El código en Visual Basic se almacena en módulos. Cada módulo está subdividido en distintas secciones; una para cada objeto del módulo. Cada sección de código puede contener uno o más procedimientos distintos. Y un procedimiento esta formado por comentarios, declaraciones de constantes y de variables, expresiones, sentencias y llamadas a procedimientos.

Variables.

Una variable contiene un valor que puede modificarse a lo largo de la ejecución de la aplicación. Cada variable tiene atributos propios, como:

- **Nombre.**- es el nombre que utilizamos para referirnos a la variable en la aplicación.
- **Tipo.**- El tipo determina qué clase de valores puede almacenar una variable.
- **Ámbito.**- El ámbito de una variable especifica en qué parte de la aplicación es conocida la variable y por lo tanto puede utilizarse.

Nombres de variables

El nombre de una variable tiene que comenzar por una letra, puede tener hasta 255 caracteres de longitud y debe ser único dentro de su ámbito. Los caracteres pueden ser letras, dígitos, el carácter de subrayado y los caracteres de declaración del tipo de la variable (% , & , ! , # , @ , y \$).

No se puede utilizar el punto ni otros caracteres que tienen un significado especial para Visual Basic; por ejemplo el paréntesis. Los caracteres de declaración de tipo, cuando se utilicen, tienen que ocupar la última posición.

El nombre de una variable no puede ser una palabra reservada. Una *palabra reservada* tiene un significado especial para Visual Basic. Son palabras reservadas las sentencias predefinidas (como **For**), los nombres de funciones (como **Val**), métodos (como **Hide**), propiedades (como **Caption**), tipos (como **Long**) y de operadores (como **And**).

Tipos de datos intrínsecos

Todas las variables tienen un tipo de dato que determina la clase de datos que pueden almacenar. Una variable puede ser de alguno de los tipos siguientes:

Tipo	Descripción	Car-tipo	Rango
Integer	Entero (2 bytes)	%	-32.768 a 32.767
Long	Entero largo (4 bytes)	&	-2.147.483.648 a 2.147.483.647
Single	Coma flotante. Precisión simple (4 bytes)	!	-340E+38 a 3,40E+38
Double	Coma flotante. Precisión doble (8 bytes)	#	-1,79D+308 a 1,79D+308
Currency	Número con punto Decimal fijo (8 bytes)	@	+/- 922.3373203.6685.477, 5807
String	Cadena de caracteres de longitud fija: (1 byte por carácter)	(ninguno)	Hasta 64K aproximadamente

String	Cadena de caracteres de longitud variable: (10 bytes + 1 byte por carácter)		Hasta 2 ³¹ caracteres aproximadamente
Byte	Carácter (1 byte)	(ninguno)	0 a 255
Boolean	Boolean (2 bytes)	(ninguno)	True o False
Date	Fecha/Hora (8 bytes)	(ninguno)	1/Enero/100 a 31/Diciembre/9.999
Object	Referencia a un objeto (4 bytes)	(ninguno)	Cualquier referencia a tipo Object
Variant (por omisión)	Con números: 16 bytes	(ninguno)	Cualquier valor numérico hasta el intervalo de un tipo Double
Variant (por omisión)	Con caracteres: 22 bytes + 1 byte por carácter	(ninguno)	El mismo intervalo que para un tipo String de longitud variable
Decimal	Números con 0 a 28 decimales (14 bytes) (No se puede declarar una variable de este tipo. Sólo se puede utilizar con un Variant)	(ninguno)	Valor mayor con 0 decs: +/-79.228.162.514.264. 337.593.543.950.335 Valor mayor con 28 dec: +/-7.9228162514264 337593543950335

Un *byte* son ocho bits y un bit se representa mediante un 0 o un 1, símbolos que componen el sistema binario empleado por los ordenadores. Un múltiplo es el K; 1Kbyte es igual a 2¹⁰ bytes (2 es la base del sistema binario y 1K=2¹⁰=1024). Cada carácter del código de caracteres ANSI se representa por un byte.

Un número expresado en coma flotante es un número fraccionario con punto decimal variable. En el tipo **Currency** el punto decimal es fijo (cuatro decimales).

El valor real en precisión simple 3,40E+38 equivale a 3,40 por 10 elevado a 38. Análogamente, el valor real en precisión doble 1,79D+308 equivale a 1,79 por 10 elevado a 308.

True significa verdad y **False** significa falso.

Declaración de variables

Antes de utilizar una variable, es aconsejable declarar su tipo. Una forma de hacerlo la sentencia **Dim** (o una de las palabras **Public**, **Private** o **Static**). Cualquier declaración de éstas inicia las variables numéricas con el valor cero y las variables alfanuméricas con el carácter nulo.

Módulos de Visual Basic

El código de una aplicación Visual Basic se edita en módulos. Los módulos tienen incorporadas funciones automáticas de formato y comprobación de sintaxis. Hay tres tipos de módulos: de formularios, estándar y de clase. Para agregar uno de estos módulos en una aplicación, simplemente tenemos que ejecutar la orden correspondiente del menú *Proyecto: Agregar formulario* para un formulario, *Agregar módulo* para un módulo estándar o *Agregar módulo de clase* para una clase.

Módulos de formulario

Es un fichero con extensión *.frm* que puede contener descripciones del formulario y sus controles, así como los valores de sus propiedades, declaraciones de constantes, variables y procedimientos externos a nivel del formulario, así como procedimientos conducidos por eventos y procedimientos generales.

Módulos estándar

Es un fichero con extensión *.bas* que sólo contiene declaraciones y definiciones de procedimientos, tipos y datos a los que tienen acceso otros módulos de la aplicación. Las declaraciones a nivel de módulo y las definiciones en un módulo estándar son públicas (**Public**) por omisión.

Módulos de clase

Es un fichero con extensión *.cls* que contiene la definición de una clase de objetos; esto es, la definición de cada una de sus propiedades y métodos.

Ámbito de variables

Se entiende por ámbito o alcance de una variable el espacio de la aplicación donde la variable es visible y por lo tanto se puede utilizar.

La tabla siguiente resume cómo debe declararse una variable en función del espacio donde deseamos que sea visible:

<i>Ámbito</i>	<i>Declaración</i>
Local	Dim , Static , o ReDim (dentro de un procedimiento)
Módulo	Dim o Private (sección de declaraciones del módulo)
Global	Public (sección de declaraciones de un módulo)

A nivel el módulo, no hay diferencia entre **Dim** y **Private**, pero se aconseja utilizar **Private** en contraste como **Public**. En un procedimiento no tiene lugar esta observación, ya que no se puede utilizar ni **Private** ni **Public**.

Un formulario (*.frm*) incluye controles más código y que un módulo (*.bas*) o una clase (*.cls*) incluyen sólo código.

VARIABLES LOCALES

Una *variable local* se reconoce solamente en el procedimiento en el que está definida. Fuera de ese procedimiento, la variable no es conocida. Su utilización más común es intervenir en cálculos intermedios.

Para declarar una variable local a un procedimiento, coloque la sentencia **Dim** correspondiente dentro del mismo.

Una variable local es reiniciada cada vez que entra en el procedimiento. En otras palabras, una variable local no conserva su valor entre una llamada al procedimiento y la siguiente. Para hacer que esto suceda, hay que declarar la variable estática. Visual Basic inicia una variable estática solamente la primera vez que se llama al procedimiento. Para declarar una variable estática, utilice la palabra clave **Static** en lugar de **Dim**.

Para hacer que todas las variables de un procedimiento sean estáticas, puede también proceder declarando el procedimiento estático.

Si una variable aparece en un procedimiento y no está explícitamente declarada, es por omisión local. En cualquier caso, se aconseja declarar todas las variables explícitamente.

VARIABLES UTILIZADAS DENTRO DEL MÓDULO

Una *variable declarada a nivel del módulo* (formulario, módulo estándar o clase) puede ser compartida por todos los procedimientos de ese módulo. Una variable a nivel del módulo hay que declararla con **Dim** y **Private** en la sección de declaraciones del módulo (sección *General*). Para editar esta sección, hay que abrir la ventana de código del formulario, de un módulo estándar o de una clase, para lo que tiene que dirigirse al explorador de proyectos, seleccionar el módulo y hacer clic en el botón *Ver código*. Después en la ventana de código seleccionará "(General)" de la lista de la izquierda, lista de objetos y "(Declaraciones)" de la lista de la derecha, lista de procedimientos.

Este tipo de variables son por omisión *estáticas*.

VARIABLES GLOBALES

Una *variable global* es una variable declarada a nivel del módulo pero a la que se puede acceder desde cualquier otro módulo.

Para hacer que una variable sea global o pública, hay que declarar **Public** en la sección de declaraciones del módulo. Para ello, si el módulo ya existe, selecciónelo en la ventana *Proyecto* y haga clic en el botón *Ver código*; y si no existe, créelo ejecutando la orden correspondiente del menú *Proyecto*.

Operadores.

La tabla que se muestra a continuación presenta el conjunto de operadores que soporta Visual Basic ordenados de mayor a menor prioridad respecto a cómo se evalúan cuando varios de ellos intervienen en una misma expresión. Los ordenadores que aparecen sobre una misma línea tienen igual prioridad. Las operaciones que se incluyan entre paréntesis se evalúan antes, ejecutándose primero los paréntesis más internos.

Tipo	Operación	Operador
Aritmético	Exponenciación	^
	Cambio de signo	-
	Multiplicación y división	*, /
	División entera	\
	Resto de la división entera	Mod
	Suma y resta	+, -
Concatenación	Concatenar o enlazar	&
Relacional	Igual, distinto, menor, mayor ...	=, <>, <, >, <=, >=
Otros	Comparar dos expresiones de cars.	Like
	Comparar dos referencias a objetos	Is
Lógico	Negación	Not
	And	And
	Or inclusiva	Or
	Or exclusiva	Xor
	Equivalencia (opuesto a Xor)	Eqv
	Implicación	Imp
	(falso si el primer operando verdadero y segundo operando falso)	

Sentencias.

Una sentencia es una línea de texto que indica una o más operaciones a realizar. Una línea puede tener varias sentencias, separadas unas de otras por dos puntos:

total = cantidad * precio: suma = suma + total

Una sentencia Visual Basic puede escribirse en varias líneas físicas utilizando el carácter de continuación de línea (un espacio en blanco seguido del carácter de subrayado). Por ejemplo:

PagoMensual = CantidadPrest*
 (Interés / (1 - (1 / ((1 + Interés) ^ [Meses]))))

La sentencia más común en Visual Basic es la sentencia de asignación. Su forma general es

variable = expresión

que indica que el valor que resulte de evaluar la expresión tiene que ser almacenado en la variable especificada.

Una variable **Variant** no es una variable sin tipo; más bien, es una variable que puede cambiar su tipo libremente (una variable genérica).

5.6 Procedimientos

Una aplicación no es más que un conjunto de sentencias destinadas a resolver un problema determinado. La forma en la que se organizan las sentencias y el orden en que se ejecutan dan lugar a la estructura de la aplicación.

Puesto que una aplicación Visual Basic consta de objetos, la estructura de su código define lo que se ve en pantalla. Así, un formulario no es más que una representación de las propiedades que definen su apariencia y su comportamiento intrínseco. Apariencia y comportamiento está regidos por un código contenido en lo que hemos denominado módulo de formulario

Cada módulo de formulario contiene procedimientos conducidos por eventos; esto es, procedimientos que responden a los eventos que se produzcan en relación con el formulario; bien sobre el propio formulario, o bien sobre sus controles.

Para hacer que una aplicación responda a los eventos que sobre ella se puedan producir, uniremos a cada uno de sus objetos el código que deseamos se ejecute como respuesta al evento ocurrido. Este código asociado con cada objeto y escrito para un determinado evento recibe el nombre de ***procedimiento conducido por un evento***.

Además de procedimientos conducidos por eventos, los módulos de formulario pueden contener procedimientos generales que se ejecutan como respuesta a una llamada desde cualquier procedimiento conducido por un evento. Estos procedimientos se pueden colocar en el propio módulo de formulario, aunque lo más común es colocarlos en un módulo estándar; entre otras cosas, para ponerlos a disposición de otros formularios.

Cuando sea necesario crear objetos que se puedan llamar desde los procedimientos de aplicación, se utilizará un módulo de clase.

La programación con procedimientos tiene dos ventajas importantes:

- Los procedimientos permiten dividir la programación de una aplicación en unidades de código lógicas separadas, lo que facilita su depuración y mantenimiento.
- Los procedimientos utilizados en un programa pueden actuar como bloques básicos en otros programas, normalmente con una mínima modificación o incluso sin cambios.

En Visual Basic se utilizan varios tipos de procedimientos:

- Los procedimientos **Sub** que no devuelven ningún valor.
- Los procedimientos **Function** que devuelven un valor.
- Los procedimientos **Property** que permiten crear propiedades para una clase.

En desarrollos realizados hasta ahora, los procedimientos conducidos por eventos son **Sub**.

A su vez, podemos clasificar los procedimientos en:

- Procedimientos intrínsecos. Son los procedimientos predefinidos en Visual Basic.

Procedimientos extrínsecos. Son los procedimientos definidos por el usuario.

5.7 Proyectos de Visual Basic

Un proyecto es un grupo de todos los archivos que conforman un programa. Entre ellos puede haber formularios, módulos (bloques de código que no están unidos a un formulario), gráficos y controles ActiveX. Otros tipos de archivos pueden formar parte de un proyecto.

En un proyecto cuando se crea un programa, cada formulario, módulo, gráfico y control ActiveX queda guardado como un archivo individual.

La información detallada de todos los archivos del programa se almacena en el archivo del proyecto. Este archivo enumera todos los archivos que utiliza el programa, aparte de los detalles como el nombre del proyecto y la forma en que se debe configurar el IDE mientras trabaja en el proyecto. Los archivos de proyecto tienen la extensión, VBP.

5.8 Estrategia de programación bajo Visual Basic.

La creación de un programa bajo Visual Basic conlleva los siguientes pasos:

1. Creación de una interfaz de usuario. Esta interfaz será la principal vía de comunicación hombre - máquina, tanto para salida de datos como para entrada. Será necesario partir de una ventana - Formulario - a la que le iremos añadiendo los controles necesarios que nos permitan la interacción con el código del programa.
2. Definición de las propiedades de los controles (Objetos) que hayamos colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.
3. Generación del código asociado a los eventos que ocurran a estos objetos. A la respuesta a estos eventos (click, doble click, una tecla pulsada, etc.) le llamamos Procedimiento, y deberá generarse de acuerdo a las necesidades del programa.
4. Generación del código del programa. Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos no responden a un evento acaecido a un objeto, sino que responden a un evento producido durante la ejecución del programa.

Capítulo 6. Descripción del programa.

6.1 Introducción.

La presente aplicación es una herramienta destinada a auxiliar a empresarios, inversionistas y tomadores de decisiones en la evaluación de proyectos de inversión. Para ello, a partir de los conceptos de **Inversión Neta, Ingresos operativos y Egresos operativos de un proyecto o negocio**, el programa calcula los parámetros de decisión discutidos en el Capítulo 4: **Tasa Interna de Rendimiento (TIR), Valor (VPN) Presente Neto y Periodo de Retorno de la Inversión.**

Como toda aplicación en ambiente Windows, el programa consta de una interfaz gráfica conformada por seis(6) módulos de formulario o pantallas que el usuario va desplegando conforme procede a la captura de datos y el cálculo de los parámetros de decisión. El nombre con que estos formularios se identifican en el código de la aplicación es:

1. **Evalua.frm;**
2. **Ingreso.frm;**
3. **Egreso.frm;**
4. **FlujoDeEfectivoNeto.frm;**
5. **InversionNeta.frm;**
6. **Renta.frm;**

Así mismo el programa consta de un Módulo estándar denominado: **ModuleInv.bas** el cual contiene todos los procedimientos y funciones que son comunes al resto de los módulos.

Al igual que otras aplicaciones de Visual Basic los formularios contienen objetos y código asociado a ellos, los cuales iremos describiendo en este capítulo.

Los formularios: **Evalua, Ingreso, Egreso e InversionNeta** se utilizan para la captura de los datos generales del proyecto así como de los conceptos e importes de ingreso, egreso e inversión neta del proyecto de inversión.

