



**Escuela Superior Politécnica  
del Litoral**

**Facultad de Ingeniería  
en Electricidad**

**“InvestMaster para Windows”**

**Proyecto de Tópico de Grado  
Previa a la obtención del título de  
Ingeniero en Computación**

**Presentado por:**

**Alvarado Marin, Jimmy Abel  
Molina Izquierdo, Charles Giovanni  
Naranjo Sanchez, Alice Bertha  
Yee Guin, Ricardo**

**Guayaquil - Ecuador  
1994**

## *AGRADECIMIENTO:*

Queremos dar un especial agradecimiento al Ing. Sixto García A., Director del Tópico de Graduación por que gracias a este proyecto pudimos comprender que la preparación académica de nuestra área no termina aún y que queda mucho por aprender principalmente de una carrera tan compleja como lo es el área financiera, involucrada en este trabajo, el cual con mucho sacrificio pudimos sacar adelante.

Gracias también a la Compañía Humphrey & Calfivar S.A., en la persona del Dr. Moisés Tacle y el Ing. Juan Alvarado, por facilitarnos la información necesaria para la culminación de este proyecto.

## *DEDICATORIA :*

Con mucho cariño a nuestros padres que sin escatimar sacrificio alguno, nos apoyaron durante toda nuestra carrera, para poder alcanzar nuestras metas.

A nuestros hermanos para que este trabajo constituya un reflejo del modelo de superación que debe existir en todas sus metas planteadas.

*DECLARACION EXPRESA*

"La responsabilidad por los hechos, ideas y doctrinas expuestas en este proyecto, nos corresponde exclusivamente; y, el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de Exámenes y Títulos Profesionales )

\_\_\_\_\_  
Molina Charles

\_\_\_\_\_  
Alvarado Jimmy

\_\_\_\_\_  
Naranjo Alice

\_\_\_\_\_  
Yee Guin Ricardo



*MIEMBROS DEL TRIBUNAL*

---

Armando Altamirano  
Subdecano de la Facultad de  
Ingeniería en Electricidad

---

Ing. Sixto García  
Director del Tópico de Graduación

---

Ing. Guido Caicedo  
Miembro del Tribunal

---

Ing. Mónica Villavicencio  
Miembro del Tribunal

## INDICE

	Pág.
Introducción.....	7
Fundamentos de Lenguaje C++ orientado a objeto bajo Windows.....	9
Descripción de los Módulos del Sistema	
Módulo 1: Riqueza Relativa, Retorno, Niveles de Riqueza.....	14
Riqueza Relativa	
Retorno	
Niveles de Riqueza	
Módulo 2: El valor del dinero.....	18
Flujo de Caja Simple	
Anualidades	
Flujo de Caja Irregular	
Módulo 3: Cálculos Estadísticos.....	21
Módulo 4: Valoración de Bonos.....	24
Análisis de Bonos	
Bonos de Descuento	
Tasa de crecimiento	
Módulo 5: Valoración del Stock.....	28
Módulo 6: Análisis Financiero.....	30
Hoja de Balance	
Estado de Pérdidas y Ganancias	
Indices Financieros	
Módulo 7: Análisis de Cartera.....	35
Módulo 8: Análisis de Regresión.....	37
Módulo 9: Modelo de Costo de Acarreo.....	40
Componentes básicos del sistema.....	42
Manejo de Archivos	
Sistema de Ayuda	
Impresión de Reportes	
Código del Programa Fuente.....	43

# INVESTMASTER

## Introducción

El sistema diseñado se conoce con el nombre de InvestMaster. Es un proyecto financiero que permite obtener un análisis rápido y efectivo de algunos aspectos que intervienen en la toma de decisiones financieras de una empresa.

Esta aplicación desarrollada consta de nueve módulos los mismos que permiten obtener de forma eficiente los complicados cálculos manuales que se realizan en el área de las Finanzas.

Esta entrega constituye la versión 1.0 de InvestMaster.

Este trabajo constituye un gran esfuerzo en la implementación de software en Lenguaje C++ orientado a objetos bajo Windows. En una segunda versión se puede mejorar lo realizado y además incluir un módulo adicional conocido como Valoración de Opciones.

## Definición del Menú Principal

### *Opción Ayuda*

- Acerca
- Salir

### *Opción Módulos*

- Riqueza relativa, retorno y riesgo

- Valor del dinero en el tiempo

  - Flujo de Caja Simple

  - Anualidades

  - Flujo de Caja Irregular

- Cálculos Estadísticos

- Valoración de bonos

  - Análisis de Bonos/Cupones

  - Análisis de Bonos de Descuento

  - Cálculos de Tasa de Interés

- Valoración de Stock

Análisis Financiero

Hoja de Balance

Estado Pérdidas y Ganancias

Indices Financieros

Análisis de Cartera

Análisis de Regresión

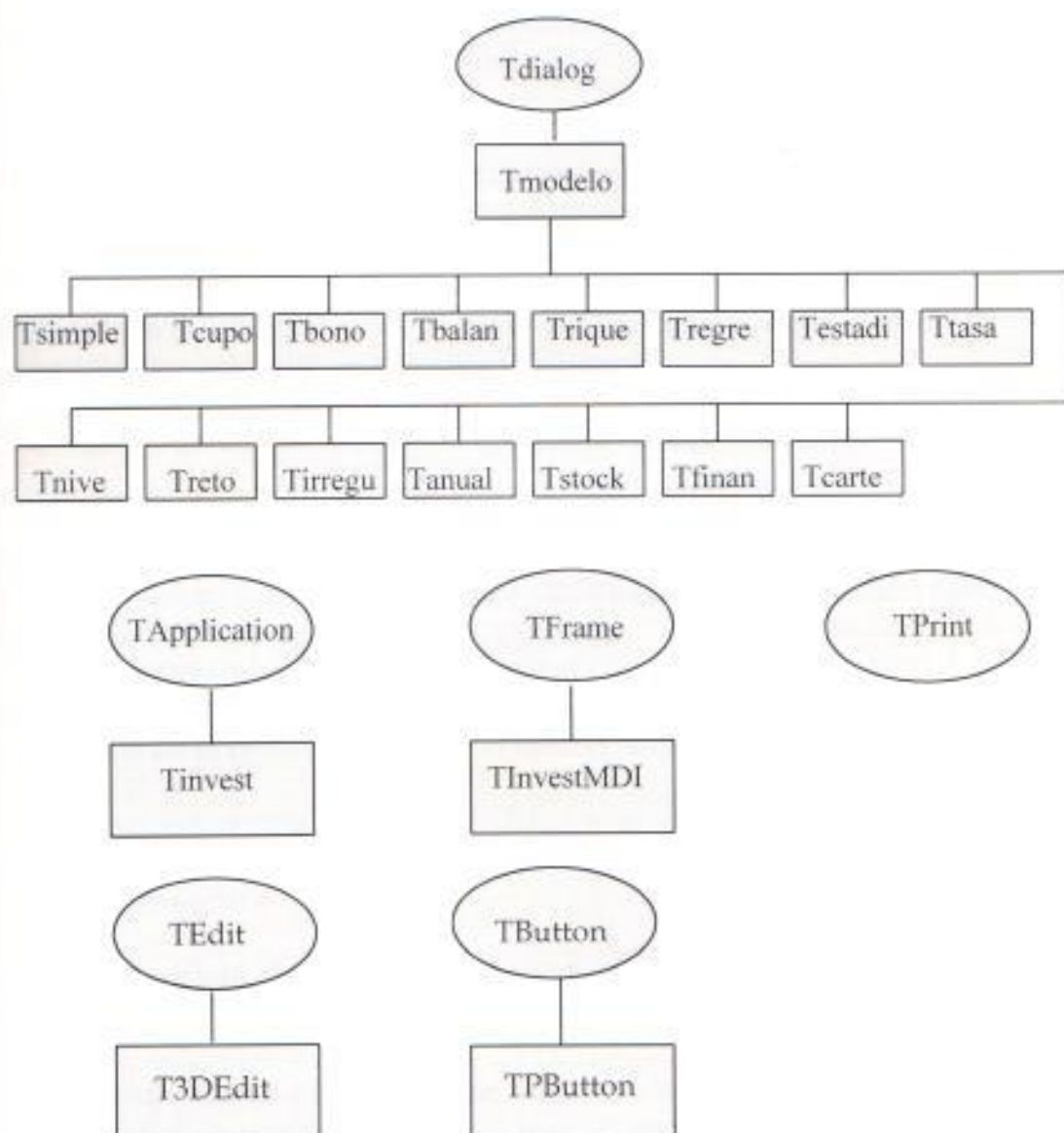
Modelo de costo de acarreo

## Fundamentos de Lenguaje C++ orientado a objeto bajo Windows

### Diseño orientado a Objetos.

Para propósito de la implementación orientada a objetos de la aplicación, se hizo uso de las clases predefinidos de Borland, los cuales se encuentran en la librería *OWL* (Objects Windows Library), las cuales facilitan el trabajo y reducen el tiempo de programación.

A continuación se muestra un esquema de las clases utilizadas:





La clase *Tinvest* es derivada de la clase *Taplicacion*, una instancia de esta clase hace las veces del objeto aplicación.

La clase *TInvestMDI* es heredada de la clase *Tframe*, una instancia de esta clase hace las veces de la ventana principal de la aplicación. Esta ventana es del tipo MDI (multiple document interface), y contiene a las ventanas heredadas de *Tdialog*, que se describen mas adelante. Se lo implementó como MDI, para que futuras versiones que quieran implementar como ventanas hijas a las ventanas de entrada de datos, no tengan que redefinir a la ventana principal.

La clase *Tmodelo* es derivada de la clase *Tdialog*, y contiene la definición de variables globales a las ventanas de diálogos de cada módulo de la aplicación, así como tambien las funciones básicas de abrir y cerrar archivos, todas las clases derivadas de *Tmodelo* pueden utilizar sus funciones de forma pública.

De la clase *Tdialog* se utilizan las siguientes funciones:

*ExecuteDialog*, que genera una instancia de la clase.

*EndDialog*, que llama a la función destructora de la clase.

*SetDialogItemText*, que setea el texto de un control especificando su identificador asignado en el archivo de recursos.

*GetDialogItemText*, que recupera los datos ingresados en la caja de dialogo, luego de que una acción de lectura ha sido ejecutada.

Cada uno de los módulos tiene su propia ventana de entrada de datos, en esta versión están implementadas con cajas de diálogos, cada una es tratada de manera independiente, adicionalmente cada una está provista de funciones específicas; por ejemplo: el control estilo botón "Calcular", dispara un evento que ejecuta un procedimiento virtual de cálculo, cuya acción es diferente para cada ventana de diálogo, heredada de la primera, pero con el mismo nombre de función virtual.

Las funciones virtuales que se utilizan y se detallan a continuación, responden al evento de presionar el botón respectivo en la caja de diálogo respectiva:

*ProcCalcular()*: Realiza el cálculo respectivo al módulo al que pertenece, los resultados son mostrados por medio de ventanas de mensajes.

*ProcAbrir()*: Muestra la ventana de selección de archivo para apertura, en la cual el usuario puede seleccionar el archivo de datos a cargar. Cada módulo tiene una extensión de archivo propio.

*ProcGrabar()*: Muestra una ventana de selección de archivo para grabar, en la cual el usuario puede seleccionar el archivo de datos en el cual se van a grabar. Cada módulo tiene una extensión de archivo propio.

La clase **Tprint**, esta clase está provista de funciones públicas que controlan las impresiones de los cálculos obtenidos de las clases de diálogos anteriores. A continuación describimos cada una de ellas:

*TInitPrint()*: Esta función obtiene primeramente el manejador del dispositivo de contexto de la impresora, opcionalmente displaya la pantalla de seteo, así mismo inicializa las variables internas de la clase.

*GetPageSize()*: Obtiene el tamaño de la página, ancho y largo.

*GetLineSpace()*: Obtiene la altura de una línea.

*GetTab()*: Obtiene tres veces el valor del ancho máximo de un carácter.

*ClosePrint()*: Finaliza el trabajo de impresión.

*EndPage()*: Envía una orden a la impresora para que imprima una nueva página.

Adicionalmente tenemos las variables públicas de la clase: *scalex* y *scaley*, que sirven para escalar los puntos de impresión de la pantalla al tamaño específico de la hoja a imprimir.

Las funciones de la clase **Tprint** actúan como funciones amigas de las clases heredadas de **Tdialog**.

Para validación de entradas se hizo necesaria la implementación de una nueva clase, derivada de **TEdit**, la cual es provista por Borland. Esta nueva clase se llama **T3DEdit**, y para esta versión, tiene características como:

- Sólo permite ingreso de dígitos.
- Acepta negativos si se le especifica.
- Grafica el contorno para dar efecto 3D (tercera dimensión).

A continuación la estructura de la clase:

```

class T3DEdit :public TEdit
{
public:
    T3DEdit(PTWindowsObject Padre, int id, LPSTR texto, int x,int y,int w,int
h,WORD max, BOOL readonly);
    T3DEdit(PTWindowsObject Padre, int id, WORD max, BOOL negativo);
    virtual void WMPaint(RTMessage Msg)=[WM_FIRST + WM_PAINT];
    virtual void Caracter(RTMessage Msg) = [WM_FIRST + WM_CHAR];
    virtual void Habilitar(BOOL b);

protected:
    BOOL Punto;
    BOOL Negativo;
};

```

Así también, para mejorar la interfase de la aplicación, se implementó una nueva clase para graficar botones, TPButton, la cual es heredada de TButton, que es provista por Borland, esta clase tiene características como:

- Acepta un bitmap (32x32) para mostrar cuando no está seleccionado.
- Acepta un bitmap (32x32) para mostrarlo cuando está seleccionado.
- Acepta un bitmap (32x32) para mostrarlo cuando se encuentra deshabilitado.

A continuación la estructura de la clase:

```

class TPButton :public TButton
{
public:
    TPButton(PTWindowsObject, int,WORD,WORD,WORD);
    TPButton(PTWindowsObject,
int,LPSTR,int,int,int,int,BOOL,WORD,WORD,WORD);
    ~TPButton();
protected:
    virtual void ODA DrawEntire(DRAWITEMSTRUCT far &DrawInfo);
    virtual void ODA Focus(DRAWITEMSTRUCT far &DrawInfo);
    virtual void ODA Select(DRAWITEMSTRUCT far &DrawInfo);
    virtual void DrawCuerpo(HDC,RECT);
    virtual void DrawBordes(HDC, RECT, BOOL);
    virtual void DrawIcon_Text(HDC, RECT,BOOL, BOOL, char *);
    virtual void DrawDisabled(HDC, RECT, char *);
    HFONT hFont;
    HBITMAP hBitmap_eu;
    HBITMAP hBitmap_ed;
    HBITMAP hBitmap_d;
};

```

Para tener un mayor detalle de la implementación de cada una de las clases, se recomienda revisar el código de cada uno de los módulos en la sección respectiva.

Los archivos que conforman la aplicación es detallada a continuación:



Nombre de Archivo	Descripción
<b>invest.cpp</b>	Módulo principal
<b>invest.rc</b>	Archivo de Recursos
<b>invest.def</b>	Archivo de definición
<b>regre.cpp</b>	Módulo de Regresión
<b>print.cpp</b>	Módulo de Impresión
<b>modelo.cpp</b>	Modelo de caja de diálogo
<b>cartera.cpp</b>	Módulo de cartera
<b>stock.cpp</b>	Módulo de valoración de stock
<b>tasa.cpp</b>	Módulo de tasa de crecimiento
<b>bono.cpp</b>	Módulo de bonos de descuentos
<b>cupo.cpp</b>	Módulo de análisis de bonos/cupos
<b>acarreo.cpp</b>	Módulo de acarreo
<b>irregu.cpp</b>	Módulo de Flujo de caja irregular
<b>indi.cpp</b>	Módulo de Índices financieros
<b>anual.cpp</b>	Módulo de anualidades
<b>rique.cpp</b>	Módulo de riqueza relativa
<b>simple.cpp</b>	Módulo de flujo de caja simple
<b>nive.cpp</b>	Módulo de Niveles de riqueza
<b>retorno.cpp</b>	Módulo de retorno
<b>perdidas.cpp</b>	Módulo de estados de pérdidas y ganancias
<b>balance.cpp</b>	Módulo de Hoja de balance
<b>estadi.cpp</b>	Módulo de cálculos estadísticos
<b>3dbutton.cpp</b>	Control Picture Button
<b>3dedit.cpp</b>	Control de Edición con validación

Tabla de Archivos usados en InvestMaster para Windows vs. 1.0

Cada uno de los módulos está implementado en su propio archivo, la definición de la clase se encuentra en su respectivo archivo de cabecera (extensión hpp).

Además se utilizan otros archivos de cabecera:

*invest.h*: Registra definiciones globales para todos los módulos

*identifi.h*: Registra los identificadores de todos los controles utilizados en las cajas de diálogos y que son usados tanto en los módulos como en el archivo de recursos.

Todos estos archivos se encuentran contenido en el archivo de proyecto *invest.prj*, el cual contiene la información necesaria para generar el ejecutable de la aplicación (*invest.exe*).

## *Descripción de los Módulos del Sistema*

### **Módulo 1 : Riqueza Relativa, Retorno y Riesgo**

#### **Riqueza relativa**

Riqueza relativa es una técnica para medir el retorno de una inversión, es decir cuánto recuperamos de una inversión con relación a lo invertido. Su fórmula de cálculo es muy sencilla, pues es la relación entre el capital recuperado de la inversión para el valor invertido.

La riqueza relativa tiene como valores característicos estos son los que están alrededor de uno (1). Cuando existe un beneficio en una inversión durante un período determinado, la riqueza relativa es mayor que uno (1); pero si existe una pérdida este valor será menor que uno (1).

Si el valor recuperado es igual al valor invertido entonces la riqueza relativa tendrá un valor de uno (1). Pero hay que anotar que el dinero no es el mismo al final del período de inversión por lo que se puede considerar que el inversionista ha perdido su capital, este análisis se lo verá en el siguiente módulo.

Note además que no puede existir una riqueza relativa negativa. El peor de los casos ocurre cuando el dinero invertido se lo pierde completamente. En tal situación la riqueza relativa es cero, pero nunca negativa.

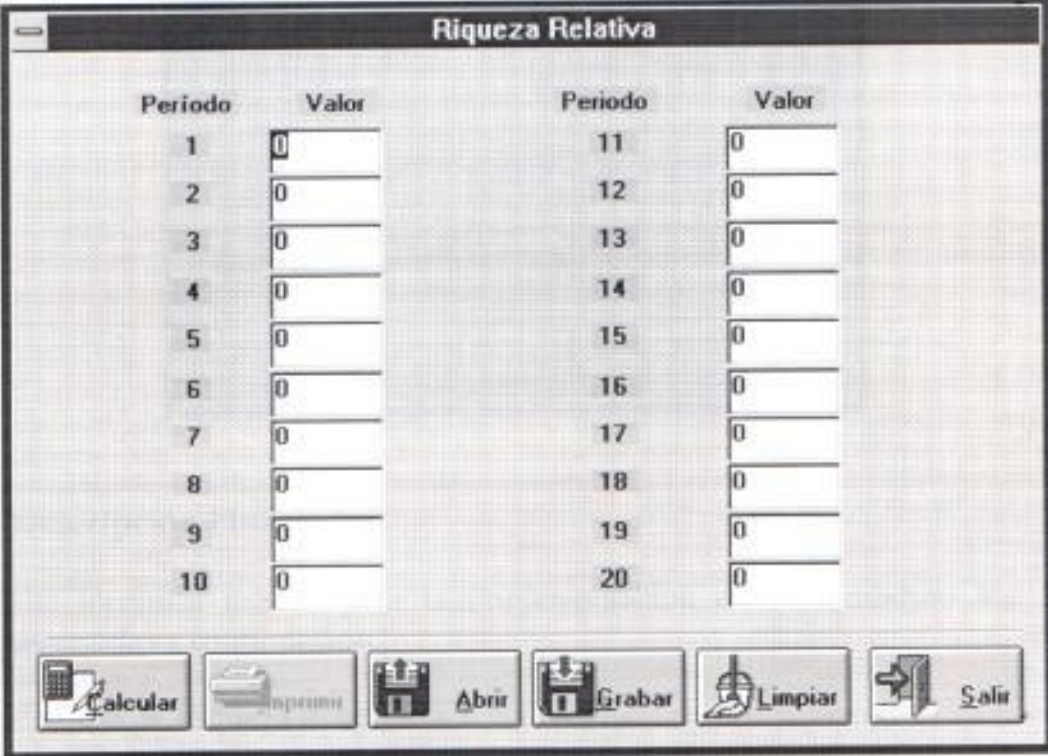
#### **Retorno y Niveles de Riqueza**

La riqueza relativa debe ser convertida en retorno o en un rendimiento a ser expresado en términos de porcentaje. Por tanto el retorno debe tener como datos de entrada valores típicos alrededor de cero. En este paquete tanto el retorno como los niveles de riqueza se inician en el período uno, puesto que en el período cero no existe retorno sino que se inicia la inversión.

