



24
24

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

USO DE LAS ESTRUCTURAS DE DATOS PARA
DESARROLLAR UN SISTEMA DE INFORMACION
PARA LA EVALUACION DE PROYECTOS

T E S I S

Que para obtener el título de

M A T E M Á T I C O

presenta

JUAN JORGE HERNANDEZ TORRES

Director de Tesis: *Ma. Guadalupe E. Ibarquengoitia G.*



México, D. F.

23 MAY 1989

1989

TESIS CON
FALLA DE ORIGEN

NO ADEUDA LIBROS
EN BIBLIOTECA
CENTRAL



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 está 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.

"USO DE LAS ESTRUCTURAS DE DATOS PARA DESARROLLAR UN SISTEMA DE INFORMACION PARA LA EVALUACION DE PROYECTOS"

I N D I C E

INTRODUCCION

CAPITULO 1

ANALISIS Y MODELACION PARA LA EVALUACION DE PROYECTOS

- 1.1 Importancia de la evaluación de proyectos
- 1.2 Modelación de la evaluación de proyectos
- 1.3 Informes asociados a la evaluación de proyectos

CAPITULO 2

SELECCION DE LOS TIPOS DE DATOS MAS ADECUADOS PARA LA EVALUACION DE PROYECTOS

- 2.1 Arreglos, registros y cadenas
- 2.2 Listas ligadas y su clasificación.
- 2.3 Comparación de arreglos y listas lineales
- 2.4. Cantidad de memoria y recuperación

CAPITULO 3

INSTRUMENTACION DEL SISTEMA DE EVALUACION DE PROYECTOS

- 3.1 Instrumentación del sistema
- 3.2 Descripción de los módulos y diagrama general del sistema
- 3.3 Archivos utilizados por el sistema

CONCLUSIONES

A N E X O S

- A1. Diagramas del sistema
- A2. Formatos de los archivos utilizados
- A3. Tabla de claves usadas
- A4. Fórmulas de cálculos

BIBLIOGRAFIA

I N T R O D U C C I O N

El gran auge que se da en el campo de la computación durante la última década ha permitido diseñar e implementar sistemas con mayor facilidad y velocidad.

Desde los lenguajes de programación de bajo nivel hasta los lenguajes de cuarta generación y demás herramienta disponible, ha sido todo un proceso de evolución en el que se han visto involucrados los diseñadores de sistemas que han tenido que resolver los problemas de acuerdo al momento en el que los abordan y con el equipo entonces disponible y elegir el mejor camino a seguir para solucionarlos. Estas soluciones deben ser lo suficientemente generales para obtener la independencia en máquina o equipo de cómputo (portabilidad), debido a que las instalaciones actuales cuentan con equipos variados.

El presente trabajo está basado en un sistema diseñado e implementado en un tiempo en el que la instalación disponible al autor era una minicomputadora de 128,000 palabras en memoria principal, en la cual estaban instalados FORTRAN IV y Pascal/1000, de ese entonces a la fecha hay mucha diferencia tanto en equipo como en herramienta para implementar soluciones. Un director General bromeaba que al pedir un equipo nuevo de cómputo, para el momento en que éste llegaba a su empresa ya había un nuevo equipo más poderoso que hacía obsoleta su flamante centro de cómputo. Para cuando se desarrollaban los sistemas ya había equipo más poderoso e incluso más barato que el anterior. La principal razón de que esto se diera es que ninguna organización puede parar ó frenar su desarrollo para esperar a ver que va salir de nuevo los años más próximos.

El principal objetivo de este trabajo fué resolver el problema de la evaluación económica de proyectos, para esto se hizo una revisión de las estructuras de datos que se tenían en los lenguajes de programación, se llevó a cabo el análisis y selección de la mejor opción para modelar y resolver el problema .

En el primer capítulo se trata brevemente la importancia de la evaluación de proyectos y la información involucrada. Así como la derivación del modelo para la implementación de la solución.

En el segundo capítulo son considerados los datos que permite manejar un computador via algún lenguaje de programación en el inicio de esta década, desde datos simples hasta estructuras de arreglos y registros conformando archivos. Se presentan las listas ligadas y dentro de este tipo de datos las listas lineales, ya que éstas son utilizadas en la implementación del sistema. Se compara entre las listas lineales y arreglos con el fin de tener un criterio del porqué de la elección de las listas lineales.

Finalmente en el capítulo tercero se muestra la manera en que se instrumentó el sistema partiendo del modelo matricial el cual tiene desventajas por lo que se propone un modelo basado en las listas lineales, que va a conllevar a una solución más eficiente. Se presenta también que estructura tiene el sistema, los módulos y archivos que usa.

En los apéndices se muestran los diagramas del sistema, los formatos de los archivos, las tablas de claves de conceptos y los cálculos necesarios para evaluar económicamente a los proyectos, todo esto con el fin de tener un panorama más completo del problema y de la forma en que se instrumentó su solución.

CAPITULO 1

ANALISIS Y MODELACION PARA LA EVALUACION DE PROYECTOS

CAPITULO 1.

ANALISIS Y MODELACION PARA LA EVALUACION DE PROYECTOS.

1.1 Importancia de la Evaluación de Proyectos.

Una de las actividades mas importantes tanto del ser humano, cómo de las organizaciones, es la toma de decisiones. Para realizar una adecuada toma de decisiones es necesario evaluar una o varias estrategias enfocadas al logro de un objetivo predeterminado.

Dentro de las organizaciones, una decisión muy importante, es la de evaluar la factibilidad de un proyecto, ya que aún cuando dicho proyecto pueda ser deseable para una organización y que la relación costo-beneficio del proyecto lo justifique plenamente, se puede dar la posibilidad de que no sea posible llevarlo a cabo en virtud de otros factores existentes tales cómo:

- Objetivos institucionales de la organización.
- Su nivel de administración.
- Capacidad económica.
- Politicas y procedimientos operativos.
- Estructura organizacional.
- Etc.

Por lo anterior es necesaria la implementación de una técnica de evaluación de proyectos que permita a las organizaciones analizar el conjunto de alternativas que se presenten así como sus consecuencias y tomar la decisión sobre la alternativa mas adecuada a los intereses de la empresa.

El desarrollo de una Evaluación de Proyectos se compone de las siguientes fases :

- I. Levantamiento de Información sobre las diferentes alternativas a evaluar.
- II. Analisis de Información.
- III. Diagnostico de alternativas.

I. Levantamiento de Información sobre las diferentes alternativas a evaluar.

Esta fase tiene por objeto recabar la información necesaria sobre cada una de las alternativas a evaluar. Esta información puede ser como:

- Características de cada alternativa.
- Recursos necesarios (Económicos, Humanos, etc.) para su desarrollo.
- Ventajas y desventajas.

En proyectos que requieren inversión, esta es información que se necesita recabar también :

- Monto de la inversión.
- Costo de la inversión.
- Formas de financiamiento.
- Costo del financiamiento.
- Tiempo de recuperación de la inversión.
- Utilidades que se esperan obtener a valor presente y proyectados al tiempo de la recuperación de la inversión.
- Etc.

II. Análisis de Información.

Esta fase consiste en analizar la información recabada contra características de la empresa, como pueden ser :

- Objetivos institucionales de la empresa.
- Nivel de administración.
- Capacidad económica.
- Políticas y procedimientos.
(Que intervengan en el proyecto).
- Estructura organizacional.
- Utilidad económica que se pretende obtener ó problema que se quiere resolver a través del proyecto comparando Costo contra Beneficio.

Cuando se trata de un proyecto de inversión el análisis deberá tomar en consideración parámetros de evaluación que sitúen sobre aspectos tales como:

- Tipo de proyecto.
 - + Nuevo
 - + Expansión
 - + Combinación de ambos.
- Rentabilidad del proyecto.
- Tiempo de recuperación de la inversión.
- Forma de financiamiento.
(Si incluye ó es con financiamiento ajeno, Costo de financiamiento).

Además que debe usarse la evaluación original llamada evaluación base y a partir de allí modificar uno o varios de los parámetros o variables que la conforman, recalcular la evaluación y obtener así escenarios que apoyan mejor el análisis para llegar a conclusiones más acertadas. Al modificar, repetir el proceso de evaluación y obtener nuevos resultados se tiene la "Evaluación del proyecto con análisis de sensibilidad".

Siendo tan importante realizar el trabajo de la evaluación de proyectos con un máximo de eficiencia y el mínimo esfuerzo y error, así como la oportunidad de resultados, se ve la conveniencia de automatizar la evaluación de proyectos.

Una vez realizado este análisis se estará en condiciones de emitir un diagnóstico con respecto a cada una de las alternativas a seguir, para apoyar la toma de decisiones.

III. Diagnóstico de alternativas.

Esta etapa consiste en la redacción de un informe sobre la evaluación de cada alternativa incluyendo en él :

- Antecedentes del proyecto.
- Objetivo.
- Problema que se va a resolver ó utilidad económica que se pretende lograr.
- Alcances del trabajo realizado.
- Reportes e informes del computador.

-diagnóstico sobre cada alternativa.

-Alternativa que se sugiere como la más adecuada y su justificación.

1.2 Modelación de la evaluación de proyectos

Para modelar cualquier situación ó problema es necesario determinar las variables a considerar, así como su interacción de modo que sea lo más apegado a la realidad. Para el caso de la evaluación económica de proyectos en general se manejan tres tipos de datos, a saber :

- i) Periodos
- ii) Conceptos
- iii) Productos

i) Periodos

Un proyecto de inversión puede durar varios años para llevarse a cabo, a esto se le llama vida del proyecto.

La vida del proyecto puede dividirse en dos etapas:

-PREOPERATIVA

-OPERATIVA

La etapa preoperativa comprende toda la preparación necesaria para la etapa operativa. La preparación puede constar del estudio de los recursos propios y financiamiento para la compra e instalación de maquinaria y equipo, etc.

En la etapa operativa se obtienen los beneficios del proyecto y se debe evaluar en cuanto tiempo se recupera la inversión aportada, como parámetro de evaluación. La vida de los proyectos es finita y se estima en periodos de tiempo años, semestres, meses etc. Se hacen estimaciones ó proyecciones que cubran todo el proyecto en ambas etapas (preoperativa y operativa).

ii) Conceptos

Los conceptos que se manejan al evaluar un proyecto son resultado de estudios de mercado, de costos y rendimiento de maquinaria y equipo etc. por lo que se obtienen conceptos como los siguientes:

- Inversiones
- Gastos y productos financieros

- Depreciaciones y amortizaciones
- Volúmenes de producción
- Costos de producción
- Precios de venta
- Gastos de comercialización
- Gastos administrativos
- Impuestos
- Utilidades generadas

iii) Productos

Los proyectos pueden contemplar el manejo de varios productos, por lo que, algunos de estos conceptos se manejan a nivel de producto como volúmenes y costos de producción, precios de venta ó algunos gastos.