La captura de información se lleva a cabo introduciendo los datos en cajas de texto (**TextBox**), o en tablas de datos (**MSHFlexGrid**). La ejecución de acciones se lleva a cabo mediante botones de comando (**CommandButton**), barras de herramientas (**ToolBar**) y menús de instrucciones. La salida de resultados se hace a través de cajas de texto (**TextBox**) y tablas de datos (**MSHFlexGrid**) en la pantalla, y a través de la impresora.

6.2. Formularios.

6.2.1 Módulo de Inicio (Evalua.frm)

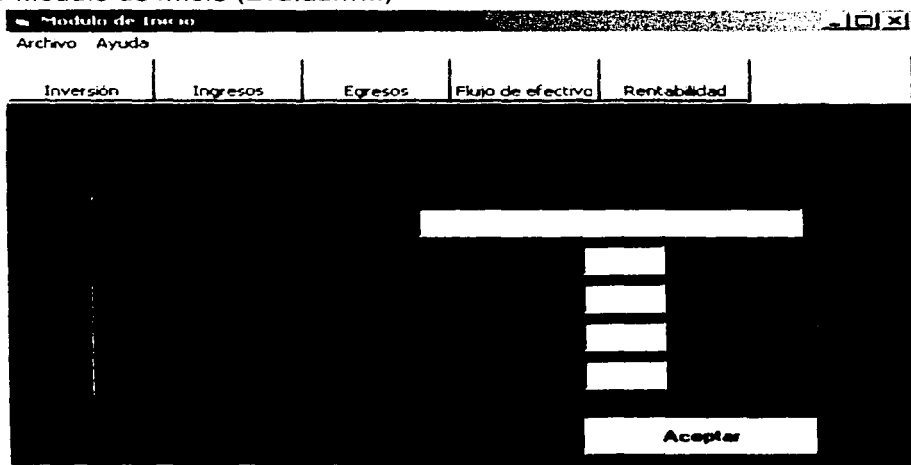


figura 6 1 Imagen del formulario Evalua frm

El formulario *Evalua* que se muestra en la fig. 6.1 es el primero que se despliega al arrancar la aplicación. Este formulario sirve para capturar los datos iniciales del proyecto. *Nombre, cantidad de conceptos de ingreso, de egreso y de inversión neta*, así como la *vida útil* del proyecto.

El formulario consta de los siguientes objetos de Visual Basic: siete etiquetas (*Label*), cinco cuadros de texto (*TextBox*), un botón de comando (*CommandButton*), una barra de herramientas (*ToolBar*) con cinco botones y un menú de opciones para *Abrir* y *Guardar* archivos.

Esencialmente el formulario es una pantalla de captura de datos. En ella se capturan los datos que dimensionan las matrices de flujo de efectivo que conforman el corazón de la aplicación. Como se observa en la pantalla los datos que hay que proporcionar son:

- *Nombre del proyecto;*
- *Número de conceptos de ingreso;*
- *Número de conceptos de egreso;*
- *Número de conceptos de inversión.*

TESIS CON
FALLA DE ORIGEN

Número de períodos del proyecto (vida útil).

Para introducir datos, se coloca el cursor en la caja de texto (*TextBox*) correspondiente y se escribe en ella usando el teclado de la computadora. Una vez que se han proporcionado los datos se oprime el botón **Aceptar**. Con excepción del nombre del proyecto, que puede conformarse de letras y números, el resto de los datos debe ser un número. Si se introduce un dato no numérico, el programa despliega una caja de texto advirtiendo del error.

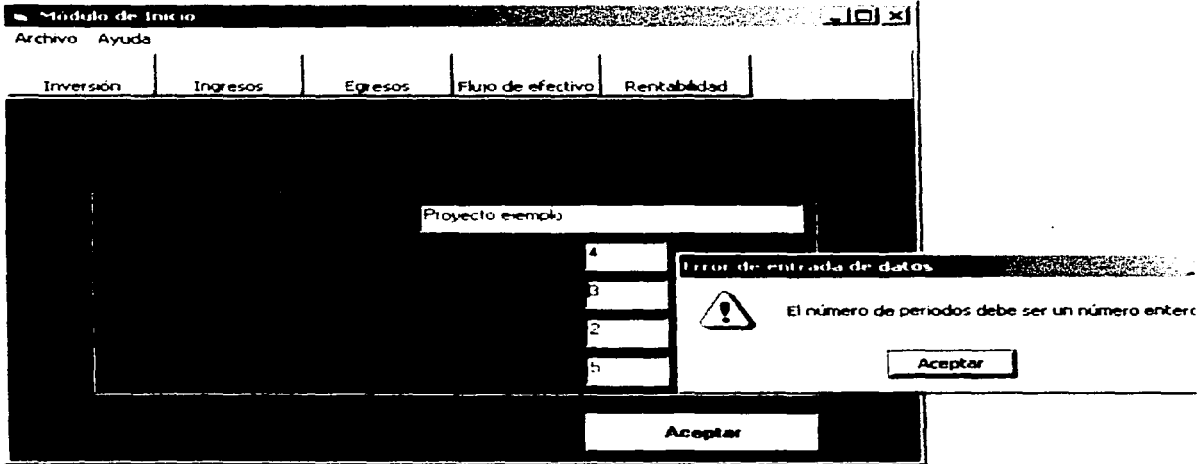


Fig. 6.2 Mensaje de error en la captura de datos

Una vez introducidos todos los datos, el programa preguntará si se desean grabar los datos y crear un Proyecto. Para ello cuenta con un *CommonDialog Box* que abre una caja de diálogo común de Windows para grabar en el directorio seleccionado por el usuario. El programa establece por defecto el directorio *ProyInver*, pero este puede ser modificado por el usuario.

TESIS CON
FALLA DE ORIGEN

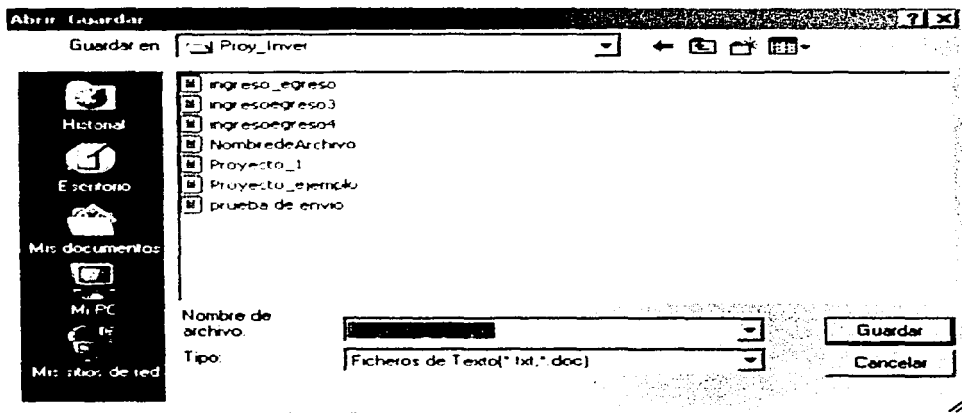


figura 6.3 Caja de Dialogo *Guardar*

Cuando se ha creado y guardado el proyecto, el programa despliega el mensaje de la figura 6.4 indicando que se puede continuar con la captura de datos en cualquiera de los módulos de la aplicación

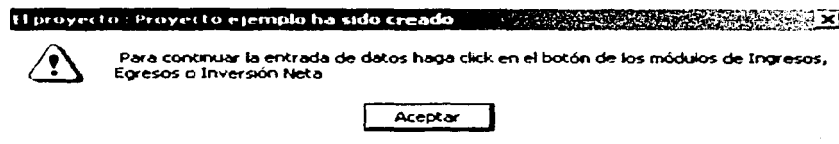


Figura 6.4 Mensaje de proyecto creado

Si al iniciar la aplicación se desea trabajar con un proyecto previamente grabado, entonces deberá seleccionarse **Abrir** en el Menú **Archivo** y el programa desplegará una caja de diálogo como la mostrada en la figura 6.5.

TESIS CON
FALLA DE ORIGEN

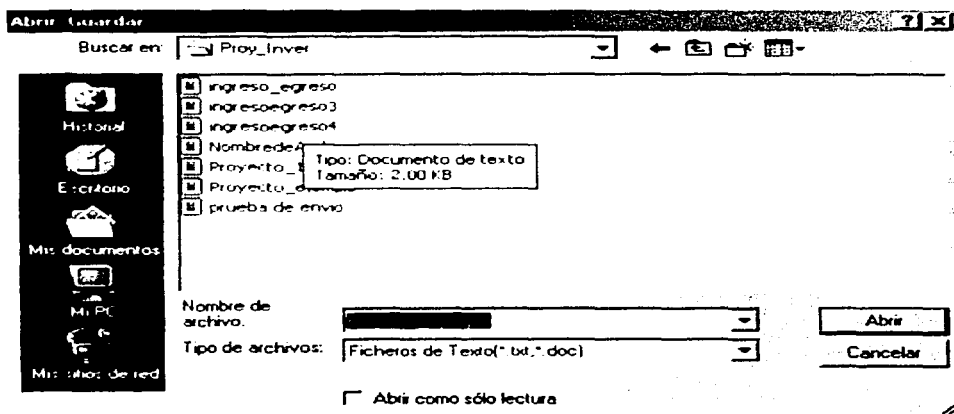


figura 6.5 Caja de Diálogo **Abrir**

Una vez abierto el archivo del proyecto seleccionado, entonces es posible llamar a los Módulos de Ingresos, Egresos, Inversión Neta, Flujo de Efectivo y Rentabilidad, pulsando los correspondientes botones en la barra de herramientas en la parte superior del formulario. De esta manera se está en posibilidad de iniciar la captura, modificar o simplemente visualizar los datos que conforman los flujos de efectivo del proyecto.

A continuación se presenta el código del módulo.

Evalua.frm (código del módulo)

Dim str As String

Private Sub Command1_Click()
On Error GoTo 100

*Lectura de datos generales

NombreDeArchivo = TextNombre.Text

If IsNumeric(Text1.Text) Then

nci = CInt(Text1.Text)

GoTo 20

Else:

str = "El número de conceptos de ingreso"

TESIS CON
FALLA DE ORIGEN

```

GoTo 10
End If

20 If IsNumeric(Text2.Text) Then
nce = CInt(Text2.Text)
GoTo 30
Else:
str = "El numero de conceptos de egreso"
GoTo 10
End If

30 If IsNumeric(Text3.Text) Then
ncinv = CInt(Text3.Text)
GoTo 40
Else:
str = "El número de conceptos de inversión"
GoTo 10
End If

40 If IsNumeric(Text4.Text) Then
nper = CInt(Text4.Text)
GoTo 50
Else:
str = "El número de periodos"
GoTo 10
End If

50 ' Dimensionamiento de variables
dimmat = nci + nce + ncinv + 4
ReDim CiStr(nci), CeStr(nce), CinvStr(ncinv)
ReDim MatFlujoN(dimmat, nper)

Call MenuGuardarArchivo_Click

MsgBox " Para continuar la entrada de datos haga click en el botón de los
módulos de Ingresos, Egresos o Inversión Neta ", vbExclamation, "El proyecto : " &
NombreDeArchivo & " ha sido creado"

Exit Sub
10 MsgBox str & " debe ser un número entero.", vbExclamation, "Error de entrada
de datos"
100 If Err.Number = cd!Cancel Then Exit Sub
End Sub

```

```

Private Sub MenuFlujo_Click()
FlujoDeEfectivoNeto.Show vbModeless
End Sub

```

```
Private Sub MenuIngresos_Click()
Ingreso.Show vbModeless
End Sub
```

```
Private Sub MenuEgresos_Click()
Egreso.Show vbModeless
End Sub
```

```
Private Sub MenuInvNeta_Click()
InversionNeta.Show vbModeless
End Sub
```

```
Private Sub MenuRentabilidad_Click()
Renta.Show vbModeless
End Sub
```

```
Private Sub MenuAbrirArchivo_Click()
On Error GoTo 100
    CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar

'Filtros
    CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
        ""*.txt;*.doc|Todos los ficheros(*.*)|*.*"
'Filtro por defecto
    CommonDialog1.FilterIndex = 1

'Visualizar la caja de diálogo

    CommonDialog1.ShowOpen
    n = FreeFile
    Open CommonDialog1.FileTitle For Random As n

    Get #n, 1, NombreDeArchivo
    Get #n, , nci
    Get #n, , nce
    Get #n, , ncinv
    Get #n, , nper
    TextNombre.Text = NombreDeArchivo
    Text1.Text = nci
    Text2.Text = nce
    Text3.Text = ncinv
    Text4.Text = nper
```

'Esta instrucción permite abrir y actualizar el formulario de flujo de efectivo cuando se capturan datos en este módulo.

FlujoDeEfectivoNeto.MenuRecuperaDatos_Click

```
100 If Err.Number = cdlCancel Then Exit Sub  
End Sub
```

Private Sub MenuGuardarArchivo_Click()

On Error GoTo 100

CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar

```
CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _  
    ""*.txt;*.doc|Todos los ficheros(*.*)|*.*"
```

'Filtro por defecto

```
    CommonDialog1.FilterIndex = 1
```

'Visualizar la caja de diálogo

```
    CommonDialog1.ShowSave
```

```
    n = FreeFile
```

'Abrir el fichero para escribir

```
    Open CommonDialog1.FileTitle For Random As n
```

```
    CommonDialog1.Flags = &H2
```

```
    Put #n, 1, NombreDeArchivo
```

```
    Put #n, , nci
```

```
    Put #n, , nce
```

```
    Put #n, , ncin
```

```
    Put #n, , nper
```

```
    Close #n
```

```
100 If Err.Number = cdlCancel Then Exit Sub  
End Sub
```

Private Sub MenuImprimir_Click()

CommonDialog2.ShowPrinter

PrintForm

End Sub

Private Sub MenuSalir_Click()

Close #n

End

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)

Select Case Button.Key

```
    Case "BtIng"
```

```
        Ingreso.Show vbModeless
```

```
    Case "BtEgr"
```

Egreso.Show vbModeless
Case "BtInvNeta"
MenuInvNeta Click
Case "BtFlujo"
MenuFlujo Click
Case "BtRenta"
MenuRentabilidad Click

End Select
End Sub

TESIS CON
FALLA DE ORIGEN

6.2.2 Inversión (Inversión Neta)

figura 6.6 Imagen del formulario InversionNeta.frm

Para acceder al módulo de Inversión Neta se debe pulsar el botón **Inversión** en la barra de herramientas del módulo inicial. Al hacerlo, se despliega la pantalla que aparece en la figura 6.6

El formulario consta de un menú de opciones, una barra de herramientas, una tabla de datos y una caja de texto en donde se muestra el nombre del proyecto activo.

Para iniciar la captura de datos se deberá elegir **Iniciar** en el menú **Archivo**. Al hacerlo el formulario aparecerá con el número de renglones y columnas con que se dimensionó en el módulo de inicio. Si el número de conceptos de Inversión Neta es n , el número de renglones de la tabla de datos será $n+2$. Así mismo, si el número de periodos del proyecto es p , el número de columnas de la tabla es $p+2$.

Si el proyecto es de nueva creación y no se han introducido datos de la inversión neta, la tabla aparecerá en blanco, con excepción del primer renglón (en amarillo)

-en el que aparecerá el encabezado “Concepto de Inversión Neta / Periodo” y los números 0,1,...,p correspondientes a los años de vida útil del proyecto- y el último renglón de la primera columna – en que aparecerá la leyenda “ Suma de Inversión Neta”.