## Interfases

Las interfases usadas para este módulos son las que se muestran a continuación :

Ingreso de Datos :



Periodo	Valor	Periodo	Valor
1	0	11	0
2	0	12	0
3	0	13	0
4	0	14	0
5	0	15	0
6	0	16	0
7	0	17	0
8	0	18	0
9	0	19	0
10	0	20	0

Calcular Imprimir Abrir Grabar Limpiar Salir



Resultados :

CALCULOS TOTALES DEL RETORNO		
Riqueza 1	6.5000	
Riqueza 2	8.8900	
Riqueza 3	4.4000	
	<b>Media Aritmética</b>	<b>Media Geométrica</b>
Riqueza Relativa	3.2983	2.5170
Retorno	2.2983	1.5170
<b>Riesgo de retorno de media aritmética</b>		
Desviación estándar	2.6687	
Varianza	7.1218	
OK		

### Descripción Financiera

Las fórmulas usadas para la implementación de este módulo son resumidas a continuación :

$$\text{Retorno} = \text{Riqueza Relativa} - 1$$

$$\text{Riqueza Relativa} = \text{Nivel de Riqueza}(i) / \text{Nivel de Riqueza}(i-1)$$

Cálculos comunes:

#### Media Aritmética de la Riqueza Relativa

$$\mu_{RR} = \Sigma (\text{Riqueza Relativa}) / n$$

#### Media Aritmética del Retorno

$$\mu_{\text{Retorno}} = \mu_{RR} - 1$$

**Media Geométrica de Riqueza Relativa**

$$\phi_{RR} = \sqrt[n]{\text{Riqueza Relativa 1} \times \dots \times \text{Riqueza Relativa n}}$$

**Media Geométrica de Retorno**

$$\phi_{\text{Retorno}} = \phi_{RR} - 1$$

**Varianza**

$$V = \frac{\sum (\text{Retorno} - \mu_{\text{Retorno}})^2}{n}$$

**Desviación estándar**

$$\sigma = \sqrt{V}$$

## Módulo 2: El valor del dinero en el tiempo

Este módulo involucra 3 submódulos los cuales son descritos a continuación:

### Submódulo 1: Flujo de Caja Simple

En este submódulo se permite el ingreso de 3 de los 4 datos, uno de los cuales es calculado por InvestMaster cuando se presione el botón de calcular.

### Submódulo 2: Anualidades

En este segundo submódulo se ingresa ya sea el valor presente o el futuro o no se llena ninguno.

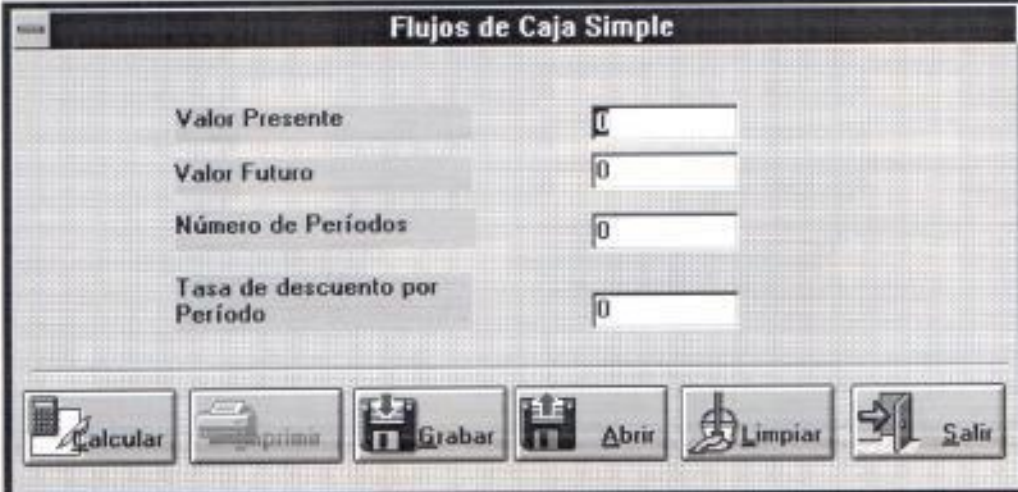
Si se especificó uno de los anteriores se deberá ingresar 2 de los siguientes campos. Al presionar el botón de calcular este mostrará los valores faltantes.

### Submódulo 3: Pagos Irregulares

En este submódulo se ingresan todos los pagos con el valor presente o la tasa de descuento. Calcular resuelve con fórmulas el valor no ingresado.

### Interfases:

#### Submódulo 1: Flujo de Caja Simple



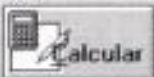

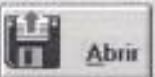
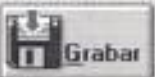

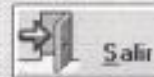
Valor Presente	0
Valor Futuro	0
Número de Períodos	0
Tasa de descuento por Período	0

Calcular Imprimir Grabar Abrir Limpiar Salir

## Submódulo 2: Anualidades

**Anualidades**

Valor Presente	0
Valor Futuro	0
Pagos	0
Número de Períodos	0
Tasa de Descuento	0

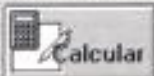

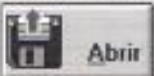
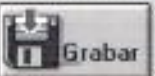
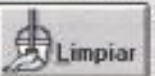

 Calcular
  Imprimir
  Abrir
  Grabar
  Limpiar
  Salir

## Submódulo 3: Flujo de Caja Irregular

**Flujos de Caja Irregular**

Períodos	Pagos	Períodos	Pagos	Períodos	Pagos
1	0	11	0	21	0
2	0	12	0	22	0
3	0	13	0	23	0
4	0	14	0	24	0
5	0	15	0	25	0
6	0	16	0	26	0
7	0	17	0	27	0
8	0	18	0	28	0
9	0	19	0	29	0
10	0	20	0	30	0

Valor Presente 0      Tasa de descuento por período 0

 Calcular
  Imprimir
  Abrir
  Grabar
  Limpiar
  Salir

## Descripción Financiera

### Submódulo 1: Flujo de Caja Simple

$$\text{Valor Futuro} = \text{Valor Presente} * (1 + \text{Tasa})^t$$

donde t = Número de periodos

### Submódulo 2: Anualidades

### Submódulo 3: Flujo de Caja Irregular

$$\text{Valor Presente} = \sum \text{Pagos} / (1 + \text{Tasa})^t$$

$$\text{Valor Futuro} = \sum \text{Pagos} * (1 + \text{Tasa})^t$$

donde t = número de periodos



### Módulo 3: Análisis Estadístico

Este módulo permite obtener los valores estadísticos correspondientes a dos variables (variable A, variable B). Los valores estadísticos que obtiene el módulo al realizar los cálculos son :

- La Media
- La Desviación Estándar
- La Varianza
- El Skewness
- El Kurtosis
- El T-test
- La Covarianza
- La Correlación

Estos cálculos son muy importantes y poseen una gran interpretación . Uno de los más importantes es la extensión de la covarianza o correlación entre los activos individuales comprendidos en cartera:

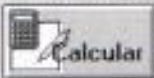

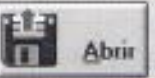



Así pues las medidas de riesgo pueden además ser expresadas usando el coeficiente de correlación en lugar de la covarianza.

Interfases :

Ingreso de Datos

**Análisis Estadístico**

	Variable A	Variable B		Variable A	Variable B
1	0	0	11	0	0
2	0	0	12	0	0
3	0	0	13	0	0
4	0	0	14	0	0
5	0	0	15	0	0
6	0	0	16	0	0
7	0	0	17	0	0
8	0	0	18	0	0
9	0	0	19	0	0
10	0	0	20	0	0

 Calcular
  Imprimir
  Abrir
  Grabar
  Limpiar
  Salir

Resultados :

**CALCULOS ESTADISTICOS**

Media A=	1.0000	Media B=	1.0200
Desviación		Desviación	
Estándar A=	0.2055	Estándar B=	0.2251
Varianza A=	0.0422	Varianza B=	0.0507
SKEWNESS A=	-0.0000	SKEWNESS B=	-0.1263
KURTOSIS A=	12.9017	KURTOSIS B=	11.6938
t-test Media (A)=	15.3897		
t-test Media (B)=	14.3298		
tMed(A)=tMed(B) =	-0.2075		
Covarianza=	0.0133		
Correlación=	0.2883		
t-test Correlación=	0.8515		

OK

**Descripción Financiera****Media :**

$$\mu = \sum x_i / n$$

**Varianza :**

$$\sigma = \sum (x - \mu) / (n - 1)$$

**Desviación Estándar :**

$$\beta = \sqrt{\sigma}$$

**Skewness :  $\sigma\mu\chi\beta$** 

$$\text{SKEW} = \sum (x - \mu) / \sigma$$

**Kurtosis :**

$$\text{KURT} = \sum (x - \mu) / \sigma$$

**T-test :**

$$\text{T-test Media} = x - \mu / \sigma * n$$

**T-test A,B :**

$$\text{T-test} = (\mu / \sigma * (n - 1) + \sigma * (n - 1) * * n + n) - (\mu / n + n - 2 ** n)$$

**Covarianza :**

$$\text{COV} = \sum (x - \mu) * (y - \mu) / n - 1$$

**Correlación :**

$$\text{CORR} = \text{COV} / \sigma_x \sigma_y$$

**T-correlación :**

$$\text{T-test} = \text{CORR}n - 2 / 1 - \text{CORR}$$

## **Módulo 4: Valoración de bonos**

### **Submódulo 1: Análisis de Cupón / Bonos**

Cuando un bono madura o vence, el tenedor de un bono debe hacer los pagos correspondientes establecidos entre la fecha original de emisión y la fecha de vencimiento; estos pagos son conocidos como Cupones.

En este submódulo se deberán ingresar obligatoriamente todos los 5 primeros datos, el Precio o el Retorno al Vencimiento pueden ser especificados; en caso contrario el programa calculará el que no se ingresó, así como la duración y el Retorno al Vencimiento aproximado.

### **Submódulo 2: Análisis de Bonos de Descuento**

Este bono promete un cierto pago simple a una fecha especificada en el futuro y es vendido por un valor menor que de dicho pago futuro. Normalmente el pago futuro es el valor facial del bono. La diferencia entre el valor facial y el precio vendido es el bono de descuento.

Para trabajar con este módulo se requiere el valor facial y los días de vencimiento. De los 3 campos restantes se deberá ingresar uno de ellos, los demás serán calculados cuando se seleccione esta opción.

### **Submódulo 3: Tasas de Interés**

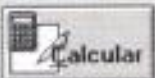





En este caso se deberán ingresar los 4 primeros datos. El programa calculará únicamente la Tasa de Interés.



Interfases :

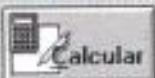





Submódulo 1 : Análisis de Cupón/ Bonos

Valoración Cupón/Bonos	
Valor Facial	<input type="text" value="0"/>
Pagos de cupón por año	<input type="text" value="0"/>
Años de vencimiento	<input type="text" value="0"/>
Cupones remanentes este año	<input type="text" value="0"/>
Tasa de cupón anual [ % ]	<input type="text" value="0"/>
Precio	<input type="text" value="0"/>
Retorno al vencimiento	<input type="text" value="0"/>

 **Calcular**
 **Imprimir**
 **Abrir**
 **Grabar**
 **Limpiar**
 **Salir**







Submódulo 2 : Análisis de Bonos de Descuento

Valoración de bonos de descuento	
Valor Facial	<input type="text" value="0"/>
Días de Vencimiento	<input type="text" value="0"/>
Precio del dólar	<input type="text" value="0"/>
Descuento dólar	<input type="text" value="0"/>
Rendimiento del Descuento	<input type="text" value="0"/>

 **Calcular**
 **Imprimir**
 **Abrir**
 **Grabar**
 **Limpiar**
 **Salir**

## Submódulo 3: Tasas de Interés

Tasa de crecimiento		
	Bono período corto	Bono período largo
Retorno al Vencimiento	<input type="text" value="0"/>	<input type="text" value="0"/>
Tiempo de vencimiento en meses	<input type="text" value="0"/>	<input type="text" value="0"/>

 Calcular
  Imprimir
  Abrir
  Grabar
  Limpiar
  Salir

## Descripción Financiera

## Submódulo 1: Análisis de Cupón/ Bonos

Precio :

$$P = \frac{Ct}{(1+r)^t} + \frac{VF}{(1+r)^t}$$

donde,

- t = tiempo en años
- r = retorno al vencimiento
- FV = Valor facial
- Ct = Valor del bono aplicado con la tasa  $Ct = FV * \text{tasa}$

## Retorno aproximado al vencimiento :

$$RA = \left( \frac{Ct + (FV - P) / n}{(FV + P) / 2} \right)$$

$$RA = \left( \frac{Ct + (FV - P) / n}{(FV + 2 * P) / 3} \right)$$

## Duración :

$$D = \frac{t * Ct}{(1+r) * P}$$

**Submódulo 2: Análisis de Bonos de Descuento**

Precio :

$$P = FV - DISC = FV \cdot (1 - d \cdot t) / 360$$

Descuento :

$$DISC = FV \cdot d \cdot t / 360$$

Retorno descuento :

$$d = 360 \cdot (FV - P) / FV(t)$$

donde t= días de vencimiento

Equivalente Retorno bono :

$$EBY = 365 \cdot d / 360 - d \cdot t$$

**Submódulo 3: Tasas de Interés**

$$\text{Tasa de Crecimiento} = \frac{(1+b)^{d-c} - 1}{(1+a)}$$

donde,

a= retorno al venc. período corto

b= retorno al venc. período largo

c= tiempo de venc. período corto

d= tiempo de venc. período largo

## Módulo 5: Modelo del Dividendo

Este módulo permite al usuario ingresar máximo 8 dividendos futuros. Permite hallar el precio de stock ingresando previamente, la tasa de descuento y la de crecimiento.

Para cualquier inversión se conoce que el Valor de la inversión depende de los flujos de caja que serán generados por dicha inversión, el tiempo de esos flujos de caja y la tasa de descuento aplicada a esos flujos.

El tiempo corriente se asume que es el tiempo antes del tiempo cero. Esto implica que un dividendo a ser recibido en el período uno debe ser descontado por sólo un período. Se asume que los dividendos crecen a una tasa de crecimiento a partir del último periodo ingresado por el usuario.

### Interfases :

#### Valoración de Stock

Período	Dividendo
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0

Precio Corriente	0
Tasa de descuento	0
Tasa de crecimiento	0

Calcular	Imprimir	Abrir	Grabar	Limpiar	Salir
----------	----------	-------	--------	---------	-------



## Descripción Financiera

Precio corriente :

$$P_{n-1} = D_n / k - g$$

$$PC = D_1 / (1+k) + P_{n-1} / (1+k)$$

Es decir:

$$PC = D_1 / (1+k) + D_n / (k-g) * (1+k)$$

donde ,

k= tasa de descuento

g= tasa de crecimiento

## **Módulo 6: Análisis financiero**

Este módulo realiza una hoja de balance, un Estado de Pérdidas y Ganancias y una determinación de todos los Índices Financieros.

### **Submódulo 1: Hoja de Balance**

En la hoja de Balance deberán ser ingresados todos los datos de los activos, pasivos y patrimonios. Investmaster realizará el cálculo de los totales repectivos cuadrando de esta manera el Balance únicamente con los datos ingresados por el usuario.

### **Submódulo 2: Estado de Pérdidas y Ganancias**

El programa realizará cálculos de todos los valores que se encuentran con letras mayúsculas. Es decir por ejemplo: las Ventas Netas.

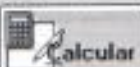



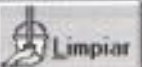
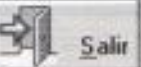
### **Submódulo 3: Índices Financieros**

En base a todos los datos ingresados en los submódulos anteriores. Investmaster mostrará al seleccionar este submódulo todos los índices financieros calculados sin necesidad de que se ingrese ningún valor por el usuario.

Interfases :

Submódulo 1 : Hoja de Balance

ACTIVOS		PASIVOS Y PATRIMONIOS	
<b>ACTIVO CORRIENTE</b>		<b>PASIVO CORRIENTE</b>	
Caja-Bancos	0	Cuentas por Pagar	0
Inventario	0	Documentos por Pagar	0
Documentos por Cobrar	0	Impuestos por Pagar	0
Cuentas por Cobrar	0	Gastos Acumulados	0
Otros	0	<b>Total Pasivo Corriente</b>	
<b>Total Activo Corriente</b>		<b>PASIVO FIJO</b>	
<b>ACTIVOS FIJOS</b>		Documentos por Pagar	0
Terrenos	0	Hipotecas por Pagar	0
Planta y Equipos	0	Otros	0
Depreciacion Acumulada	0	<b>Total Pasivo Fijo</b>	
Otros	0	<b>PATRIMONIO</b>	
<b>Total Activos Fijos</b>		Acciones Preferentes	0
<b>Total Activos</b>		Acciones Comunes	0
		Ganancias Retenidas	0
		<b>Total Patrimonio</b>	
		<b>Total Pasivos y Patrimonio</b>	

 **Calcular**
 **Imprimir**
 **Abrir**
 **Grabar**
 **Limpiar**
 **Salir**


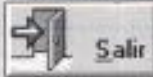
## Submódulo 2: Estado de Pérdidas y Ganancias

Estado de Pérdidas y Ganancias	
Ventas Brutas	0
Devolución y Descuentos	0
Ventas Netas	
Costo Mercaderías Vendidas	0
Utilidad Bruta Operacional	
Gasto de Adm., Vtas y Generales	0
Gasto Depreciación	0
Utilidad antes Intereses e Impuestos	
Gasto de Interés	0
Utilidad Antes Impuestos	
Impuestos	0
Utilidad Neta	

Calculador    Imprimir    Abrir    Grabar    Limpiar    Salir

## Submódulo 3 : Índices Financieros

Índices Financieros	
Índice de Liquidez :	1.3915
Prueba del ácido :	0.7594
Rotación total de activos :	17.1390
Rotación de activos fijos :	1.7139
Período promedio de cobro :	1.0680
Rotación de Inventario :	37.7313
Índice de deudas a activos :	0.6570
Índice de deudas a inversión :	1.9155
Margen de utilidad:	0.0019
Tiempo-Interes ganado :	1.0798
Retorno de activos :	0.0029
Retorno de Inversión :	0.0084

## Descripción Financiera

## Submódulo 1 : Hoja de Balance

TOTAL ACTIVO CORRIENTE = Suma de activos corrientes

TOTAL ACTIVOS FIJOS= Suma de activos fijos -Depreciación

TOTAL ACTIVOS = Suma de Activos

TOTAL PASIVO CORRIENTE= Suma de pasivos corrientes

TOTAL PASIVOS FIJOS= Suma de pasivos

TOTAL PATRIMONIO= Suma de patrimonio

TOTAL PASIVOS Y PATRIMONIO = Suma de Pasivos y Patrimonios



**Submódulo 2: Estado de Pérdidas y Ganancias**

VENTAS NETAS= Ventas brutas - Dev y Descuentos

UTILIDAD OPERACIONAL BRUTA = Ventas Netas - Costo de Merc. Vend.

UTILIDAD ANTES INTERESES E IMPUESTOS= Utilidad Op. Bruta - Gastos de Adm., Ventas y Grales-Gastos de Depreciación

UTILIDAD ANTES IMPUESTOS=Utilidad antes int. e impuestos - Gastos de interés neto

UTILIDAD NETA= Utilidad antes impuestos - Impuestos

**Submódulo 3: Indices financieros**

Indice de liquidez = Activo Corriente / Pasivo Corriente

Prueba Acida = ( Activo Corriente - Inventario ) / Pasivo Corriente

Margen de Utilidad = Utilidad Total / Ventas totales

Retorno de Activos = Utilidad neta / Total activo

Retorno de Inversión = Utilidad neta / Total patrimonio

Tiempo interés ganado = Utilidad antes interes e impuestos / Gastos interes

Tasa de Deudas de activo total = Pasivo total / Activo total

Deuda a inversión = Pasivo total / Patrimonio total

Periodo promedio de cobro = ( Cuentas por cobrar / Ventas netas ) / 360

Retorno de inventario = Ventas netas / Inventario

## Módulo 7: Análisis de Cartera

Este módulo requiere algunos de los cálculos estadísticos citados anteriormente que permiten manejar e interpretar una cartera.