Estos conceptos pueden ser de captura, cálculo, sensibilizados y de paso. El dato sensibilizado es el dato de captura o de cálculo que es modificado para tener varios escenarios de un mismo proyecto; y el de paso es un dato auxiliar para calcular un dato final.

Es necesario disponer de un modelo que permita tener un buen control sobre esta información. El control debe incluir:

- Identificación (Que tipo de dato es)
- Ubicación precisa que permita:
 - + Registrar o almacenar
 - + Recuperar para revisar, modificar o aún eliminar
- Independencia entre datos
- Secuencia para manipular datos de acuerdo al tratamiento que sea necesario, aún para emitir resultados de las evaluaciones.

El planteamiento del modelo se hace en terminos matriciales por tres razones, la primera, conceptualmente es fácil visualizar la solución del problema de evaluación de proyectos.

La segunda, las características de las matrices permiten tener un buen control sobre la información.

Y la tercera, porque las matrices están disponibles en cualquier instalación de cómputo, dada la universalidad de esta herramienta.

El modelo básico para trabajar el problema es mediante una matriz de tres dimensiones (periodos, conceptos y productos), porque este modelo proporciona todo lo necesario para llevar a cabo la evaluación, ya que cubre los requisitos de control mencionados.

Es necesario un diseño que utilice los conceptos de entrada (datos), aplique las fórmulas de cálculo necesarias y permita generar los resultados deseados.

Diseño :

CONCEPTOS ----- "mas" ----- FORMULAS ----- "igual a" ----- RESULTADOS

Este diseño toma como base el concepto matricial, con esto conservamos la referencia a un dato dado, asociándolo a tres aspectos.

- Qué concepto
- Qué producto
- En qué período

Estas referencias surgen de una forma natural de los tres tipos de datos generales que se deben utilizar, con lo anterior se garantiza que la identificación, ubicación e independencia de datos se cumplan.

1.3 Informes asociados a la evaluación de proyectos

Los datos asociados a la evaluación de proyectos están agrupados en reportes de evaluación y estados financieros, los cuales son informes que muestran la situación operativa y financiera para el proyecto a una fecha dada ó por un periodo de tiempo determinado. estos datos que contiene cada reporte son listados a continuación dentro de cada estado financiero.

1) ESTADO DE RESULTADOS

Reporta el neto de los ingresos y egresos de cada período, como resultado de la operación del negocio proyectado para un período de tiempo dado (por ejemplo: un mes ó un año) compuesto por los siguientes rubros:

Total de Ventas
Total de Costos

Utilidad Bruta

Total de Gastos Admón. y Ventas

Gastos Generales
Gastos de Exploración
Depreciación Planta y Equipo
Depreciación de Servicio
Amortización Gastos Financieros
Amortización Gastos Preoperativos Capitalizables
Total de Gastos

Utilidad en Operación

Gastos Financieros
Productos Financieros

Utilidad Antes ISR

Subsidio ISR

Utilidad Gravable

Impuesto Sobre la Renta (ISR)

Reparto de Utilidades (PTU)

Utilidad Neta

2) FLUJO DE EFECTIVO

Muestra el efectivo en caja, bancos e inversiones del proyecto en cada periodo. Con los conceptos siguientes :

Utilidad Neta

Partidas Virtuales:

(que disminuyen la utilidad de resultados pero que no son una salida de efectivo)

Depreciación de Planta y Equipo

Amortización de servicios

Amortización de Gastos Financieros Capitalizables

Amortización de Gastos Preoperativos Capitalizables

Préstamo Capital de Trabajo

Liquidación de Capital de Trabajo

Préstamo Inversión Activo Fijo

Otros Ingresos

Total de Entradas de Efectivo

Inversión en Activo Fijo

Gastos Financieros Capitalizables

Gastos Preoperativos Capitalizables

Inversión en Capital de Trabajo

Pagos de Pasivo a Largo Plazo

Pagos de Pasivo a Corto Plazo

Otros Egresos

Total Salidas de Efectivo

Flujo de Efectivo

3) VARIOS PRODUCTOS.

Con datos de volúmenes de ventas, precios, costos y gastos incurridos para integrar las ventas netas (Ver cálculos del anexo A4.).

Volumen de Ventas Producto _i
Precios de Ventas Producto _i
Gastos de Administración Producto _i
Gastos de Venta Producto _i
Costos de Empaque Producto _i
Costos Variables Producto _i
Costos Fijos

Donde *i* es un entero que representa el número de producto.

4) GASTOS FINANCIEROS.

Cuando se requiere de un financiamiento para el proyecto, es necesario medir el impacto de los intereses y pagos de principal sobre los resultados del proyecto. Este informe consta de :

Intereses sobre saldos insolutos ó globales
Préstamo
Saldo inicial
Saldo final
Pagos de intereses y capital a inicio, mediados o fin de periodo
Amortización anual
Gastos financieros
Gastos financieros Capitalizables
Amortización de Gastos financieros Capitalizables

5) CAPITAL DE TRABAJO

Para operar (i.e. producir, vender etc.), el negocio requiere comprar materia prima, pagar mano de obra etc. por lo que es necesario contar con cierta inversión que es dedicada a este fin, esta inversión es llamada capital de trabajo. Se determinan políticas para calcular el capital de trabajo necesario para la operación y se contempla la recuperación del capital de trabajo.

Este estado incluye los siguientes conceptos:

| | |
|------------------------------|----------------------------------|
| Caja y Bancos | N periodos de costo fijo |
| Cuentas por cobrar | N periodos de ventas |
| Inventario de materia prima | N periodos del costo variable |
| Producto en proceso | N días del costo total |
| Producto en terminado | N días de ventas |
| IVA por pagar | 15 % de ventas |
| Gastos financieros por pagar | N periodos de gastos financieros |
| Cuentas por pagar | N periodos del costo variable |

N es un entero. (normalmente estos periodos se toman con duración de un mes).

6) PARAMETROS DE EVALUACION.

Los parámetros de evaluación muestran información acerca de la rentabilidad, tiempo de recuperación de la inversión, tasa máxima que soporta el proyecto.

El valor Neto Presente trae al presente flujos de efectivo futuros, actualizando por inflación (Factor de actualización). La TIR (Tasa Interna de Retorno) muestra la tasa máxima de interés que soporta el proyecto, el ROI (Return on investment) muestra el rendimiento de la inversión y el PAY OUT TIME el tiempo necesario para recuperar la inversión.

Conceptos que forman los parametros de evaluación :

Valor Neto Presente Flujo Neto Actualizado con Financiamiento

Valor Neto Presente Factor Actualizado sin Financiamiento

Valor Neto Presente con Financiamiento

Valor Neto Presente sin Financiamiento

TIR sin Financiamiento Flujo Neto Actualizado

TIR sin Financiamiento Factor Actualizado

ROI retorno de la inversión

PAY OUT TIME (tiempo en que se recupera la inversión)

En el anexo A4 se presentan las fórmulas por medio de la cuales se pueden calcular todos los datos requeridos en los informes mencionados .

Para mayor información y justificación se puede consultar la bibliografía al final de este trabajo.

CAPITULO 2

SELECCION DE LOS TIPOS DE DATOS MAS ADECUADOS PARA LA EVALUACION DE PROYECTOS

CAPITULO 2. SELECCION DE LOS TIPOS DE DATOS MAS ADECUADOS PARA LA EVALUACION DE PROYECTOS

Para representar datos en un computador, se tienen varias posibilidades desde los tipos mas simples de datos a estructuras mas complejas, evidentemente los tipos mas simples de datos se van uniendo para formar las estructuras de datos. El presente trabajo analiza brevemente los tipos de datos en los lenguajes de programación de alto nivel, cómo ALGOL, COBOL, Pascal etc.

Cómo tipos simples, básicos, tenemos los caracteres, números enteros, reales y booleanos.

Estos se representan en la máquina mediante alguna codificación especial como el ASCII para los caracteres; los enteros se representan mediante el código binario de su valor y en general los reales como mantisa y característica. Los valores booleanos, cuyos únicos valores son verdadero ó falso son representados por los valores 0 y 1 binarios.

Al agrupar estos tipos de datos simples se pueden tener arreglos, registros, cadenas y listas ligadas que forman lo que es una estructura de datos, entendiendo ésta como un colección de variables sobre las cuales se tiene un buen control de acceso, independencia entre ellas y relacionadas de un modo común (relación que les permite formar parte de la estructura de datos).

La celda de memoria es el bloque básico de construcción y puede ser representado por una localidad en cuyo interior aloja los valores de los datos.

Las estructuras de datos son creadas agregando celdas que van quedando relacionadas ó bien, definiendo el número de celdas a usar, así como las características de las celdas en cuanto al tipo de dato, longitud ó rango de valores etc.

El mecanismo más simple de agrupación de datos en una estructura, en la mayoría de los lenguajes de programación, es el arreglo.

2.1 Arreglos, registros y cadenas.

Una estructura estática de datos es aquella en la que el tipo de información que contiene y el número de elementos son predefinidos. Estas características se conservan a lo largo de la ejecución del programa que las usa.

Los arreglos y los registros que se trabajan en memoria principal son ejemplos de estas estructuras.

ARREGLOS

La forma más simple, antigua y común de estructurar los datos son los arreglos.

Es simple porque permiten referirse a un bloque contiguo de datos usando un sólo identificador, seleccionando en forma sencilla a uno de sus elementos. (su principal característica es que los elementos se indexan)

La idea de los arreglos tan usados en programación surge, desde los inicios de los lenguajes de programación, del concepto matemático de un Vector X de elementos x_1, x_2, \dots, x_N y esta idea se puede extender fácilmente para arreglos o matrices de dos ó mas dimensiones.

Un ARREGLO es una colección ordenada de elementos homogéneos llamados celdas, las cuales son mapeadas a un conjunto índice (que puede estar formado por números enteros) de modo que a cada celda corresponde un y sólo un elemento del índice.

Lo común es trabajar con arreglos de valores reales, enteros y ocasionalmente complejos ó de algún otro tipo de datos. El trabajar con arreglos permite organizar una colección de variables primitivas de tipos idénticos en un bloque más grande e identificarlos usando un sólo nombre de variable.

En resumen los arreglos se caracterizan por:

- Homogeneidad de sus elementos o entradas.
- Orden impuesto sobre sus elementos.
- Para referirse a algún elemento en particular se usan índices.
- El tamaño y dimensión se determinan desde antes de la ejecución del bloque del programa que los usa.

- Su almacenamiento en memoria es en forma secuencial y contigua, aún cuando hay instrumentaciones que no están restringidas a esta característica.