The screenshot shows a dialog box titled "Inversión neta del proyecto" with a menu bar containing "Archivo" and "Ayuda". Below the menu bar are buttons for "Iniciar", "Abrir", and "Guardar", and a text field containing "Proyecto ejemplo". The main area contains a table with the following structure:

Concepto de Inversión Neta / Periodo	0	1	2	3	4
Suma de Inversión Neta					

Figura 6.7 Formulario correspondiente al proyecto proyecto ejemplo de la figura 6.2

En este momento deben introducirse los conceptos de inversión y sus correspondientes importes, de igual forma en que se haría en cualquier hoja de cálculo. Al ir introduciendo los datos irán apareciendo en el último renglón, los valores correspondientes a la suma de la inversión neta para cada periodo

The screenshot shows the same dialog box as Figure 6.7, but with data entered into the table. The table now contains the following data:

Concepto de Inversión Neta / Periodo	0	1	2	3	4
Muejardas de ejemplo	1.200.000,00	550.000,00	0,00	0,00	0,00
Equipos de ejemplo	0,00	0,00	0,00	0,00	0,00
Suma de Inversión Neta	1.200.000,00	550.000,00	0,00	0,00	0,00

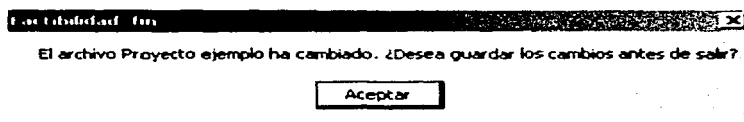
Figura 6.8 Datos de inversión neta del proyecto: Proyecto ejemplo

Cuando al introducir los datos se comete algún error, éste se puede corregir borrándolo con la tecla **Suprimir**, con la tecla de retroceso, o sobrescribiendo el valor. Al hacerlo, el nuevo valor corregido de la suma de inversión neta irá apareciendo en la columna correspondiente.

Una vez capturados todos los datos de inversión pueden guardarse como parte del proyecto eligiendo **Guardar** en el menú **Archivo**. Al hacerlo aparecerá una caja de diálogo como la mostrada en la figura 6.3

TESIS CON
FALLA DE ORIGEN

Si los datos fueron modificados al intentar salir del módulo el programa preguntará si se desean guardar los cambios.



Si el proyecto ha sido creado con anterioridad, se puede elegir Abrir en el menú Archivo. Al hacerlo, los datos del proyecto correspondientes a la inversión neta se desplegarán en la pantalla. Para modificarlos se deberá seguir el procedimiento descrito con anterioridad.

A continuación se presenta el código del módulo

Inversión.frm (código del módulo)

Dim ndr As Integer, indcol As Integer, pos As Integer, nrows As Integer

Private Sub MenuCargaDatos_Click()

Modulo = "Inversión Neta"
IndiceInversion = 1

dimmat = nci + nce + ncinv + 4
ReDim MatFlujoN(dimmat, nper), CiStr(nci + nce + ncinv + 3)
For ndr = 1 To ncinv
For indcol = 1 To nper
MatFlujoN(indr, indcol) = 0
Next indcol
Next ndr

If NombreDeArchivo = "" Then
MsgBox " Debe abrir o iniciar primero un proyecto en el Módulo de inicio ",
vbExclamation, "Error de entrada de datos"
Exit Sub
Else

ncom = ncinv
Call CargaCuadrícula(MSHFlexGridIng)
For col = 0 To nper
MSHFlexGridIng.col = col
MSHFlexGridIng.CellBackColor = &HC0FFFF
Next col

TESIS CON
FALLA DE ORIGEN

End If

End Sub

Private Sub MenuGuardar_Click()

Modulo = "Inversión Neta"

'Si ocurre un error ejecutar ManipulaErrorGuardar

On Error GoTo 100

'genera un error cuando se pulsa cancelar

CommonDialog1.CancelError = True

'Filtros

CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
 "* *.txt;*.doc|Todos los ficheros(*.*)|*.*"

'Filtro por defecto

CommonDialog1.FilterIndex = 1

'Visualizar la caja de diálogo

CommonDialog1.ShowSave

n = FreeFile

'Abrir el fichero para escribir

Open CommonDialog1.FileTitle For Random As n

'Call Guardar(n)

dimmat = nci + nce + ncinv + 4

ReDim MatFlujoN(dimmat, 1 To nper)

ReDim CiStr(nci + nce + ncinv + 3)

Call Guardar(MSHFlexGridng)

SalirGuardar:

Close #n

IndiceModif = 0

IndiceInversion = 0

100 Call ManipulaErrorGuardar

End Sub

Private Sub MenuImprimir_Click()

Modulo = "Ingreso"

CommonDialog2.ShowPrinter

Toolbar1.Visible = False

PrintForm

20 Printer.EndDoc

Toolbar1.Visible = True

End Sub

Public Sub MenuRecuperaDatos_Click()

Modulo = "Inversión Neta"

IndiceInversion = 1

```
'Si ocurre un error ejecutar ManipulaErrorGuardar
  On Error GoTo 100
  CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar
```

```
'Filtros
  CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)" & _
    "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
```

```
'Filtro por defecto
  CommonDialog1.FilterIndex = 1

  If NombreDeArchivo <> "" Then GoTo 99
```

```
'Visualizar la caja de diálogo
  CommonDialog1.ShowOpen
  n = FreeFile
  Open CommonDialog1.FileTitle For Random As n
  NombreDeArchivo = CommonDialog1.FileTitle
```

```
99 Text1.Text = NombreDeArchivo
```

```
  Get #n, 2, nci
  Get #n, 3, nce
  Get #n, 4, ncinv
  Get #n, 5, nper
```

```
  dimmat = nci + nce + ncinv + 4
  ReDim MatFlujoN(dimmat, nper)
  ReDim CiStr(nci + nce + ncinv + 3)
  ncom = ncinv
```

```
102 Call CargaCuadrícula(MSHFlexGridIng)
  For col = 0 To nper
    MSHFlexGridIng.col = col
    MSHFlexGridIng.CellBackColor = &HC0FFFF
  Next col
100 Call Recuperar(MSHFlexGridIng)
```

SalirAbrir:

'Esta instrucción permite abrir y actualizar el formulario de flujo de efectivo cuando se capturan datos en este módulo.

FlujoDeEfectivoNeto.MenuRecuperaDatos_Click

```
  Exit Sub
```

103 ManipulaErrorAbrir:

'Manipular el error.

```
  If Err.Number = cd!Cancel Then Exit Sub 'Se pulsó Cancelar
```

End Sub

Private Sub MenuSalir_Click()

If IndiceModif = 1 Then
MsgBox " El archivo " & NombreDeArchivo & " ha cambiado. ¿Desea guardar los cambios antes de salir?"

Call MenuGuardar_Click

ReDim CiStr(nci + nce + ncin + 5)

If IndiceEgreso = 1 Then

Modulo = "Egreso"

MSHFlexGridIng.Rows = nce + 2

Open NombreDeArchivo For Random As n

Call Guardar(Egreso.MSHFlexGridIng)

Else

If IndiceInversion = 1 Then

Modulo = "Inversión Neta"

MSHFlexGridIng.Rows = ncin + 2

Call Guardar(InversionNeta.MSHFlexGridIng)

Close #n

Else

Close #n

End If

End If

End If

End

End Sub

Private Sub MSHFlexGridIng_KeyPress(KeyAscii As Integer)

Modulo = "Inversión Neta"

Call PresTeclado(MSHFlexGridIng, KeyAscii)

End Sub

Private Sub MSHFlexGridIng_KeyUp(KeyCode As Integer, Shift As Integer)

Modulo = "Inversión Neta"

Call UpTecla(MSHFlexGridIng, KeyCode, Shift)

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)

Modulo = "Inversión Neta"

Select Case Button.Key

```
Case "BtCarDat"  
MenuCargaDatos_Click  
Case "BtAbrir"  
MenuRecuperaDatos_Click  
Case "BtEditar"  
'MenuEgresos_Click  
Case "BtGuardarDat"  
MenuGuardar_Click
```

```
End Select  
End Sub
```

6.2.3 Ingreso (Ingresos operativos del proyecto)

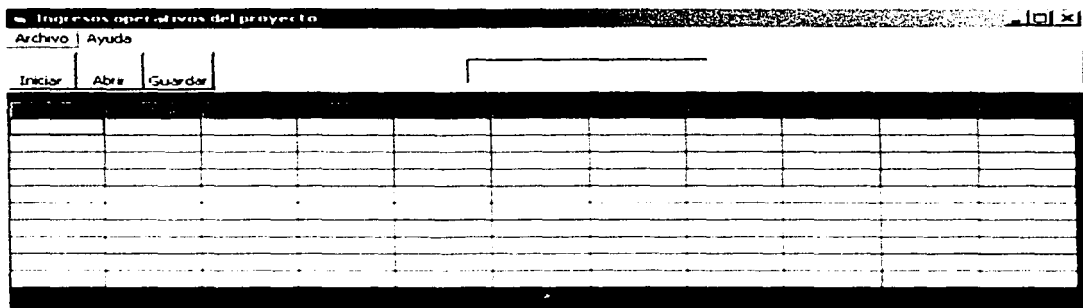


figura 6.9 imagen del formulario Ingreso frm

Para acceder al módulo de Ingresos se debe pulsar el botón **Ingresos** en la barra de herramientas del módulo inicial. Al hacerlo, se despliega la pantalla que aparece en la figura 6.9

TESIS CON
FALLA DE ORIGEN

El formulario consta de un menú de opciones, una barra de herramientas, una tabla de datos y una caja de texto en donde se muestra el nombre del proyecto activo

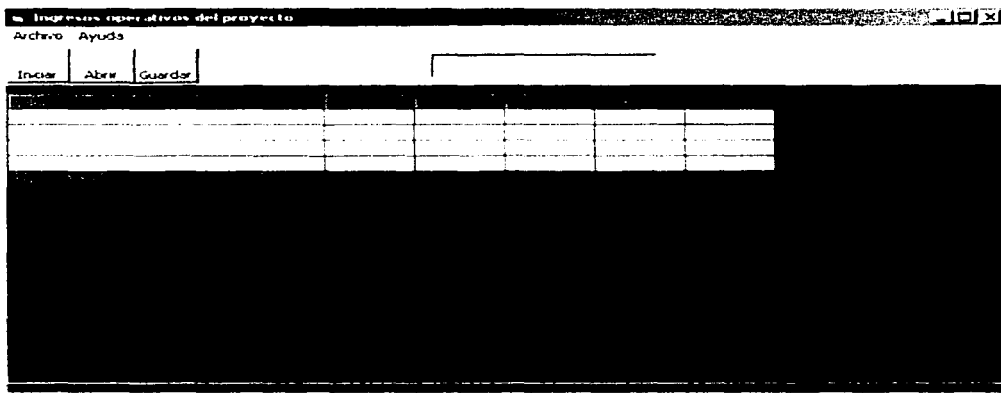


Figura 6 10 Formulario correspondiente al proyecto Proyecto ejemplo de la figura 6.2

Para iniciar la captura de datos se deberá elegir **Iniciar** en el menú **Archivo**. Al hacerlo el formulario aparecerá con el número de renglones y columnas con que se dimensionó en el módulo de inicio. Si el número de conceptos de Ingreso es n_i , el número de renglones de la tabla de datos será n_i+2 . Así mismo, si el número de periodos del proyecto es p , el número de columnas de la tabla es $p+2$.

Si el proyecto es de nueva creación y no se han introducido datos de ingresos, la tabla aparecerá en blanco, con excepción del primer renglón (en azul) -en el que aparecerán el encabezado "**Concepto de Ingreso / Periodo**" y los números $0,1,\dots,p$ correspondientes a los años de vida útil del proyecto- y el último renglón de la primera columna - en que aparecerá la leyenda "**Suma de Ingresos**".

En este momento deben introducirse los conceptos de ingreso y sus correspondientes importes, de igual forma en que se haría en cualquier hoja de cálculo. Al ir introduciendo los datos irán apareciendo en el último renglón, los valores correspondientes a la suma de ingresos para cada periodo.

TESIS CON
FALLA DE ORIGEN

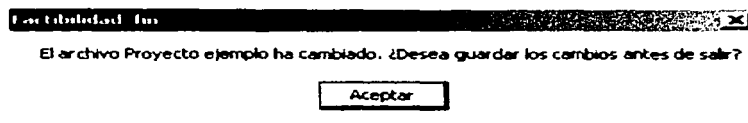
Ingresos operativos del proyecto					
Archivo		Ayuda			
Iniciar	Abrir	Guardar	Proyecto ejemplo		
Ventas producto 1	1000	2000	2000	3500	2500
Ventas producto 2	1500	1500	4000		
Ventas producto 3	35000	3500	4500		355
Ventas a crédito	500	500			

Figura 6.11 Datos ingresos del proyecto Proyecto ejemplo.

Cuando al introducir los datos se comete algún error, éste se puede corregir borrándolo con la tecla **Suprimir**, con la tecla de retroceso, o sobre escribiendo el valor. Al hacerlo, el nuevo valor corregido de la suma de ingresos irá apareciendo en la columna correspondiente.

Una vez capturados todos los datos de inversión pueden guardarse como parte del proyecto eligiendo **Guardar** en el menú **Archivo**. Al hacerlo aparecerá una caja de diálogo como la mostrada en la figura 6.3.

Si los datos fueron modificados al intentar salir del módulo el programa preguntará si se desean guardar los cambios.



A continuación se presenta el código del módulo.

TESIS CON
FALLA DE CALIDAD

Ingreso (código del módulo)

Dim indr As Integer, indcol As Integer, pos As Integer, nrows As Integer

Private Sub MenuCargaDatos_Click()

```
Modulo = "Ingreso"
dimmat = nci + nce + ncinv + 4
ReDim MatFlujoN(dimmat, nper), C1Str(0 To nci + 1)
For indr = 1 To nci
For indcol = 1 To nper
MatFlujoN(indr, indcol) = 0
Next indcol
Next indr
```

```
If NombreDeArchivo = "" Then
MsgBox " Debe abrir o iniciar primero un proyecto en el Módulo de inicio ",
vbExclamation, "Error de entrada de datos"
```

```
Exit Sub
```

```
Else
```

```
    ncom = nci
    Call CargaCuadrícula(MSHFlexGridIng)
    For col = 0 To nper
        MSHFlexGridIng.col = col
        MSHFlexGridIng.CellBackColor = &HC0C000
    Next col
```

```
End If
```

```
End Sub
```

Private Sub MenuEDitaDatos_Click()

```
End Sub
```

Private Sub MenuGuardar_Click()

```
Modulo = "Ingreso"
```

```
'Si ocurre un error ejecutar ManipulaErrorGuardar
```

```
On Error GoTo 100
```

```
'genera un error cuando se pulsa cancelar
```

```
CommonDialog1.CancelError = True
```

```
'Filtros
```

```
CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
    "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
```

```
'Filtro por defecto
```

```
CommonDialog1.FilterIndex = 1
```

```
'Visualizar la caja de diálogo
```



```
CommonDialog1.ShowSave
n = FreeFile
'Abrir el fichero para escribir
Open CommonDialog1.FileName For Random As n
```

```
dimmat = nci + nce + ncinv + 4
ReDim MatFlujoN(0 To dimmat, 1 To nper)
ReDim CiStr(0 To nci + 1)
```

```
Call Guardar(MSHFlexGridIng)
```

```
SalirGuardar:
Close #n
IndiceModif = 0
100 Call ManipulaErrorGuardar
End Sub
```

```
Private Sub MenuImprimir_Click()
Modulo = "Ingreso"
CommonDialog2.ShowPrinter
Toolbar1.Visible = False
PrintForm
20 Printer.EndDoc
Toolbar1.Visible = True
End Sub
```

```
Public Sub MenuRecuperaDatos_Click()

Modulo = "Ingreso"

'Si ocurre un error ejecutar ManipulaErrorGuardar
On Error GoTo 100
CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar

'Filtros
CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)" & _
    "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
'Filtro por defecto
CommonDialog1.FilterIndex = 1

If NombreDeArchivo <> "" Then GoTo 99

'Visualizar la caja de diálogo
CommonDialog1.ShowOpen
n = FreeFile
Open CommonDialog1.FileName For Random As n
```

```
NombreDeArchivo = CommonDialog1.FileName  
99 Text1.Text = NombreDeArchivo
```

```
101 Get #n, 2, nci
```

```
Get #n, 3, nce
```

```
Get #n, 4, ncinv
```

```
Get #n, 5, nper
```

```
dimmat = nci + nce + ncinv + 4
```

```
ReDim MatFlujoN(dimmat, nper)
```

```
ReDim CiStr(nci + 1)
```

```
ncom = nci
```

```
102 Call CargaCuadrícula(MSHFlexGridIng)
```

```
For col = 0 To nper
```

```
MSHFlexGridIng.col = col
```

```
MSHFlexGridIng.CellBackColor = &HC0C000
```

```
Next col
```

```
100 Call Recuperar(MSHFlexGridIng)
```

SalirAbrir:

'Esta instrucción permite abrir y actualizar el formulario de flujo de efectivo cuando se capturan datos en este módulo

FlujoDeEfectivoNeto.MenuRecuperaDatos_Click

```
Exit Sub
```

```
103 ManipulaErrorAbrir:
```

```
'Manipular el error.
```

```
If Err.Number = cdICancel Then Exit Sub 'Se pulsó Cancelar
```

```
End Sub
```

```
Private Sub MenuSalir_Click()
```

```
If IndiceModif = 1 Then
```

```
MsgBox " El archivo " & NombreDeArchivo & " ha cambiado. ¿Desea guardar los cambios antes de salir?"
```

```
Call MenuGuardar_Click
```

```
ReDim CiStr(nci + nce + ncinv + 5)
```

```
If IndiceEgreso = 1 Then
```

```

Modulo = "Egreso"
MSHFlexGridIng.Rows = nce + 2
Open NombreDeArchivo For Random As n
Call Guardar(Egreso.MSHFlexGridIng)
Else

If IndiceInversion = 1 Then
  Modulo = "Inversión Neta"
  MSHFlexGridIng.Rows = ncin + 2
  Call Guardar(InversionNeta.MSHFlexGridIng)
  Close #n
Else
  Close #n
End If
End If
End If
  End
End Sub

```

```

Private Sub MSHFlexGridIng_KeyPress(KeyAscii As Integer)

```

```

  Modulo = "Ingreso"
  Call PresTeclado(MSHFlexGridIng, KeyAscii)

```

```

End Sub

```

```

Private Sub MSHFlexGridIng_KeyUp(KeyCode As Integer, Shift As Integer)

```

```

  Modulo = "Ingreso"
  Call UpTecla(MSHFlexGridIng, KeyCode, Shift)

```

```

End Sub

```

```

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)

```

```

  Modulo = "Ingreso"
  Select Case Button.Key

    Case "BtCarDat"
      MenuCargaDatos_Click
    Case "BtAbrir"
      MenuRecuperaDatos_Click
    Case "BtGuardarDat"
      MenuGuardar_Click

```

```

End Select
End Sub

```

6.2.4 Egreso(Egresos operativos del proyecto)

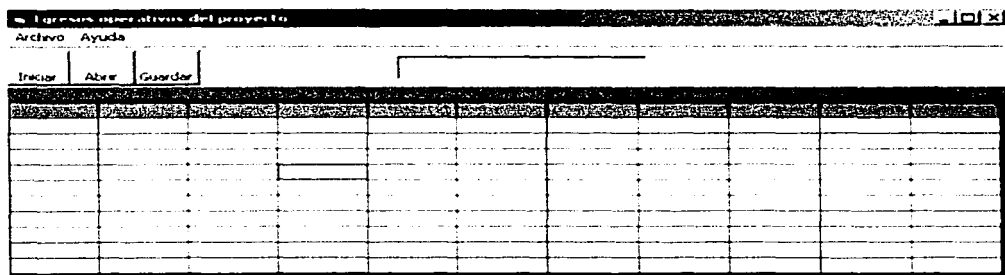


figura 6.12 Imagen del formulario Egreso frm

Para acceder al módulo de Egresos se debe pulsar el botón Egresos en la barra de herramientas del módulo inicial. Al hacerlo, se despliega la pantalla que aparece en la figura 6.12

El formulario consta de un menú de opciones, una barra de herramientas, una tabla de datos y una caja de texto en donde se muestra el nombre del proyecto activo

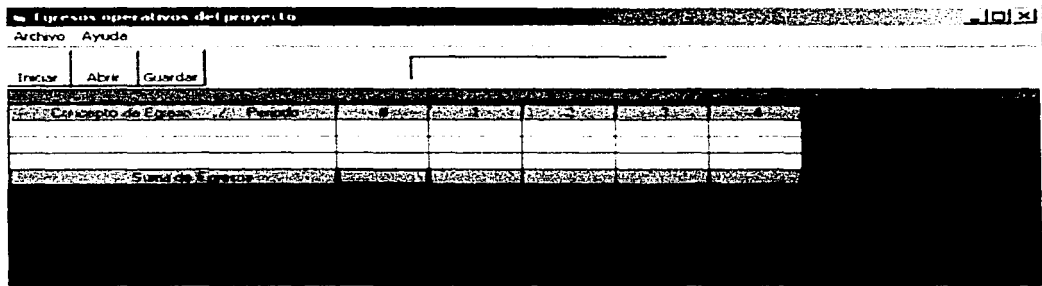


Figura 6.13 Formulario correspondiente al proyecto: Proyecto ejemplo de la figura 6.2

Para iniciar la captura de datos se deberá elegir Iniciar en el menú Archivo. Al hacerlo el formulario aparecerá con el número de renglones y columnas con que se dimensionó en el módulo de inicio. Si el número de conceptos de egreso es ne , el número de renglones de la tabla de datos será $ne+2$. Así mismo, si el número de periodos del proyecto es p , el número de columnas de la tabla es $p+2$.

Si el proyecto es de nueva creación y no se han introducido datos de ingresos, la tabla aparecerá en blanco, con excepción del primer renglón (en azul) -en el que aparecerán el encabezado "Concepto de Egreso / Periodo" y los números 0,1,...,p correspondientes a los años de vida útil del proyecto- y el último renglón de la primera columna – en que aparecerá la leyenda " Suma de Egresos".

En este momento deben introducirse los conceptos de egreso y sus correspondientes importes, de igual forma en que se haría en cualquier hoja de cálculo. Al ir introduciendo los datos irán apareciendo en el último renglón, los valores correspondientes a la suma de egresos para cada periodo

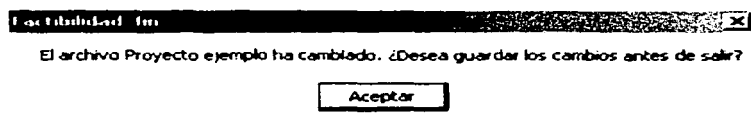
Concepto de Egreso / Periodo	0	1	2	3	4
Mano de obra	850	850	850	850	850
Materiales	1500	1500	1500	1500	1500
Herrajes	60	60	60	60	60
Suma de Egresos	2410	2410	2410	2410	2410

Figura 6.14 Datos de egresos del proyecto: Proyecto ejemplo.

Cuando al introducir los datos se comete algún error, éste se puede corregir borrándolo con la tecla Suprimir, con la tecla de retroceso, o sobre escribiendo el valor. Al hacerlo, el nuevo valor corregido de la suma de ingresos irá apareciendo en la columna correspondiente.

Una vez capturados todos los datos de inversión pueden guardarse como parte del proyecto eligiendo Guardar en el menú Archivo. Al hacerlo aparecerá una caja de diálogo como la mostrada en la figura 6.3

Si los datos fueron modificados al intentar salir del módulo el programa preguntará si se desean guardar los cambios.



A continuación se presenta el código del módulo.

Egreso (código del módulo)

Dim textosuma As String

Private Sub MenuCargaDatos_Click()

```
Modulo = "Egreso"
```

```
IndiceEgreso = 1
```

```
    dimmat = nci + nce + ncinv + 4
```

```
    ReDim MatFlujoN(dimmat, nper), CiStr(nci + nce + 2)
```

```
    For indr = 1 To nce
```

```
        For indcol = 1 To nper
```

```
            MatFlujoN(indr, indcol) = 0
```

```
        Next indcol
```

```
    Next indr
```

```
If NombreDeArchivo = "" Then
```

```
MsgBox " Debe abrir o iniciar primero un proyecto en el Módulo de inicio ",
```

```
vbExclamation, "Error de entrada de datos"
```

```
Exit Sub
```

```
Else
```

```
    ncom = nce
```

```
    Call CargaCuadrícula(MSHFlexGridIng)
```

```
    For col = 0 To nper
```

```
        MSHFlexGridIng.col = col
```

```
        MSHFlexGridIng.CellBackColor = &HFF&
```

```
    Next col
```

```
End If
```

```
End Sub
```

Private Sub MenuEditar_Click()

```
End Sub
```

Private Sub MenuGuardar_Click()

```
    Modulo = "Egreso"
```

```
'Si ocurre un error ejecutar ManipulaErrorGuardar
```

```
    On Error GoTo 100
```

```
'genera un error cuando se pulsa cancelar
```

```
    CommonDialog1.CancelError = True
```

```
'Filtros
```

```
    CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
```

```

        "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
'Filtro por defecto
    CommonDialog1.FilterIndex = 1
'Visualizar la caja de diálogo
    CommonDialog1.ShowSave
        n = FreeFile
'Abrir el fichero para escribir
    Open CommonDialog1.FileName For Random As n

    dimmat = nci + nce + ncinv + 4
    ReDim MatFlujoN(0 To dimmat, 1 To nper)
    ReDim CiStr(0 To nci + nce + 2)

    Call Guardar(MSHFlexGridIng)

SalirGuardar:
    Close #n
    IndiceModif = 0
    IndiceEgreso = 0
100 Call ManipulaErrorGuardar
End Sub

```

```
Public Sub MenuRecuperaDatos_Click()
```

```

    Modulo = "Egreso"
    IndiceEgreso = 1
'Si ocurre un error ejecutar ManipulaErrorGuardar
    On Error GoTo 100
    CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar

'Filtros
    CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
        "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
'Filtro por defecto
    CommonDialog1.FilterIndex = 1

    If NombreDeArchivo <> "" Then GoTo 99

'Visualizar la caja de diálogo
    CommonDialog1.ShowOpen
        n = FreeFile
    Open CommonDialog1.FileName For Random As n
    NombreDeArchivo = CommonDialog1.FileName
99 Text1.Text = NombreDeArchivo

    Get #n, 2, nci
    Get #n, 3, nce

```

```

Get #n, 4, ncinv
Get #n, 5, nper
dimmat = nci + nce + ncinv + 4
ReDim MatFlujoN(dimmat, nper)
ReDim CiStr(0 To nci + nce + 2)
ncom = nce
102 Call CargaCuadrícula(MSHFlexGridIng)
For col = 0 To nper
MSHFlexGridIng.col = col
MSHFlexGridIng.CellBackColor = &HFF&
Next col
100 Call Recuperar(MSHFlexGridIng)

```

SalirAbrir:

'Esta instrucción permite abrir y actualizar el formulario de flujo de efectivo cuando se capturan datos en este módulo.

FlujoDeEfectivoNeto.MenuRecuperaDatos_Click

Exit Sub

103 ManipulaErrorAbrir:

'Manipular el error.

If Err.Number = cdCancel Then Exit Sub 'Se pulsó Cancelar

End Sub

Private Sub MenuImprimir_Click()

Modulo = "Egreso"

CommonDialog2.ShowPrinter

Toolbar1.Visible = False

PrintForm

20 Printer.EndDoc

Toolbar1.Visible = True

End Sub

Private Sub MenuSalir_Click()

If IndiceModif = 1 Then

MsgBox " El archivo " & NombreDeArchivo & " ha cambiado. ¿Desea guardar los cambios antes de salir?"

Call MenuGuardar_Click

ReDim CiStr(nci + nce + ncinv + 5)

If IndiceIngreso = 1 Then

Modulo = "Ingreso"

MSHFlexGridIng.Rows = nce + 2


```
Open NombreDeArchivo For Random As n
Call Guardar(Ingreso.MSHFlexGridIng)
Else
If IndiceInversion = 1 Then
Modulo = "Inversión Neta"
MSHFlexGridIng.Rows = ncinv + 2
Call Guardar(InversionNeta.MSHFlexGridIng)
Close #n
Else
Close #n
End If
End If
End If
End Sub
```

```
Private Sub MSHFlexGridIng_KeyPress(KeyAscii As Integer)
Modulo = "Egreso"

ReDim CiStr(0 To nci + nce + 2)
Call PresTeclado(MSHFlexGridIng, KeyAscii)

End Sub
```

```
Private Sub MSHFlexGridIng_KeyUp(KeyCode As Integer, Shift As Integer)

Modulo = "Egreso"
Call UpTecla(MSHFlexGridIng, KeyCode, Shift)

End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Modulo = "Egreso"
Select Case Button.Key
Case "BtCarDat"
MenuCargaDatos_Click
Case "BtAbrir"
MenuRecuperaDatos_Click
Case "BtGuardarDat"
MenuGuardar_Click
End Select
End Sub
```

6.2.5. FlujoDeEfectivoNeto. (Flujo de efectivo neto del proyecto).

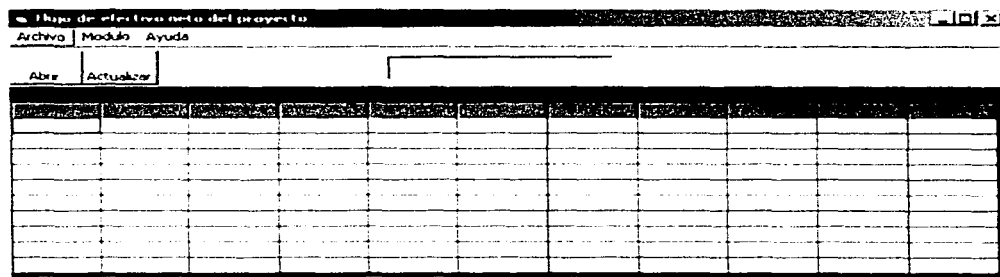


Figura 6.15 imagen del formulario FlujoDeEfectivoNeto frm

Para acceder al módulo de flujo de efectivo se debe pulsar el botón **Flujo de efectivo** en la barra de herramientas del módulo inicial. Al hacerlo, se despliega la pantalla que aparece en la figura 6.15

El formulario consta de un menú de opciones, una barra de herramientas, una tabla de datos y una caja de texto en donde se muestra el nombre del proyecto activo

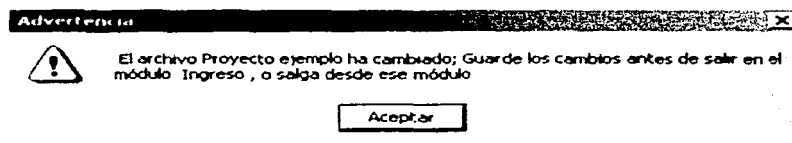
	1	2	3	4	5
Ventas producto 1	1.000,00	2.000,00	2.000,00	2.500,00	2.500,00
Ventas producto 2	1.500,00	1.500,00	2.500,00	2.500,00	2.500,00
Ventas producto 3	35.000,00	3.500,00	4.500,00	255,00	2.500,00
Ventas a crédito	500,00	500,00	400,00	400,00	400,00
Suma de Ingresos	38.000,00	7.500,00	9.400,00	5.655,00	7.900,00
Materia prima	850,00	160,00	1170,00	880,00	900,00
Materiales	1.500,00	1.500,00	1.500,00	1.500,00	1.700,00
Electricidad	60,00	65,00	75,00	90,00	100,00
Suma de Egresos	2.410,00	2.425,00	2.448,00	2.570,00	2.700,00
Maquinaría y equipo	0,00	500,00	0,00	0,00	0,00
Equipo de oficina	200,00	45,00	0,00	0,00	0,00
Suma de Inversión Neta	200,00	545,00	0,00	0,00	0,00

Figura 6.16 imagen del formulario del proyecto 'Proyecto ejemplo'

El módulo **Flujo de Efectivo** es una tabla resumen del flujo de efectivo neto del proyecto. No es posible modificar los datos que aparecen en él.

Una vez que se ha abierto el formulario, si existe un proyecto activo, la tabla mostrará los valores por periodo de cada uno de los conceptos de inversión, ingreso y egreso del proyecto, así como del flujo de efectivo neto del proyecto. Si se desea modificar alguno de los valores de la tabla deberá abrirse el formulario correspondiente, modificar el valor de acuerdo al procedimiento descrito para los módulos anteriores. Al hacerlo se mostrarán inmediatamente los nuevos valores de la tabla.

Si se hizo alguna modificación en cualquiera de los formularios de Inversión Neta, Ingreso o Egreso, al salir del formulario el programa mostrará el siguiente mensaje.