Investmaster deberá permitir el ingreso de hasta 5 stocks con su respectivo retorno, desviación estándar, pesos y correlaciones.


Cuando se ingresan 2 stocks los cálculos son simples, pero cuando se ingresan más de 2, se presenta un análisis de cartera múltiple, cuyas fórmulas son más complejas.


Una vez que los datos son ingresados se podrá calcular el retorno esperado en cartera y la desviación estándar de cartera.


### Interfases:


Ingreso de Datos:


Análisis de Cartera									
No	E(R)	Desviación Estándar	Pesos	CORRELACIONES					
				1	2	3	4	5	
1	0	0	0	1	1.0000				
2	0	0	0	2	0	1.0000			
3	0	0	0	3	0	0	1.0000		
4	0	0	0	4	0	0	0	1.0000	
5	0	0	0	5	0	0	0	0	1.0000


 **Calcular**

 **Imprimir**

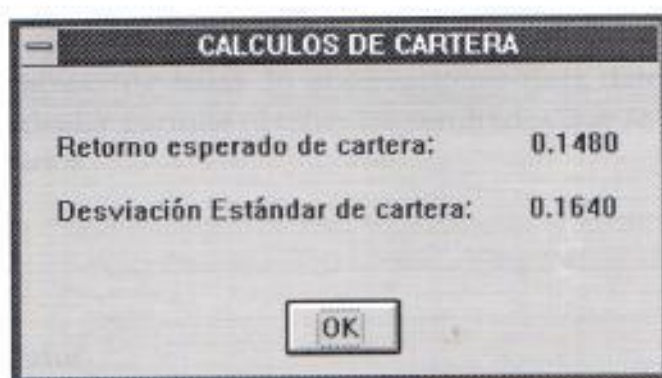
 **Abrir**

 **Grabar**

 **Limpiar**

 **Salir**

Resultados:



## Descripción Financiera

### Submódulo 1 : Análisis de Cartera simple

Retorno esperado de Cartera :

$$E(R_p) = W_a E(R_a) + W_b E(R_b)$$

donde,

$W_a, b$  son los pesos

$R_a, b$  son los retornos respectivos

Desviación Estándar de cartera

$$\sigma = W_a * \sigma_a + W_b * \sigma_b + 2 * W_a * W_b * \sigma_{a,b}$$

### Submódulo 2 : Análisis de Cartera Múltiple

Retorno esperado de Cartera

$$E(RP) = W_i * E(R_i)$$

Desviación Estándar de Cartera:

$$\sigma = W_i W_j \sigma_{i,j}$$



## Módulo 8: Análisis de Regresión

Investmaster realiza los cálculos de regresión fácilmente. La regresión permite el ingreso de hasta 36 observaciones para datos de  $x,y$ . Al seleccionar calcular permite obtener los resultados que se mostrarán en el cuadro inferior.

Interfases :

Ingreso de Datos

Análisis de Regresión								
	Var X	Var Y		Var X	Var Y		Var X	Var Y
1:	0	0	13:	0	0	25:	0	0
2:	0	0	14:	0	0	26:	0	0
3:	0	0	15:	0	0	27:	0	0
4:	0	0	16:	0	0	28:	0	0
5:	0	0	17:	0	0	29:	0	0
6:	0	0	18:	0	0	30:	0	0
7:	0	0	19:	0	0	31:	0	0
8:	0	0	20:	0	0	32:	0	0
9:	0	0	21:	0	0	33:	0	0
10:	0	0	22:	0	0	34:	0	0
11:	0	0	23:	0	0	35:	0	0
12:	0	0	24:	0	0	36:	0	0

 Calcular	 Imprimir	 Abrir	 Grabar	 Limpiar	 Salir
--	--	---	--	--	---

## Resultados :

CALCULOS REALIZADOS	
Beta=	1.6791
Alfa=	0.0170
Error Estándar BETA=	0.6841
Error Estándar ALFA=	0.0684
t_beta=	2.4546
t_alfa=	0.2488
Grados de libertad=	10.00
SUMA DE ERRORES CUADRADOS	
Error=	0.0432
Regresión=	0.0260
Total=	0.0692
R-cuadrado=	0.3760
Error Estándar de Estimación =	0.0657
OK	

## Descripción Financiera

$$\text{Retorno promedio } x = \Sigma x / n$$

$$\text{Retorno promedio } y = \Sigma y / n$$

$$\text{Desviación } x = x - \text{Retorno promedio } x$$

$$\text{Desviación } y = y - \text{Retorno promedio } y$$

$$\text{Varianza } x = \text{Desv } x / n$$

$$\text{Varianza } y = \text{Desv } y / n$$

$$\text{Desv. est. } x = \sqrt{\text{Varianza } x}$$

$$\text{Desv. est. } y = \sqrt{\text{Varianza } y}$$

$$\text{Desviación } x,y = \text{Desv } x * \text{Desv } y$$

$$\text{Covarianza } x,y = \text{Desviación } x,y / n$$

$$\text{BETA} = \text{Covarianza } x,y / \text{Varianza } x$$

$$\text{ALFA} = \text{Retorno promedio } y - \text{BETA} * \text{Retorno promedio } x$$

$$\text{CORRELACION} = \text{Covarianza } x,y / \text{Desv. Est } x \text{ Desv Est. } y$$

$$\text{R-Cuadrada} = (\text{Correlación } x,y)^2$$

$$a = (\sum y) (\sum x) - (x) (\sum xy) / n(\sum x) - (x)$$

$$b = n(\sum xy) - (\sum x) (\sum y) / n(\sum x) - (x)$$

Error Estándar de estimación

$$S_{xx} = \sum x^2 - 1/n(\sum x)^2$$

$$S_{xy} = \sum xy - 1/n(\sum x)(\sum y)$$

$$S_{yy} = \sum y^2 - 1/n(\sum y)^2$$

$$S_e = S_{yy} - bS_{xy} / n - 2$$

Grados de libertad = n-2

$$\text{Error Estándar de BETA} = S_e / (S_{xx})$$

$$t\text{-stat BETA} = b / \text{Error Estándar de BETA}$$

$$\text{Error Estándar de ALFA} = S_e \sqrt{1/n + \sum (x-x_i)^2 / S_{xx}}$$

$$t\text{-stat ALFA} = a / \text{Error Estándar de ALFA}$$

$$\text{Error} = S_{yy} - bS_{xy}$$

$$\text{Regresión} = bS_{xy}$$

$$\text{Total} = \text{Error} + \text{Regresión}$$

## Módulo 9: Futuro y Costo de Acarreo

El módulo de análisis de futuro y costo de acarreo permite especificar 3 datos de los 4 valores que incluyen la relación costo de acarreo y calcular el cuarto.

### Interfases

Ingreso de Datos:

**Costo de Acarreo**

Precio Futuro o Precio Futuro Lejano	0
Precio Disponible o Precio Futuro Disponible	0
Costo de Acarreo	0
Tiempo (días) entre 2 fechas de entrega	0

Calcular Imprimir Abrir Grabar Limpiar Salir

Resultado:

**CALCULOS COSTO DE ACARREO**

El Precio Futuro es: 925.7917

OK

### Descripción Financiera

Precio futuro

$$FP = SP * ( 1 + CC )$$

donde, SP es el precio disponible  
CC el costo de acarreo  
t el tiempo en días

Precio Disponible

$$SP = FP / ( 1 + CC )$$

Costo de acarreo

$$CC = ( FP / SP ) - 1$$

Tiempo

$$t = \ln FP / SP * 365 / \ln(1+CC)$$



## *Componentes básicos del sistema*

### *Manejo de Archivos*

En el diseño del sistema se usa el manejo de archivos los cuales permiten grabar y recuperar la información almacenada. El sistema utiliza una extensión específica para cada módulo la cual permite al usuario reconocer rápidamente su trabajo almacenado. Así mismo permite tener una información más ordenada.

### *Sistema de Ayuda*

Se permite tener un sistema de ayuda que identifica la función del sistema, le permite conocer el tipo de información que debe ser almacenada o ingresada en cada módulo y las salidas que se esperan obtener.

Este sistema ha sido diseñado de manera tal que el usuario puede acceder a la información deseada ingresando una de las palabras del menú que hagan referencia al módulo deseado o por medio de una búsqueda alfabética.

En este diseño se hace uso del sistema de ayudas del Window.

### *Impresión de Reportes*

Los resultados obtenidos de cada módulo pueden ser presentados por pantalla o impresos.

Los resultados se pueden imprimir por impresora o pueden ser recuperados en un utilitario, por ejemplo Microsoft Word.

## Código del programa fuente

```
# define WIN31
# include <windows.h>
# include <mdi.h>
# include <applicat.h>

# include "identifi.h"
# include "invest.h"
# include "printer.hpp"
# include "cartera.hpp"
# include "stock.hpp"
# include "tasa.hpp"
# include "bono.hpp"
# include "cupo.hpp"
# include "acarreo.hpp"
# include "irregu.hpp"
# include "anual.hpp"
# include "simple.hpp"
# include "ruve.hpp"
# include "retorno.hpp"
# include "perdidas.hpp"
# include "balance.hpp"
# include "estadi.hpp"
# include "rique.hpp"
# include "regre.hpp"
# include "indi.hpp"

# define CHILD_MENU_POS 0
# define ERRORNUM -1
# define or |
# define and &&

/***** Variables Globales de la Aplicación *****/
LPSTR PROGRAM_NAME="InvestMaster para Windows";

int fbalan=1,fpyg=1;
//hoja de balance static
double afx[10];
double afy[12],cc=0.0,in=0.0;
double ac=0.0,afijo=0.0,pc=0.0,pfijo=0.0,totac=0.0,topyp=0.0,patri=0.0; // hoja de Balance

//perdidas
double epx[7];
char epg1[12],epg2[12],epg3[12],epg4[12],epg5[12],epg6[12],epg7[12];
double vn,ub,uai2,uai,un;

/***** Definicion de la clase para el manejo de la aplicacion *****/
class TInvest:public TApplication
{
public:
TInvest(LPSTR,HINSTANCE,HINSTANCE,LPSTR,int);
virtual void InitInstance();
virtual void InitMainWindow(void);
};
```

```

/***** About *****/

class TAbout:public TDialog
{
public:
    virtual void SetupWindow();
    TAbout(PTWindowsObject padre, LPSTR AName):TDialog(padre, AName){};
};

void TAbout::SetupWindow()
{
    long memoria;
    int recursos;
    char Buffer[256];

    TDialog::SetupWindow();
    SetCaption("InvestMaster para Windows");
    memoria = GetFreeSpace(0)/1024;
    wsprintf(Buffer,"%lu KB Libres",memoria);
    SetDlgItemText(HWindow, IDM_MEMORIA, Buffer);

    recursos = GetFreeSystemResources(0);
    wsprintf(Buffer,"%i %% Libres",recursos);
    SetDlgItemText(HWindow, IDM_RECURSO, Buffer);
}

/***** Definicion de la clase para la ventana MDI padre *****/

class TInvestMDI:public TMDIFrame
{
public:
    TInvestMDI(LPSTR,int);
    ~TInvestMDI();
    virtual void GetWindowClass(WNDCLASS &);
    virtual void OpenAnual() = [CM_FIRST + IDM_ANUAL];
    virtual void OpenIrregu() = [CM_FIRST + IDM_IRREGU];
    virtual void OpenAcarreo() = [CM_FIRST + IDM_ACAR];
    virtual void OpenCupo() = [CM_FIRST + IDM_CUPO];
    virtual void OpenBono() = [CM_FIRST + IDM_BONO];
    virtual void OpenTasa() = [CM_FIRST + IDM_TASA];
    virtual void OpenStock() = [CM_FIRST + IDM_STOCK];
    virtual void OpenCarte() = [CM_FIRST + IDM_CARTE];
    virtual void OpenEstadi() = [CM_FIRST + IDM_ESTADI];
    virtual void OpenBalances() = [CM_FIRST + IDM_BALAN];
    virtual void OpenPerdidas() = [CM_FIRST + IDM_PERDI];
    virtual void OpenRetorno() = [CM_FIRST + IDM_RETO];
    virtual void OpenNive() = [CM_FIRST + IDM_NIVE];
    virtual void OperSimple() = [CM_FIRST + IDM_SIMPLE];
    virtual void OperRique() = [CM_FIRST + IDM_RIQUE];
    virtual void OpenIndi() = [CM_FIRST + IDM_INDI];
    virtual void About() = [CM_FIRST + IDM_ABOUT];
    virtual void OpenRegre() = [CM_FIRST + IDM_REGRE];
    virtual void OpenHelp() = [CM_FIRST + IDM_HELP];
    virtual void CMExit(RTMessage Msg) = [CM_FIRST + CM_EXIT];

protected:
    virtual void SetupWindow(void);
    int tipo;
    HMENU hMenu;
};

void TInvestMDI::OpenHelp()

```

```

    WinHelp(HWindow,"invest.hlp",HELP_CONTENTS,0L);
    return;
}

void TInvestMDI::SetupWindow(void)
{
    TMDIframe::SetupWindow();
    return;
}

void TInvestMDI::OpenSimple()
{
    GetApplication()->ExecDialog( new TSimple(this,"SimpleBox"));
    return;
}

void TInvestMDI::OpenRegre()
{
    GetApplication()->ExecDialog( new TRegre(this,"RegreBox"));
    return;
}

void TInvestMDI::OpenRique()
{
    GetApplication()->ExecDialog( new TRique(this,"RiqueBox"));
    return;
}

void TInvestMDI::OpenIndi()
{
    if (fbalan==1 && fpyg==1)
    {
        MessageBeep(-1);
        MessageBox(NULL,"No se han ingresado datos en Módulos \n Balance ni Pérdidas y Ganancias", " MENSAJE DE ERROR",MB_ICONSTOP|MB_OK);
    }
    else
    {
        if (fbalan==0 && fpyg==1)
        {
            MessageBeep(-1);
            MessageBox(NULL,"No se han ingresado datos en Módulo Pérdidas y Ganancias", " MENSAJE DE ERROR",MB_ICONSTOP|MB_OK);
        }
        else
        {
            if (fbalan==1 && fpyg==0)
            {
                MessageBeep(-1);
                MessageBox(NULL,"No se han ingresado datos en Módulo Hoja de Balance", " MENSAJE DE ERROR",MB_ICONSTOP|MB_OK);
            }
            else
            {
                if (fbalan==0 && fpyg==0)
                {
                    if (pc==0.0 || vn==0.0 || afijo==0.0 || ac==0.0 || m==0.0 || totac==0.0 || patri==0.0 || ep[5]==0.0)
                    {
                        MessageBeep(-1);
                        MessageBox(NULL,"No se puede calcular indices \n No se han ingresado datos necesarios para los cálculos", "Error en Ingreso de datos",MB_OK|MB_ICONHAND);
                    }
                    else
                    {
                        GetApplication()->ExecDialog( new
TIndi(this,"IndiBox"));
                    }
                }
            }
        }
    }
}

```

```
    }  
    }  
}  
  
void TInvestMDI::OpenNive()  
{  
    GetApplication()->ExecDialog( new TNive(this,"NiveBox"));  
    return;  
}  
  
void TInvestMDI::OpenAnual()  
{  
    GetApplication()->ExecDialog( new TAnual(this,"AnualBox"));  
    return;  
}  
  
void TInvestMDI::OpenIrregu()  
{  
    GetApplication()->ExecDialog( new TIrregu(this,"IreguBox"));  
    return;  
}  
  
void TInvestMDI::OpenAcarreo()  
{  
    GetApplication()->ExecDialog( new TAcarreo(this,"AcarrBox"));  
    return;  
}  
  
void TInvestMDI::OpenCupo()  
{  
    GetApplication()->ExecDialog( new TCupo(this,"CupoBox"));  
    return;  
}  
  
void TInvestMDI::OpenBono()  
{  
    GetApplication()->ExecDialog( new TBono(this,"BonoBox"));  
    return;  
}  
  
void TInvestMDI::OpenTasa()  
{  
    GetApplication()->ExecDialog( new TTasa(this,"TasaBox"));  
    return;  
}  
  
void TInvestMDI::OpenStock()  
{  
    GetApplication()->ExecDialog( new TStock(this,"StockBox"));  
    return;  
}  
  
void TInvestMDI::OpenCarte()  
{  
    GetApplication()->ExecDialog( new TCarte(this,"CarteBox"));  
    return;  
}  
  
void TInvestMDI::OpenPerdidas()  
{  
    GetApplication()->ExecDialog( new TPerdidas(this,"PerdiBox"));  
    return;  
}
```



```

void TInvestMDI::OpenRetorno()
{
    GetApplication()->ExecDialog( new TRetorno(this,"RetoBox"));
    return;
}

void TInvestMDI::OpenEstadi()
{
    GetApplication()->ExecDialog( new TEstadi(this,"EstadiBox"));
    return;
}

void TInvestMDI::OpenBalance()
{
    GetApplication()->ExecDialog( new TBalance(this,"FinanBox"));
    return;
}

void TInvestMDI::About()
{
    GetApplication()->ExecDialog( new TAbout(this,"About"));
    return;
}

TInvestMDI::TInvestMDI(LPSTR Name,int MenuId):TMDIFrame(Name,MenuId)
{
    this->ChildMenuPos=CHILD_MENU_POS;
    AssignMenu("PieMenu");
    GetApplication()->ExecDialog( new TDialog(this,"PORTADA"));
    return;
}

TInvestMDI::~TInvestMDI()
{
    DestroyMenu(hMenu);
}

void TInvestMDI::CMExit(RTMessage Msg)
{
    MessageBeep(-1);
    if(MessageBox(NULL,"¿Desea finalizar su sesión en InvestMaster ?"," SALIR DE
INVESTMASTER",MB_ICONQUESTION | MB_YESNO)==IDYES)
    {
        MessageBeep(-1);
        TMDIFrame::CMExit(Msg);
    }
    return;
}

void TInvestMDI::GetWindowClass(WNDCLASS &Win)
{
    TWindow::GetWindowClass(Win);
    Win.hIcon=LoadIcon(GetApplication()->hInstance,"INVEST");
    return;
}

/***** Funciones para el manejo de la aplicacion *****/

void TInvest::InitInstance()
{
    TApplication::InitInstance();
    HAccTable = LoadAccelerators(hInstance, "invest_acc");
    return;
}

```

```

TInvest::TInvest(LPSTR Name,HINSTANCE hInst,HINSTANCE hPrev,LPSTR comando,int
show):TApplication(Name,hInst,hPrev,comando,show)
{
    return;
}

void TInvest::InitMainWindow(void)
{
    WNDCLASS Win;
    MainWindow= new TInvestMDI(PROGRAM_NAME,0);
    return;
}

/***** WinMain *****/
int PASCAL WinMain(HINSTANCE hInst,HINSTANCE hPrev,LPSTR comando,int Show)
{
    TInvest App(PROGRAM_NAME,hInst,hPrev,comando,SW_SHOWMAXIMIZED);
    App.Run();
    return App.Status;
}

```

## Funciones Principales

```

/*****
/*
/*          FUNCION MODELO
/*
/*
*****/

#include "modelo.hpp"

void TModelo::ProcSalir()
{
    EndDialog(HWindow,1);
    return;
}

BOOL TModelo::PedirNombArchivo(BOOL lbTipo,char szExt[3])
{
    OPENFILENAME ofnTemp;
    DWORD Errval; // Error value
    BOOL lbRc;
    char buf[5]; // Error buffer
    char Errstr[50]="GetOpenFileName retornó error #";
    char szTemp[256]; // = "All Files (*.dat)\0*.dat\0";

    sprintf(szTemp,"All Files (*.%s)\0*.%s\0",szExt);
    szTemp[18]='*';
    szTemp[19]='.';
    szTemp[20]=szExt[0];
    szTemp[21]=szExt[1];
    szTemp[22]=szExt[2];
    szTemp[23]='\0';
    szTemp[24]='\0';
    strcpy(szName,"");
    ofnTemp.lStructSize = sizeof( OPENFILENAME );
    ofnTemp.hwndOwner = HWindow; // An invalid hWnd causes non-modality
    ofnTemp.hInstance = 0;
    ofnTemp.lpstrFilter = (LPSTR)szTemp; // See previous note concerning string