Cada compilador o intérprete provee un mecanismo para calcular la localidad en memoria de un elemento en particular especificado por el índice; igual ocurre para los arreglos multidimensionales. Esta localidad se calcula como función de la localidad que corresponde al primer elemento, más un cierto desplazamiento que depende del índice del elemento deseado. A este mecanismo se le conoce como "Polinomio de direccionamiento".

Estas características permiten modelar infinidad de situaciones y problemas, de allí, que sean tan usados en la computación.

Hay varias aplicaciones comunes para los cuales esta estructura es inadecuada:

- Para el caso de un gran volumen de información ordenada en la que se desea insertar un elemento intermedio y se tienen que recorrer el resto de los elementos de donde se inserte en adelante. Esto involucra más tiempo de proceso y se puede volver inoperante.
- Para representar matrices poco densas ó ralas, es decir matrices en las que gran parte de sus elementos ó entradas no son utilizados, con el consiguiente desperdicio de espacio en memoria.

REGISTROS

Hay muchos problemas cuya solución es simplificada agrupando tipos distintos de datos en una sola estructura, de tal modo que las relaciones significantes entre los componentes son más claras y pueden ser utilizadas con más facilidad, estas agrupaciones serán llamadas registros y sus componentes no son homogéneos como en el caso de la definición original de los arreglos en los lenguajes de programación.

Cada registro consiste de una o más celdas ó campos y cada campo contiene un dato, o bien está a su vez dividido en subcampos que contienen datos. Esta subdivisión de campos en subcampos puede presentarse en cualquier nivel, por ejemplo, un campo llamado Fecha puede estar dividido en los subcampos Día, Mes, Año y cuyos tipos de datos son numéricos (Entero positivo).

Los registros permiten organizar elementos de tipos arbitrarios de datos en uno, por ejemplo :

- Números complejos (Compuesto de dos números reales).
- Ejes de coordenadas (Formado por dos ó mas números).
- Proceso de datos con información cómo el nombre, apellidos paterno y materno, fecha de nacimiento, sexo, estado civil.

Los campos pueden ser de tipos y tamaños distintos, generalmente cada campo es identificado por su nombre para efectos de recuperación o modificación del contenido, no por índices cómo en los arreglos.

En algunos lenguajes de programación se puede definir una estructura estática de arreglos cuyas entradas sean registros como en el caso de Pascal y COBOL.

Se puede hablar de :

- Registros Internos (alojados en memoria principal).
- Registros Externos (Almacenados en archivos en algún dispositivo periférico de la máquina).

Una de las formas clásicas de organizar registros es una estructura llamada ARCHIVO, el cual es una colección de celdas de un tipo particular llamadas registros. Este tipo de estructura asume que los registros están almacenados en algún medio externo como son los dispositivos magnéticos como disco, cinta, disco flexible, etc. y que se pueden llevar a memoria principal ya sea registro por registro o en bloques ó todo el archivo (si su tamaño, número y magnitud de registros, lo permite).

Si el registro está almacenado en un medio magnético como parte de un archivo, es diferente del registro que está almacenado en memoria principal (haciendo las funciones de buffer) o del registro que es un elemento de un arreglo almacenado en memoria principal. Esta diferencia radica en :

- Los registros en periféricos tienen carácter de permantes mientras que en memoria existen en ella mientras su configuración lo permita por ej. al apagar la máquina se pierden.

- En memoria es más rápido el acceso de información que desde los periféricos y si estos son tratados como memoria de acceso aleatorio (RAM DISC) tienen características de memoria. Para un acceso secuencial, por ej. en cinta, el problema surge al momento de leer un registro dado, ya que hay necesidad de pasar todos los registros que anteceden al deseado, sin embargo cuando se van a procesar todos los registros en el orden en que están grabados y este tipo de acceso es muy útil.

Hay estructuras de datos que son creadas y manipuladas en la memoria y su manejo es más flexible y rápido pero entonces no se habla de archivos, sino de estructuras de datos dinámicas ya que aumentan ó disminuyen según las necesidades de espacio durante la ejecución del programa que las usa.

Se pueden tener registros variantes en cuanto a la manera de interpretar la información que vaya a ser almacenada generalmente mediante el uso de selectores de campos que dependen del tipo de información a ser utilizada, se "selecciona" la máscara o formato con el que se va a interpretar la información.

Ejemplo de un registro variante:

dependiendo del valor del campo selector S:

S=1 entonces el campo C1 es un valor entero.

S=2 entonces el campo C1 es un texto de cinco posiciones.

Este tipo de registros variantes es una ventaja que tienen algunas versiones de Pascal.

CADENAS

En el proceso de información no sólo se manejan datos numéricos, también se trabaja con caracteres que asociados forman una colección llamada cadena de caracteres. Se accesan por medio de un identificador, este tipo de datos es tan utilizado que se ha

desarrollado herramienta especial para su manejo y hay lenguajes de alto nivel que tienen instrucciones poderosas para el manejo de cadenas.

2.2 Listas ligadas y su clasificación

La necesidad de contar con una estructura de datos más flexible que los arreglos de tamaño fijo para representar datos es frecuente ya que es posible que los datos sean leídos desde un dispositivo de entrada y la cantidad total de datos no sea conocida o en el caso en que los datos sean generados internamente por un programa de una manera impredecible.

Para permitir la representación natural y la manipulación de tales datos, muchos lenguajes de programación permiten estructuras que aumentan (disminuyen), en cuanto al número de sus elementos, durante la ejecución del programa a esto se le llama crecer, (decrecer) dinámicamente.

Este crecimiento dinámico de las estructuras que van en función a las propias necesidades del volumen de datos requerido y a la memoria disponible de la máquina.

El enlace entre registros que conforman la estructura es mediante el uso de un tipo de variable llamado apuntador, que es una liga o referencia de registros, a los que se llamará nodos.

La dirección de un nodo es llamado apuntador de este nodo y es la localidad de memoria donde se ubica su primer palabra por lo que un apuntador no es sino un medio de identificar y manipular un nodo en particular.

El contenido del apuntador puede ser transparente al programador, sin embargo éste puede recorrer nodos enlazados por apuntadores y conocer el contenido de dichos nodos.

A este tipo de estructuras que modifican su tamaño durante la ejecución del programa que les usa, se les llama estructuras dinámicas.

La manipulación de apuntadores permite asignar el contenido de un apuntador P a otro apuntador Q.

$Q \Leftarrow P$ (Q apunta al nodo donde apunta P)

Es conveniente que el contenido del nodo pueda ser asignado a otro nodo con el uso de los apuntadores.

$Q^{\wedge} \Leftarrow P$ (El nodo al que apunta Q tendrá el mismo contenido que el nodo al que apunta P).

Para crear y explotar las estructuras dinámicas existen técnicas y lo más común es pensar en nodos que son homogéneos entre sí, para evitar dificultades adicionales ya que la creación de cada nodo involucra la generación de su descriptor respectivo (entendiendo por descriptor aquella área de memoria que tiene las características del nodo y la ubicación en memoria de cada uno de sus campos) y sus ligas.

Las estructuras dinámicas reciben diferentes nombres como por ejemplo: Stacks, colas, listas, arboles, redes etc. según las características particulares de creación y uso como se analiza más adelante. Cabe hacer notar que estas estructuras se pueden simular utilizando estructuras estáticas de datos como vectores o arreglos, pero su eficiencia en cuanto a espacio ocupado en memoria no es óptimo.

LISTAS LINEALES

La forma más simple de enlazar un conjunto de nodos es alinearlos, formando una lista y en este caso lo mínimo requerido es de un apuntador por nodo para hacer referencia a su sucesor, lo que da a la lista un carácter secuencial (para llegar al nodo N es necesario recorrer primero N-1 nodos). Matemáticamente una lista es una sucesión de cero o más elementos de un tipo dado, con frecuencia se representa a las listas como una sucesión de elementos $a(1), a(2), \dots, a(N)$ con $N > 0$.

Si $N > 0$ entonces $a(1)$ es el primer elemento de la lista y $a(N)$ es el último, y aplicando las listas a programación se derivan las siguientes propiedades; primero que una lista es finita, y segundo que hay un orden (consecutividad) en los elementos, i.e.

$a(I)$ es precedida por $a(1), \dots, a(I-1)$ $(1 \leq I \leq N)$

Una propiedad adicional a considerar estriba en que el número de elementos de la lista aumentará de acuerdo a las necesidades del usuario de estas, en este caso se habla de un crecimiento (decrecimiento) dinámico, y es conveniente postular que después de $a(N)$ hay un elemento final que sólo indica que la lista ya no tiene más elementos, que puede ser llamado FIN.

Por lo que si dada $a(I)$ el elemento siguiente es FIN entonces $a(I) = a(N)$ se trata del último nodo.

Sean L una lista de N elementos, p una posición de la lista (p puede no estar en L) y x un nodo a operar.

Las operaciones más representativas en las listas son :

Inserta(x,p,L).

Inserta el nodo x en la posición p en la lista L

leyendo la lista desde el inicio hasta el elemento que tiene la posición p

i.e.

Si L es $a(1), a(2), \dots, a(N)$.

Se transforma en :

$L = a(1), a(2), \dots, a(p-1), x, a(p+1), \dots, a(N)$.

(x queda insertado entre $a(p-1)$ y $a(p+1)$, ocupando la posición p)

Si $p+1$ es el final de la lista (FIN); $p = a(N)$

$L = a(1), a(2), \dots, a(p-1), a(p)$

L Se transforma en :

$L = a(1), a(2), \dots, a(p-1), x$

(x queda insertado al final de la lista ocupando la posición p y se recorren las posiciones de la lista original de modo que $a(p)$ se transforma en $a(p-1)$)

Si $p > N$, p no existe en la lista por lo que x no se puede insertar en L .

Localiza(x, L).

Esta función regresa la posición de x en L

Si x aparece más de una vez entonces el resultado es la primera ocurrencia de x

Si no existe x entonces $FIN(L)$ es el resultado.

Recupera(p, L).

Esta función regresa el elemento de la posición p en la lista L .

No hay resultado (indefinido) si $p=FIN(L)$ ó si L no tiene la posición p .

Elimina(p,L).

Borra el nodo de la posición p

Si L es $a(1), \dots, a(N)$

Se transforma en $L = a(1), a(2), \dots, a(p-1), a(p+1), \dots, a(N)$

Si p es el final de la lista (FIN) o L no tiene posición en L el resultado es indefinido.

Siguiente nodo(p,L) y nodo previo (p,L)

Regresa la posición siguiente (anterior) al nodo cuya posición es p en L.

Si $p = \text{FIN}(L)$ el nodo siguiente es indefinido

Si $p = a(1)$ el nodo anterior es indefinido.