Ello implica que se deberá salir de la aplicación desde ese módulo y grabar los cambios antes de salir

A continuación se muestra el código del módulo

FlujoDeEfectivoNeto (código del módulo)

Dim ndr As Integer, indcol As Integer, pos As Integer, rows As Integer, flujoefec As Double

```
Private Sub MenuImprimir_Click()
Modulo = "Ingreso"
CommonDialog2.ShowPrinter
Toolbar1.Visible = False
PrintForm
20 Printer.EndDoc
Toolbar1.Visible = True
End Sub
```

TESIS CON
FALLA DE ORIGEN

```
Public Sub MenuRecuperaDatos_Click()
'La variable guia permite identificar si se está modificando
'algún formulario de entrada de datos
If guia <> "" Then Exit Sub
Modulo = "Flujo de efectivo"
```

```
'Si ocurre un error ejecutar ManipulaErrorGuardar
On Error GoTo 100
CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar
```

```
'Filtros
  CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)|" & _
    ""*.txt;*.doc|Todos los ficheros(*.*)|*.*"
'Filtro por defecto
  CommonDialog1.FilterIndex = 1

  If NombreDeArchivo <> "" Then GoTo 99
'Visualizar la caja de diálogo
  CommonDialog1.ShowOpen
  n = FreeFile
  Open CommonDialog1.FileName For Random As n
  NombreDeArchivo = CommonDialog1.FileName
99 Text1.Text = NombreDeArchivo
```

```
Get #n, 1, NombreDeArchivo
Get #n, 2, nci
Get #n, 3, nce
Get #n, 4, ncinv
Get #n, 5, nper
```

```
dimmat = nci + nce + ncinv + 5
ReDim MatFlujoN(dimmat, nper)
ReDim CiStr(nci + nce + ncinv + 5)
ncom = nci + nce + ncinv + 3
```

```
102 Call CargaCuadrícula(MSHFlexGridIng)
```

```
100 Call Recuperar(MSHFlexGridIng)
```

```
reng = nci + nce + ncinv + 4
For col = 0 To nper
  MSHFlexGridIng.col = col
  MSHFlexGridIng.Row = nci + 1
  MSHFlexGridIng.CellBackColor = &HFFC0C0
  MSHFlexGridIng.Row = nci + nce + 2
  MSHFlexGridIng.CellBackColor = &HFFC0C0
  MSHFlexGridIng.Row = nci + nce + ncinv + 3
  MSHFlexGridIng.CellBackColor = &HFFC0C0
```

```
flujoefec = 0
MSHFlexGridIng.Row = reng
MSHFlexGridIng.col = col
MSHFlexGridIng.CellBackColor = &HFF8080
```

```

    flujofec = MatFlujoN(nci + 1, col) - MatFlujoN(nci + nce + 2, col) - MatFlujoN(nci
+ nce + ncinv + 3, col)
    If col = 0 Then GoTo 10
    formato = "standard"
    MSHFlexGridIng.Text = Format(flujofec, formato)
10 Next col

```

SalirAbrir:

```

'Close #n
Exit Sub

```

103 ManipulaErrorAbrir:

'Manipular el error.

```

If Err.Number = cdlCancel Then Exit Sub 'Se pulsó Cancelar

```

End Sub

Private Sub MenuSalir_Click()

```

If IndiceModif = 1 Then
    texto = Modulo
    MsgBox " El archivo " & NombreDeArchivo & " ha cambiado; Guarde los
cambios antes de salir en el módulo " & texto & " , o salga desde ese módulo " ,
vbExclamation, " Advertencia"
    Select Case Modulo
        Case "Ingreso"
            Ingreso.Show vbModeless
        Case "Egreso"
            Egreso.Show vbModeless
        Case "Inversión Neta"
            InversionNeta.Show vbModeless
    End Select
    Exit Sub
Else
    Close #n
End If

```

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)

```

    Modulo = "Ingreso"
    Select Case Button.Key

```

```

'Case "BtCarDat"
'MenuCargaDatos_Click
Case "BtAbrir"

```


indica su etiqueta, el valor de la tasa de interés (tasa de descuento) con que será evaluado el proyecto.

Con el formulario vacío, como se muestra en la figura 6.18, deberá pulsarse el botón **Abrir** de la barra de herramientas, o elegir **Abrir** en el menú **Archivo**. Si no existe un proyecto activo, el programa desplegará una **Caja de Diálogo Guardar/Abrir** como la mostrada en la figura 6.4 Deberá entonces elegirse el nombre del proyecto que se desea abrir. En el caso de existir un proyecto activo, al pulsar **Abrir** el programa mostrará un formulario como el de la figura 6.18.

Concepto / Periodo	0	1	2	3	4
Ingresos Operativos	88.000,00	7.400,00	9.400,00	4.650,00	7.000,00
Egresos operativos	2.310,00	2.400,00	2.450,00	2.520,00	2.700,00
Inversión neta	200.000,00	545.000,00	0,00	0,00	0,00
Flujo de efectivo neto	184.310,00	5.000,00	6.950,00	2.130,00	5.000,00

Tasa de interés %

Valor presente neto

Tasa interna de rendimiento

Periodo de recuperación de la inversión

Figura 6.18 Imagen del formulario del proyecto: Proyecto ejemplo

El formulario contiene una tabla que muestra las sumas de: ingresos operativos, egresos operativos, inversión neta y flujo de efectivo neto, para cada periodo de la vida útil del proyecto.

En este momento deberá introducirse, en la caja de texto con la etiqueta "Tasa de interés %" la tasa de descuento con que será utilizada para el cálculo del valor presente neto, la tasa interna de rendimiento y el periodo de recuperación de la inversión. Para conocer el valor de estos parámetros deberá pulsarse el botón correspondiente. Al hacerlo el formulario mostrará dos renglones adicionales mostrando el **valor presente del flujo de efectivo** y el **valor presente del flujo de efectivo acumulado**. En las cajas de texto junto a los botones de comando se mostrará el valor correspondiente a: **Valor presente neto**, **Tasa interna de rendimiento** y **Periodo de recuperación de la inversión**.

TESIS CON
FALLA DE ORIGEN

Rentabilidad del proyecto

Archivo

Proyecto ejemplo

Concepto / Período	0	1	2	3	4
Ingresos Operativos	89.000,00	7.500,00	7.400,00	7.275,00	7.000,00
Egresos Operativos	2.410,00	2.425,00	2.450,00	2.470,00	2.700,00
Inversión neta	7.200.000,00	549.000,00	0,00	0,00	0,00
Flujo de efectivo neto	1.164.410,00	5.99.575,00	5.950,00	3.805,00	5.300,00
Valor presente del flujo de efectivo	1.164.410,00	514.215,24	5.413,65	2.864,34	4.278,05
Valor presente del flujo de efectivo de unidades	1.164.410,00	1.670.625,24	1.672.921,36	1.659.656,44	1.665.378,39

Tasa de interés: 2

Valor presente neto: 1.665.378,39

Tasa interna de rendimiento:

Periodo de recuperación de la inversión: 25,729

Figura 6-19 Imagen del formulario del proyecto Proyecto ejemplo

Para salir del formulario deberá pulsarse **Salir** en el menú **Archivo**.

A continuación se presenta el código del módulo.

Rentabilidad (código del módulo)

Dim interes As Single

Dim VP() As Currency, VPcol As Currency, flujofec() As Double, TiR As Double

Private Sub CdBIvp_Click()

 interes = CSng(Text1.Text)

 ReDim VP(nper)

 If Text1.Text = "" Then

 MsgBox "Introduzca el valor de la tasa de interés ", vbExclamation, "Error de entrada de datos"

 Exit Sub

 Else

 interes = CSng(Text1.Text) / 100

 MSHFlexGridIng.Rows = 7


```

MSHFlexGridIng.col = 0
MSHFlexGridIng.Row = 5
MSHFlexGridIng.Text = "Valor presente del flujo de efectivo "
MSHFlexGridIng.CellBackColor = &HC0E0FF
VP(0) = 0
For col = 1 To nper
VPcol = flujoefec(col) / ((1 + interes) ^ (col - 1))
VP(col) = VP(col - 1) + VPcol
MSHFlexGridIng.col = col
MSHFlexGridIng.Row = 5
MSHFlexGridIng.Text = Format(VPcol, formato)
MSHFlexGridIng.Row = 6
MSHFlexGridIng.Text = Format(VP(col), formato)
MSHFlexGridIng.CellBackColor = &HC0FFFF
Next col
MSHFlexGridIng.Row = 6
MSHFlexGridIng.col = 0
MSHFlexGridIng.Text = "Valor presente del flujo de efectivo acumulado"
MSHFlexGridIng.CellBackColor = &HC0FFFF
Text2.Text = Format(VP(nper), formato)
End If

```

End Sub

```

Private Sub CdTIR_Click()
If Text1.Text = "" Then
MsgBox " Introduzca el valor de la tasa de interés ", vbExclamation, "Error de
entrada de datos"
Exit Sub
Else
TiR = IRR(flujoefec)
Text3.Text = Format(TiR * 100, formato)
End If
End Sub

```

```

Private Sub CdTrec_Click()
If Text1.Text = "" Then
MsgBox " Introduzca el valor de la tasa de interés ", vbExclamation, "Error de
entrada de datos"
Exit Sub
Else
For col = 2 To nper
If flujoefec(col) > 0 And flujoefec(col - 1) < 0 Then
trec = (col - 2) + (-VP(col - 1) / (flujoefec(col) / ((1 + interes) ^ (col - 1))))
Text4.Text = Format(trec, formato)
Else

```

```
End If
Next col
End If
```

```
End Sub
```

```
-----
Private Sub MenuActualizarRenta_Click()
'+++++++esta subrutina está pendiente de desarrollar+++++++
'For reng = 0 To 6
'For col = 1 To nper
'MSHFlexGridIng.Row = reng
'MSHFlexGridIng.col = col
```

```
End Sub
```

```
-----
Private Sub MenuDatosRenta_Click()
```

```
Modulo = "Rentabilidad"
```

```
'Si ocurre un error ejecutar ManipulaErrorGuardar
'On Error GoTo 100
CommonDialog1.CancelError = True 'genera un error cuando se pulsa cancelar
```

```
If NombreDeArchivo = "" Then
```

```
MsgBox " Debe abrir o iniciar primero un proyecto en el Módulo de inicio ",
vbExclamation, "Error de entrada de datos"
Exit Sub
```

```
Else
```

```
ReDim flujoefec(1 To nper)
```

```
'Filtros
```

```
CommonDialog1.Filter = "Ficheros de Texto(*.txt;*.doc)" & _
    "*.txt;*.doc|Todos los ficheros(*.*)|*.*"
```

```
'Filtro por defecto
```

```
CommonDialog1.FilterIndex = 1
```

```
If NombreDeArchivo <> "" Then GoTo 99
```

```
'Visualizar la caja de diálogo
```

```
CommonDialog1.ShowOpen
```

```
n = FreeFile
```

```
Open CommonDialog1.FileTitle For Random As n
```

```
NombreDeArchivo = CommonDialog1.FileTitle
```

```
99 Text5.Text = NombreDeArchivo
```

```
Get #n, 1, NombreDeArchivo
Get #n, 2, nci
Get #n, 3, nce
Get #n, 4, ncinv
Get #n, 5, nper
```

```
dimmat = nci + nce + ncinv + 5
ReDim MatFlujoN(dimmat, nper)
ReDim CiStr(nci + nce + ncinv + 5)
ncom = nci + nce + ncinv + 3
```

```
col = 0
MSHFlexGridIng.ColWidth(0) = 3400
MSHFlexGridIng.Cols = nper + 1
MSHFlexGridIng.Rows = 5
MSHFlexGridIng.ColAlignmentFixed(col) = flexAlignCenterCenter
MSHFlexGridIng.col = 0
MSHFlexGridIng.Row = 0
MSHFlexGridIng.Text = " Concepto " & " / Periodo"
```

```
For col = 1 To nper
MSHFlexGridIng.col = col
MSHFlexGridIng.CellAlignment = flexAlignCenterCenter
MSHFlexGridIng.Text = col - 1
Next col
```

```
MSHFlexGridIng.col = 0
MSHFlexGridIng.Row = 1
MSHFlexGridIng.Text = " Ingresos Operativos"
```

```
pos = 6 + (nci * (nper + 1)) + 1
For col = 1 To nper
Get #n, pos, MatFlujoN(nci + 1, col)
MSHFlexGridIng.Row = 1
MSHFlexGridIng.col = col
formato = "#,###,###,##0.00"
MSHFlexGridIng.Text = Format(MatFlujoN(nci + 1, col), formato)
```

```
pos = pos + 1
Next col
```

```
MSHFlexGridIng.Row = 2
MSHFlexGridIng.col = 0
MSHFlexGridIng.Text = " Egresos operativos"
```

```
pos = 6 + ((nci + nce + 1) * (nper + 1)) + 1
```

```
For col = 1 To nper
Get #n, pos, MatFlujoN(nci + nce + 2, col)
MSHFlexGridIng.Row = 2
MSHFlexGridIng.col = col
MSHFlexGridIng.Text = Format(MatFlujoN(nci + nce + 2, col), formato)
pos = pos + 1
Next col
```

```
MSHFlexGridIng.Row = 3
MSHFlexGridIng.col = 0
MSHFlexGridIng.Text = "Inversión neta"
```

```
pos = 6 + ((nci + nce + ncinv + 2) * (nper + 1)) + 1
For col = 1 To nper
Get #n, pos, MatFlujoN(nci + nce + ncinv + 3, col)
MSHFlexGridIng.Row = 3
MSHFlexGridIng.col = col
MSHFlexGridIng.Text = Format(MatFlujoN(nci + nce + ncinv + 3, col), formato)
pos = pos + 1
Next col
```

```
MSHFlexGridIng.Row = 4
MSHFlexGridIng.col = 0
MSHFlexGridIng.CellBackColor = &HC0FFFF
MSHFlexGridIng.Text = "Flujo de efectivo neto"
```

```
For col = 1 To nper
flujoefec(col) = MatFlujoN(nci + 1, col) - MatFlujoN(nci + nce + 2, col) -
MatFlujoN(nci + nce + ncinv + 3, col)
MSHFlexGridIng.col = col
MSHFlexGridIng.Text = Format(flujoefec(col), formato)
MSHFlexGridIng.CellBackColor = &HC0FFFF
Next col
SalirAbrir:
```

```
Close #n
Exit Sub
```

```
103 ManipulaErrorAbrir:
'Manipular el error.
If Err.Number = cdlCancel Then Exit Sub 'Se pulsó Cancelar
```

```
End If
End Sub
```

```

Private Sub MenuImprimir_Click()
Modulo = "Ingreso"
  CommonDialog2.ShowPrinter
  Toolbar1.Visible = False
  PrintForm
20 Printer.EndDoc
  Toolbar1.Visible = True
End Sub

```

```

Private Sub MenuSalir_Click()
  If IndiceModif = 1 Then
    texto = Modulo
    MsgBox " El archivo " & NombreDeArchivo & " ha cambiado; Guarde los
cambios antes de salir en el módulo " & texto & " , o salga desde ese módulo " ,
vbExclamation, " Advertencia"
    Select Case Modulo
      Case "Ingreso"
        Ingreso.Show vbModeless
      Case "Egreso"
        Egreso.Show vbModeless
      Case "Inversión Neta"
        InversiónNeta.Show vbModeless
    End Select
    Exit Sub
  Else
    Close #n
  End If
End Sub

```

```

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Modulo = "Ingreso"
Select Case Button.Key
  Case "BtAbrir"
    MenuDatosRenta_Click
  Case "BtActualiza"
    MenuActualizarRenta_Click
End Select
End Sub

```

6.2.7 Modulo Estándar (ModuleInv).

Además de los módulos descritos, el programa contiene un módulo estándar llamado **ModuleInv**, el cual contiene las declaraciones de variables globales, así como los procedimientos y funciones utilizadas por los otros módulos. Este módulo

está oculto al usuario de la aplicación, ya que no tiene ningún formulario asociado a él, solo código.