```

```

ofnTemp.lpstrCustomFilter = NULL;
ofnTemp.nMaxCustFilter = 0;
ofnTemp.nFilterIndex = 1;
ofnTemp.lpstrFile = (LPSTR)szName;
ofnTemp.nMaxFile = sizeof( szName );
ofnTemp.lpstrFileTitle = NULL;
ofnTemp.nMaxFileTitle = 0;
ofnTemp.lpstrInitialDir = NULL;
ofnTemp.nFileOffset = 0;
ofnTemp.nFileExtension = 0;
ofnTemp.lpstrDefExt = (LPSTR)szExt;
ofnTemp.lCustData = NULL;
ofnTemp.lpfHook = NULL;
ofnTemp.lpTemplateName = NULL;

if( lbTipo ) // TRUE = Grabar
{
    ofnTemp.lpstrTitle = "Grabar Archivo";
    ofnTemp.Flags = OFN_FILEMUSTEXIST | OFN_HIDEREADONLY |
OFN_PATHMUSTEXIST | OFN_OVERWRITEPROMPT;
    lbRc = GetSaveFileName( &ofnTemp );
}
else //FALSE = Abrir
{
    ofnTemp.lpstrTitle = "Abrir Archivo";
    ofnTemp.Flags = OFN_FILEMUSTEXIST | OFN_HIDEREADONLY |
OFN_PATHMUSTEXIST;
    lbRc = GetOpenFileName( &ofnTemp );
}
if (lbRc != TRUE)
{
    Errval=CommDlgExtendedError();
    if(Errval!=0) // 0 value means user selected Cancel
    {
        wsprintf(buf,"%ld",Errval);
        strcat(Errstr,buf);
        MessageBox(HWindow,
Errstr,"WARNING",MB_OK | MB_ICONSTOP);
    }
}
return lbRc;
}

```

```

/*****
/*          FUNCIONES PARA <PICTURE BUTTONS>
/*          *****/
#include "3dbutton.hpp"

```

```

TPButton::TPButton(PWindowsObject padre, int id, WORD bmp_eu, WORD bmp_ed, WORD bmp_d)
:TButton(padre,id,NULL)
{
    hBitmap_eu= 0; /* Bitmap enable up */
    hBitmap_ed= 0; /* Bitmap enable down */
    hBitmap_d = 0; /* Bitmap disable */

    if(bmp_eu)
    {
        hBitmap_eu = LoadBitmap(GetApplication()-
>hInstance,MAKEINTRESOURCE(bmp_eu));
        hBitmap_d = LoadBitmap(GetApplication()-
>hInstance,MAKEINTRESOURCE(bmp_eu));
    }
}

```

```

    }
    if (bmp_ed)
        hBitmap_ed = LoadBitmap(GetApplication()-
>hInstance, MAKEINTRESOURCE(bmp_ed));

    Attr.Style = BS_OWNERDRAW | WS_CHILD | WS_VISIBLE | WS_CLIPSIBLINGS |
WS_GROUP | WS_TABSTOP;
    /* Creación del tipo de letra de los botones */
    hFont = CreateFont(11, 0, 0, 0, FW_HEAVY, 0, 0, 0, 0, 0, 18, "MS Sans Serif");
}

TPButton::TPButton(PWindowsObject padre, int id, LPSTR texto, int x, int y, int w, int h, BOOL
esdefault, WORD bmp_eu, WORD bmp_ed, WORD bmp_d)
: TButton(padre, id, texto, x, y, w, h, esdefault, NULL)
{
    hBitmap_eu = 0; /* Bitmap enable up */
    hBitmap_ed = 0; /* Bitmap enable down */
    hBitmap_d = 0; /* Bitmap disable */

    if (bmp_eu)
    {
        hBitmap_eu = LoadBitmap(GetApplication()-
>hInstance, MAKEINTRESOURCE(bmp_eu));
        hBitmap_d = hBitmap_eu;
    }
    if (bmp_ed)
        hBitmap_ed = LoadBitmap(GetApplication()-
>hInstance, MAKEINTRESOURCE(bmp_ed));

    Attr.Style = BS_OWNERDRAW | WS_CHILD | WS_VISIBLE | WS_CLIPSIBLINGS |
WS_GROUP | WS_TABSTOP;
    /* Creación del tipo de letra de los botones */
    hFont = CreateFont(11, 0, 0, 0, FW_HEAVY, 0, 0, 0, 0, 0, 18, "MS Sans Serif");
}

TPButton::~TPButton()
{
    if (hBitmap_eu)
        DeleteObject(hBitmap_eu);
    if (hBitmap_ed)
        DeleteObject(hBitmap_ed);
    if (hBitmap_d)
        DeleteObject(hBitmap_d);

    if (hFont)
        DeleteObject(hFont);
}

void TPButton::ODADrawEntire(DRAWITEMSTRUCT far &b)
{
    char sz[80];
    BOOL foco = FALSE;
    BOOL estado = TRUE;

    GetWindowText(b.hwndItem, sz, sizeof sz);

    if ( (b.itemState & ODS_FOCUS) == ODS_FOCUS)
        foco = TRUE;
    if ( (b.itemState & ODS_SELECTED) == ODS_SELECTED)
        estado = FALSE;

    DrawCuerpo(b.hDC, b.rclItem);
    DrawBordes(b.hDC, b.rclItem, estado);
    DrawIcon_Text(b.hDC, b.rclItem, estado, foco, sz);
    ReleaseDC(HWindow, b.hDC);
}

```



```

void TPButton::ODAFocus(DRAWITEMSTRUCT far &b)
{
    RECT rcText;
    rcText = b.rcItem;

    rcText.left+=35;
    rcText.top+=5;
    rcText.bottom-=5;
    rcText.right-=5;
    DrawFocusRect(b.hDC,&rcText);
//    ReleaseDC(HWindow,b.hDC);
}

void TPButton::ODASelect(DRAWITEMSTRUCT far &b)
{
    char sz[80];
    BOOL foco = FALSE;
    BOOL estado = TRUE;

    GetWindowText(b.hwndItem,sz,sizeof sz);

    if( (b.itemState & ODS_FOCUS) == ODS_FOCUS)
        foco= TRUE;
    if((b.itemState & ODS_SELECTED) == ODS_SELECTED)
        estado = FALSE;

    DrawCuerpo(b.hDC, b.rcItem);
    DrawBordes(b.hDC,b.rcItem,estado);
    DrawIcon_Text(b.hDC, b.rcItem, estado, foco, sz);
//    ReleaseDC(HWindow,b.hDC);
}

void TPButton::DrawCuerpo(HDC hDC, RECT rc)
{
    HPEN OldPen;
    SetBkMode(hDC,TRANSPARENT);

    FillRect(hDC,&rc,(HBRUSH)GetStockObject(LTGRAY_BRUSH));
    OldPen = SelectObject(hDC,GetStockObject(BLACK_PEN));

    MoveTo(hDC,rc.left+1,rc.top);
    LineTo(hDC,rc.right-1,rc.top);
    MoveTo(hDC,rc.right-1,rc.top+1);
    LineTo(hDC,rc.right-1,rc.bottom-1);
    MoveTo(hDC,rc.right-2,rc.bottom-1);
    LineTo(hDC,rc.left,rc.bottom-1);
    MoveTo(hDC,rc.left,rc.bottom-2);
    LineTo(hDC,rc.left,rc.top);

    SelectObject(hDC, OldPen);
}

void TPButton::DrawBordes(HDC hDC, RECT rc, BOOL estado)
{
    HPEN PenD;
    HPEN OldPen;

    PenD = CreatePen(PS_SOLID,1,RGB(128,128,128));
    if(estado)
    {
        OldPen = SelectObject(hDC,GetStockObject(WHITE_PEN));
        MoveTo(hDC,rc.left+1,rc.bottom-3);
        LineTo(hDC,rc.left+1,rc.top+1);
        LineTo(hDC,rc.right-2,rc.top+1);
        MoveTo(hDC,rc.left+2,rc.bottom-4);
        LineTo(hDC,rc.left+2,rc.top+2);
        LineTo(hDC,rc.right-3,rc.top+2);
    }
}

```



```

        SelectObject(hDC, PenD);
        MoveTo(hDC, rc.left+2, rc.bottom-2);
        LineTo(hDC, rc.right-2, rc.bottom-2);
        LineTo(hDC, rc.right-2, rc.top+1);
        MoveTo(hDC, rc.left+3, rc.bottom-3);
        LineTo(hDC, rc.right-3, rc.bottom-3);
        LineTo(hDC, rc.right-3, rc.top+2);
    }
    else
    {
        OldPen = SelectObject(hDC, PenD);
        MoveTo(hDC, rc.left+1, rc.bottom-2);
        LineTo(hDC, rc.left+1, rc.top+1);
        LineTo(hDC, rc.right-2, rc.top+1);
        MoveTo(hDC, rc.left+2, rc.bottom-2);
        LineTo(hDC, rc.left+2, rc.top+2);
        LineTo(hDC, rc.right-1, rc.top+2);
    }

    SelectObject(hDC, OldPen);
    DeleteObject(PenD);
}

void TPButton::DrawIcon_Text(HDC hDC, RECT rc, BOOL status, BOOL foco, char *sz)
{
    RECT rcText = rc;
    HFONT OldFont;

    OldFont = SelectObject(hDC, hFont);
    if( status) /*Liberado */
    {
        if(hBitmap_eu)
        {
            HDC hMemoryDC;
            HBITMAP hOldBmp;

            BITMAP Bitmap;

            GetObject(hBitmap_eu, sizeof(BITMAP), (LPSTR)&Bitmap);
            hMemoryDC = CreateCompatibleDC(hDC);
            hOldBmp = (HBITMAP)SelectObject(hMemoryDC, hBitmap_eu);

            BitBlt(hDC, rc.left+3, rc.top+3, Bitmap.bmWidth, Bitmap.bmHeight, hMemoryDC, 0, 0, SRCCOPY);
            SelectObject(hMemoryDC, hOldBmp);
            DeleteDC(hMemoryDC);
        }
        rcText.left+=38;
        rcText.top+=8;
        rcText.bottom-=8;
        rcText.right-=6;
        DrawText(hDC, sz, -1, &rcText, DT_RIGHT | DT_BOTTOM | DT_SINGLELINE |
DT_NOCLIP);
        if( foco)
            DrawFocusRect(hDC, &rcText);
    }
    else
    {
        if(hBitmap_ed)
        {
            HDC hMemoryDC;
            HBITMAP hOldBmp;
            BITMAP Bitmap;

            GetObject(hBitmap_ed, sizeof(BITMAP), (LPSTR)&Bitmap);

```

```

        hMemoryDC = CreateCompatibleDC(hDC);
        hOldBmp = (HBITMAP)SelectObject(hMemoryDC,hBitmap_ed);

        BitBlt(hDC,rc.left+6,rc.top+6,Bitmap.bmWidth,Bitmap.bmHeight,hMemoryDC,0,0,SRCCOPY);
        SelectObject(hMemoryDC,hOldBmp);
        DeleteDC(hMemoryDC);
    }

    rcText.left+=40;
    rcText.top+=10;
    rcText.bottom-=6;
    rcText.right-=4;
    DrawText(hDC,sz,-1, &rcText, DT_RIGHT | DT_BOTTOM | DT_SINGLELINE |
DT_NOCLIP);
    if( foco )
        DrawFocusRect(hDC,&rcText);
    }
    SelectObject(hDC, OldFont);
}

```

```

/*-----*/
/*          FUNCIONES PARA <EDIT BUTTONS>          */
/*-----*/

```

```

#include "3dedit.hpp"
#include <stdio.h>
#include <string.h>

#define and &&
#define or ||
/*
T3DEdit::T3DEdit(PWindowsObject padre, int id, LPSTR texto, int x, int y, int w, int h, WORD max,
BOOL readonly)
:TEdit(padre,id,texto,x+2,y+2,w-4,h-4,max,readonly)
{
    Attr.Style = ES_AUTOHSCROLL | ES_RIGHT | WS_CHILD | WS_VISIBLE | WS_TABSTOP;
    if (readonly)
        Attr.Style = Attr.Style | ES_READONLY;
    Punto = FALSE;
}
*/

T3DEdit::T3DEdit(PWindowsObject padre, int id, WORD max, BOOL negativo)
:TEdit(padre, id, max)
{
    // Attr.Style = ES_AUTOHSCROLL | ES_RIGHT | WS_CHILD | WS_VISIBLE | WS_TABSTOP;
    Punto = FALSE;
    Negativo = negativo;
}

void T3DEdit::Habilitar(BOOL estado)
{
    EnableWindow(HWindow,estado);
}

void T3DEdit::WMPaint(RTMessage Msg)
{
    RECT rc;
    HDC hDC,

```

```

    HPEN PenG, PenN, PenB, PenD,
    HPEN OldPen;

    HDC = GetDC(Parent->HWindow);
    rc.left = Attr.X-2;
    rc.top = Attr.Y-2;
    rc.right = Attr.X + Attr.W + 2;
    rc.bottom = Attr.Y + Attr.H + 2;

    PenG = CreatePen(PS_SOLID,1,RGB(192,192,192));
    PenN = CreatePen(PS_SOLID,1,RGB(0,0,0));
    PenB = CreatePen(PS_SOLID,1,RGB(255,255,255));
    PenD = CreatePen(PS_SOLID,1,RGB(128,128,128));

    OldPen = SelectObject(HDC, PenD);
    MoveTo(HDC, rc.left, rc.bottom-2);
    LineTo(HDC, rc.left, rc.top);
    LineTo(HDC, rc.right-1, rc.top);

    SelectObject(HDC, PenN);
    MoveTo(HDC, rc.left+1, rc.bottom-2);
    LineTo(HDC, rc.left+1, rc.top+1);
    LineTo(HDC, rc.right-1, rc.top+1);

    SelectObject(HDC, PenG);
    MoveTo(HDC, rc.left+2, rc.bottom-2);
    LineTo(HDC, rc.right-2, rc.bottom-2);
    LineTo(HDC, rc.right-2, rc.top+1);

    SelectObject(HDC, PenB);
    MoveTo(HDC, rc.left, rc.bottom-1);
    LineTo(HDC, rc.right-1, rc.bottom-1);
    LineTo(HDC, rc.right-1, rc.top);

    SelectObject(HDC, OldPen);
    DeleteObject(PenG);
    DeleteObject(PenB);
    DeleteObject(PenN);
    DeleteObject(PenD);
    ReleaseDC(Parent->HWindow, HDC);

    DefWndProc(Msg);
}

void T3DEdit::Caracter(RTMessage Msg)
{
    char c;
    char cadena[80];
    char cadena2[80];
    int ini, fin, len;
    int i;

    c = (char)Msg.WParam;

    GetText(cadena, 12);
    GetSelection(ini, fin);
    len = GetTextLen();

    // sprintf(cadena2, "Longitud = %d", len);
    // MessageBox(HWindow, cadena2, "Atencion", MB_OK);

    if( (c>='0' and c<='9') or c=='.' or c=='+' or c=='-' or c=='\b' )
    {
        if( c=='.' )
    }

```

```

        if(Punto )
            Msg.WParam = 0;
        else
            Punto = TRUE;
    }
    else if (c=="\b")
    {
        if(ini != fin)
        {
            for(i=ini; i<fin; i++)
                if( cadena[i]=='\n')
                    Punto = FALSE;
        }
        else if(ini>=0 && cadena[ini-1]=='\n')
            Punto = FALSE;
    }
    else if (c=="+" || c=="-")
    {
        if( c=="-" and !Negativo)
            Msg.WParam = 0;
        else
        {
            if(ini != 0)
                Msg.WParam = 0;
            else
            {
                BOOL band=FALSE;
                for( i=0; i<len; i++)
                    if(cadena[i]=='+' or cadena[i]=='-')
                        band=TRUE;

                if( band )
                    Msg.WParam = 0;
            }
        }
    }
}
}
else
    Msg.WParam = 0;

DefWndProc(Msg);
}

void T3DEdit::SetupWindow()
{
    TEdit::SetupWindow();
}

```

## Módulo de Acarreo

```
#include "acarreo.hpp"
```

```
void TAcarreo::Proclmprimir()
```

```
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();

```



```

Tab= Pr.GetTab();
Pr.StartDoc();
Pr.StartPage();
strcpy( strLine, "InvestMaster Para Windows.(Versión
1.0)");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "MODULO: Costo de Acarreo");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "Precio Futuro:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",pf);
TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Precio Disponible:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",pd);
TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Costo de Acarreo:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",cc);
TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Tiempo entre 2 fechas de entrega:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",td);
TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);

```



```

        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con
éxito.",
                                                                "InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
    }

void TAcarreo::ProcGrabar()
{
    if( PedirNombArchivo( TRUE, "aca" ))
    {
        GetDlgItemText(HWindow,F_CS1,Buffer,12);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,F_CS2,Buffer,12);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,F_CS3,Buffer,12);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,F_CS4,Buffer,12);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
    }
}

void TAcarreo::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "aca" ))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS4,Buffer);
    }
    return;
}

void TAcarreo::ProcCalcular()
{
    int cs;
    int n=0;

    pd=cc=td=t=den=pf=prelog1=prelog2=0.00;
    for(cs=1;cs<5;cs++)
    {
        fcs[cs]=0.0;
    }
}

```

```

        GetDlgItemText(HWindow,F_CS1,Buffer,12);
        fcs[1]=atof(Buffer);
        GetDlgItemText(HWindow,F_CS2,Buffer,12);
        fcs[2]=atof(Buffer);
        GetDlgItemText(HWindow,F_CS3,Buffer,12);
        fcs[3]=atof(Buffer);
        GetDlgItemText(HWindow,F_CS4,Buffer,12);
        fcs[4]=atof(Buffer);

for(cs=1;cs<5;cs++)
{
    if(fcs[cs]!=0.0)
        n=n+1;
}

if (n==4)
{
    MessageBox(HWindow,"Ingresó todos los datos"
,"OBSERVACION",MB_OK);
    EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
}
else
{
    if(n==0||n<3)
    {
        MessageBox(HWindow,"Debe ingresar por lo menos 3
datos","OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }
    else
    {
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
        if(fcs[1]==0.0)
        {
            pd=fcs[2]; cc=fcs[3]; td=fcs[4];
            t=td/365.0; den=1.0+cc;
            pf=pd*pow(den,t);
            sprintf(SizeText," El Precio Futuro es: %.4f",pf);
            MessageBox(HWindow,SizeText,"CALCULOS COSTO DE
ACARREO",MB_OK);
        }
        else{
            if(fcs[2]==0.0)
            {
                pf=fcs[1]; cc=fcs[3]; td=fcs[4];
                den=1.0+cc; t=td/365.0;
                pd=pf/pow(den,t);
                sprintf(SizeText," El Precio Disponible es: %.4f",pd);
                MessageBox(HWindow,SizeText,"CALCULOS COSTO DE
ACARREO",MB_OK);
            }
            else{
                if(fcs[3]==0.0)
                {

```

```
    pf=fcs[1]; pd=fcs[2]; td=fcs[4];
    den=pf/pd; t=td/365.0;
    cc=pow(den,1/t)-1.0;
        sprintf(SizeText," El Costo de Acarreo es: %.4f",cc);
        MessageBox(HWindow,SizeText,"CALCULOS COSTO DE
ACARREO",MB_OK);
    }
    else
    {
        pf=fcs[1]; pd=fcs[2]; cc=fcs[3];
        prelog1=log(pf/pd);
        prelog2=log(1.0+cc);
        t=(prelog1/prelog2)*365.0;
        td=t;
            sprintf(SizeText,"El tiempo en días es: %.4f",t);
            MessageBox(HWindow,SizeText,"CALCULOS COSTO DE ACARREO",MB_OK);
    }
}
}
}
}
}
```

## Módulo de Anualidades

```
#include "anual.hpp"

void TAnual::ProclImprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Anualidades");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "Valor Presente:");
```

```

        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%8.2f",vpte);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Valor Futuro:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%8.2f",vf);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Pagos :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%8.2f",pago);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Número de periodos :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%8.2f",np);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Tasa de descuento :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%8.2f",tasaa);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
                    "InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION);
}

```

```
void TAnual::ProcGrabar()
```

```
{
    if( PedirNombArchivo( TRUE, "anu" ))
    {
        GetDlgItemText(HWindow,A_NU1,Buffer,12);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,A_NU2,Buffer,12);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,A_NU3,Buffer,12);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,A_NU4,Buffer,12);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,A_NU5,Buffer,12);
    }
}