Ambas operaciones no definidas si L no tiene posición p.

Convierte en nulo(L)

Esta función transforma a L en una lista vacía y regresa como resultado $\text{FIN}(L)$.

Primer(L)

Su resultado es el primer elemento de la lista, si ésta es vacía regresa como valor $\text{FIN}(L)$.

Imprime(L)

Imprime los elementos de la lista en el orden de ocurrencia.

El número máximo de nodos depende del tamaño de cada uno y de la memoria disponible en la máquina.

Clasificación de las listas lineales.

Por la forma de insertar, suprimir y obtener nodos en una lista lineal, podemos tener cualquiera de estas estructuras :

-STACK o PILA es una lista lineal para la que todas las inserciones, lecturas y supresiones son hechos por un sólo extremo de la lista.

-Una COLA es una lista lineal para la cual todas las inserciones son hechas en un extremo de la lista y en el otro extremo se llevan a cabo las supresiones.

-Una COLA DE ENTRADA/SALIDA es una lista lineal para la cual todas las inserciones y supresiones son realizadas en ambos extremos de la lista. Lo que es insertado por un lado tiene que ser suprimido por el otro.

-Una LISTA LIGADA CIRCULAR es una lista lineal pero con la variante de que el enlace del último nodo hace referencia al primer elemento.

-LISTA DOBLEMENTE LIGADA la lista lineal permite el recorrido desde el primer nodo al último, más no en el sentido inverso (Desde el último al primero); sin embargo al dotar de dos enlaces a cada nodo, un enlace que haga referencia al sucesor y el otro al antecesor, se logra que las búsquedas puedan efectuarse en los dos sentidos.

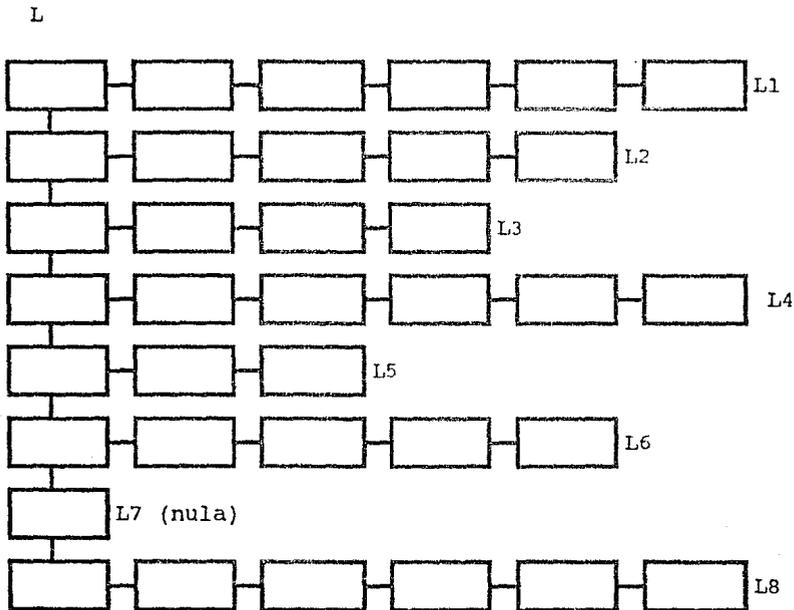
Esta clasificación no es única ya que al momento de determinar o diseñar la estructura se pueden hacer combinaciones que dependen de las necesidades del proceso de la información.

Se puede sofisticar tanto cómo sea necesario en la construcción de las estructuras que usan apuntadores.

Cómo ejemplo de la gran flexibilidad de las estructuras tenemos una lista lineal cuyos elementos son listas lineales, claro esta que esta ya no es una lista lineal sino una estructura más compleja.

También los arboles, redes y gráficas son ejemplos de esta gran flexibilidad.

Para ilustrar gráficamente esto, sean L, L_1, L_2, \dots, L_n listas ligadas donde L es la lista base (sus nodos son el nodo inicial de L_1 a L_n) el dibujo tiene 8 sublistas ($n=8$) y una de ellas es nula (sin elementos).



2.3 Comparación de arreglos y listas lineales

La comparación entre ambas herramientas (arreglos y listas lineales) es necesaria para hablar de las ventajas y desventajas que su uso implica, y es necesario conocerlas para tomar la decisión más adecuada en la programación de sistemas. Para hacer la comparación se van a tomar en cuenta las operaciones más comunes.

INSERCIÓN Y SUPRESIÓN

Son más fáciles de hacer con listas ligadas pues sólo reorganizan apuntadores para insertar, mover y suprimir. En datos estáticos la inserción en información ordenada es difícil pues puede involucrar movimiento de un gran número de nodos o datos al recorrer los elementos, además al suprimir existen espacios sin ocupar mientras que en datos dinámicos se puede reutilizar ese espacio en memoria.

RECUPERACIÓN DE ELEMENTOS

Depende más que nada del tipo de recuperación que se implemente, si es secuencial no hay diferencia entre estático y dinámico y puede ser más rápido el almacenamiento sin apuntadores.

Si la recuperación involucra un sujeto de búsqueda (Llave) de una forma aleatoria se puede crear una lista de llaves con las direcciones de las localidades ó sea un directorio. Por lo que es más eficiente en el almacenamiento dinámico buscando primero en el directorio y luego en el almacenamiento.

COMBINACIÓN Y SEPARACIÓN

Son más fáciles de hacer en listas ligadas ya que sólo se requiere modificar unos cuantos apuntadores, con el almacenamiento estático se requiere una reorganización considerable de espacio moviendo elementos lo que es más difícil y lento.

Usando apuntadores es más fácil unir dos listas, dividir una lista en dos, etc.

2.4 Cantidad de memoria y recuperación

El almacenamiento ligado requiere más espacio, pues cada apuntador lo necesita, sin embargo este espacio extra puede ser más aparente que real por otros factores, cómo el hecho de que un mismo espacio se puede compartir entre varias listas ligadas sin dejar espacios libres entre ellas.

Algunas razones de necesitar menos espacio son :

- Se puede usar el espacio de cada nodo no utilizado.
- La misma información puede ser común a varios nodos y con las ligas se puede compartir, sin tener que duplicar, usando en total menos espacio que en el almacenamiento estático, es decir, que se utiliza correctamente el espacio de que se dispone.
- Algunos algoritmos para datos estáticos pueden ser poco eficientes cuando se reserva mucha más memoria de la que realmente se utiliza. Además habría que dejar varias localidades para tener un nivel de eficiencia razonable.

Al tener los datos en memoria principal se tiene un óptimo de velocidad de proceso, pero es conveniente optimizar el uso de la memoria en razón de las necesidades de proceso.

CAPITULO 3

INSTRUMENTACION DEL SISTEMA DE EVALUACION DE PROYECTOS

CAPITULO 3. INSTRUMENTACION DEL SISTEMA DE EVALUACION DE PROYECTOS

Con el diseño del modelo basado en una matriz de tres dimensiones es posible instrumentar el sistema de evaluación de proyectos, sin embargo, por las consideraciones dadas en el capítulo dos hay algunas deficiencias inherentes descritas a continuación.

3.1 Instrumentación del sistema.

Como el objetivo es implantar una solución automatizada en un computador, el enfoque dado a continuación esta basado en este supuesto.

En primer lugar se va a desperdiciar espacio porque no todos los conceptos se manejan a nivel de producto y no se generan todos los conceptos para cada evaluación, ya que esto depende del tipo de proyecto.

Para ilustrar el desperdicio de espacio sea X una matriz tridimensional cuyas dimensiones son $100 \times 50 \times 20$ (100 conceptos 50 periodos y 20 productos) y que cada celda de X ocupa un espacio S .

Pero el proyecto a evaluar tiene 40 conceptos, 10 periodos y 5 productos.

El espacio total de espacio disponible para X es :

$$100 * 50 * 20 = 100,000 \text{ espacios}$$

El espacio utilizado por el proyecto es:

$$40 * 10 * 5 = 2,000 \text{ espacios}$$

Se estan desperdiciando 98,000 espacios que representan el 98 % del espacio total disponible.

Este desperdicio de espacio afecta la implantación de la solución a la evaluación en una computadora.

Supongamos que tenemos un proyecto con 101 conceptos, 50 periodos y 20 productos. El espacio disponible, de acuerdo a la definición de X , es insuficiente.

Podemos tener una matriz de 200 conceptos de modo que sea suficiente espacio para el proyecto ya este cuenta con 150 conceptos, pero deseamos insertar en la matriz un concepto adicional en la posición cuarta (entonces tenemos que recorrer 147 conceptos) con lo cual esto no es manejable por su gran deficiencia.

La actualización de los conceptos y productos en X involucran el movimiento o corrimiento de la información dentro de la matriz ó de lo contrario se maneja un catálogo rígido de conceptos y productos además de que las operaciones para cálculos involucradas se repiten con toda las celdas que tienen ceros.

Además no todos los lenguajes de programación permiten definir el tamaño de la matriz al momento de ejecución.

Por los inconvenientes que tiene esta herramienta para la modelación, y posterior instrumentación automatizada, Se propone una alternativa que proporcione un buen control sobre la información y que además no tenga los problemas que involucran el uso de una matriz tridimensional.

Para que no se desperdicie espacio en memoria lo que debe ocurrir es que si un dato es cero, lo más conveniente es que no ocupe un lugar en memoria, de esa manera sólo existirá el dato que sea diferente de cero.

Y si deseamos eliminar el problema de actualizar datos que se tiene con las matrices, lo más conveniente es que se rompa la secuencia de los datos, se actualize e inmediatamente se restablezca dicha secuencia.

Por la presentación de los tipos de estructuras de datos del capítulo dos y las conclusiones anteriores, se consideró la alternativa de resolver el problema por medio de listas ligadas simulando una matriz con el siguiente diseño :

a)

| | |
|----------------|-----------|
| REGISTRO DATOS | APUNTADOR |
|----------------|-----------|

Se repite N veces, tantas como sea necesario, con la condición de que están enlazados de modo secuencial por una referencia a cada uno de ellos. teniendo como límite la capacidad de la memoria.

El REGISTRO DATOS está conformado por los siguientes campos :

CLVE_CONCEPT es la clave correspondiente del concepto (si el concepto es cero no se carga y los programas asumen valor cero)

COLUMNA es usado para denotar al número de producto, al número de concepto y para algunos casos es usado como clave auxiliar.

REGLON es el número de periodo y V_DATO corresponde al valor numérico para el concepto (producto) para un periodo dado.

Para una mejor idea de este registro se puede consultar el anexo A2. de FORMATOS DE ARCHIVOS.