ModuleInv está formado por una sección de declaraciones, una función: **Public Function fnsuma()** y ocho procedimientos Sub: **Public Sub IniFin()**, **Public Sub CargaCuadrícula(MSHFlexGridIng)**, **Public Sub Recuperar(MSHFlexGridIng)**, **Public Sub PresTeclado(MSHFlexGridIng, KeyAscii)**, **Public Sub UpTecla(MSHFlexGridIng, KeyCode, Shift)**, **Public Sub actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto)**, **Public Sub ManipulaErrorGuardar()**. El desempeño de cada función y procedimiento en el programa se describe en la tabla 6.1

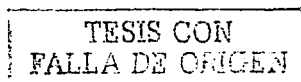
Procedimiento / función	Tarea
Public Function fnsuma()	Lleva a cabo las sumas de conceptos de ingreso, egreso e inversión neta.
Public Sub IniFin()	Establece los límites inicial y final de los contadores For Next, en función de la cantidad de conceptos de cada módulo
Public Sub CargaCuadrícula(MSHFlexGridIng)	Carga los datos a la Tabla de datos (MSHFlexGrid) de cada módulo.
Public Sub Recuperar(MSHFlexGridIng)	Recupera los datos de la tabla de un archivo de texto y los carga a la tabla de datos (MSHFlexGrid) de cada módulo
Public Sub Guardar(MSHFlexGridIng)	Guarda los datos de la tabla de datos (MSHFlexGrid) en un archivo de texto
Public Sub PresTeclado(MSHFlexGridIng, KeyAscii)	Permite formar cadenas de caracteres al oprimir teclas en el teclado y capturar así datos en la tabla de datos (MSHFlexGrid).
Public Sub UpTecla(MSHFlexGridIng, KeyCode, Shift)	Permite borrar caracteres con la tecla de retroceso y la tecla <i>Suprimir</i>
Public Sub actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto)	Actualiza los datos de la tabla de flujo de efectivo.
Public Sub actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto), Public Sub	Intercepta errores del teclado evitando suspender la ejecución del programa

Tabla 6.1 Funciones y procedimientos del programa.

A continuación se presenta el código del módulo.

ModuleInv (código del módulo).

Declaración de cantidad de conceptos y periodos
 Dim textosuma As String, textoingreso As String, textoegreso As String,
 textoinversion As String
 Public MatFlujoN() As Currency



```

Public CiStr() As String, CeStr() As String, CinvStr() As String, formato As String,
guia As String
Public nci As Integer, nce As Integer, ncin As Integer, nper As Integer, n As
Integer, indice As Integer
Public reng As Integer, retcol As Integer, retreng As Integer, dimmat As Integer
Public suma, InvNeta As Double
Public l As Integer, ancho As Integer, IndiceInversion As Integer, IndiceEgreso
As Integer, IndiceIngreso As Integer
Public NombreDeArchivo As String, Modulo As String, Proyecto As String
Dim nreng As Integer, nc As Integer, nreg As Integer, npos As Integer
Dim nindice As Integer, Nind1 As Integer, Nind2 As Integer
Public ini As Integer, fin As Integer, col As Integer, ncom As Integer, IndiceModif
As Integer

```

```

Public Sub IniFin()
Select Case Modulo
Case "Ingreso"
ini = 1
fin = nci
nreng = nci + 2
Case "Egreso"
ini = nci + 2
fin = (nci + nce) + 1
nreng = nce + 2
Case "Inversión Neta"
ini = nci + nce + 3
fin = nci + nce + ncin + 2
nreng = ncin + 2
Case "FlujoNeto"
End Select
End Sub

```

```

Public Function fnsuma()
Call IniFin
suma = 0
For reng = ini To fin
suma = suma + MatFlujoN(reng, col)
fnsuma = suma
Next reng
End Function

```

```

Public Sub CargaCuadrícula(MSHFlexGridIng)
col = 0
MSHFlexGridIng.ColWidth(0) = 3400
If Modulo = "Inversión Neta" Then MSHFlexGridIng.ColWidth(0) = 3500

```

```

MSHFlexGridIng.Cols = nper + 1
MSHFlexGridIng.Rows = ncom + 2
MSHFlexGridIng.ColAlignmentFixed(col) = flexAlignCenterCenter
MSHFlexGridIng.col = 0
MSHFlexGridIng.Row = 0
MSHFlexGridIng.Text = " Concepto de " & Modulo & " / Periodo"

```

```

For col = 1 To nper
MSHFlexGridIng.col = col
MSHFlexGridIng.CellAlignment = flexAlignCenterCenter
MSHFlexGridIng.Text = col - 1
Next col

```

```

MSHFlexGridIng.Row = ncom + 1
MSHFlexGridIng.col = 0
MSHFlexGridIng.CellAlignment = flexAlignCenterCenter
MSHFlexGridIng.Text = " Suma de " & Modulo & "s " & ""
If Modulo = "Inversión Neta" Then MSHFlexGridIng.Text = " Suma de " &
Modulo & ""
If Modulo = "Flujo de efectivo" Then MSHFlexGridIng.Text = " Flujo de efectivo
neto"
End Sub

```

```

Public Sub Recuperar(MSHFlexGridIng)

```

```

Select Case Modulo
Case "Ingreso"
pos = 6
Nind1 = nci
Nind2 = 0
Case "Egreso"
pos = 6 + (nci + 1) + ((nci + 1) * nper)
Nind1 = nce
Nind2 = nci + 1
Case "Inversión Neta"
pos = 6 + (nper + 1) * (nci + nce + 2)
Nind1 = ncin
Nind2 = nci + nce + 2
Case "Flujo de efectivo"
pos = 6
Nind1 = ncom - 1
Nind2 = 0
Case "Rentabilidad"
pos = 6
Nind1 = ncom - 1
Nind2 = 0
End Select

```



```

For reng = 1 To Nind1 + 1
For col = 0 To nper
MSHFlexGridIng.Row = reng
MSHFlexGridIng.col = col

If col = 0 Then
Get #n, pos, CiStr(reng + Nind2)
MSHFlexGridIng.Text = CiStr(reng + Nind2)
Else
Get #n, pos, MatFlujoN(reng + Nind2, col)
formato = "standard"
MSHFlexGridIng.Text = Format(MatFlujoN(reng + Nind2, col), formato)
End If
pos = pos + 1
Next col
Next reng

```

End Sub

Public Sub Guardar(MSHFlexGridIng)

```

Select Case Modulo
Case "Ingreso"
pos = 6
Nind1 = nci
Nind2 = 0
Case "Egreso"
pos = 6 + (nper + 1) * (nci + 1)
Nind1 = nce
Nind2 = nci + 1
Case "Inversión Neta"
pos = 6 + (nper + 1) * (nci + nce + 2)
Nind1 = ncinv
Nind2 = nci + nce + 2
End Select

```

```

For reng = 1 To (Nind1 + 1)
For col = 0 To nper
MSHFlexGridIng.Row = reng
MSHFlexGridIng.col = col
If col = 0 Then
CiStr(reng + Nind2) = MSHFlexGridIng.Text
Put #n, pos, CiStr(reng + Nind2)
Else
MatFlujoN(reng + Nind2, col) = CCur(MSHFlexGridIng.Text)
Put #n, pos, MatFlujoN(reng + Nind2, col)
End If

```

```
pos = pos + 1
Next col
Next reng
```

End Sub

Public Sub PresTeclado(MSHFlexGridIng, KeyAscii)

```
    guia = Modulo
    FlujoDeEfectivoNeto.MenuRecuperaDatos_Click
1   Modulo = guia

    Select Case Modulo
    Case "Ingreso"
        Nind1 = nci
        Nind2 = 0
    Case "Egreso"
        Nind1 = nce
        Nind2 = nci + 1
    Case "Inversión Neta"
        Nind1 = ncinv
        Nind2 = nci + nce + 2
    End Select

    reng = MSHFlexGridIng.Row
    If reng = Nind1 + 1 Then Exit Sub
    retreng = reng
    col = MSHFlexGridIng.col
    retcol = col
    If KeyAscii >= 32 And KeyAscii <= 127 Then
    If MSHFlexGridIng.Text = "0" Then
        Dim str As String
        str = "" & Chr(KeyAscii)
        MSHFlexGridIng.Text = "" & Chr(KeyAscii)
    Else
        MSHFlexGridIng.Text = MSHFlexGridIng.Text & Chr(KeyAscii)
    End If

    If col = 0 Then GoTo 10
    MatFlujoN(reng + Nind2, col) = CCur(MSHFlexGridIng.Text)

'Suma de ingresos por periodo

    MatFlujoN(Nind1 + Nind2 + 1, col) = fnsuma()

    col = retcol
```

```
reng = retreng
MSHFlexGridIng.col = col
MSHFlexGridIng.Row = Nind1 + 1
MSHFlexGridIng.Text = Format((MatFlujoN(Nind1 + Nind2 + 1, col)), formato)
```

```
End If
GoTo 20
```

```
10 CiStr(reng + Nind2) = MSHFlexGridIng.Text
```

```
20 MSHFlexGridIng.col = retcol
MSHFlexGridIng.Row = retreng
```

```
Call actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto)
```

```
End Sub
```

```
Public Sub UpTecla(MSHFlexGridIng, KeyCode, Shift)
```

```
Select Case Modulo
```

```
Case "Ingreso"
```

```
Nind1 = nci
```

```
Nind2 = 0
```

```
Case "Egreso"
```

```
Nind1 = nce
```

```
Nind2 = nci + 1
```

```
Case "Inversión Neta"
```

```
Nind1 = ncinv
```

```
Nind2 = nci + nce + 2
```

```
End Select
```

```
reng = MSHFlexGridIng.Row
```

```
If reng = Nind1 + 1 Then Exit Sub
```

```
retreng = reng
```

```
col = MSHFlexGridIng.col
```

```
retcol = col
```

```
Select Case KeyCode
```

```
Case vbKeyDelete
```

```
If col = 0 Then
```

```
MSHFlexGridIng.Text = ""
```

```
CiStr(reng + Nind2) = MSHFlexGridIng.Text
```

```
Exit Sub
```

```
Else
```

```
MSHFlexGridIng.Text = 0
```

```
GoTo 10
```

```
End If
```

```

Case vbKeyBack
    MSHFlexGridIng.col = retcol
    MSHFlexGridIng.Row = retreng
If Len(MSHFlexGridIng.Text) > 0 Then
    MSHFlexGridIng.Text = _
    Left(MSHFlexGridIng.Text, Len(MSHFlexGridIng.Text) - 1)
    If Len(MSHFlexGridIng.Text) = 0 Then
        MSHFlexGridIng.Text = 0
    End If
If col = 0 Then GoTo 30

10 MatFlujoN(reng + Nind2, col) = CCur(MSHFlexGridIng.Text)
    col = retcol

'Suma de ingresos por periodo
    MatFlujoN(reng + Nind2, col) = MSHFlexGridIng.Text
    MatFlujoN(Nind1 + Nind2 + 1, col) = fnsuma()

'Carga de datos a la cuadrícula

    col = retcol
    reng = retreng
    MSHFlexGridIng.col = col
    MSHFlexGridIng.Row = Nind1 + 1
    MSHFlexGridIng.Text = MatFlujoN(Nind1 + Nind2 + 1, col)
    End If

30 MSHFlexGridIng.col = retcol
    MSHFlexGridIng.Row = retreng

    Call actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto)
    MSHFlexGridIng.col = retcol
    MSHFlexGridIng.Row = retreng

    End Select

    IndiceModif = 1
End Sub

```

```

Public Sub actualiza(MSHFlexGridIng, FlujoDeEfectivoNeto)

```

```

    Select Case Modulo

```

```

    Case "Ingreso"

```

```
reng1 = MSHFlexGridIng.Row  
reng2 = nci + 1  
reng3 = nci + 1
```

```
Case "Egreso"
```

```
reng1 = MSHFlexGridIng.Row + nci + 1  
reng2 = nce + 1  
reng3 = nci + nce + 2
```

```
Case "Inversión Neta"
```

```
reng1 = MSHFlexGridIng.Row + nci + nce + 2  
reng2 = ncinv + 1  
reng3 = nci + nce + ncinv + 3  
End Select
```

```
'Actualización del valor modificado
```

```
FlujoDeEfectivoNeto.MSHFlexGridIng.Row = reng1  
FlujoDeEfectivoNeto.MSHFlexGridIng.col = MSHFlexGridIng.col
```

```
If col = 0 Then
```

```
FlujoDeEfectivoNeto.MSHFlexGridIng = MSHFlexGridIng.Text
```

```
Exit Sub
```

```
Else
```

```
FlujoDeEfectivoNeto.MSHFlexGridIng.Text = Format(MSHFlexGridIng.Text,  
formato)
```

```
'Actualización de la suma
```

```
MSHFlexGridIng.Row = reng2  
MSHFlexGridIng.col = col  
textosuma = MSHFlexGridIng.Text
```

```
FlujoDeEfectivoNeto.MSHFlexGridIng.Row = reng3  
FlujoDeEfectivoNeto.MSHFlexGridIng.Text = Format(textosuma, formato)
```

```
Select Case Modulo
```

```
Case "Ingreso"
```

```
If guia <> "" Then
```

```
Egreso.MSHFlexGridIng.Row = nce + 1
```

```
Egreso.MSHFlexGridIng.col = col
```

```
textoegreso = Egreso.MSHFlexGridIng.Text
```

```

If textoegreso = "" Then textoegreso = 0
InversionNeta.MSHFlexGridIng.Row = ncinv + 1
InversionNeta.MSHFlexGridIng.col = col
textoinversion = InversionNeta.MSHFlexGridIng.Text
If textoinversion = "" Then textoinversion = 0
textosuma = Ingreso.MSHFlexGridIng.Text - textoegreso - textoinversion
Else
textosuma = MSHFlexGridIng.Text - MatFlujoN(nci + nce + 2, col) -
MatFlujoN(nci + nce + ncinv + 3, col)
End If

```

Case "Egreso"

```

If guia <> "" Then
Ingreso.MSHFlexGridIng.Row = nci + 1
Ingreso.MSHFlexGridIng.col = col
textoiingreso = Ingreso.MSHFlexGridIng.Text
If textoiingreso = "" Then textoiingreso = 0
InversionNeta.MSHFlexGridIng.Row = ncinv + 1
InversionNeta.MSHFlexGridIng.col = col
textoinversion = InversionNeta.MSHFlexGridIng.Text
If textoinversion = "" Then textoinversion = 0
textosuma = textoiingreso - MSHFlexGridIng.Text - textoinversion
Else
textosuma = -MSHFlexGridIng.Text + MatFlujoN(nci + 1, col) - MatFlujoN(nci +
nce + ncinv + 3, col)
End If

```

Case "Inversión Neta"

```

If guia <> "" Then
Ingreso.MSHFlexGridIng.Row = nci + 1
Ingreso.MSHFlexGridIng.col = col
textoiingreso = Ingreso.MSHFlexGridIng.Text
If textoiingreso = "" Then textoiingreso = 0
Egreso.MSHFlexGridIng.Row = nce + 1
Egreso.MSHFlexGridIng.col = col
textoegreso = Egreso.MSHFlexGridIng.Text
If textoegreso = "" Then textoegreso = 0
textosuma = textoiingreso - textoegreso - MSHFlexGridIng.Text
Else
'textosuma = -MSHFlexGridIng.Text + MatFlujoN(nci + 1, col) - MatFlujoN(nci +
nce + 2, col)
End If

```

End Select

```

reng = nci + nce + ncinv + 4
FlujoDeEfectivoNeto.MSHFlexGridIng.Row = reng
FlujoDeEfectivoNeto.MSHFlexGridIng.col = col

```

```
FlujoDeEfectivoNeto.MSHFlexGridIng.Text = Format(textosuma, formato)
```

```
MSHFlexGridIng.col = retcol
```

```
MSHFlexGridIng.Row = retreng
```

```
End If
```

```
End Sub
```

```
Public Sub ManipulaErrorGuardar()
```

```
ManipulaErrorGuardar:
```

```
'Manipular el error.
```

```
  If Err.Number = cdlCancel Then Exit Sub 'Se pulsó Cancelar
```

```
'CdlCancel se obtiene coin el examinador de objetos
```

```
  MsgBox Err.Description
```

```
End Sub
```

Capítulo 7.0 Ejemplos.

Ejemplo 1.- La empresa "A" está planeando sustituir la maquinaria con que actualmente fabrica balones de balones de futbol . La maquinaria para iniciar el proyecto costara \$200,000 si el pago se realiza de contado. Los servicios de empaque y transporte costarán \$12,000, el seguro de traslado \$3,000, la instalación \$4,000 y la puesta en operación \$1,000.

La empresa "A" venderá todos sus balones al menudeo de contado, por lo cual no generará cuentas por cobrar. Así mismo, estima que, con objeto de conseguir mejores precios de materia prima, pagará a sus proveedores de contado, por lo que no se producirán cuentas por pagar.