```



```

        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,A_NU5,Buffer,12);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
    }
    return;
}

void TAnual::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "anu" ))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_NU1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_NU2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_NU3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_NU4,Buffer);
        GetPrivateProfileString("Datos","dato5","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_NU5,Buffer);
        GetPrivateProfileString("Datos","dato6","",Buffer,12,szName);
    }
    return;
}

void TAnual::ProcCalcular()
{
    int a;
    int n=0;

    pago=npf=min=max=tasaa=fpag=vfut=ftasa = 0.0;
    for(a=1;a<6;a++)
    {
        anu[a]=0.0;
    }

    GetDlgItemText(HWindow,A_NU1,Buffer,12);
    anu[1]=atof(Buffer);
    GetDlgItemText(HWindow,A_NU2,Buffer,12);
    anu[2]=atof(Buffer);
    GetDlgItemText(HWindow,A_NU3,Buffer,12);
    anu[3]=atof(Buffer);
    GetDlgItemText(HWindow,A_NU4,Buffer,12);
    anu[4]=atof(Buffer);
    GetDlgItemText(HWindow,A_NU5,Buffer,12);
    anu[5]=atof(Buffer);

    for(a=1;a<6;a++)
    {
        if(anu[a]!=0.0)
            n=n+1;
    }

    if(n==5)
        MessageBox(HWindow,"Ingresó todos los datos","OBSERVACION",MB_OK);
    else
    {
        if(n==0 || n<3 || n==4)
        {
            MessageBox(HWindow,"Debe ingresar 3 datos","OBSERVACION",MB_OK);
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
        }
        else // n es 3
        {
            if (anu[1]!=0.0 && anu[2]!=0.0)
            {
                MessageBeep(-1);
            }
        }
    }
}

```



```

        MessageBox(NULL,"Debe ingresar valor presente o valor futuro, no los
dos","OBSERVACION",MB_OK|MB_ICONHAND);
    }
    else
    {
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
        EnableMenuItem(HWindow,IDM_GRAF2_1_6, MF_ENABLED);
        if (anu[1]==0.0 && anu[2]==0.0)
        {
            tasaa=anu[5]; np=anu[4]; pago=anu[3];
            for(a=1;a<np+1;a++)
            {
                den=pow((1.0+tasaa),a);
                vpte=vpte+(pago/den);
            } //for
            den=0.0;
            npf=np-1;
            for(a=1;a<np;a++)
            {
                den=pow((1.0+tasaa),a);
                vfut=vfut+(pago*den);
            } //for
            vf=vfut+pago;
            //for

            sprintf(SizeText,"\n El valor presente es:      %.2f \n El valor futuro es:      %.2f \n
",vpte,vf);
            MessageBox(HWindow,SizeText,"RESULTADOS",MB_OK);
            tasa2=tasaa; pago2=pago;
        } //if de anu1 y anu2
        else
        {
            if(anu[1]==0.0) // presente
            {
                if(anu[3]==0.0) // pagos
                {
                    np=anu[4]; tasaa=anu[5]; vfut=anu[2];

                    pago=(vfut/np)-6.6875;
                    do{
                        den=0.0;
                        vfutpre=0.0;
                        pago=pago+1;
                        for(a=1;a<np;a++)
                        {
                            den=pow((1.0+tasaa),a);
                            vfutpre=vfutpre+(pago*den);
                        } //for
                        comp=vfutpre+pago;
                    }while(comp<=vfut);

                    for(a=1;a<np+1;a++)
                    {
                        den=pow((1.0+tasaa),a);
                        vpte=vpte+(pago/den);
                    } //for

                    sprintf(SizeText,"\n El Valor presente es:      %.2f \n Pagos:
%.2f \n ",vpte,pago);
                    MessageBox(HWindow,SizeText,"CALCULOS ANUALIDADES",MB_OK);
                    tasa2=tasaa; pago2=pago;
                } // if pagos
            } else // tasa
            {
                if (anu[5]==0)
                {
                    vfut=anu[2]; pago=anu[3]; np=anu[4]; tasaa=0.0;
                }
            }
        }
    }
}

```

```

        do
            den=0.0; comp=0.0;
            vfutpre=0.0;
            tasaa=tasaa+0.01;
            for(a=1;a<np;a++)
            {
                den=pow((1.0+tasaa),a);
                vfutpre=vfutpre+(pago*den);
            }
            comp=vfutpre+pago;
        }while(comp<=vfut);

        den=0.0;
        vpte=0.0;
        for(a=1;a<np+1;a++)
        {
            den=pow((1.0+tasaa),a);
            vpte=vpte+(pago/den);
        }
        printf(SizeText,"\n El Valor presente es:          %.2f \n La tasa de
descuento es:      %.3f \n",vpte,tasaa);
        MessageBox(HWindow,SizeText, "RESULTADOS",MB_OK);
        tasa2=tasaa; pago2=pago;

    } // if tasa

    else
    {
        if (anu[4]==0.0)
        {
            vfutpre=anu[2]; pago=anu[3]; tasaa=anu[5];
            vf=0.0;
            a=0;
            do
            {
                comp=0.0;
                a=a+1;
                np=a+1;
                den=pow((1.0+tasaa),a);
                vf=vf+(pago*den);
                comp=vf+pago;
            }while(comp<=vfutpre);

            //calculo valor presente/////
            den=0.0;
            vpte=0.0;
            for(a=1;a<np+2;a++)
            {
                den=pow((1.0+tasaa),a);
                vpte=vpte+(pago/den);
            }
            printf(SizeText,"\n El valor presente es:          %.2f \n El número de periodos es:
%.2f \n ",vpte,np);
            MessageBox(HWindow,SizeText, "RESULTADOS",MB_OK);
            tasa2=tasaa; pago2=pago;
        } //if de periodos
    } //else de numero de periodos
    } // if de valor presente
    else
    {
        if(anu[2]==0.0) // futuro
        {
            if(anu[3]==0.0) //pagos
            {
                np=anu[4]; tasaa=anu[5]; vpte=anu[1];

                pago=((vpte)/np)-0.7925;
            }
        }
    }
}

```

```

do{
    den=0.0;
    vprepre=0.0;
    pago=pago+1;
    for(a=1;a<np+1;a++)
    {
        den=pow((1.0+tasaa),a);
        vprepre=vprepre+(pago/den);
    } //for
}while(vprepre<=vpte);
pago=pago-1;
for(a=1;a<np;a++)
{
    den=pow((1.0+tasaa),a);
    vfut=vfut+(pago*den);
} //for
vf=vfut+pago;
sprintf(SizeText,"\n El Valor futuro es:           %.2f \n Pagos:
%.2f \n",vf,pago);
MessageBox(HWindow,SizeText,"CALCULOS ANUALIDADES",MB_OK);
tasaa=tasaa; pago2=pago;
} // if pagos

else // tasa
{
    if (anu[5]==0.0)
    {
        vpte=anu[1]; pago=anu[3]; np=anu[4]; tasaa=0.0;

        do{
            den=0.0;
            vprepre=0.0;
        comp=0.0;
            tasaa=tasaa+0.01;
            for(a=1;a<np+1;a++)
            {
                den=pow((1.0+tasaa),a);
                vprepre=vprepre+(pago/den);
            }
            comp=vprepre;
        }while(comp>=vpte);

        den=0.0;
        vpte=0.0;
        for(a=1;a<np;a++)
        {
            den=pow((1.0+tasaa),a);
            vfut=vfut+(pago*den);
        }
        vf=vfut+pago;
        sprintf(SizeText,"\n El valor futuro es:           %.2f \n La tasa de descuento
es:  %.3f \n",vf,tasaa);
        MessageBox(HWindow,SizeText,"RESULTADOS",MB_OK);
        tasaa=tasaa; pago2=pago;

    } // if tasa

else //numero de periodos
{
    if (anu[4]==0.0)
    {
        vpte=anu[1]; pago=anu[3]; tasaa=anu[5];
        vprepre=0.0; den=0.0;
        a=0;
        do
        {
            comp=0.0;

```

```

        a=a+1;
        np=a-1;
        den=pow((1.0+tasaa),a);
        vprepre=vprepre+(pago/den);

        }while(vprepre<=vpte);

        // calculo valor futuro/////
        den=0.0;
        vf=0.0; vfutpre=0.0;
        for(a=1;a<np;a++)
        {
            den=pow((1.0+tasaa),a);
            vfutpre=vfutpre+(pago*den);
        }
        vf=vfutpre+pago;
        sprintf(SizeText,"\n El valor futuro es:          %.2f \n El número de periodos es:
%.2f \n ",vf,np);
        MessageBox(HWindow,SizeText, "RESULTADOS",MB_OK);
        tasa2=tasaa; pago2=pago;
        //if numero de periodos
        //else de numero de periodos
        //else de tasa
        // if valor futuro
        //else de valor futuro
        // else de valor presente=0
        }
        //n=3
        // else gral
    }
}

```

## Módulo de Balance

```

#include "balance.hpp"

extern int fbalan;

extern double afx[10];
extern double afy[12],cc,in;
extern double ac,afijo,pc,pfijo,totac,topyp,patri; // hoja de Balance

void TBalance::Proclmpimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,

    strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis Financiero (Balance).");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,

    strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
}

```



```

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8),
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8),
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "TOTALES DE BALANCE");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Total Activo Corriente =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",ac);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Total Activo Fijo =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",afijo);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Total Activos ==>");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",totac);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Total Pasivo Corriente =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",pc);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Total Pasivo Fijo =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",pfijo);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Total Patrimonio =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",patri);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Total Pasivos y Patrimonio ==>");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",topyp);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

```



```

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
                                                            "InvestMaster"      para
Windows 1.0", MB_OK | MB_ICONINFORMATION );

        return;
    }

void TBalance::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "bal"))
    {
        GetDlgItemText(HWindow,A_FX1,Buffer,12);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,A_FX2,Buffer,12);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,A_FX3,Buffer,12);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,A_FX4,Buffer,12);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,A_FX5,Buffer,12);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,A_FX6,Buffer,12);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,A_FX7,Buffer,12);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
        GetDlgItemText(HWindow,A_FX8,Buffer,12);
        WritePrivateProfileString("Datos","dato8",Buffer,szName);
        GetDlgItemText(HWindow,A_FX9,Buffer,12);
        WritePrivateProfileString("Datos","dato9",Buffer,szName);
        GetDlgItemText(HWindow,A_FY1,Buffer,12);
        WritePrivateProfileString("Datos","dato10",Buffer,szName);
        GetDlgItemText(HWindow,A_FY2,Buffer,12);
        WritePrivateProfileString("Datos","dato11",Buffer,szName);
        GetDlgItemText(HWindow,A_FY3,Buffer,12);
        WritePrivateProfileString("Datos","dato12",Buffer,szName);
        GetDlgItemText(HWindow,A_FY4,Buffer,12);
        WritePrivateProfileString("Datos","dato13",Buffer,szName);
        GetDlgItemText(HWindow,A_FY5,Buffer,12);
        WritePrivateProfileString("Datos","dato14",Buffer,szName);
        GetDlgItemText(HWindow,A_FY6,Buffer,12);
        WritePrivateProfileString("Datos","dato15",Buffer,szName);
        GetDlgItemText(HWindow,A_FY7,Buffer,12);
        WritePrivateProfileString("Datos","dato16",Buffer,szName);
        GetDlgItemText(HWindow,A_FY8,Buffer,12);
        WritePrivateProfileString("Datos","dato17",Buffer,szName);
        GetDlgItemText(HWindow,A_FY9,Buffer,12);
        WritePrivateProfileString("Datos","dato18",Buffer,szName);
        GetDlgItemText(HWindow,A_FY10,Buffer,12);
        WritePrivateProfileString("Datos","dato19",Buffer,szName);

    }

    return;
}

void TBalance::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "bal"))
    {

```

```

    GetPrivateProfileString("Datos","dato1","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX1,Buffer);
    GetPrivateProfileString("Datos","dato2","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX2,Buffer);
    GetPrivateProfileString("Datos","dato3","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX3,Buffer);
    GetPrivateProfileString("Datos","dato4","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX4,Buffer);
    GetPrivateProfileString("Datos","dato5","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX5,Buffer);
    GetPrivateProfileString("Datos","dato6","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX6,Buffer);
    GetPrivateProfileString("Datos","dato7","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX7,Buffer);
    GetPrivateProfileString("Datos","dato8","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX8,Buffer);
    GetPrivateProfileString("Datos","dato9","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FX9,Buffer);
    GetPrivateProfileString("Datos","dato10","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY1,Buffer);
    GetPrivateProfileString("Datos","dato11","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY2,Buffer);
    GetPrivateProfileString("Datos","dato12","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY3,Buffer);
    GetPrivateProfileString("Datos","dato13","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY4,Buffer);
    GetPrivateProfileString("Datos","dato14","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY5,Buffer);
    GetPrivateProfileString("Datos","dato15","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY6,Buffer);
    GetPrivateProfileString("Datos","dato16","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY7,Buffer);
    GetPrivateProfileString("Datos","dato17","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY8,Buffer);
    GetPrivateProfileString("Datos","dato18","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY9,Buffer);
    GetPrivateProfileString("Datos","dato19","",Buffer,12,szName);
    SetDlgItemText(HWindow,A_FY10,Buffer);
}
return;
}

void TBalance::ProcCalcula()
{
    int fx,fy,n,m=0,p=0;

    char afix1[12],afix2[12],afix3[12],afix4[12],afix5[12],afix6[12],afix7[12],afix8[12],
    char afix9[12];

    char afiy1[12],afiy2[12],afiy3[12],afiy4[12],afiy5[12],afiy6[12],afiy7[12],afiy8[12],
    char afiy9[12],afiy10[12],afiy11[12];

    for(fx=0;fx<10;fx++) {
        afix[fx]=0.0;
    }

    for(fy=0,fy<11,fy++) {
        afiy[fy]=0.0;
    }

    GetDlgItemText(HWindow,A_FX1,afix1,12);
    afix[0]=atof(afix1);
    GetDlgItemText(HWindow,A_FX2,afix2,12);
    afix[1]=atof(afix2);
    GetDlgItemText(HWindow,A_FX3,afix3,12);
    afix[2]=atof(afix3);
    GetDlgItemText(HWindow,A_FX4,afix4,12);

```

```

afx[3]=atof(afix4);
    GetDlgItemText(HWindow,A_FX5,afix5,12);
afx[4]=atof(afix5);
    GetDlgItemText(HWindow,A_FX6,afix6,12);
afx[5]=atof(afix6);
    GetDlgItemText(HWindow,A_FX7,afix7,12);
afx[6]=atof(afix7);
    GetDlgItemText(HWindow,A_FX8,afix8,12);
afx[7]=atof(afix8);
    GetDlgItemText(HWindow,A_FX9,afix9,12);
afx[8]=atof(afix9);

    GetDlgItemText(HWindow,A_FY1,afiy1,12);
afy[0]=atof(afiy1);
    GetDlgItemText(HWindow,A_FY2,afiy2,12);
afy[1]=atof(afiy2);
    GetDlgItemText(HWindow,A_FY3,afiy3,12);
afy[2]=atof(afiy3);
    GetDlgItemText(HWindow,A_FY4,afiy4,12);
afy[3]=atof(afiy4);
    GetDlgItemText(HWindow,A_FY5,afiy5,12);
afy[4]=atof(afiy5);
    GetDlgItemText(HWindow,A_FY6,afiy6,12);
afy[5]=atof(afiy6);
    GetDlgItemText(HWindow,A_FY7,afiy7,12);
afy[6]=atof(afiy7);
    GetDlgItemText(HWindow,A_FY8,afiy8,12);
afy[7]=atof(afiy8);
    GetDlgItemText(HWindow,A_FY9,afiy9,12);
afy[8]=atof(afiy9);
    GetDlgItemText(HWindow,A_FY10,afiy10,12);
afy[9]=atof(afiy10);

ac=0.0;
afijo=0.0;
pc=0.0;
pfijo=0.0;
patri=0.0;
totac=0.0; toppy=0.0;
for(fx=0;fx<9;fx++)
{
    if(afx[fx] != 0.0)
        m=m+1;
}
for(fx=0;fx<10;fx++)
{
    if(afy[fx]!=0.0)
        p=p+1;
}

if ((m==0) && (p==0))
{
    fbalan=1;
    MessageBox(HWindow,"Debe ingresar datos para calcular"," OBSERVACION",MB_OK);
}
else
{
    fbalan=0;

    cc=afx[3];
    in=afx[1];

    for(fx=0;fx<5;fx++)
    {
        if(afx[fx] != 0.0)
            ac=ac+afx[fx];
    }
}

```



```

    }
    for(fx=5;fx<7;fx++)
    {
        if(afx[fx] != 0.0)
            afijo=afijo+afx[fx];
    }
    afijo= afijo+afx[8]-afx[7];
    for(fy=0;fy<4;fy++)
    {
        if(afy[fy] != 0.0)
            pc=pc+afy[fy];
    }
    for(fy=4;fy<7;fy++)
    {
        if(afy[fy] != 0.0)
            pfijo=pfijo+afy[fy];
    }
    for(fy=7;fy<10;fy++)
    {
        if(afy[fy] != 0.0)
            patri=patri+afy[fy];
    }

    topyp=pc+pfijo+patri;
    totac=ac+afijo;
    if(totac!=topyp)
    {
        MessageBeep(-1);
        MessageBox(HWindow,"Error en Ingreso de Datos, el Balance no cuadra",
MENSAJE DE ERROR", MB_ICONSTOP);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }
    else{
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
        sprintf(SizeText,"%6.2f",ac);
        SetDlgItemText(HWindow,ID_AC,SizeText);

        sprintf(SizeText,"%6.2f",afijo);
        SetDlgItemText(HWindow,ID_AFIJO,SizeText);

        sprintf(SizeText,"%6.2f",totac);
        SetDlgItemText(HWindow,ID_TOTAC,SizeText);

        sprintf(SizeText,"%6.2f",pc);
        SetDlgItemText(HWindow,ID_PC,SizeText);

        sprintf(SizeText,"%6.2f",pfijo);
        SetDlgItemText(HWindow,ID_PFIJO,SizeText);

        sprintf(SizeText,"%6.2f",patri);
        SetDlgItemText(HWindow,ID_PATRI,SizeText);

        sprintf(SizeText,"%6.2f",topyp);
        SetDlgItemText(HWindow,ID_TOPYP,SizeText);

        //          sprintf(SizeText,"TOTAL ACTIVO CORRIENTE \t\t %6.2f \n TOTAL
ACTIVO FIJO \t\t %6.2f \n TOTAL ACTIVOS\t\t\t %2f \n \n TOTAL PASIVO CORRIENTE\t\t %2f
\n TOTAL PASIVO FIJO\t\t %2f \n TOTAL PATRIMONIO\t\t %2f \n TOTAL PASIVOS Y\t\t\t
%2f \n PATRIMONIO ",ac,afijo,totac,pc,pfijo,patri,topyp);

        //   MessageBox(HWindow,SizeText,"TOTALES DEL BALANCE",MB_OK);
    }
}
}

```

## Módulo Bono

```

#include "bono.hpp"

void TBono::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis de Bonos de Descuento");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "RESULTADOS OBTENIDOS:");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Valor Facial :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    sprintf(strLine,"%15.2f",fv);
    TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Días de vencimiento :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    sprintf(strLine,"%15.2f",t);
    TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Años de vencimiento :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    sprintf(strLine,"%15.2f",t);
    TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,

strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Precio del dólar:");

```



```

        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
                sprintf(strLine,"%15.2f",precio);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Descuento del dólar :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        sprintf(strLine,"%15.2f",disc);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Rendimiento del descuento :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        sprintf(strLine,"%15.2f",d);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Valor de un punto base :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        sprintf(strLine,"%15.2f",vb);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Equivalente del interés simple :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        sprintf(strLine,"%15.2f",EBY);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Equivalente de rendimiento del bono :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        sprintf(strLine,"%15.2f",inte);
        TextOut( hPr, 15*Tab, CurrentLine*LineSpace, strLine,
        strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
                    "InvestMaster", para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TBono::ProcGrabar()
{
    if( PedirNombArchivo( TRUE, "bon" ))
    {
        GetDlgItemText(HWindow,A_NUI,Buffer,10);
        WritePrivateProfileString("Datos","anu1",Buffer,szName);

```

```

        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        WritePrivateProfileString("Datos","anu2",Buffer,szName);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        WritePrivateProfileString("Datos","anu3",Buffer,szName);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        WritePrivateProfileString("Datos","anu4",Buffer,szName);
        GetDlgItemText(HWindow,A_NU5,Buffer,10);
        WritePrivateProfileString("Datos","anu5",Buffer,szName);

    }
    return;
}

void TBono::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "bon" ))
    {
        GetPrivateProfileString("Datos","anu1","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU1,Buffer);
        GetPrivateProfileString("Datos","anu2","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU2,Buffer);
        GetPrivateProfileString("Datos","anu3","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU3,Buffer);
        GetPrivateProfileString("Datos","anu4","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU4,Buffer);
        GetPrivateProfileString("Datos","anu5","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU5,Buffer);
    }
}

void TBono::ProcCalcular()
{
    int a;
    int n=0;

    fv=nump=t=precio=disc=d=vb=vbnum=vbden=den1=num1=den=num=inte=tanto=por=EBY =
0.0;
    for(a=1;a<6;a++)
    {
        anu[a]=0.0;

        GetDlgItemText(HWindow,A_NU1,Buffer,10);
        anu[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        anu[2]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        anu[3]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        anu[4]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU5,Buffer,10);
        anu[5]=atof(Buffer);

        for(a=1;a<6;a++)
        {
            if(anu[a]!=0.0)
                n=n+1;
        }

        if(n==5)
        {
            MessageBox(HWindow,"Ingresó todos los datos","OBSERVACION",MB_OK);
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
        }
        else
        {
            if(n==4)