Como complemento al diseño, para el control de diferentes productos se tiene:

b)

| | |
|--------------------|-----------|
| REGISTRO PRODUCTOS | APUNTADOR |
|--------------------|-----------|

El REGISTRO DE PRODUCTOS tiene CLVE_CHAR la Clave del concepto CLVE_NUM es el número del producto y NOMB_PROD su nombre. Este registro sirve para generar la tabla de todos los conceptos.

El anexo A3. muestra la tabla de claves de conceptos usados.

Una vez se tiene la definición de variables a utilizar, los cálculos y resultados a obtener así como el diseño de la estructura de datos en que se va a modelar el problema de evaluar los proyectos de inversión, es necesario revisar las características del equipo en que se ha de implantar finalmente el sistema.

Al considerar que equipo y herramientas de programación están disponibles en la instalación, se puede elegir el lenguaje de programación se va a utilizar, que formatos son adecuados para los registros a procesar para una mayor velocidad de transmisión entre periféricos y memoria, de que manera se va a capturar, si en línea o en batch, etc.

Estos elementos en conjunto nos dan un mejor criterio para el diseño e implementación del sistema.

La instrumentación del sistema fue llevada a cabo en el año de 1982 y para desarrollar el sistema se tenían las siguientes opciones:

1. FORTRAN IV
2. SISINF
3. Pascal/1000

En una minicomputadora HP-1000 de 128,000 palabras de memoria principal.

La opción tomada fue usar Pascal/1000 por ser más poderoso que Fortran IV y puesto que SISINF es un lenguaje de programación asociado a un banco de datos para aplicaciones administrativas (fue desarrollado en Monterrey, N.L.) por lo que para las rutinas de cálculo habría problemas para su instrumentación.

En cuanto a la implementación con Pascal/1000, se utilizaron estructuras de datos dinámicas del tipo de listas ligadas de acuerdo al diseño mostrado. las rutinas de creación y manipulación estuvieron definidas para este diseño.

Para manejar :

- i) Periodos (secciones de tiempo)
- ii) Conceptos (nombres y valores en los diferentes periodos)
- iii) Productos (conceptos asociados a los productos)

Eliminando los inconvenientes de desperdicio de espacio y poca flexibilidad de actualización del modelo matricial.

Este diseño se implementó con variables dinámicas y con las operaciones asociadas a las listas. Se usan los conceptos y la forma de asociarlos para calcular los estados financieros y los parámetros de evaluación de acuerdo a la sección 1.3 del capítulo 1.

Para las secciones 3.2 y 3.3 es conveniente apoyarse en los diagramas mostrados en el anexo A1.

3.2 Descripción de módulos y diagrama general del sistema.

Con esta alternativa de selección de la estructura de datos se pasa al diseño general del sistema de evaluación de proyectos. Dicho sistema consta de los siguientes módulos:

- CAPTURA
- CALCULOS
- REPORTES
- SENSIBILIDAD

Y maneja, principalmente, cuatro archivos :

- DATOS
- RESULTADOS (O DE PASO)
- PRODUCTOS
- GASTOS FINANCIEROS

El archivo de DATOS se genera con el módulo de captura y el nombre del archivo es determinado por el usuario al momento de ejecutar la captura para manejar varios productos en varios proyectos, esto permite manejar varios proyectos o varias versiones de un mismo proyecto.

En el Archivo de RESULTADOS, se almacenan los resultados de la evaluación, su nombre es definido por el usuario al momento de ejecutar el módulo de cálculo del sistema para manejar varios proyectos, este módulo usa los archivos de datos y de productos.

El archivo de RESULTADOS es utilizado por el módulo de reportes y de sensibilidad.

El archivo de PRODUCTOS, con claves y nombres de productos del proyecto se genera en el módulo de captura, el usuario determina el nombre de este archivo y se usa para llevar su control.

La estructura de cada módulo es la siguiente :

i. Módulo de captura

Su principal función es generar y actualizar al archivo con datos para los cálculos de evaluación de proyectos (AEPDEF) y generar al archivo con nombres y claves de los productos del proyecto (AEPCLV).

Permite :

- a. Capturar-Modificar los datos del proyecto
- b. Cálculo de cambio de inversión
- c. Cálculo y reporte de gastos financieros

ii. Módulo de cálculos

Toma los datos, para los cálculos, de AEPDEF y AEPCLV. El resultado de esos cálculos los deja almacenados en el archivo AEPASO.

permite calcular:

- a. Varios productos
- b. Estados de resultados
- c. Capital de trabajo
(captura políticas para el capital de trabajo)
- d. Estados de flujo de efectivo

Y se van a calcular los siguientes parámetros de evaluación

- e. Recuperación de la inversión (R O I)
- f. Tasa interna de retorno (T I R)
- g. Valor neto presente (V N P)
- h. Tiempo de recuperación (PAY-OUT TIME)

iii. Módulo de reportes

En este módulo se permite obtener los resultados de los cálculos en la terminal y en papel, por lo que es posible seleccionar los reportes y el archivo de salida (terminal, impresora). Para efectos de comunicación con otros productos permite generar los resultados a un archivo ASCII.

Para obtener los reportes no hay necesidad de efectuar los cálculos de estados financieros y parámetros de evaluación, ya que al calcularlos se guardan en el archivo de datos y cálculos del proyecto llamado AEPASO.

iv. Módulo de sensibilidad

Toma los datos, para los cálculos de sensibilidad, de los archivos AEPASO y AEPCLV. El resultado de esos cálculos se pueden obtener en la terminal y opcionalmente en papel o en un archivo ASCII.

Permite sensibilizar :

(Modificar valores al momento de ejecución del sistema)

- a. Volúmenes de venta
- b. Precios de venta
- c. Costos
- d. Gastos
- e. Inversión

NO MODIFICA LOS ARCHIVOS DE LA EVALUACION BASE

Y para el módulo de sensibilidad se tiene la evaluación base (datos y resultados originales de la evaluación) de esa manera, se puede sensibilizar con todas las opciones deseadas para tener todas las evaluaciones sensibilizadas. Es un archivo en disco y se genera con el módulo de cálculos.

3.3 Archivos utilizados por el sistema

El sistema utiliza los siguientes archivos:

| <u>N o m b r e</u> | <u>D e s c r i p c i ó n</u> |
|--------------------|--|
| INPUT | Es la terminal desde la que se va a ejecutar el sistema, además todos los reportes se pueden desplegar en la terminal y algunos reportes son especiales para terminal. |
| | Archivo utilizado para captura y modificación de datos de proyecto. |
| OUTPUT | Es la impresora para reportes. |
| AEPGFN | Archivo con datos para el cálculo de los gastos financieros. |
| | Se crea por medio del editor de la máquina y se almacena en disco. |

N o m b r e

D e s c r i p c i ó n

AEPDEF

Archivo con datos para el cálculo de la evaluación económica de los proyectos.

Los datos son:

i. Datos generales del proyecto como:

Nombre del proyecto,
duración del proyecto,
periodo del proyecto,
etc.

ii Datos económicos del proyecto como:

Volúmenes de venta, precios de
venta, costos, gastos,
depreciaciones, inversiones,
financiamientos, etc.

Es un archivo en disco y se genera con el módulo de captura.

AEPCLV

Archivo con claves y nombres de los productos del proyecto.

Es un archivo en disco y se genera con el módulo de captura.

AEPASO

Archivo con datos para la evaluación del proyecto y sus resultados.

C O N C L U S I O N E S

"CONCLUSIONES"

El diseño y uso de las estructuras de datos se puede convertir en un aliado muy importante para la solución de problemas o desarrollo de sistemas en campos tan variados como:

- Administración
- Ingeniería
- Medicina
- Ciencia
- Finanzas
- Etc.

Para desarrollar los sistemas de información es importante analizar el problema a resolver y conocer la herramienta (software disponible, su implementación y el equipo) a utilizar y la manera en que se va a emplear, así como sus limitaciones para optimizar los resultados finales.

Es importante hacer notar que este sistema fue diseñado e implementado en el año de 1982, época en la que en México las instalaciones típicas de computadoras no eran tan poderosas, ni de tanta capacidad de proceso ni tan veloces en proceso y acceso a dispositivos como actualmente. La capacidad de almacenamiento en la memoria principal era de un millón de caracteres, (tan sólo la computadora personal en que se elaboró el texto del presente trabajo tiene una capacidad de cuatro millones caracteres en memoria principal "vía ram disc" y con un procesador de 12 megahertz que indica una mayor velocidad de proceso). Los lenguajes utilizados no eran tan poderosos ni tan fáciles de usar como los lenguajes de cuarta generación (lenguajes superpuestos a lenguajes alto nivel) que integran los conceptos de Bases de Datos o las herramientas proporcionadas por las casas que desarrollan software y que van dirigidas al usuario final, como es el caso de las hojas de cálculo.

La computadora en que fué implantado el sistema fue una HP-1000 (Minicomputadora) de 128 Kw de memoria (128,000 palabras) usando el lenguaje de programación Pascal/1000. A pesar de las limitaciones de la época, el diseño conceptual de la solución permanece a pesar de las nuevas herramientas. Cabe aclarar que este sistema fue implantado posteriormente en una HP-3000 usando un paquete llamado IFPS (Interactive Financial and Planning System), el cual está enfocado al usuario final y está basado en un modelo matricial. La implantación del sistema fue hecha en tres meses, mientras que en Pascal/1000 fué trabajo de un año. En una hoja de cálculo el trabajo se puede desarrollar en dos semanas, considerando el análisis original del problema y cambiando la herramienta de implementación.

En el caso de estos paquetes orientados a usuarios finales muy utilizados en México como las hojas de cálculo, la definición de variables y la forma de operarlas, actualmente está totalmente definida, mientras que en Pascal hubo necesidad de crear una herramienta propia, mediante el diseño de rutinas y funciones de creación, asociación de datos leídos en terminal con las estructuras definidas, acceso a los elementos de la lista para: manipulación, actualización, control, reporte, liberación de áreas de memoria ya no utilizadas etc. De haber existido en ese momento el acceso a áreas de periféricos externos en Pascal por medio de apuntadores se podría haber diseñado una base de datos para resolver problemas de un modo más general.

De esto se concluye que la gran ventaja de estas herramientas estriba en cuanto a tiempo de desarrollo, ya que la primer implementación del sistema quedo de tal forma que es muy fácil de usar y eficiente en tiempo de proceso (actualmente se utiliza en Monterrey).

Es interesante hacer notar que los apuntadores sólo hacen referencia a localidades de memoria y está restringido a ella, en lugar de referirse incluso a áreas de periféricos, esto sería una gran ventaja al poder trabajar en discos externos para diseñar prácticamente cualquier base de datos, y con ello un generador de Bases de Datos.