Se estima que el primer año de operaciones se venderán 1,500 balones tipo A y 600 balones tipo B. A partir del segundo año y hasta el quinto año (último año de la vida económica del negocio), la empresa estima vender 2,400 balones del tipo A y 750 tipo B. Los precios de venta estimados para cada tipo de balón son los siguientes:

PRECIO POR BALON		
AÑO	Tipo A	Tipo B
1	\$ 40	\$ 170
2	\$ 60	\$ 180
3	\$ 70	\$ 200
4	\$ 90	\$ 200
5	\$ 100	\$ 200

La fábrica estima que los costos variables del negocio están integrados por: sueldos y salarios de obreros (\$5/balón tipo A y \$10/balón tipo B), materia prima y otros materiales (\$10/balón tipo A y \$25/balón tipo B) y, por último, servicios necesarios para la producción: agua, luz y otros (\$2/balón). Se estima que los costos variables se incrementarán un 40% el primer año y un 10% cada uno los años siguientes.

Los costos fijos del negocio son los siguientes: sueldos y salarios del personal administrativo (\$60,000 anuales), otros materiales (\$5,000 anuales), y otros servicios (\$10,000 anuales). Se estima que estos costos se incrementen un 20% en el segundo año de operaciones y que no habrá incrementos en los años posteriores.

¿Debe aceptarse el proyecto, si la tasa de descuento de los flujos de efectivo es de 15?

TESIS CON
FALLA DE ORIGEN

Solución: En el módulo de inicio creamos un proyecto al que llamaremos Ejemplo 1 Fábrica de balones. Capturemos los datos que aparecen en la pantalla. De acuerdo con el enunciado del proyecto hay dos conceptos de ingreso, seis conceptos de egreso y cinco de inversión neta. La vida económica del proyecto es de 5 años (el año 0 más cuatro años adicionales)

Módulo de Inicio

Archivo Ayuda

Inversión Ingresos Egresos Flujo de efectivo Rentabilidad

Ejemplo1 Fábrica de balones:

2

6

5

5

Aceptar

Figura 7 1 Pantalla de inicio del ejemplo1

Una vez creado el proyecto procedemos a capturar los datos del proyecto. Las pantallas correspondientes a la inversión, ingresos, egresos y flujo de efectivo se muestran a continuación.

TESIS CON
FALLA DE ORIGEN

Inversión neta del proyecto					
Archivo Ayuda					
Iniciar		Abrir		Guardar	
Ejemplo1 Fábrica de balones					
Concepto de Inversión Neta / Periodo	0	1	2	3	4
Compra de maquinaria	200000	0.00	0.00	0.00	0.00
Empaque y transporte	12000	0.00	0.00	0.00	0.00
Seguro de traslado	3000	0.00	0.00	0.00	0.00
Instalación	4000	0.00	0.00	0.00	0.00
Puesta en operación	1000	0.00	0.00	0.00	0.00
Suma de Inversión Neta	220,000.00	0.00	0.00	0.00	0.00

Figura 7 2 Inversión neta del proyecto 1.

Ingresos operativos del proyecto					
Archivo Ayuda					
Iniciar		Abrir		Guardar	
Ejemplo1 Fábrica de balones					
Ventas balones tipo A	60,000.00	144,000.00	168,000.00	216,000.00	240,000.00
Ventas balones tipoB	102,000.00	135,000.00	150,000.00	150,000.00	150,000.00

Figura 7 3 Ingresos del Ejemplo 1

TESIS CON
FALLA DE ORIGEN

Egresos operativos del proyecto

Archivo Ayuda

Iniciar Abrir Guardar

Ejemplo1 Fabrica de balones

Concepto de Egreso					
Sueldos y salarios de obreros	13,500.00	27,300.00	30,030.00	33,033.00	36,336.00
Materias primas	30,000.00	59,850.00	65,835.00	72,419.00	79,660.00
Servicios	4,200.00	5,880.00	6,468.00	7,115.00	7,826.00
Sueldos y salarios personal administrativo	60000	72000	72000	72000	72000
Otros materiales	5000	6000	6000	6000	6000
Otros servicios	10000	12000	12000	12000	12000
Suma de Egresos	122,700.00	183,030.00	192,333.00	202,567.00	213,822.00

Figura 7.4 Egresos del ejemplo 1

Flujo de efectivo neto del proyecto

Archivo Modulo Ayuda

Abrir Actualizar

Ejemplo1 Fabrica de balones

Ventas balones tipo A	50,000.00	144,000.00	168,000.00	215,000.00	240,000.00
Ventas balones tipo B	102,000.00	135,000.00	150,000.00	153,000.00	150,000.00
Suma de Ingresos	162,000.00	279,000.00	318,000.00	368,000.00	390,000.00
Sueldos y salarios de obreros	13,500.00	27,300.00	30,030.00	33,033.00	36,336.00
Materias primas	30,000.00	59,850.00	65,835.00	72,419.00	79,660.00
Servicios	4,200.00	5,880.00	6,468.00	7,115.00	7,826.00
Sueldos y salarios personal administrativo	60,000.00	72,000.00	72,000.00	72,000.00	72,000.00
Otros materiales	5,000.00	6,000.00	6,000.00	6,000.00	6,000.00
Otros servicios	10,000.00	12,000.00	12,000.00	12,000.00	12,000.00
Suma de Egresos	122,700.00	183,030.00	192,333.00	202,567.00	213,822.00
Compra de maquinaria	200,000.00	0.00	0.00	0.00	0.00
Empaque y transporte	12,000.00	0.00	0.00	0.00	0.00
Seguro de traslado	3,000.00	0.00	0.00	0.00	0.00
Instalacion	4,000.00	0.00	0.00	0.00	0.00
Puesta en operacion	1,000.00	0.00	0.00	0.00	0.00
Suma de Inversión Neta	220,000.00	0.00	0.00	0.00	0.00

Figura 7.5 Flujo de efectivo neto del ejemplo 1.

TESIS CON
FALLA DE ORIGEN

Una vez capturados los datos procedemos a evaluar el proyecto en el módulo "Rentabilidad del Proyecto", considerando una tasa de descuento del 15%.

Como podemos observar en la figura 7.6. El valor presente neto del proyecto es positivo, la tasa interna de rendimiento es superior al costo de capital (15%) y el periodo de recuperación de la inversión es de 2 años (menor a la vida económica del proyecto). Los tres criterios indican que debe aceptarse el proyecto.

Rentabilidad del proyecto					
Archivo					
Abre	Actualizar	Ejemplo1: Fabrica de balones			
Concepto / Periodo	0	1	2	3	4
Ingresos Operativos	162 000 00	279 000 00	318 000 00	366 000 00	460 000 00
Egresos operativos	122 700 00	193 000 00	192 333 00	202 567 00	213 822 00
Inversión neta	220 000 00	0 00	0 00	0 00	0 00
Flujo de efectivo neto	180 700 00	86 970 00	125 667 00	163 433 00	176 178 00
Valor presente del flujo de efectivo	180 700 00	81 452 17	85 022 31	107 459 95	100 730 34
Valor presente del flujo de efectivo acumulado	180 700 00	97 247 83	2 225 52	109 234 33	205 964 67

Tasa de interés	15
Valor presente neto	205 964 67
Tasa interna de rendimiento	58 03
Periodo de recuperación de la inversión	2 01

Figura 7.6 Rentabilidad del ejemplo1.

Ejemplo2. Supongamos que la fábrica de balones tiene la opción de adquirir la maquinaria a en tres anualidades, sin embargo la maquinaria tendría en este caso un valor de \$230.000. ¿Cuál es, en esta situación, la rentabilidad del proyecto?

TESIS CON
FALLA DE ORIGEN

Solución. En este caso solo debemos modificar los datos correspondientes a la inversión neta y volver a ejecutar el programa, con lo que la rentabilidad del proyecto se modificaría como se muestra en las figuras siguientes.

Inversión neta del proyecto						
Archivo Ayuda						
Iniciar	Abrir	Guardar	Ejemplo2 Fabrica de balones			
Concepto de Inversion Neta / Periodo	0	1	2	3	4	
Compra de maquinaria	90,000.00	90,000.00	90,000.00	0.00	0.00	
Empaque y transporte	12,000.00	0.00	0.00	0.00	0.00	
Seguro de traslado	3,000.00	0.00	0.00	0.00	0.00	
Instalación	4,000.00	0.00	0.00	0.00	0.00	
Puesta en operación	1,000.00	0.00	0.00	0.00	0.00	
Suma de Inversión Neta	110,000.00	90,000.00	90,000.00	0.00	0.00	

Figura 7 7 Inversion neta del ejemplo2

Flujo de efectivo neto del proyecto						
Archivo Modulo Ayuda						
Abrir	Actualizar	Ejemplo3 Fabrica de balones				
Ventas balones tipo A	60,000.00	144,000.00	168,000.00	216,000.00	240,000.00	
Ventas balones tipoB	102,000.00	135,000.00	150,000.00	150,000.00	150,000.00	
Suma de Ingresos	162,000.00	279,000.00	318,000.00	366,000.00	390,000.00	
Sueldos y salarios de obreros	13,500.00	27,300.00	30,030.00	33,033.00	36,336.00	
Materias primas	30,000.00	59,850.00	65,835.00	72,419.00	78,660.00	
Servicios	4,200.00	5,880.00	6,468.00	7,115.00	7,826.00	
Sueldos y salarios personal administrativo	60,000.00	72,000.00	72,000.00	72,000.00	72,000.00	
Otros materiales	5,000.00	6,000.00	6,000.00	6,000.00	6,000.00	
Otros servicios	10,000.00	12,000.00	12,000.00	12,000.00	12,000.00	
Suma de Egresos	122,700.00	183,030.00	192,333.00	202,567.00	213,822.00	
Compra de maquinaria	90,000.00	90,000.00	90,000.00	0.00	0.00	
Empaque y transporte	12,000.00	0.00	0.00	0.00	0.00	
Seguro de traslado	3,000.00	0.00	0.00	0.00	0.00	
Instalación	4,000.00	0.00	0.00	0.00	0.00	
Puesta en operación	1,000.00	0.00	0.00	0.00	0.00	
Suma de Inversión Neta	110,000.00	90,000.00	90,000.00	0.00	0.00	

Figura 7 8 Flujo de efectivo del ejemplo 2.

TESIS CON
FALLA DE ORIGEN

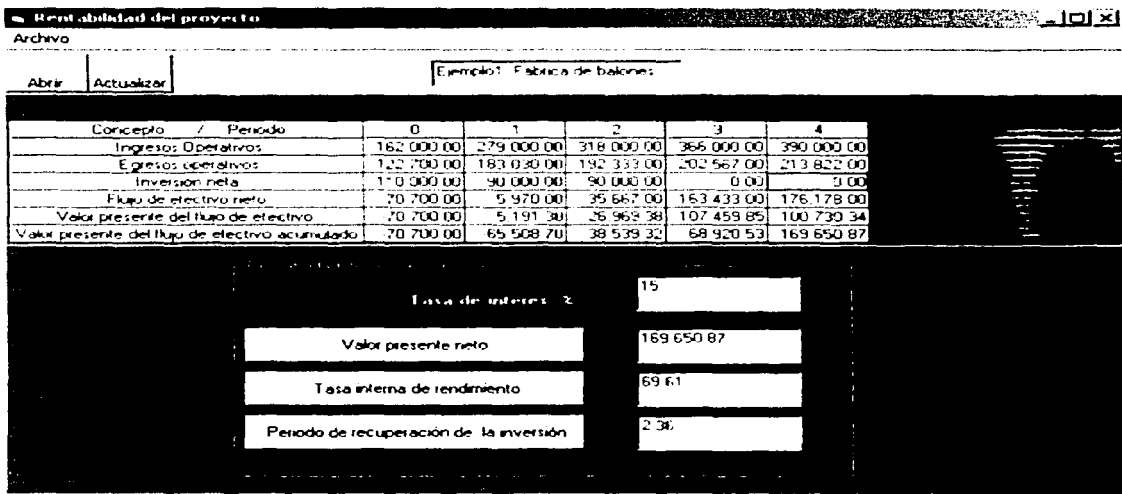


Figura 7.9 Rentabilidad del ejemplo 2.

Como se observa en la figura 7.6. El incremento en el pago de la maquinaria disminuye en un 5% el valor presente neto del proyecto lo que no afecta la aceptación del proyecto.

Ejemplo3. Con el propósito de recuperar parte de la utilidad perdida por el pago a crédito de la maquinaria. La directiva de la fábrica planea vender la maquinaria vieja en 50,000 (que es su valor en libros) en el año 2 de operación. ¿Cómo mejora esto la rentabilidad del proyecto?

Solución: Partiendo de los datos del ejemplo2, debemos crear un nuevo proyecto en el cual cambaremos lo referente a inversión neta, en donde agregaremos un nuevo concepto: "Venta de maquinaria usada".

TESIS CON
FALLA DE ORIGEN

Inversión neta del proyecto					
Archivo Ayuda					
Iniciar	Abrir	Guardar	Ejemplo3 Fábrica de balones		
Concepto de Inversión Neta / Periodo	0	1	2	3	4
Compra de maquinaria	90,000 00	90,000 00	90,000 00	0 00	0 00
Empaque y transporte	12,000 00	0 00	0 00	0 00	0 00
Seguro de traslado	3,000 00	0 00	0 00	0 00	0 00
Instalación	4,000 00	0 00	0 00	0 00	0 00
Puesta en operación	1,000 00	0 00	0 00	0 00	0 00
Venta de maquinaria usada	0 00	0 00	-50,000 00	0 00	0 00
Suma de Inversión Neta	110,000 00	90,000 00	-50,000 00	0 00	0 00

Figura 7 10 Inversión neta del Ejemplo 3

Flujo de efectivo neto del proyecto					
Archivo Modulo Ayuda					
Abrir	Actualizar	Ejemplo3 Fábrica de balones			
Ventas balones tipo A	60,000 00	144,000 00	168,000 00	216,000 00	240,000 00
Ventas balones tipoB	102,000 00	135,000 00	150,000 00	150,000 00	150,000 00
Suma de Ingresos	162,000 00	279,000 00	318,000 00	366,000 00	390,000 00
Sueldos y salarios de obreros	13,500 00	27,330 00	30,030 00	33,033 00	36,336 00
Materias primas	30,000 00	59,850 00	65,835 00	72,419 00	79,660 00
Servicios	4,200 00	5,880 00	6,468 00	7,115 00	7,826 00
Sueldos y salarios personal administrativo	60,000 00	72,000 00	72,000 00	72,000 00	72,000 00
Otros materiales	5,000 00	6,000 00	6,000 00	6,000 00	6,000 00
Otros servicios	10,000 00	12,000 00	12,000 00	12,000 00	12,000 00
Suma de Egresos	122,700 00	183,030 00	192,333 00	202,567 00	213,822 00
Compra de maquinaria	90,000 00	90,000 00	90,000 00	0 00	0 00
Empaque y transporte	12,000 00	0 00	0 00	0 00	0 00
Seguro de traslado	3,000 00	0 00	0 00	0 00	0 00
Instalación	4,000 00	0 00	0 00	0 00	0 00
Puesta en operación	1,000 00	0 00	0 00	0 00	0 00
Venta de maquinaria usada	0 00	0 00	-50,000 00	0 00	0 00
Suma de Inversión Neta	110,000 00	90,000 00	-50,000 00	0 00	0 00

Figura 7 11 Flujo de efectivo del Ejemplo 3

TESIS CON
FALLA DE ORIGEN

Figura 7 12 Rentabilidad del ejemplo 3.

Como podemos apreciar en la figura 7.11, la rentabilidad del proyecto mejora sustancialmente con la venta de los activos.