```

```

    {
        MessageBox(HWindow,"Debe ingresar sólo 3 datos","OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }
else
    {
        if(n==0 || n<3 || (anu[1]==0.0 && anu[2]==0.0))
        {
            MessageBox(HWindow,"Debe ingresar los 2 datos iniciales y cualquiera de los
siguientes","OBSERVACION",MB_OK);
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
        }
        else // n es 3
        {
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
            if (anu[1]!=0.0 && anu[2]!=0.0 && anu[3]!=0.0)
            {
                fv=anu[1]; t=anu[2]; precio=anu[3];
                num=360*(fv-precio); den=fv*t;
                d=num/den;
                disc=fv-precio;
                num1=365.0*d;
                den1=360.0-d*t;
                EBY=num1/den1;

                vnum=(fv*0.01)/100.0;
                vden=t/360.0;
                vb=vnum*vden;

                tanto=1.0-(fv/precio);
                por=tanto/100.0;
                inte=EBY+por;
                sprintf(SizeText,"Descuento dólar \t \t \t %.2f \n Rendimiento descuento \t \t %.5f \n Valor
de un punto base \t \t %.2f \n Equivalente interés simple \t \t %.5f \n Equivalente rendimiento
bono \t \t %.5f",disc,d,vb,EBY,inte);
                MessageBox(HWindow,SizeText,"RESULTADOS BONOS DE DESCUENTO",MB_OK);
            }
            else
            {
                if (anu[1]!=0.0 && anu[2]!=0.0 && anu[4]!=0.0)
                {
                    fv=anu[1]; t=anu[2]; disc=anu[4];
                    precio=fv-disc;
                    num=360*(fv-precio); den=fv*t;
                    d=num/den;

                    num1=365.0*d;
                    den1=360.0-d*t;
                    EBY=num1/den1;

                    vnum=(fv*0.01)/100.0;
                    vden=t/360.0;
                    vb=vnum*vden;

                    tanto=1.0-(fv/precio);
                    por=tanto/100.0;
                    inte=EBY+por;
                    sprintf(SizeText,"Precio del dólar \t \t \t %.2f \n Rendimiento descuento \t \t %.5f \n Valor
de un punto base \t \t %.2f \n Equivalente interés simple \t \t %.5f \n Equivalente rendimiento
bono \t \t %.5f",precio,d,vb,EBY,inte);
                    MessageBox(HWindow,SizeText,"RESULTADOS BONOS DE DESCUENTO",MB_OK);
                }
            }
        }
        else
        {
            fv=anu[1]; t=anu[2]; d=anu[5];

            nump=(d*t)/360.0;

```



```
num=1-nump;
precio=fv*num;
disc=fv*nump;
num1=365.0*d;
den1=360.0-d*t;
EBY=num1/den1;

vbnun=(fv*0.01)/100.0;
vbden=t/360.0;
vb=vbnun*vbden;

tanto=1.0-(fv/precio);
por=tanto/100.0;
inte=EBY+por;
sprintf(SizeText,"Precio del dólar \t \t \t %.2f \n Descuento dólar \t \t \t %.2f \n Valor de
un punto base \t \t \t %.2f \n Equivalente interés simple \t \t \t %.5f \n Equivalente rendimiento bono
\t %.5f",precio,disc,vb,EBY,inte);
MessageBox(HWindow,SizeText,"RESULTADOS BONOS DE DESCUENTO",MB_OK);
}
}
}
}
}
}
}
```

## Módulo Cartera

```
# include "cartera.hpp"

void TCarte::ProcGrabar()
{
    if( PedirNombArchivo( TRUE,"car" ))
    {
        GetDlgItemText(HWindow,A_E1,Buffer,8);
        WritePrivateProfileString("Datos","ca1",Buffer,szName);
        GetDlgItemText(HWindow,A_E2,Buffer,8);
        WritePrivateProfileString("Datos","ca2",Buffer,szName);
        GetDlgItemText(HWindow,A_E3,Buffer,8);
        WritePrivateProfileString("Datos","ca3",Buffer,szName);
        GetDlgItemText(HWindow,A_E4,Buffer,8);
        WritePrivateProfileString("Datos","ca4",Buffer,szName);
        GetDlgItemText(HWindow,A_E5,Buffer,8);
        WritePrivateProfileString("Datos","ca5",Buffer,szName);
        GetDlgItemText(HWindow,A_E6,Buffer,8);
        WritePrivateProfileString("Datos","ca6",Buffer,szName);
        GetDlgItemText(HWindow,A_E7,Buffer,8);
        WritePrivateProfileString("Datos","ca7",Buffer,szName);
        GetDlgItemText(HWindow,A_E8,Buffer,8);
        WritePrivateProfileString("Datos","ca8",Buffer,szName);
        GetDlgItemText(HWindow,A_E9,Buffer,8);
        WritePrivateProfileString("Datos","ca9",Buffer,szName);
        GetDlgItemText(HWindow,A_E10,Buffer,8);
        WritePrivateProfileString("Datos","ca10",Buffer,szName);
        GetDlgItemText(HWindow,A_E11,Buffer,8);
        WritePrivateProfileString("Datos","ca11",Buffer,szName);
        GetDlgItemText(HWindow,A_E12,Buffer,8);
        WritePrivateProfileString("Datos","ca12",Buffer,szName);
        GetDlgItemText(HWindow,A_E13,Buffer,8);
        WritePrivateProfileString("Datos","ca13",Buffer,szName);
        GetDlgItemText(HWindow,A_E14,Buffer,8);
        WritePrivateProfileString("Datos","ca14",Buffer,szName);
        GetDlgItemText(HWindow,A_E15,Buffer,8);
        WritePrivateProfileString("Datos","ca15",Buffer,szName);
        GetDlgItemText(HWindow,A_EY1,Buffer,8);
        WritePrivateProfileString("Datos","ca16",Buffer,szName);
    }
}
```

```

    GetDlgItemText(HWindow,A_EY2,Buffer,8);
    WritePrivateProfileString("Dados","ca17",Buffer,szName);
    GetDlgItemText(HWindow,A_EY3,Buffer,8);
    WritePrivateProfileString("Dados","ca18",Buffer,szName);
    GetDlgItemText(HWindow,A_EY4,Buffer,8);
    WritePrivateProfileString("Dados","ca19",Buffer,szName);
    GetDlgItemText(HWindow,A_EY5,Buffer,8);
    WritePrivateProfileString("Dados","ca20",Buffer,szName);
    GetDlgItemText(HWindow,A_EY6,Buffer,8);
    WritePrivateProfileString("Dados","ca21",Buffer,szName);
    GetDlgItemText(HWindow,A_EY7,Buffer,8);
    WritePrivateProfileString("Dados","ca22",Buffer,szName);
    GetDlgItemText(HWindow,A_EY8,Buffer,8);
    WritePrivateProfileString("Dados","ca23",Buffer,szName);
    GetDlgItemText(HWindow,A_EY9,Buffer,8);
    WritePrivateProfileString("Dados","ca24",Buffer,szName);
    GetDlgItemText(HWindow,A_EY10,Buffer,8);
    WritePrivateProfileString("Dados","ca25",Buffer,szName);
    GetDlgItemText(HWindow,A_EY11,Buffer,8);
    WritePrivateProfileString("Dados","ca26",Buffer,szName);
    GetDlgItemText(HWindow,A_EY12,Buffer,8);
    WritePrivateProfileString("Dados","ca27",Buffer,szName);
    GetDlgItemText(HWindow,A_EY13,Buffer,8);
    WritePrivateProfileString("Dados","ca28",Buffer,szName);
    GetDlgItemText(HWindow,A_EY14,Buffer,8);
    WritePrivateProfileString("Dados","ca29",Buffer,szName);
    GetDlgItemText(HWindow,A_EY15,Buffer,8);
    WritePrivateProfileString("Dados","ca30",Buffer,szName);
}
return;
}

void TCarte::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "car" ))
    {
        GetPrivateProfileString("Dados","ca1","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E1,Buffer);
        GetPrivateProfileString("Dados","ca2","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E2,Buffer);
        GetPrivateProfileString("Dados","ca3","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E3,Buffer);
        GetPrivateProfileString("Dados","ca4","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E4,Buffer);
        GetPrivateProfileString("Dados","ca5","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E5,Buffer);
        GetPrivateProfileString("Dados","ca6","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E6,Buffer);
        GetPrivateProfileString("Dados","ca7","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E7,Buffer);
        GetPrivateProfileString("Dados","ca8","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E8,Buffer);
        GetPrivateProfileString("Dados","ca9","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E9,Buffer);
        GetPrivateProfileString("Dados","ca10","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E10,Buffer);
        GetPrivateProfileString("Dados","ca11","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E11,Buffer);
        GetPrivateProfileString("Dados","ca12","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E12,Buffer);
        GetPrivateProfileString("Dados","ca13","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E13,Buffer);
        GetPrivateProfileString("Dados","ca14","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E14,Buffer);
        GetPrivateProfileString("Dados","ca15","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_E15,Buffer);
        GetPrivateProfileString("Dados","ca16","",Buffer,8,szName);
        SetDlgItemText(HWindow,A_EY1,Buffer);
    }
}

```



```

    GetPrivateProfileString("Datos","ca17","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY2,Buffer);
    GetPrivateProfileString("Datos","ca18","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY3,Buffer);
    GetPrivateProfileString("Datos","ca19","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY4,Buffer);
    GetPrivateProfileString("Datos","ca20","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY5,Buffer);
    GetPrivateProfileString("Datos","ca21","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY6,Buffer);
    GetPrivateProfileString("Datos","ca22","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY7,Buffer);
    GetPrivateProfileString("Datos","ca23","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY8,Buffer);
    GetPrivateProfileString("Datos","ca24","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY9,Buffer);
    GetPrivateProfileString("Datos","ca25","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY10,Buffer);
    GetPrivateProfileString("Datos","ca26","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY11,Buffer);
    GetPrivateProfileString("Datos","ca27","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY12,Buffer);
    GetPrivateProfileString("Datos","ca28","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY13,Buffer);
    GetPrivateProfileString("Datos","ca29","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY14,Buffer);
    GetPrivateProfileString("Datos","ca30","",Buffer,8,szName);
    SetDlgItemText(HWindow,A_EY15,Buffer);
}
return;
}

void TCarte::ProcImprimir()
{
    HANDLE hLibrary;

    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();

    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis de Cartera");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "RESULTADOS OBTENIDOS.");

```

```

    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "E(R)");
    TextOut( hPr, 8*Tab, CurrentLine*LineSpace, strLine,
    strcpy( strLine, "Desviación Estándar");
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
    strcpy( strLine, "Pesos");
    TextOut( hPr, 20*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "Periodo 1");
    TextOut( hPr,Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",er[1]);
    TextOut( hPr, 4*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",sd[1]);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",peso[1]);
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine++;
    strcpy( strLine, "Periodo 2");
    TextOut( hPr,Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",er[2]);
    TextOut( hPr, 4*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",sd[2]);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",peso[2]);
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine++;
    strcpy( strLine, "Periodo 3");
    TextOut( hPr,Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",er[3]);
    TextOut( hPr, 4*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",sd[3]);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",peso[3]);
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "Retorno esperado de cartera :");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",retorno);
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "Desviación estándar de cartera :");

```

```

        TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%15.2f",desv);
        TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,25*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
"InvestMaster" para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TCarte::ProcCalcular()
{
int c,j,n,i,e=0,des=0,pes=0;

        mul2=desv=retorno = 0.00;
for(c=1;c<6;c++)
{
        er[c]=0.0;
        sd[c]=0.0;
        peso[c]=0.0;
}

for(c=1;c<6;c++)
{
for(j=1;j<6;j++)
{
        corr[c][j]=0.0;
        sigma[c][j]=0.0;
}
}

        GetDlgItemText(HWindow,A_E1,Buffer,8);
        er[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_E2,Buffer,8);
        er[2]=atof(Buffer);
        GetDlgItemText(HWindow,A_E3,Buffer,8);
        er[3]=atof(Buffer);
        GetDlgItemText(HWindow,A_E4,Buffer,8);
        er[4]=atof(Buffer);
        GetDlgItemText(HWindow,A_E5,Buffer,8);
        er[5]=atof(Buffer);
        GetDlgItemText(HWindow,A_E6,Buffer,8);
        sd[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_E7,Buffer,8);
        sd[2]=atof(Buffer);
        GetDlgItemText(HWindow,A_E8,Buffer,8);
        sd[3]=atof(Buffer);
        GetDlgItemText(HWindow,A_E9,Buffer,8);
        sd[4]=atof(Buffer);
        GetDlgItemText(HWindow,A_E10,Buffer,8);
        sd[5]=atof(Buffer);
        GetDlgItemText(HWindow,A_E11,Buffer,8);
        peso[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_E12,Buffer,8);

```



```

    peso[2]=atof(Buffer);
    GetDlgItemText(HWindow,A_E13,Buffer,8);
    peso[3]=atof(Buffer);
    GetDlgItemText(HWindow,A_E14,Buffer,8);
    peso[4]=atof(Buffer);
    GetDlgItemText(HWindow,A_E15,Buffer,8);
    peso[5]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY1,Buffer,8);
    corr[1][1]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY2,Buffer,8);
    corr[1][2]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY3,Buffer,8);
    corr[1][3]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY4,Buffer,8);
    corr[1][4]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY5,Buffer,8);
    corr[1][5]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY6,Buffer,8);
    corr[2][2]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY7,Buffer,8);
    corr[2][3]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY8,Buffer,8);
    corr[2][4]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY9,Buffer,8);
    corr[2][5]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY10,Buffer,8);
    corr[3][3]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY11,Buffer,8);
    corr[3][4]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY12,Buffer,8);
    corr[3][5]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY13,Buffer,8);
    corr[4][4]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY14,Buffer,8);
    corr[4][5]=atof(Buffer);
    GetDlgItemText(HWindow,A_EY15,Buffer,8);
    corr[5][5]=atof(Buffer);

    for(c=1;c<6;c++)
    {
        if(er[c]!=0.0)
            e=e+1;
        if(sd[c]!=0.0)
            des=des+1;
        if(peso[c]!=0.0)
            pes=pes+1;
    }
    if(e==0 || des==0 || pes==0 || e!=des || e!=pes || pes!=des)
    {
        MessageBox(HWindow,"Debe ingresar datos para calcular","OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }
    else
    {
        if (er[2]==0)
        {
            MessageBox(HWindow,"Debe ingresar mínimo 2 stocks para calcular","OBSERVACION",MB_OK);
            EnableWindow(GetDlgItem(HWindow,IDM_PRINT), FALSE);
        }
        else
        {
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
            n=e;corr[2][1]=corr[1][2]; corr[3][1]=corr[1][3];corr[3][2]=corr[2][3];
            corr[4][1]=corr[1][4];corr[4][2]=corr[2][4];corr[4][3]=corr[3][4];
            corr[5][1]=corr[1][5];corr[5][2]=corr[2][5];corr[5][3]=corr[3][5];corr[5][4]=corr[4][5];
            for(i=1;i<n+1;i++)

```

```

    }
    //desviacion estandar
    mul2=0.0;
    for(i=1;i<n+1;i++)
    {
        for(j=1;j<n+1;j++)
            mul2=mul2+peso[i]*peso[j]*sigma[i][j];
    }

    desv=sqrt(fabs(mul2));

    //retorno
    for(j=1;j<n+1;j++)
        retorno=retorno+peso[j]*er[j];
    sprintf(SizeText,"\n Retorno esperado de cartera:      %.4f \n \n Desviación Estándar de
cartera:      %.4f \n",retorno,desv);
    MessageBox(HWindow,SizeText,"CALCULOS DE CARTERA",MB_OK);
}
}
}

```

## Módulo de Cupones

```

#include "cupo.hpp"

void TCupo::Proclmpimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis de Bonos/Cupones");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "RESULTADOS OBTENIDOS:");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Valor Facial :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"% 15 2f",bon);

```



```

strlen(strLine));
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "Pagos anuales :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",anu[2]);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Años de vencimiento :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",t);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Tasa anual (%):");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
    sprintf(strLine,"%15.2f",tasa);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Precio :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",precio);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Duración :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",x/precio);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Retorno al vencimiento :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",ret);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    strcpy( strLine, "Retorno aproximado al");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine++;
    strcpy( strLine, "vencimiento :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.2f",aprox);
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    CurrentLine+=2;
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);

```

```

        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impressão finalizada com êxito.",
                    "InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
    }

void TCupo::ProcGrabar()
{
    if( PedirNombArchivo( FALSE, "cup" ) )
    {
        GetDlgItemText(HWindow,A_NU1,Buffer,10);
        WritePrivateProfileString("Datos","anu1",Buffer,szName);
        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        WritePrivateProfileString("Datos","anu2",Buffer,szName);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        WritePrivateProfileString("Datos","anu3",Buffer,szName);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        WritePrivateProfileString("Datos","anu4",Buffer,szName);
        GetDlgItemText(HWindow,A_NU5,Buffer,10);
        WritePrivateProfileString("Datos","anu5",Buffer,szName);
        GetDlgItemText(HWindow,A_NU6,Buffer,10);
        WritePrivateProfileString("Datos","anu6",Buffer,szName);
        GetDlgItemText(HWindow,A_NU7,Buffer,10);
        WritePrivateProfileString("Datos","anu7",Buffer,szName);
    }
}

void TCupo::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "cup" ) )
    {
        GetPrivateProfileString("Datos","anu1","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU1,Buffer);
        GetPrivateProfileString("Datos","anu2","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU2,Buffer);
        GetPrivateProfileString("Datos","anu3","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU3,Buffer);
        GetPrivateProfileString("Datos","anu4","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU4,Buffer);
        GetPrivateProfileString("Datos","anu5","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU5,Buffer);
        GetPrivateProfileString("Datos","anu6","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU6,Buffer);
        GetPrivateProfileString("Datos","anu7","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU7,Buffer);
    }

    return;
}

void TCupo::ProcCalcular()
{
    int a;
    int n=0,i,w;

    x=c=bon=t=tasa=b=u=v=cons=aprox=y=d=r=z=ret=precio=0.0;
    for(a=1;a<6;a++)
    {
        anu[a]=0.0;
    }

    GetDlgItemText(HWindow,A_NU1,Buffer,10);

```

```

        anu[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        anu[2]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        anu[3]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        anu[4]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU5,Buffer,10);
        anu[5]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU6,Buffer,10);
        anu[6]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU7,Buffer,10);
        anu[7]=atof(Buffer);

    for(a=1;a<8;a++)
    {
        if(anu[a]!=0.0)
            n=n+1;
    }

    if(n==7)
    {
        MessageBox(HWindow,"Ingresó todos los datos","OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }

    else
    {
        if(n==0 || n<6)
        {
            MessageBox(HWindow,"Debe ingresar 6 datos","OBSERVACION",MB_OK);
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
        }
        else
        {
            EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
            if (anu[6]==0.0)
            {
                tasa=anu[5]; bon=anu[1]; t=anu[3]; r=anu[7];
                c=bon*tasa/100.0;
                cons=1.0+r;
                w=int(t);
                for(i=1;i<w;i++)
                {
                    a=a+c/pow(cons,i);
                    x=x+(c/pow(cons,i))*i;
                }
                d=c+bon;
                b=a+d/pow(cons,t);
                y=x+(c/pow(cons,t))*t;
                z=c+(bon-b)/t;
                u=(bon+b)/2.0;
                aprox=z/u;
                sprintf(SizeText,"Precio: %.2f \n Duración: %.2f \n Retorno aproximado al vencimiento:
                %.3f",b,y/b,aprox);
                MessageBox(HWindow,SizeText,"RESULTADOS",MB_OK);
            }
            else
            {
                if (anu[7]==0.0)
                {
                    tasa=anu[5]; bon=anu[1]; t=anu[3]; precio=anu[6]; x=0.0;
                    c=bon*tasa/100.0;
                    z=c+(bon-precio)/t;
                    u=(bon+precio)/2.0;
                    v=(bon+2*precio)/3.0;
                    aprox=z/u;
                    ret=z/v;
                    w=int(t);