Aunque la tendencia es cada vez más hacia el uso de software especializado (paquetes), para desarrollar con mayor rapidez las aplicaciones, este no puede cubrir el 100% de las necesidades, por lo que en tales casos se puede utilizar herramienta adicional al paquete (o paquetes) como un lenguaje de programación poderoso y estructurado.

El conocimiento de las estructuras de datos y sus propiedades es básico para los profesionales de la computación ya que pueden instrumentar una gran cantidad de sistemas utilizandolas en prácticamente cualquier lenguaje de programación haciendo más poderosas y eficientes dichas instrumentaciones. Además las estructuras de datos son base de herramientas más poderosas y fáciles de usar como los bancos de datos y las bases de datos.

En la importancia de la herramienta se considera también la disponibilidad (casi cualquier instalación puede tener COBOL o Pascal) para efectos de portabilidad, aún cuando la tendencia mundial apunta a sistemas operativos como UNIX como estándares mundiales, esto es un proceso que de llevarse a cabo se lleva varios años en darse (dependiendo del nivel de desarrollo económico de cada país, principalmente).

A N E X O S

A N E X O S

Estos anexos incluyen :

- A1. Diagramas de sistema
- A2. Formatos de los archivos utilizados en el sistema.
- A3. Tabla de claves usadas y explicación breve.
- A4. Fórmulas de cálculos.

A1. DIAGRAMAS DE SISTEMA

El sistema de evaluación de proyectos está conformado por cuatro módulos:

- i. Captura datos y cálculo gastos financieros y cambio de inversión (CAPT_).
- ii. Cálculos de estados financieros y parámetros de evaluación (CALC_).
- iii. Sensibilidad a los proyectos y reportes de sensibilidad (ANASE).
- iv. Reportes de estados financieros y parámetros de evaluación.

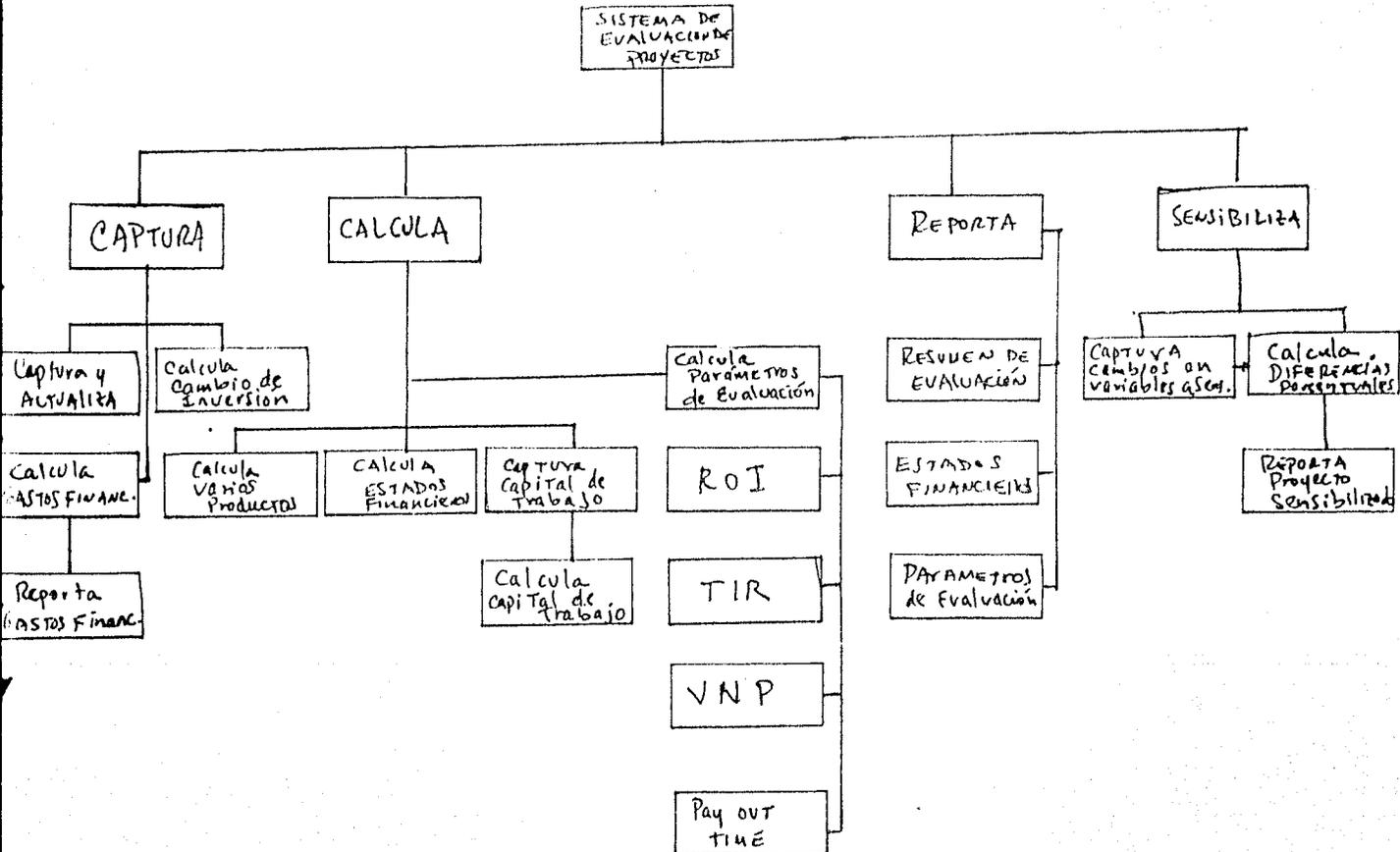
El módulo CAPT_ recibe datos via terminal (terminal ó por medio de un archivo de datos para gastos financieros) y genera dos archivos: AEPDEF (datos del proyecto) y AEPCLV (claves varios productos), opcionalmente, genera el reporte de gastos financieros.

El módulo CALC_ toma los datos de AEPDEF y AEPCLV para sus cálculos y genera a AEPASO (proyecto evaluado).

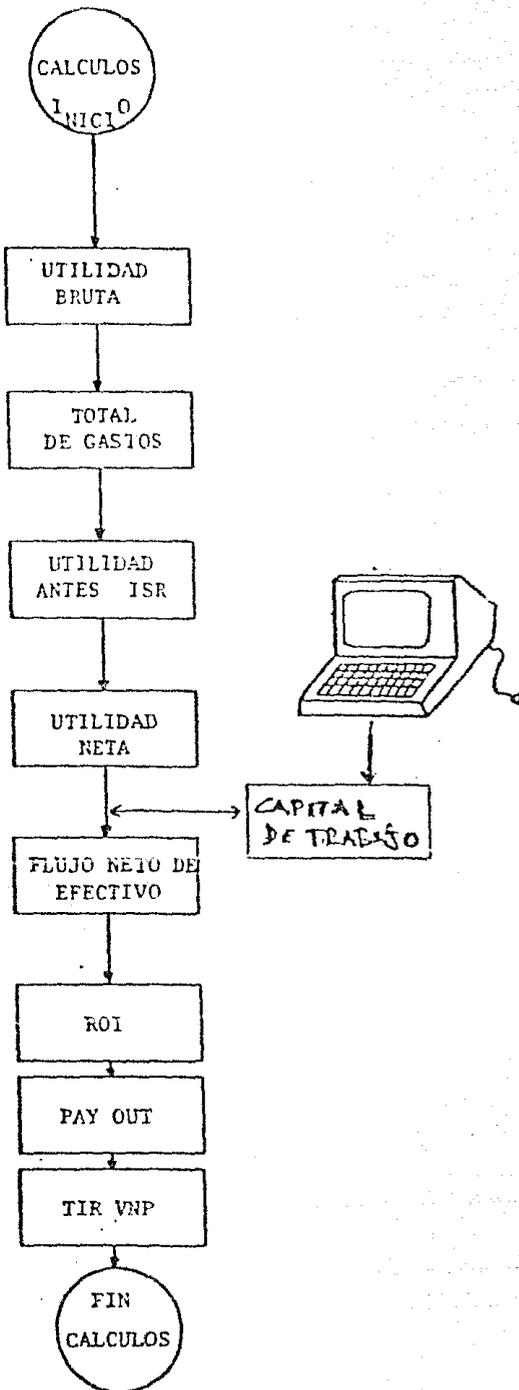
El módulo ANASE toma datos de AEPASO (datos y cálculos del proyecto evaluado) y AEPCLV para dar distintos escenarios de un mismo proyecto, y opcionalmente se sacan los reportes a papel o a terminal.

Y el módulo REP_ se usa para reportar a papel ó a terminal los resultados de la evaluación.