Ejemplo4. Autopista Cardel – Veracruz. A los accionistas de la autopista Cardel – Veracruz se les ha propuesto ampliar la concesión de su actual carretera con la condición de construir un tramo adicional de carretera con una inversión de poco más de 179 millones de pesos en la que al 40% de inversión la aportará el Gobierno del Estado. Los accionistas estiman que el nuevo proyecto les permitiría

Rentabilidad del proyecto						
Archivo						
Abre	Actualizar	Ejemplo3: Fabrica de balones				
Concepto / Periodo	0	1	2	3	4	
Ingresos Operativos	162 000 00	278 000 00	318 000 00	355 000 00	390 000 00	
Egresos operativos	122 700 00	183 000 00	192 333 00	202 567 00	213 822 00	
Inversión neta	110 000 00	90 000 00	50 000 00	0 00	0 00	
Flujo de efectivo neto	70 700 00	5 970 00	175 667 00	163 433 00	176 178 00	
Valor presente del flujo de efectivo	70 700 00	5 191 30	132 829 49	107 459 95	100 730 34	
Valor presente del flujo de efectivo acumulado	-70 700 00	-65 508 70	67 320 79	174 760 64	275 510 98	

Tasa de interés, %	15
Valor presente neto	275 510 98
Tasa interna de rendimiento	108 41
Periodo de recuperación de la inversión	1 42

pagar 23 55 millones de pesos de la primera etapa que no fue posible recuperar, debido al alto costo del dinero en el periodo 1995 al 2000. Si el costo de capital es de 7 5%. ¿deben aceptar el proyecto?

Solución: Mediante el uso de la aplicación podemos observar que el valor presente neto del proyecto es negativo y que la tasa interna de retorno es menor al costo de capital por lo que los inversionistas deben financiar el proyecto a una tasa menor al 6% o rechazar el proyecto. (Los importes de las pantallas están en millones de pesos)

Inversión neta del proyecto											
Archivo Ayuda											
Iniciar	Abrir	Guardar	Autopista Cardel Veracruz Etapa								
Concepto de Inversión Neta / Periodo	0	1	2	3	4	5	6	7	8	9	10
Aportación de capital ACCSA	107.956 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00
Aportación de capital Gobierno del Estado	71.970 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00
Suma de Inversión Neta	179.926 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00	0 00

Figura 7.13 Inversión de la Carretera Cardel – Veracruz

Ingresos operativos del proyecto						
Archivo Ayuda						
Iniciar	Abrir	Guardar	Cardel Veracruz 2.txt			
Concepto de Ingresos	Periodo	0	1	2	3	4
Carretera La Cruzera		0 00	97.207 00	109.296 00	117.621 00	129.383 00
Carretera Cardel		0 00	7.687 00	8.456 00	9.301 00	10.231 00
Suma de Ingresos						

Ingresos operativos del proyecto						
Archivo Ayuda						
Iniciar	Abrir	Guardar	Cardel Veracruz 2.txt			
Concepto de Ingresos	Periodo	0	1	2	3	4
Carretera La Cruzera		156.553 00	172.208 00	189.429 00	208.372 00	229.209 00
Carretera San Julian		12.379 00	13.617 00	14.979 00	16.447 00	18.125 00
Suma de Ingresos						

Figura 7.14 Ingresos de la carretera Cardel – Veracruz.

TESIS CON
FALLA DE ORIGEN

Egresos operativos del proyecto

Archivo Ayuda

Iniciar Abrir Guardar Autopista Cardel Veracruz Etapa 2

Concepto de Egresos	Periodo	1996	1997	1998	1999	2000
Costo de operación de la carretera	0.00	129,880.00	14,288.00	15,715.00	17,287.00	19,015.00
Costo de operación de la empresa	0.00	3,037.00	3,365.00	3,701.00	4,071.00	4,479.00
Pago de participación al Gobi. del Estado	0.00	41,958.00	47,102.00	50,797.00	55,846.00	61,430.00
Contribuciones SCT	0.00	1,049.00	1,154.00	1,269.00	1,398.00	1,538.00
Costo de mantenimiento de la carretera	0.00	16,392.00	18,031.00	19,834.00	21,817.00	23,999.00
Impuestos	0.00	0.00	2,204.00	2,291.00	3,523.00	2,757.00
Pagos de la pensión etapa	0.00	11,775.00	11,775.00	0.00	0.00	0.00
Suma de Egresos	0.00	204,091.00	257,313.00	299,607.00	362,947.00	416,716.00

Egresos operativos del proyecto

Archivo Ayuda

Iniciar Abrir Guardar Autopista Cardel Veracruz Etapa 2

Concepto de Egresos	Periodo	1996	1997	1998	1999	2000
Costo de operación de la carretera	20,917.00	23,008.00	26,308.00	27,840.00	30,624.00	
Costo de operación de la empresa	4,929.00	5,419.00	5,961.00	6,557.00	7,213.00	
Pago de participación al Gobi. del Estado	67,573.00	74,330.00	81,763.00	89,928.00	98,934.00	
Contribuciones SCT	1,589.00	1,858.00	2,044.00	2,248.00	2,473.00	
Costo de mantenimiento de la carretera	26,491.00	29,009.00	31,943.00	35,137.00	37,408.00	
Impuestos	3,253.00	3,362.00	3,595.00	4,062.00	4,595.00	
Pagos de la pensión etapa	0.00	0.00	0.00	0.00	0.00	
Suma de Egresos	124,560.00	146,586.00	151,614.00	165,822.00	173,747.00	

Figura 7.15 Egresos de la carretera Cardel – Veracruz

ANÁLISIS DEL
FALLA DE ORIGEN

Flujo de efectivo neto del proyecto						
Archivo Modulo Ayuda						
Abrir Actualizar		Autopista Cardel Veracruz Etapa				
	0.00	97,207.00	109,298.00	117,621.00	129,383.00	142,321.00
	0.00	7,687.00	8,456.00	9,301.00	10,231.00	11,254.00
Suma de Ingresos	0.00	104,894.00	117,754.00	126,922.00	139,614.00	153,575.00
	0.00	129,880.00	14,288.00	15,715.00	17,287.00	19,015.00
	0.00	3,037.00	3,865.00	3,701.00	4,071.00	4,479.00
	0.00	41,958.00	47,102.00	50,797.00	55,846.00	61,430.00
	0.00	1,049.00	1,154.00	1,269.00	1,398.00	1,538.00
	0.00	16,392.00	18,031.00	19,834.00	21,817.00	23,999.00
	0.00	0.00	2,204.00	2,291.00	2,523.00	2,757.00
	0.00	11,775.00	11,775.00	0.00	0.00	0.00
Suma de Egresos	0.00	204,091.00	97,919.00	93,607.00	102,942.00	113,218.00
	107,956.00	0.00	0.00	0.00	0.00	0.00
	71,970.00	0.00	0.00	0.00	0.00	0.00
Suma de Inversión Neta	179,926.00	0.00	0.00	0.00	0.00	0.00

Flujo de efectivo neto del proyecto					
Archivo Modulo Ayuda					
Abrir Actualizar		Autopista Cardel Veracruz Etapa			
	156,553.00	172,208.00	189,429.00	208,372.00	229,229.00
	12,373.00	13,617.00	14,979.00	16,447.00	18,125.00
Suma de Ingresos	168,932.00	185,825.00	204,408.00	224,819.00	247,334.00
	23,917.00	23,008.00	26,308.00	27,840.00	30,624.00
	4,929.00	5,419.00	5,961.00	6,557.00	7,213.00
	67,673.00	74,330.00	81,763.00	89,326.00	98,934.00
	1,689.00	1,856.00	2,044.00	2,248.00	2,473.00
	26,399.00	29,005.00	31,943.00	35,137.00	37,409.00
	3,053.00	3,362.00	3,595.00	4,052.00	4,595.00
	0.00	0.00	0.00	0.00	0.00
Suma de Egresos	124,560.00	136,986.00	151,614.00	165,772.00	181,247.00
	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
Suma de Inversión Neta	0.00	0.00	0.00	0.00	0.00

Figura 7.16 Flujo de efectivo de la carretera Cardel - Veracruz

TESIS CON FALLA DE ORIGEN

Rentabilidad del proyecto							
Archivo							
Abre	Actualizar	Autopista Cardel Veracruz Etapa					
Concepto / Periodo	0	1	2	3	4	5	6
Ingresos Operativos	0.00	104 894.00	117 754.00	126 922.00	139 614.00	153 675.00	169 932.00
Egresos operativos	0.00	204 091.00	97 919.00	93 607.00	102 942.00	113 218.00	124 560.00
Inversión neta	179 926.00	0.00	0.00	0.00	0.00	0.00	0.00
Flujo de efectivo neto	179 926.00	99 197.00	19 835.00	33 315.00	36 672.00	40 457.00	44 372.00
Valor presente del flujo de efectivo	179 926.00	92 276.28	17 163.87	26 817.26	27 460.01	28 111.02	28 751.36
Valor presente del flujo de efectivo acumulado	179 926.00	272 202.28	255 038.41	229 221.15	200 761.13	172 650.12	143 898.77

Rentabilidad del proyecto							
Archivo							
Abre	Actualizar	Autopista Cardel Veracruz Etapa					
Concepto / Periodo	4	5	6	7	8	9	10
Ingresos Operativos	139 614.00	153 675.00	169 932.00	185 875.00	204 409.00	224 019.00	247 334.00
Egresos operativos	102 942.00	113 218.00	124 560.00	138 989.00	151 614.00	165 772.00	181 247.00
Inversión neta	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Flujo de efectivo neto	36 672.00	40 457.00	44 372.00	48 039.00	52 794.00	59 047.00	66 087.00
Valor presente del flujo de efectivo	27 460.01	28 111.02	28 751.36	29 437.96	29 601.71	30 797.94	32 065.01
Valor presente del flujo de efectivo acumulado	200 761.13	172 650.12	143 899.77	114 460.82	84 859.11	54 061.17	21 996.16

Tasa de interés %	7.5
Valor presente neto	21 996.16
Tasa interna de rendimiento	8.00
Periodo de recuperación de la inversión	no se recupera

Figura 7.17 Rentabilidad de la carretera Cardel – Veracruz.

TESIS CON
FALLA DE ORIGEN

Capítulo 8. Conclusiones.

El programa funciona, en lo básico, adecuadamente. El cálculo de los parámetros de rentabilidad es correcto. La presentación de pantallas y el uso del programa es ágil. No obstante la aplicación requiere de posterior desarrollo.

Es necesario, primeramente, depurar el código, así mismo se requiere escribir los módulos para la salida de reporte por medio de la impresora y un procedimiento para agregar conceptos o periodos en cualquier módulo sin necesidad de regresar al módulo de inicio. Como puede verse en los listados de código del programa se incluye el título de un procedimiento *Private Sub MenuEditar_Click* que se encuentra vacío. Ello es un recordatorio de que este procedimiento requiere desarrollarse y por lo mismo no hay respuesta a la llamada al menú **Editar**, en cualquiera de los módulos.

Para dar a la aplicación un desempeño equiparable con el de programas comerciales es necesario agregar una mayor funcionalidad en lo relacionado con la edición de los textos en las tablas de datos (cortar, copiar, pegar). Debe mencionarse que esto, que parece muy simple, no lo es en lo absoluto. A medida que aumenta la funcionalidad de un programa, crece la cantidad de código asociado con él. Operaciones aparentemente sencillas, como introducir adecuadamente datos en la FlexGrid (tabla de datos), requirieron de gran cantidad de código. Esta tarea y la de guardar y recuperar adecuadamente los datos de los archivos de proyecto, representaron las actividades más difíciles de programación.

La aplicación guarda los datos de proyecto en archivos de acceso aleatorio. De hecho, para cada proyecto el programa crea una única matriz de datos que contiene los textos y cifras, excluidos los encabezados. Por ello, cuando se requiere grabar o recuperar datos de una tabla de proyecto es necesario indicar al programa la posición específica en que dicho dato o datos serán grabados y recuperados. Esta operación se ejecuta en el procedimiento *Public Sub IniFin()* que es llamado constantemente a lo largo del programa.

Los resultados obtenidos son alentadores. Podemos decir que el desarrollo de la aplicación, tiene un avance de 60%. Lo que constituye una buena base para el desarrollo de una aplicación comercial que es conveniente seguir desarrollando.

Anexo 1. Glosario de términos financieros.

Activo: Conjunto de bienes y derechos pertenecientes a una persona jurídica o física. Dentro del balance de una empresa identifica los bienes y derechos de que es titular.

Activo circulante: Aquellos activos y derechos que no permanecen en la empresa más de un ejercicio económico. Normalmente son: Caja y similares, deudores, existencias y ajustes por periodicidad.

Activo fijo: Aquellos bienes que permanecen en la empresa más de un ciclo operativo, se destinan a la actividad de la empresa y se amortizan durante la vida del bien.

Activos financieros: Son aquellos productos, pagarés, letras, bonos, acciones que el emisor utiliza para financiarse. Por tanto es el inversor quien adquiere estos activos financieros.

Beneficios empresariales: Son las ganancias que obtienen las empresas en su actividad, no todas las empresas obtienen beneficios sino que también las hay con pérdidas. Una parte de los beneficios que obtienen los reparten entre sus accionistas. Los beneficios son el apoyo y el sustento de las subidas de las empresas en las bolsas.

Costo de capital: Tasa de interés que se utiliza para "descontar" los flujos de efectivo netos del negocio para encontrar su valor presente neto

Costo de oportunidad: Flujo de efectivo que se deja de recibir como consecuencia de la aceptación del negocio o proyecto.

Depreciación: es la pérdida de valor de un bien expresado en términos monetarios.

Entidades financieras: Son aquellas encargadas de facilitar financiación a quien lo necesite, sean particulares o sociedades.

Flujo de efectivo: Cantidad de dinero que entra menos la del dinero que sale del negocio durante su vida económica.

Flujos de Efectivo Operativos: Son las entradas de dinero menos las salidas de dinero que un negocio o proyecto genera por sus operaciones, durante su vida económica.

Impuestos: Son el principal instrumento que tiene el Estado para recaudar dinero y así atender sus gastos y realizar sus inversiones. Estos impuestos pueden ser directos o indirectos

Inflación: Es la subida o bajada de los precios de los productos o servicios. Tiene una importancia vital para el inversor pues lo primero que este debe de conseguir es vencer la inflación para que de este modo su dinero no se deprecie. La inflación afecta a los tipos de interés y a la marcha de las monedas respecto de otras

Interés: Es la renta que se da al que nos ha prestado el dinero, además de devolverle el dinero hay que satisfacer una cantidad por habérselo prestado.

Inversión Neta: Cantidad de dinero que se requiere para la adquisición de los activos y bienes necesarios para iniciar y mantener en operación un negocio o proyecto

Periodo de recuperación de la inversión: Tiempo requerido para recuperar el dinero invertido inicialmente en un negocio o proyecto

Proyecto: Plan de acción para la utilización productiva de los recursos económicos de que dispone un negocio.

Rendimiento: Es la retribución que percibe el inversor por comprar activos financieros destinados a captar recursos para un emisor que los necesita.

Rentabilidad : Cambio porcentual entre la cantidad de dinero disponible al término de la operación de un proyecto y la cantidad de dinero disponible al inicio del mismo.

Riesgo: Variación en la rentabilidad (rendimiento esperado) de un negocio o proyecto.

Tasa Interna de Rendimiento: Tasa de interés obtenido por la inversión de recursos en un negocio.

Valor de Rescate de la Inversión Neta: Flujo de efectivo que se produce al final de la vida económica del negocio o proyecto, debido a la venta de los activos fijos y a la recuperación del capital de trabajo.

Valor presente neto de un flujo de efectivo: Suma de los valores presentes netos de los flujos de efectivo de cada periodo de la vida económica del proyecto

Vida económica: Periodo de tiempo que se considera relevante para fines de evaluación.

Anexo2. Bibliografía.

1. Programa de desarrollo empresarial, Módulo III, Evaluación de proyectos.
Gerardo Weihmann Illades, Rafael García García, Jaime Villaseñor Zertuche.
Instituto Tecnológico Autónomo de México (ITAM), Nacional Financiera, S.N.C.
México 1992.
2. Método del Valor Presente Neto.
Rojas, J.
Universidad de Las Américas, Puebla, Pue. 2002.
3. Present value models and investment analysis.
Lindon J. Robinson y Peter J. Barry
Michigan State University Pr. Mayo 1, 1998
4. Visual Basic 6. Curso de Programación
Francisco Javier Ceballos.
Editorial Alfaomega, México 1999
5. Visual Basic 6.
Bob Reselman, Richard Peasley y Wayne Prunchniak.
Prentice Hall 1999
6. Visual Basic Tutorial.
Gary Beene
Visual Basic Information Center
<http://www.vbinformation.com>