```

```
        for(i=1;i<w+1;i++)
        {
            x=x+(c/pow(1+aprox,i))*i;
        }

        sprintf(SizeText,"Retorno al vencimiento: %.3f \n Duración: %.2f \n Retorno aproximado al
vencimiento: %.3f",ret,x/precio,aprox);
        MessageBox(HWindow,SizeText,"RESULTADOS",MB_OK);
    } //if anu7==0
    } //else anu7
}
}
}
}
```

## Módulo de Cálculos Estadísticos

```
#include "estadi.hpp"

void TEstadi::ProclImprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis y Cálculos Estadísticos.");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "RESULTADOS OBTENIDOS:");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Media de A =");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.2f",ux);
    TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
}
```



```

CurrentLine++;
strcpy( strLine, "Media de B =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.2f",uy);
TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Desviación Estándar A =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",o1x);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Desviación Estándar B =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",o1y);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, " Varianza de A =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",odx);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, " Varianza de B =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",o2y);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Skewness de A =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",SKEWx);
TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Skewness de B =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",SKEWy);
TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Kurtosis de A =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",KURTx);
TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Kurtosis de B =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.6f",KURTy);

```



```

        TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Covarianza =");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.6f",COVxy);
        TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine++;
        strcpy( strLine, "Correlación =");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.6f",CORRxy);
        TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
        "InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );

    return;
}

void TEstadi::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "est"))
    {
        GetDlgItemText(HWindow,A_E1,Buffer,12);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,A_E2,Buffer,12);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,A_E3,Buffer,12);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,A_E4,Buffer,12);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,A_E5,Buffer,12);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,A_E6,Buffer,12);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,A_E7,Buffer,12);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
        GetDlgItemText(HWindow,A_E8,Buffer,12);
        WritePrivateProfileString("Datos","dato8",Buffer,szName);
        GetDlgItemText(HWindow,A_E9,Buffer,12);
        WritePrivateProfileString("Datos","dato9",Buffer,szName);
        GetDlgItemText(HWindow,A_E10,Buffer,12);
        WritePrivateProfileString("Datos","dato10",Buffer,szName);
        GetDlgItemText(HWindow,A_E11,Buffer,12);
        WritePrivateProfileString("Datos","dato11",Buffer,szName);
        GetDlgItemText(HWindow,A_E12,Buffer,12);
        WritePrivateProfileString("Datos","dato12",Buffer,szName);
        GetDlgItemText(HWindow,A_E13,Buffer,12);
        WritePrivateProfileString("Datos","dato13",Buffer,szName);
        GetDlgItemText(HWindow,A_E14,Buffer,12);
        WritePrivateProfileString("Datos","dato14",Buffer,szName);
        GetDlgItemText(HWindow,A_E15,Buffer,12);
        WritePrivateProfileString("Datos","dato15",Buffer,szName);
    }
}

```

```

        GetDlgItemText(HWindow,A_E16,Buffer,12);
        WritePrivateProfileString("Datos","dato16",Buffer,szName);
        GetDlgItemText(HWindow,A_E17,Buffer,12);
        WritePrivateProfileString("Datos","dato17",Buffer,szName);
        GetDlgItemText(HWindow,A_E18,Buffer,12);
        WritePrivateProfileString("Datos","dato18",Buffer,szName);
        GetDlgItemText(HWindow,A_E19,Buffer,12);
        WritePrivateProfileString("Datos","dato19",Buffer,szName);
        GetDlgItemText(HWindow,A_E20,Buffer,12);
        WritePrivateProfileString("Datos","dato20",Buffer,szName);

        GetDlgItemText(HWindow,A_EY1,Buffer,12);
        WritePrivateProfileString("Datos","dato21",Buffer,szName);
        GetDlgItemText(HWindow,A_EY2,Buffer,12);
        WritePrivateProfileString("Datos","dato22",Buffer,szName);
        GetDlgItemText(HWindow,A_EY3,Buffer,12);
        WritePrivateProfileString("Datos","dato23",Buffer,szName);
        GetDlgItemText(HWindow,A_EY4,Buffer,12);
        WritePrivateProfileString("Datos","dato24",Buffer,szName);
        GetDlgItemText(HWindow,A_EY5,Buffer,12);
        WritePrivateProfileString("Datos","dato25",Buffer,szName);
        GetDlgItemText(HWindow,A_EY6,Buffer,12);
        WritePrivateProfileString("Datos","dato26",Buffer,szName);
        GetDlgItemText(HWindow,A_EY7,Buffer,12);
        WritePrivateProfileString("Datos","dato27",Buffer,szName);
        GetDlgItemText(HWindow,A_EY8,Buffer,12);
        WritePrivateProfileString("Datos","dato28",Buffer,szName);
        GetDlgItemText(HWindow,A_EY9,Buffer,12);
        WritePrivateProfileString("Datos","dato29",Buffer,szName);
        GetDlgItemText(HWindow,A_EY10,Buffer,12);
        WritePrivateProfileString("Datos","dato30",Buffer,szName);
        GetDlgItemText(HWindow,A_EY11,Buffer,12);
        WritePrivateProfileString("Datos","dato31",Buffer,szName);
        GetDlgItemText(HWindow,A_EY12,Buffer,12);
        WritePrivateProfileString("Datos","dato32",Buffer,szName);
        GetDlgItemText(HWindow,A_EY13,Buffer,12);
        WritePrivateProfileString("Datos","dato33",Buffer,szName);
        GetDlgItemText(HWindow,A_EY14,Buffer,12);
        WritePrivateProfileString("Datos","dato34",Buffer,szName);
        GetDlgItemText(HWindow,A_EY15,Buffer,12);
        WritePrivateProfileString("Datos","dato35",Buffer,szName);
        GetDlgItemText(HWindow,A_EY16,Buffer,12);
        WritePrivateProfileString("Datos","dato36",Buffer,szName);
        GetDlgItemText(HWindow,A_EY17,Buffer,12);
        WritePrivateProfileString("Datos","dato37",Buffer,szName);
        GetDlgItemText(HWindow,A_EY18,Buffer,12);
        WritePrivateProfileString("Datos","dato38",Buffer,szName);
        GetDlgItemText(HWindow,A_EY19,Buffer,12);
        WritePrivateProfileString("Datos","dato39",Buffer,szName);
        GetDlgItemText(HWindow,A_EY20,Buffer,12);
        WritePrivateProfileString("Datos","dato40",Buffer,szName);

    }
    return;
}

void TEstadi::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "est"))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_E1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_E2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_E3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_E4,Buffer);
    }
}

```



```

GetPrivateProfileString("Datos","dato5","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E5,Buffer);
GetPrivateProfileString("Datos","dato6","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E6,Buffer);
GetPrivateProfileString("Datos","dato7","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E7,Buffer);
GetPrivateProfileString("Datos","dato8","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E8,Buffer);
GetPrivateProfileString("Datos","dato9","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E9,Buffer);
GetPrivateProfileString("Datos","dato10","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E10,Buffer);
GetPrivateProfileString("Datos","dato11","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E11,Buffer);
GetPrivateProfileString("Datos","dato12","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E12,Buffer);
GetPrivateProfileString("Datos","dato13","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E13,Buffer);
GetPrivateProfileString("Datos","dato14","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E14,Buffer);
GetPrivateProfileString("Datos","dato15","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E15,Buffer);
GetPrivateProfileString("Datos","dato16","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E16,Buffer);
GetPrivateProfileString("Datos","dato17","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E17,Buffer);
GetPrivateProfileString("Datos","dato18","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E18,Buffer);
GetPrivateProfileString("Datos","dato19","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E19,Buffer);
GetPrivateProfileString("Datos","dato20","",Buffer,12,szName);
SetDlgItemText(HWindow,A_E20,Buffer);

GetPrivateProfileString("Datos","dato21","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY1,Buffer);
GetPrivateProfileString("Datos","dato22","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY2,Buffer);
GetPrivateProfileString("Datos","dato23","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY3,Buffer);
GetPrivateProfileString("Datos","dato24","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY4,Buffer);
GetPrivateProfileString("Datos","dato25","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY5,Buffer);
GetPrivateProfileString("Datos","dato26","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY6,Buffer);
GetPrivateProfileString("Datos","dato27","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY7,Buffer);
GetPrivateProfileString("Datos","dato28","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY8,Buffer);
GetPrivateProfileString("Datos","dato29","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY9,Buffer);
GetPrivateProfileString("Datos","dato30","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY10,Buffer);
GetPrivateProfileString("Datos","dato31","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY11,Buffer);
GetPrivateProfileString("Datos","dato32","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY12,Buffer);
GetPrivateProfileString("Datos","dato33","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY13,Buffer);
GetPrivateProfileString("Datos","dato34","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY14,Buffer);
GetPrivateProfileString("Datos","dato35","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY15,Buffer);
GetPrivateProfileString("Datos","dato36","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY16,Buffer);
GetPrivateProfileString("Datos","dato37","",Buffer,12,szName);
SetDlgItemText(HWindow,A_EY17,Buffer);
GetPrivateProfileString("Datos","dato38","",Buffer,12,szName);

```

```

        SetDlgItemText(HWindow,A_EY18,Buffer);
        GetPrivateProfileString("Datos","dato39","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_EY19,Buffer);
        GetPrivateProfileString("Datos","dato40","",Buffer,12,szName);
        SetDlgItemText(HWindow,A_EY20,Buffer);
    }
    return;
}

void TEstadi::ProcCalcular()
{
    double aestx[20];
    double aesty[20];
    int v,w,n=0,m=0;

    char arr1[12],arr2[12],arr3[12],arr4[12],arr5[12],arr6[12],arr7[12],arr8[12];
    char arr9[12],arr10[12],arr11[12],arr12[12],arr13[12],arr14[12],arr15[12],arr16[12];
    char arr17[12],arr18[12],arr19[12],arr20[12];

    char arry1[12],arry2[12],arry3[12],arry4[12],arry5[12],arry6[12],arry7[12],arry8[12];
    char arry9[12],arry10[12],arry11[12],arry12[12],arry13[12],arry14[12],arry15[12],arry16[12];
    char arry17[12],arry18[12],arry19[12],arry20[12];

    double sx,sy,uxy,nx,ny,n2,adex[20],adey[20],aexy,cerx,cery,o2x,ody;
    double p,q,r,t,o3x,o3y,o4x,o4y,cubx,cuby,cuatx,cuaty,nox,noy,tmedx,tmedy,ttest,CORR2,tcor;

    // Inicializo variables de instancias
    ux=uy=odx=o2y=o1x=o1y=SKEWx=SKEWy=KURTx=KURTy=COVxy=CORRxy=0.0;

    for(v=0;v<20;v++)
    {
        aestx[v]=0.0;
        aesty[v]=0.0;
        adex[v]=0.0;
    }
    adey[v]=0.0;

    GetDlgItemText(HWindow,A_E1,arr1,12);
    aestx[0]=atof(arr1);
    GetDlgItemText(HWindow,A_E2,arr2,12);
    aestx[1]=atof(arr2);
    GetDlgItemText(HWindow,A_E3,arr3,12);
    aestx[2]=atof(arr3);
    GetDlgItemText(HWindow,A_E4,arr4,12);
    aestx[3]=atof(arr4);
    GetDlgItemText(HWindow,A_E5,arr5,12);
    aestx[4]=atof(arr5);
    GetDlgItemText(HWindow,A_E6,arr6,12);
    aestx[5]=atof(arr6);
    GetDlgItemText(HWindow,A_E7,arr7,12);
    aestx[6]=atof(arr7);
    GetDlgItemText(HWindow,A_E8,arr8,12);
    aestx[7]=atof(arr8);
    GetDlgItemText(HWindow,A_E9,arr9,12);
    aestx[8]=atof(arr9);
    GetDlgItemText(HWindow,A_E10,arr10,12);
    aestx[9]=atof(arr10);
    GetDlgItemText(HWindow,A_E11,arr11,12);
    aestx[10]=atof(arr11);
    GetDlgItemText(HWindow,A_E12,arr12,12);
    aestx[11]=atof(arr12);
    GetDlgItemText(HWindow,A_E13,arr13,12);
    aestx[12]=atof(arr13);
    GetDlgItemText(HWindow,A_E14,arr14,12);
    aestx[13]=atof(arr14);
    GetDlgItemText(HWindow,A_E15,arr15,12);
    aestx[14]=atof(arr15);

```



```

    GetDlgItemText(HWindow,A_E16,arr16,12);
aestx[15]=atof(arr16);
    GetDlgItemText(HWindow,A_E17,arr17,12);
aestx[16]=atof(arr17);
    GetDlgItemText(HWindow,A_E18,arr18,12);
aestx[17]=atof(arr18);
    GetDlgItemText(HWindow,A_E19,arr19,12);
aestx[18]=atof(arr19);
    GetDlgItemText(HWindow,A_E20,arr20,12);
aestx[19]=atof(arr20);

    GetDlgItemText(HWindow,A_EY1,arry1,12);
aesty[0]=atof(arry1);
    GetDlgItemText(HWindow,A_EY2,arry2,12);
aesty[1]=atof(arry2);
    GetDlgItemText(HWindow,A_EY3,arry3,12);
aesty[2]=atof(arry3);
    GetDlgItemText(HWindow,A_EY4,arry4,12);
aesty[3]=atof(arry4);
    GetDlgItemText(HWindow,A_EY5,arry5,12);
aesty[4]=atof(arry5);
    GetDlgItemText(HWindow,A_EY6,arry6,12);
aesty[5]=atof(arry6);
    GetDlgItemText(HWindow,A_EY7,arry7,12);
aesty[6]=atof(arry7);
    GetDlgItemText(HWindow,A_EY8,arry8,12);
aesty[7]=atof(arry8);
    GetDlgItemText(HWindow,A_EY9,arry9,12);
aesty[8]=atof(arry9);
    GetDlgItemText(HWindow,A_EY10,arry10,12);
aesty[9]=atof(arry10);
    GetDlgItemText(HWindow,A_EY11,arry11,12);
aesty[10]=atof(arry11);
    GetDlgItemText(HWindow,A_EY12,arry12,12);
aesty[11]=atof(arry12);
    GetDlgItemText(HWindow,A_EY13,arry13,12);
aesty[12]=atof(arry13);
    GetDlgItemText(HWindow,A_EY14,arry14,12);
aesty[13]=atof(arry14);
    GetDlgItemText(HWindow,A_EY15,arry15,12);
aesty[14]=atof(arry15);
    GetDlgItemText(HWindow,A_EY16,arry16,12);
aesty[15]=atof(arry16);
    GetDlgItemText(HWindow,A_EY17,arry17,12);
aesty[16]=atof(arry17);
    GetDlgItemText(HWindow,A_EY18,arry18,12);
aesty[17]=atof(arry18);
    GetDlgItemText(HWindow,A_EY19,arry19,12);
aesty[18]=atof(arry19);
    GetDlgItemText(HWindow,A_EY20,arry20,12);
aesty[19]=atof(arry20);

nx=0.0;
sx=0.0;
ux=0.0;
    for(v=0;v<20;v++)
    {
        if(aestx[v]!=0.0)
            n=n+1;
        if (aesty[v]!=0.0)
            m=m+1;
    }
    if((n==0 && m==0) || (n<2 && m<2) || (n!=0 && m==0) || (n==0 && m!=0))
    {
        MessageBox(HWindow,"Debe ingresar datos para
calcular","OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }

```

```

}
    else
    {
        EnableWindow(GetDlgItem(HWindow, IDM_IMPRIMIR), TRUE);

        for(v=0;v<20;v++)
        {
            if(aestx[v] != 0.0)
                nx=nx++;
            sx=sx+aestx[v];
        }
        ux=sx/nx;

        cerx=0.0;
        cubx=0.0;
        cuatx=0.0;
        odx=0.0;

        for(v=0;v<20;v++)
        {
            if(aestx[v] != 0.0)
                adex[v]=aestx[v]-ux;
            cerx=cerx+adex[v]*adex[v];
            cubx=cubx+adex[v]*adex[v]*adex[v];
            cuatx=cuatx+adex[v]*adex[v]*adex[v]*adex[v];
        }
        odx=cerx/(nx-1.0);
        o1x=0.0;
        o1x = sqrt(fabs(odx));
        o3x=odx*o1x;
        o4x=odx*odx;
        SKEWx=cubx/o3x;
        KURTx=cuatx/o4x;

        ny=0.0;
        sy=0.0;
        uy=0.0;
        for(w=0;w<20;w++)
        {
            if(aesty[w] != 0.0)
                ny=ny++;
            sy=sy+aesty[w];
        }
        uy=sy/ny;

        cery=0.0;
        cuby=0.0;
        cuaty=0.0;
        o2y=0.0;

        for(w=0;w<20;w++)
        {
            if(aesty[w] != 0.0)
                adey[w]=aesty[w]-uy;
            cery=cery+adey[w]*adey[w];
            cuby=cuby+adey[w]*adey[w]*adey[w];
            cuaty=cuaty+adey[w]*adey[w]*adey[w]*adey[w];
        }
        o2y=cery/(ny-1.0);
        o1y=0.0;
        o1y = sqrt(fabs(o2y));
        o3y=o2y*o1y;
        o4y=o2y*o2y;
        SKEWy=cuby/o3y;
        KURTy=cuaty/o4y;

        aexy=0.0;

```

```

for(v=0,v<20,v++)
{
    if (adex[v] != 0.0 && adeg[v] != 0.0)
        aexy=aexy+adex[v]*adeg[v];
}

COVxy=aexy/(nx-1.0);
CORRxy=COVxy/(o1x*o1y);
CORR2=CORRxy*CORRxy;
n2=nx-2.0;
nox=o1x/sqrt(fabs(nx));
noy=o1y/sqrt(fabs(ny));
tmedx=ux/nox;
tmedy=uy/noy;
uxy=ux-uy;
o2x=odx*(nx-1.0);
ody=o2y*(ny-1.0);
p=o2x+ody;
q=nx+ny-2.0;
r=nx+ny;
t=nx*ny;
ttest=uxy/sqrt(fabs((p*r)/(q*t)));
tcor=CORRxy*sqrt(fabs(n2))/sqrt(fabs(1.0-CORR2));

sprintf(SizeText,"Media A=\t %3.4f \t Media B=          %3.4f \n Desviación \t\t\t
Desviación\n Estándar A=\t %3.4f\t Estándar B=          %3.4f \n Varianza A=\t %3.4f \t Varianza B=
%3.4f \n SKEWNESS A=\t%3.4f \t SKEWNESS B=          %3.4f \n KURTOSIS A=          %3.4f KURTOSIS B=
%3.4f \n t-test Media (A)=          %3.4f \n t-test Media (B)=          %3.4f \n tMed(A)=tMed(B) = %3.4f \n
Covarianza=\t%3.4f \n          Correlación=\t%3.4f \n          t-test          Correlación=\t%3.4f
",ux,uy,o1x,o1y,odx,o2y,SKEWx,SKEWy,KURTx,KURTy,tmedx,tmedy,ttest,COVxy,CORRxy,tcor);

MessageBox(HWindow,SizeText,"CALCULOS ESTADISTICOS",MB_OK);

}
return;
}

```

## Módulo de Índices Financieros

```

#include "indi.hpp"

//Balance
extern double afx[10];
extern double afy[12],cc,in;
extern double ac,afijo,pc,pfijo,totac,topyp,patri; // hoja de Balance

//perdidas
extern double ep[7];
extern char epg1[12],epg2[12],epg3[12],epg4[12],epg5[12],epg6[12],epg7[12];
extern double vn,ub,uai2,uai,un;

void TIndi::Proclmpimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
}

```

```

Pr.StartPage();
strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "MODULO: Indices Financieros");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "Indice de Liquidez:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",il);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Prueba del ácido:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",pa);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Rotación total de activos:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",tra);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Rotación de Activos Fijos:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",raf);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Período Promedio de Cobro:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",ppc);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Rotación de Inventarios:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.4f",ri);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

```



```

        CurrentLine+=2;
        strcpy( strLine, "Indice de deudas a activos:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",ia);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Indice de deudas a Inversión:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",id);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Margen de Utilidad:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",mu);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Tiempo a Interés ganado:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",tig);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Retorno de activos:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",ra);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Retorno de inversión:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.4f",riv);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
"InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TIndi::SetupWindow()
{
    il=p=ir=pa=mub=tra=raf=ppc=n=ia=id=mu=tig=ra=riv=tp=0 0;