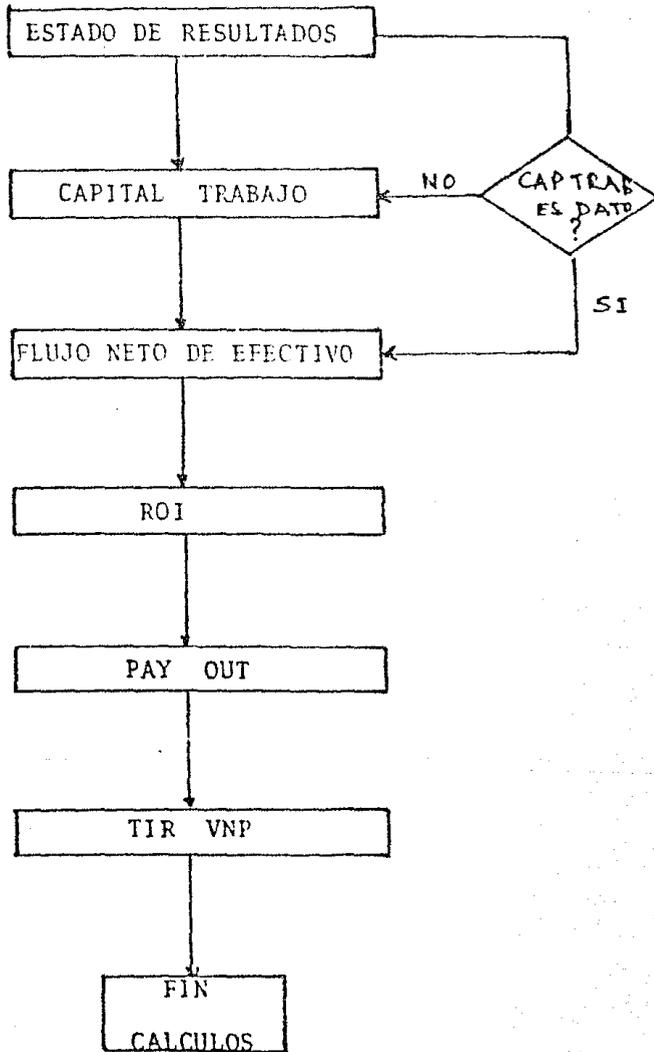
DIAGRAMA DE MODULOS



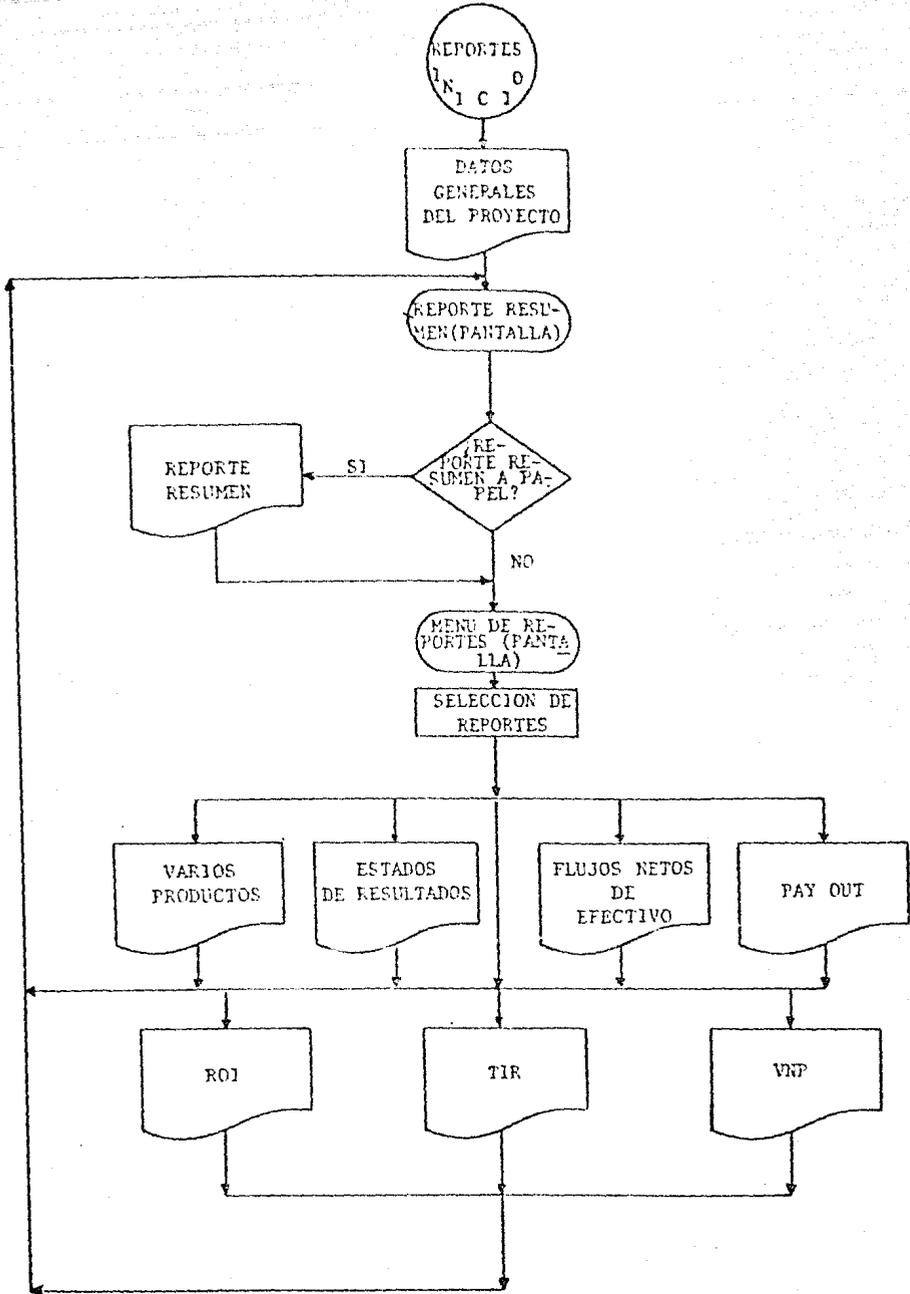
CALCULOS



CAPITAL DE TRABAJO



REPORTES



SISTEMA DE EVALUACION DE PROYECTOS

A2. FORMATOS DE ARCHIVOS

ARCHIVO: AEPDEF Datos de estados de resultados y flujo de efectivo.

 AEPASO Archivo de paso con datos del proyecto evaluado.

AEPDEF y AEPASO tienen el mismo formato.

1er. registro: Datos generales.

| NOMBRE CAMPO | TIPO | POSICION | COMENTARIOS |
|--------------|------|----------|--|
| ESP | X:1 | 1-1 | CONTROL LECTURA |
| NOMB_PROY | X:27 | 2-28 | NOMBRE PROYECTO |
| ESP | X:1 | 29-29 | ESPACIO |
| NUM_ANOS_P | I:2 | 30-31 | NUMERO AÑOS PROY |
| ESP | X:1 | 32-32 | ESPACIO |
| UNID_MON | X:1 | 33-33 | UNIDAD MONETARIA: '1' = MILES DE PESOS '2' = MILES DE DLS. |
| ESP | X:1 | 34-34 | ESPACIO |
| ANO_INIC_P | I:4 | 35-38 | AÑO INICIAL DEL PROYECTO |
| ESP | X:1 | 39-39 | ESPACIO |
| ANO_REL_P | I:3 | 40-43 | AÑO RELATIVO PROY |
| NOMENCLATURA | | X:10 | SON 10 CARACTERES ALFABETICOS. |
| | | I:3 | ES UN ENTERO DE TRES POSICIONES |

2o. registro: Datos para cálculo.

| <u>NOMBRE CAMPO</u> | <u>TIPO</u> | <u>POSICION</u> | <u>COMENTARIOS</u> |
|---------------------|-------------|-----------------|--|
| CLVE_CONCPT | X:5 | 2-6 | CLAVE DEL CONCEPTO |
| ESP | X:1 | 7-7 | ESPACIO |
| RENGLON | I:2 | 8-9 | RENGLON DE LA MATRIZ SIMULADA CON ESTRUCTURAS DE DATOS |

(EL RESTO SE REPITE SIETE VECES POR CADA REGISTRO TIPO 2)

| | | | |
|---------|--------|-------|---|
| ESP | X:1 | 10-10 | ESPACIO |
| COLUMNA | I:2 | 11-12 | COLUMNA CORRESPONDIENTE AL RENGLON DE LA MATRIZ SIMULADA |
| ESP | I:2 | 13-14 | ESPACIO |
| V_DATO | I:13:3 | 14-26 | CANTIDAD DEL CONCEPTO (DADO PARA EL RENGLO Y LA COLUMNA) |

NOMENCLATURA: I:13:3 ES UN ENTERO DE TRECE
POSICIONES CON TRES DECIMALES.

ARCHIVO AEPCLV Archivo con Nombres de los
conceptos.

| <u>NOMBRE CAMPO</u> | <u>TIPO</u> | <u>POSICION</u> | <u>COMENTARIOS</u> |
|---------------------|-------------|-----------------|--|
| ESP | X:1 | 1-1 | ESPACIO CONTROL LECTURA Y ESCRITURA |
| CLVE_CHAR | X:3 | 2-4 | CLAVE DEL CONCEPTO |
| CLVE_NUM | I:2 | 3-4 | NUMERO DE PRODUCTO |

(LOS DOS CAMPOS ANTERIORES FORMAN LA CLAVE DEL CONCEPTO)
(PARA LOS ARCHIVOS AEPDEF Y AEPASO)

| | | | |
|-----------|------|------|---------------------|
| NOMB_PROD | X:27 | 8-33 | NOMBRE DEL PRODUCTO |
|-----------|------|------|---------------------|

A.3 TABLAS DE CLAVES

ESTADO DE RESULTADOS

| CONCEPTO | CLAVE |
|--|-------|
| TOTAL DE VENTAS | TVT00 |
| TOTAL DE COSTOS | TCT00 |
| UTILIDAD BRUTA | UTB00 |
| TOTAL DE GASTOS ADMON. Y VENTAS | TGAUX |
| GASTOS GENERALES | GGN00 |
| GASTOS DE EXPLORACION | GEX00 |
| DEPRECIACION PLANTA Y EQUIPO | DPE00 |
| DEPRECIACION DE SERVICIO | DSR00 |
| AMORTIZACION GASTOS FINANCIEROS | AGF00 |
| AMORTIZACION GASTOS PREOPERATIVOS CAPIT. | AGP00 |
| TOTAL DE GASTOS | TGT00 |
| UTILIDAD EN OPERACION | UTO00 |
| GASTOS FINANCIEROS | GFN00 |
| PRODUCTOS FINANCIEROS | PFN00 |
| UTILIDAD ANTES ISR | UAI00 |
| SUBSIDIO ISR | SBI00 |
| UTILIDAD GRAVABLE | UGR00 |
| IMPUESTO SOBRE LA RENTA | ISR00 |
| REPARTO DE UTILIDADES | PTU00 |
| UTILIDAD NETA | UNAO0 |

FLUJO NETO DE EFECTIVO

| CONCEPTO | CLAVE |
|---|-------|
| UTILIDAD NETA | UNA00 |
| DEPRECIACION DE PLANTA Y EQUIPO | DPE00 |
| DEPRECIACION DE SERVICIO | DSR00 |
| AMORTIZACION DE GASTOS FINANCIEROS CAPIT. | AGF00 |
| PRESTAMO CAPITAL DE TRABAJO | PCT00 |
| LIQUID. CAPITAL DE TRABAJO | LCT00 |
| PRESTAMO INVERSION ACTIVO FIJO | PIA00 |
| OTROS INGRESOS | OIN00 |
| TOTAL DE ENTRADAS DE EFECTIVO | TEE00 |
| INVERSION EN ACTIVO FIJO | IAF00 |
| OTROS INGRESOS | OIN00 |
| TOTAL DE ENTRADAS DE EFECTIVO | TEE00 |
| INVERSION EN ACTIVO FIJO | IAF00 |
| GASTOS FINANCIEROS CAPITALIZABLES | GFC00 |
| INVERSION EN CAPITAL DE TRABAJO | ICT00 |
| PAGOS DE PASIVO A LARGO PLAZO | PPL00 |
| PAGOD DE PASIVO A CORTO PLAZO | PPC00 |
| OTROS EGRESOS | OEG00 |
| TOTAL SALIDAS DE EFECTIVO | TSE00 |
| FLUJO DE EFECTIVO | FNE00 |

VARIOS PRODUCTOS.

| CONCEPTO | CLAVE |
|-------------------------------------|-------|
| VOLUMEN DE VENTAS PRODUCTO i | VVTi |
| PRECIOS DE VENTAS PRODUCTO i | PVTi |
| COSTOS VARIABLES PRODUCTO i | CVBi |
| GASTOS DE ADMINISTRACION PRODUCTO i | GADi |
| GASTOS DE VENTA PRODUCTO i | GVTi |
| COSTOS DE EMPAQUE PRODUCTO i | CEPi |
| COSTOS FIJOS | CFJ00 |

i ES UN ENTERO ENTRE 0 Y 99

PARAMETROS DE EVALUACION.

| CONCEPTO | CLAVE |
|---|-------|
| TIR SIN FINANCIAMIENTO FLUJO NETO ACTUALIZADO | TRSFL |
| TIR SIN FINANCIAMIENTO FACTOR ACTUALIZADO | TRSFA |
| VALOR NETO PRESENTE FLUJO NETO ACTUAL. C/FINANC | VNCFL |
| VALOR NETO PRESENTE FACTOR ACTUALIZADO S/FINANC | VNCFA |
| VALOR NETO PRESENTE CON FINANCIAMIENTO | VNP00 |
| VALOR NETO PRESENTE SIN FINANCIAMIENTO | VNPSF |

SISTEMA DE EVALUACION DE PROYECTOS

A4. FORMULAS DE CALCULOS

Las relaciones entre los conceptos, para evaluar proyectos, de los principales estados financieros están dadas por las siguientes formulas:

ESTADO DE RESULTADOS

10. Total de ventas = $\frac{\text{Volumen de ventas} * \text{Precio de ventas}}{1000}$

Este calculo se hace para cada producto y la sumatoria es el total de ventas para el periodo.

20. Total de costos = $(\text{Volumen Vtas.} * \text{Costos Var})/1000 + \text{Costos fijos.}$

El volumen de ventas y costos variables es aplicado para cada producto.
Los costos fijos son por periodo

30. Utilidad bruta = Total de ventas - Total de costos.

40. Total de gastos = Gastos de administración
Gastos de venta
Gastos generales
Gastos de exploración

+ Depreciación de planta y equipo
Depreciación de servicios
Amortización de gastos preoperativos capitalizables
Amortización de gastos financieros

NOTA: Donde gastos de administración y gastos de venta se toman por porcentaje o cantidad dependiendo la opción deseada.

En el caso de gastos de administración y/o de gastos de ventas en opción porcentaje, se multiplica por el total de ventas y se divide entre 100.

50. Utilidad de operación = Utilidad bruta - total de gastos

60. Gtos. financieros = Gastos financieros - productos financieros

70. Utilidad antes ISR = Utilidad de operación - gastos financieros

8o. Utilidad gravable:

Es la utilidad antes ISR siempre y cuando sea positivo. En otro caso se irá acumulando la actividad negativa hasta que cambie de signo. Esa utilidad es la que es gravable de ISR y PTU.

9o. El ISR es el 40% de la utilidad gravable

10o. El PTU es el 10% de la utilidad gravable

Para calcular el estado de resultados sin financiamiento no se toman en cuenta los gastos financieros, los productos financieros, ni la amortización de gastos financieros capitalizables.

FLUJO DE EFECTIVO

10. Total de entradas de efectivo = Utilidad neta
Depreciación de planta y equipo
Depreciación de servicios
Préstamo para capital de trabajo
+ Liquidación capital de trabajo
Préstamo activo fijo
Otros ingresos
Amortización de gastos preoperativos capitalizables
Amortización de gastos financieros capit.
20. Total de salidas de efectivo = Inversión en activo
Inversión en capital de trabajo
Pagos de pasivo a largo plazo
Pagos de pasivo a corto plazo
+ Otros egresos
Gastos preoperativos capitalizables
Gastos financieros capitalizables
30. Flujo neto de efectivo = Total de entradas de efectivo
Total de salidas de efectivo

Para el flujo de efectivo sin financiamiento, las siguientes variaciones se hacen cero.

Para las entradas de efectivo. a) Amort. gastos financ. cap.
b) Préstamo capital de trabajo
c) Préstamo Inv. activo fijo.

Para las salidas de efectivo. a) Gtos. Financ. Capit.
b) Pagos a pasivo a largo plazo
c) Pagos de pasivo a corto plazo

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

CAPITAL DE TRABAJO

DATOS

Ventas
Precio
Costo fijo
Costo variable
Gastos financieros

$$\text{Caja y bancos} = \frac{\text{costo fijo}}{12} * \text{politica (mes)}$$

$$\text{Ctas por cobrar} = \frac{\text{Vol. de Vtas anual} * \text{precio}}{12} * \text{politica (mes)}$$

$$\text{Materia prima} = \frac{\text{volumen de Vtas anual} * \text{costo variable}}{12} * \text{politica (mes)}$$

$$\text{Prod. procesado} = \frac{(\text{Vol. de Vtas. anual} * \text{costo variable}) + \text{costo fijo}}{12} * \text{politica (mes)}$$

$$\text{Producto terminado} = \frac{\text{volumen de Vtas. anual} * \text{precio}}{12} * \text{politica (mes)}$$

$$\text{Cuentas por pagar} = \frac{\text{volumen de Vtas. anual} * \text{costo variable}}{12} * \text{politica (mes)}$$

Impuestos = $\frac{\text{volumen de Vtas. anual}}{12} * \text{precio} * \text{política}$

Gas. Fin. = $\frac{\text{gastos financieros del año}}{12} * \text{política}$
(mes)

Capital de trabajo = activo circulante - pasivo circulante

Inversión en capital de trabajo: = $\text{Capt. Trab} - \text{Cap. Trab.}$
 $i \qquad \qquad \qquad 1 - 1$

$i = 1 \dots$ número de años de proyecto

liquidación de cap. trab. en el año N = Inv. en Cap. Trab.

PARAMETROS DE EVALUACION

CALCULO DEL TIEMPO DE RECUPERACION DE LA INVERSION (PAY-OUT TIME)

(Tiempo de recuperación)

Es el tiempo en que los ingresos igualan la inversión total inicial.

Se calcula acumulando los flujos netos de efectivo en un saldo final, hasta que el saldo final sea positivo, tomando el periodo "n" en que sucedió esto y aplicamos la siguiente relación:

$$\text{Tiempo de recuperación} = (n-1) + \frac{\text{Saldo inicial periodo } n}{\text{flujo neto de efectivo periodo } n}$$

Donde $n > 0$

Nota: El tiempo de recuperación se calcula con los flujos netos de efectivo sin financiamiento.

CALCULO DE VNP

(Valor Neto Presente)

-Permite traer el valor del dinero del futuro al presente.
(Actualización de Ingresos)

-Se calcula tomando el factor de actualización y
sustituyendo en la siguiente relación.

$$\text{VNP} = \text{SUMATORIA}(i \text{ HASTA } n) \text{ DE } \frac{\text{flujo neto de Efvo.}^i}{(1 + \frac{\text{fact. act.}}{100})^i}$$

n = último año proyecto
i = inicio proyecto

CALCULO DEL ROI

Retorno sobre la inversión.

inversión total = Activo fijo
Preoperativos
Financieros capitalizables

Para calcularlo, se considera un ROI anual como sigue:

ROI año n = $\frac{\text{Utilidad en operación (período } n\text{)}}{\text{Inversión total en activo fijo}}$

Donde $0 \leq n \leq \text{Núm. periodos del proyecto}$
y el ROI total del proyecto

ROI = $\frac{\text{Promedio utilidad en operación}}{\text{Inversión total activo fijo}}$

También se calcula incluyendo el Capital de trabajo :

ROI = $\frac{\text{Promedio utilidad en operación}}{\text{Inv. Tot. Act. fijo} + \text{Capt. trabajo}}$

El ROI se calcula usando la utilidad de operación e inversión total activo fijo, con y sin financiamiento.

CALCULO DE TIR

(Tasa interna de retorno)

-Indica el rendimiento neto de la inversión

Se calcula con la relación:

$$0 = \text{SUMATORIA}(i \text{ HASTA } n) \text{ DE } (\text{flujo neto de efectivo})^i * (1 + g)^{-i}$$

inicio del proyecto $\leq i \leq$ fin del proyecto (i en periodos)

Donde: $g = \text{TIR}$ i.e. g es la variable cuyo valor hay que encontrar de modo que la sumatoria sea igual a cero.

Buscando que la sumatoria tienda a cero dando valores a " g ".
periodo

B I B L I O G R A F I A

BERTISS, A.T.
DATA STRUCTURES THEORY AND PRACTICE
ACADEMIC PRESS
1971

BWWLES KENETH, L.
MICROCOMPUTER PROBLEM SOLVING USING PASCAL
SPRINGER-VERLAY
1977

GROGONO, PETER
PROGRAMMING IN PASCAL WITH PASCAL /1000
ADDISON-WESLEY
1980

GUIBAS, L.E. MC CREIGHT
A NEW REPRESENTATION FOR LINEAR LISTS
PROC. 6th ACM SYMP THEORY OF COMP N.Y.
1977 PP 49-60

HABBERMAN, A.N.
CRITICAL COMMENTS ON THE PROGRAMMING LANGUAGE PASCAL
ACTA INFORMATICA 3(1)
1973

KNUTH, DONALD E.
FUNDAMENTAL ALGORITHMS THE ART OF COMPUTER PROGRAMMING
ADDISON WESLEY PUBLISHING COMPANY INC.
1978, VOL. I 2A EDICION

PFALTZ L. JOHN
COMPUTER DATA STRUCTURES
Mc GRAW-HILL, INC.
1977

STEENGARD-MARSDEN J.
MORE ON DYNAMICS ARRAYS IN PASCAL
SIGPLAN NOTICES
1976, MAY

TENENBAUM A.M., AUGENSTEIN M.
DATA STRUCTURES USING PASCAL
PRENICE-HALL, INC.
1981

T. YEH, RAYMOND
CURRENT TRENDS IN PROGRAMMING METHODOLOGY VOLUME IV
PRENTICE-HALL, INC.
1978

T.G. LEWIS, M.Z. SMITH
APPLYING DATA STRUCTURES
HOUGHTON MIFFLIN COMPANY, BOSTON
1976

WIRTH, NIKLAUS
ALGORITHMS + DATA STRUCTURES = PROGRAM
PRENTICE-HALL
1976

COSS BU RAUL
ANALISIS Y EVALUACION DE PROYECTOS
DE INVERSION
2a. EDICION
LIMUSA, 1986

CEPAL/AAT
MANUAL DE PROYECTOS DE DESARROLLO
ECONOMICO
ONU, 1958

W. SMITH GERALD
ENGINEERING ECONOMY: ANALYSIS OF CAPITAL
EXPEDITURES
SECOND EDITION
THE IOWA STATE UNIVERSITY PRESS, AMES, IOWA USA

JAMES T.S. PORTERFIELD
INVESTMENT DECISIONS AND CAPITAL COSTS
PRENTICE-HALL, 1965