```

```

        tp=pc+pfijo;
        il=ac/pc;
ir=ac-in;
pa=ir/pc;
mub=ub/vn;
tra=vn/ac;
    raf=vn/afijo;
    p=cc*360.0;
    ppc=p/vn;
ri=vn/in;
    ia=tp/totac;
id=tp/patri;
    mu=un/vn;
tig=uai2/epx[5];
    ra=un/totac;
    riv=un/patri;

    sprintf(SizeText,"% .4f",il);
    SetDlgItemText(HWindow,ID_INDI1,SizeText);

    sprintf(SizeText,"% .4f",pa);
    SetDlgItemText(HWindow,ID_INDI2,SizeText);

    sprintf(SizeText,"% .4f",tra);
    SetDlgItemText(HWindow,ID_INDI3,SizeText);

    sprintf(SizeText,"% .4f",raf);
    SetDlgItemText(HWindow,ID_INDI4,SizeText);

    sprintf(SizeText,"% .4f",ppc);
    SetDlgItemText(HWindow,ID_INDI5,SizeText);

    sprintf(SizeText,"% .4f",ri);
    SetDlgItemText(HWindow,ID_INDI6,SizeText);

    sprintf(SizeText,"% .4f",ia);
    SetDlgItemText(HWindow,ID_INDI7,SizeText);

    sprintf(SizeText,"% .4f",id);
    SetDlgItemText(HWindow,ID_INDI8,SizeText);

    sprintf(SizeText,"% .4f",mu);
    SetDlgItemText(HWindow,ID_INDI9,SizeText);

    sprintf(SizeText,"% .4f",tig);
    SetDlgItemText(HWindow,ID_INDI10,SizeText);

    sprintf(SizeText,"% .4f",ra);
    SetDlgItemText(HWindow,ID_INDI11,SizeText);

    sprintf(SizeText,"% .4f",riv);
    SetDlgItemText(HWindow,ID_INDI12,SizeText);

    return;
}

```

## Módulo de Fluxo de Caja Irregular

```

#include "irregu.hpp"

void Tirregu::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;

```

```

int Tab;

TPrint Pr;
hPr= Pr.GetDC();
LineSpace= Pr.GetLineSpace();
Tab= Pr.GetTab();
Pr.StartDoc();
Pr.StartPage();
strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "MODULO: Flujo de Caja Irregular");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "Pago 1 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",fir[1]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Pago 2 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",fir[2]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Pago 3 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",fir[3]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Pago 4 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",fir[4]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Valor Presente :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%8.2f",vp);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Tasa de descuento :");

```



```

        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%8.2f",tasair);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
"InvestMaster" para
Windows 1.0", MB_OK | MB_ICONINFORMATION );

}

void Tirregu::ProcGrabar()
{
    if( PedirNombArchivo( TRUE, "irr" ))
    {
        GetDlgItemText(HWindow,F_IR1,Buffer,8);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,F_IR2,Buffer,8);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,F_IR3,Buffer,8);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,F_IR4,Buffer,8);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,F_IR5,Buffer,8);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,F_IR6,Buffer,8);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,F_IR7,Buffer,8);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
        GetDlgItemText(HWindow,F_IR8,Buffer,8);
        WritePrivateProfileString("Datos","dato8",Buffer,szName);
        GetDlgItemText(HWindow,F_IR9,Buffer,8);
        WritePrivateProfileString("Datos","dato9",Buffer,szName);
        GetDlgItemText(HWindow,F_IR10,Buffer,8);
        WritePrivateProfileString("Datos","dato10",Buffer,szName);
        GetDlgItemText(HWindow,F_IR11,Buffer,8);
        WritePrivateProfileString("Datos","dato11",Buffer,szName);
        GetDlgItemText(HWindow,F_IR12,Buffer,8);
        WritePrivateProfileString("Datos","dato12",Buffer,szName);
        GetDlgItemText(HWindow,F_IR13,Buffer,8);
        WritePrivateProfileString("Datos","dato13",Buffer,szName);
        GetDlgItemText(HWindow,F_IR14,Buffer,8);
        WritePrivateProfileString("Datos","dato14",Buffer,szName);
        GetDlgItemText(HWindow,F_IR15,Buffer,8);
        WritePrivateProfileString("Datos","dato15",Buffer,szName);
        GetDlgItemText(HWindow,F_IR16,Buffer,8);
        WritePrivateProfileString("Datos","dato16",Buffer,szName);
        GetDlgItemText(HWindow,F_IR17,Buffer,8);
        WritePrivateProfileString("Datos","dato17",Buffer,szName);
        GetDlgItemText(HWindow,F_IR18,Buffer,8);
        WritePrivateProfileString("Datos","dato18",Buffer,szName);
        GetDlgItemText(HWindow,F_IR19,Buffer,8);
        WritePrivateProfileString("Datos","dato19",Buffer,szName);
        GetDlgItemText(HWindow,F_IR20,Buffer,8);
        WritePrivateProfileString("Datos","dato20",Buffer,szName);
        GetDlgItemText(HWindow,F_IR21,Buffer,8);
    }
}

```



```

WritePrivateProfileString("Datos","dato21",Buffer,szName);
GetDlgItemText(HWindow,F_IR22,Buffer,8);
WritePrivateProfileString("Datos","dato22",Buffer,szName);
GetDlgItemText(HWindow,F_IR23,Buffer,8);
WritePrivateProfileString("Datos","dato23",Buffer,szName);
GetDlgItemText(HWindow,F_IR24,Buffer,8);
WritePrivateProfileString("Datos","dato24",Buffer,szName);
GetDlgItemText(HWindow,F_IR25,Buffer,8);
WritePrivateProfileString("Datos","dato25",Buffer,szName);
GetDlgItemText(HWindow,F_IR26,Buffer,8);
WritePrivateProfileString("Datos","dato26",Buffer,szName);
GetDlgItemText(HWindow,F_IR27,Buffer,8);
WritePrivateProfileString("Datos","dato27",Buffer,szName);
GetDlgItemText(HWindow,F_IR28,Buffer,8);
WritePrivateProfileString("Datos","dato28",Buffer,szName);
GetDlgItemText(HWindow,F_IR29,Buffer,8);
WritePrivateProfileString("Datos","dato29",Buffer,szName);
GetDlgItemText(HWindow,F_IR30,Buffer,8);
WritePrivateProfileString("Datos","dato30",Buffer,szName);
GetDlgItemText(HWindow,F_IR31,Buffer,8);
WritePrivateProfileString("Datos","dato31",Buffer,szName);
GetDlgItemText(HWindow,F_IR32,Buffer,8);
WritePrivateProfileString("Datos","dato32",Buffer,szName);
}
}

void Tirregu::ProcAbrir()
{
    if( PedirNombArchivo( FALSE, "irr" ))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR4,Buffer);
        GetPrivateProfileString("Datos","dato5","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR5,Buffer);
        GetPrivateProfileString("Datos","dato6","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR6,Buffer);
        GetPrivateProfileString("Datos","dato7","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR7,Buffer);
        GetPrivateProfileString("Datos","dato8","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR8,Buffer);
        GetPrivateProfileString("Datos","dato9","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR9,Buffer);
        GetPrivateProfileString("Datos","dato10","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR10,Buffer);
        GetPrivateProfileString("Datos","dato11","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR11,Buffer);
        GetPrivateProfileString("Datos","dato12","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR12,Buffer);
        GetPrivateProfileString("Datos","dato13","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR13,Buffer);
        GetPrivateProfileString("Datos","dato14","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR14,Buffer);
        GetPrivateProfileString("Datos","dato15","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR15,Buffer);
        GetPrivateProfileString("Datos","dato16","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR16,Buffer);
        GetPrivateProfileString("Datos","dato17","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR17,Buffer);
        GetPrivateProfileString("Datos","dato18","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR18,Buffer);
        GetPrivateProfileString("Datos","dato19","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR19,Buffer);
    }
}

```

```

        GetPrivateProfileString("Datos","dato20","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR20,Buffer);
        GetPrivateProfileString("Datos","dato21","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR21,Buffer);
        GetPrivateProfileString("Datos","dato22","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR22,Buffer);
        GetPrivateProfileString("Datos","dato23","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR23,Buffer);
        GetPrivateProfileString("Datos","dato24","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR24,Buffer);
        GetPrivateProfileString("Datos","dato25","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR25,Buffer);
        GetPrivateProfileString("Datos","dato26","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR26,Buffer);
        GetPrivateProfileString("Datos","dato27","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR27,Buffer);
        GetPrivateProfileString("Datos","dato28","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR28,Buffer);
        GetPrivateProfileString("Datos","dato29","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR29,Buffer);
        GetPrivateProfileString("Datos","dato30","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR30,Buffer);
        GetPrivateProfileString("Datos","dato31","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR31,Buffer);
        GetPrivateProfileString("Datos","dato32","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR32,Buffer);
    }
    return;
}

void Tlregu::ProcCalcular()
{
    int ir,band=0,n=0;

    vp=den=r=vprev=tasair=min=max=0.0;
    for(ir=1;ir<33;ir++)
    {
        fir[ir]=0.0;
    }

    GetDlgItemText(HWindow,F_IR1,Buffer,8);
    fir[1]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR2,Buffer,8);
    fir[2]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR3,Buffer,8);
    fir[3]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR4,Buffer,8);
    fir[4]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR5,Buffer,8);
    fir[5]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR6,Buffer,8);
    fir[6]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR7,Buffer,8);
    fir[7]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR8,Buffer,8);
    fir[8]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR9,Buffer,8);
    fir[9]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR10,Buffer,8);
    fir[10]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR11,Buffer,8);
    fir[11]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR12,Buffer,8);
    fir[12]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR13,Buffer,8);
    fir[13]=atof(Buffer);
    GetDlgItemText(HWindow,F_IR14,Buffer,8);

```

```

GetDlgItemText(HWindow,F_IR14,Buffer,8);
fir[14]=atof(Buffer);
GetDlgItemText(HWindow,F_IR15,Buffer,8);
fir[15]=atof(Buffer);
GetDlgItemText(HWindow,F_IR16,Buffer,8);
fir[16]=atof(Buffer);
GetDlgItemText(HWindow,F_IR17,Buffer,8);
fir[17]=atof(Buffer);
GetDlgItemText(HWindow,F_IR18,Buffer,8);
fir[18]=atof(Buffer);
GetDlgItemText(HWindow,F_IR19,Buffer,8);
fir[19]=atof(Buffer);
GetDlgItemText(HWindow,F_IR20,Buffer,8);
fir[20]=atof(Buffer);
GetDlgItemText(HWindow,F_IR21,Buffer,8);
fir[21]=atof(Buffer);
GetDlgItemText(HWindow,F_IR22,Buffer,8);
fir[22]=atof(Buffer);
GetDlgItemText(HWindow,F_IR23,Buffer,8);
fir[23]=atof(Buffer);
GetDlgItemText(HWindow,F_IR24,Buffer,8);
fir[24]=atof(Buffer);
GetDlgItemText(HWindow,F_IR25,Buffer,8);
fir[25]=atof(Buffer);
GetDlgItemText(HWindow,F_IR26,Buffer,8);
fir[26]=atof(Buffer);
GetDlgItemText(HWindow,F_IR27,Buffer,8);
fir[27]=atof(Buffer);
GetDlgItemText(HWindow,F_IR28,Buffer,8);
fir[28]=atof(Buffer);
GetDlgItemText(HWindow,F_IR29,Buffer,8);
fir[29]=atof(Buffer);
GetDlgItemText(HWindow,F_IR30,Buffer,8);
fir[30]=atof(Buffer);
GetDlgItemText(HWindow,F_IR31,Buffer,8);
fir[31]=atof(Buffer);
GetDlgItemText(HWindow,F_IR32,Buffer,8);
fir[32]=atof(Buffer);

for(ir=1;ir<31;ir++)
{
if(fir[ir]!=0.0)
n=n+1;
}
if((n==0)&&(fir[31]==0) && (fir[32]==0))
{
MessageBox(HWindow,"Debe ingresar datos","OBSERVACION",MB_OK);
EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
}
else
{
if(n==0)
{
MessageBox(HWindow,"Debe ingresar pagos","OBSERVACION",MB_OK);
EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
}
else
{
if(fir[31]==0.0 && fir[32]==0.0)
{
MessageBox(NULL,"Faltan datos ..... \n \nIngrese el valor presente o la tasa de
descuento","ERROR EN INGRESO",MB_OK);
EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
}
else
{ //Calculo valor presente porque no lo ingresó
EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
if(fir[31]==0.0)

```



```

    {
        r=fir[32];
        for(ir=1;ir<31;ir++)
        {
            if (fir[ir]!= 0.0)
            {
                den=pow((1.0+r),ir);
                vp=vp+(fir[ir]/den);
            } //if
        } //for
        sprintf(SizeText,"El valor presente es:  %.2f ",vp);
        MessageBox(HWindow,SizeText,"RESULTADOS",MB_OK);
    } //if

    else
    {
        min=fir[31]-0.01;
        max=fir[31]+0.01;
        r=0.01;
    do{
        for(ir=1;ir<n+1;ir++)
        {
            den=pow((1.0+r),ir);
            vprev=vprev+(fir[ir]/den);
        }
        if(vprev>=min && vprev<=max)
        {
            band=1;
            tasair=r;
            break;
        }
        else
            band=0;
        den=0.0;
        vprev=0.0;
        r=r+0.01;
    }while(r<1.0);

    if (band==1){
        sprintf(SizeText,"La tasa de descuento es:  %.4f ",tasair);
        MessageBox(HWindow,SizeText, "RESULTADOS",MB_OK);
    }
    else
        MessageBox(NULL,"No existe tasa...Error en datos","RESULTADOS",MB_OK);
}
}
}
}

```

## Módulo de Nivel

```

#include "nive.hpp"

void TNive::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

```



```

TPrint Pr;
hPr= Pr.GetDC();
LineSpace= Pr.GetLineSpace();
Tab= Pr.GetTab();
Pr.StartDoc();
Pr.StartPage();
// GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
strcpy( strLine, "InvestMaster Para Windows (Versión 1.0)");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "MODULO: Análisis de Niveles de Riqueza.");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "CALCULO DE NIVELES DE RIQUEZA:");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Retorno 1:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[0]);
TextOut( hPr, 5*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

strcpy( strLine, "Riqueza 1:=");
TextOut( hPr, 13*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",wr[0]);
TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Retorno 2:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[1]);
TextOut( hPr, 5*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

strcpy( strLine, "Riqueza 2:");
TextOut( hPr, 13*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",wr[1]);
TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Retorno 3:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[2]);
TextOut( hPr, 5*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

strcpy( strLine, "Riqueza 3:");
TextOut( hPr, 13*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

```

```

    sprintf(strLine,"%15.4f",wr[2]);
    TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "MEDIA ARITMETICA");
    TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    strcpy( strLine, "MEDIA GEOMETRICA");
    TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "Riqueza Relativa:");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",ma);
    TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",mg);
    TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "Retorno :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",mar);
    TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",mgr);
    TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "RIESGO DE RETORNO DE MEDIA
ARITMETICA");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Desviación Estándar :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",des);
    TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Varianza :");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    sprintf(strLine,"%15.4f",var);
    TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    Pr.EndPage();
    Pr.EndDoc();
    DeleteDC(hPr);
    MessageBox( HWindow, "Impresión finalizada con éxito.",
"InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );

```

```

}

void TNive::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "niv"))
    {
        GetDlgItemText(HWindow,R_QX1,Buffer,10);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,R_QX2,Buffer,10);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,R_QX3,Buffer,10);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,R_QX4,Buffer,10);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,R_QX5,Buffer,10);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,R_QX6,Buffer,10);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,R_QX7,Buffer,10);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
        GetDlgItemText(HWindow,R_QX8,Buffer,10);
        WritePrivateProfileString("Datos","dato8",Buffer,szName);
        GetDlgItemText(HWindow,R_QX9,Buffer,10);
        WritePrivateProfileString("Datos","dato9",Buffer,szName);
        GetDlgItemText(HWindow,R_QX10,Buffer,10);
        WritePrivateProfileString("Datos","dato10",Buffer,szName);
        GetDlgItemText(HWindow,R_QX11,Buffer,10);
        WritePrivateProfileString("Datos","dato11",Buffer,szName);
        GetDlgItemText(HWindow,R_QX12,Buffer,10);
        WritePrivateProfileString("Datos","dato12",Buffer,szName);
        GetDlgItemText(HWindow,R_QX13,Buffer,10);
        WritePrivateProfileString("Datos","dato13",Buffer,szName);
        GetDlgItemText(HWindow,R_QX14,Buffer,10);
        WritePrivateProfileString("Datos","dato14",Buffer,szName);
        GetDlgItemText(HWindow,R_QX15,Buffer,10);
        WritePrivateProfileString("Datos","dato15",Buffer,szName);
        GetDlgItemText(HWindow,R_QX16,Buffer,10);
        WritePrivateProfileString("Datos","dato16",Buffer,szName);
        GetDlgItemText(HWindow,R_QX17,Buffer,10);
        WritePrivateProfileString("Datos","dato17",Buffer,szName);
        GetDlgItemText(HWindow,R_QX18,Buffer,10);
        WritePrivateProfileString("Datos","dato18",Buffer,szName);
        GetDlgItemText(HWindow,R_QX19,Buffer,10);
        WritePrivateProfileString("Datos","dato19",Buffer,szName);
        GetDlgItemText(HWindow,R_QX20,Buffer,10);
        WritePrivateProfileString("Datos","dato20",Buffer,szName);
    }
    return;
}

void TNive::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "niv"))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX4,Buffer);
        GetPrivateProfileString("Datos","dato5","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX5,Buffer);
        GetPrivateProfileString("Datos","dato6","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX6,Buffer);
        GetPrivateProfileString("Datos","dato7","",Buffer,10,szName);
        SetDlgItemText(HWindow,R_QX7,Buffer);
    }
}

```



```

    GetPrivateProfileString("Dados","dato8","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX8,Buffer);
    GetPrivateProfileString("Dados","dato9","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX9,Buffer);
    GetPrivateProfileString("Dados","dato10","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX10,Buffer);
    GetPrivateProfileString("Dados","dato11","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX11,Buffer);
    GetPrivateProfileString("Dados","dato12","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX12,Buffer);
    GetPrivateProfileString("Dados","dato13","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX13,Buffer);
    GetPrivateProfileString("Dados","dato14","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX14,Buffer);
    GetPrivateProfileString("Dados","dato15","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX15,Buffer);
    GetPrivateProfileString("Dados","dato16","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX16,Buffer);
    GetPrivateProfileString("Dados","dato17","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX17,Buffer);
    GetPrivateProfileString("Dados","dato18","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX18,Buffer);
    GetPrivateProfileString("Dados","dato19","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX19,Buffer);
    GetPrivateProfileString("Dados","dato20","",Buffer,10,szName);
    SetDlgItemText(HWindow,R_QX20,Buffer);
}
return;
}

void TNive::ProcCalcular()
{
    int rx;
    double rqx[20];

    n=ret=w|=ma=cma=mar=mg=mgr=var=des=mul=mvar= 0.0;
    for(rx=0;rx<20;rx++)
    {
        rqx[rx]=0.0;
        r[rx]=0.0;
        wr[rx]=0.0;
    }

    GetDlgItemText(HWindow,R_QX1,Buffer,10);
    rqx[0]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX2,Buffer,10);
    rqx[1]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX3,Buffer,10);
    rqx[2]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX4,Buffer,10);
    rqx[3]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX5,Buffer,10);
    rqx[4]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX6,Buffer,10);
    rqx[5]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX7,Buffer,10);
    rqx[6]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX8,Buffer,10);
    rqx[7]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX9,Buffer,10);
    rqx[8]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX10,Buffer,10);
    rqx[9]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX11,Buffer,10);
    rqx[10]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX12,Buffer,10);
    rqx[11]=atof(Buffer);

```



```

        GetDlgItemText(HWindow,R_QX13,Buffer,10);
        rqx[12]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX14,Buffer,10);
        rqx[13]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX15,Buffer,10);
        rqx[14]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX16,Buffer,10);
        rqx[15]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX17,Buffer,10);
        rqx[16]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX18,Buffer,10);
        rqx[17]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX19,Buffer,10);
        rqx[18]=atof(Buffer);
        GetDlgItemText(HWindow,R_QX20,Buffer,10);
        rqx[19]=atof(Buffer);

//Calculos comunes
ma=0.0; mar=0.0; mg=0.0; mgr=0.0; var=0.0; des=0.0;
mul=1.0,wl=0.0,ret=0.0,mvar=0.0,cma=0.0;

for(rx=0;rx<20;rx++)
{
    if(rqx[rx] != 0.0)
        n=n+1.0;
}

    if (n==0 || n<2){
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
        MessageBox(HWindow,"Debe ingresar datos","OBSERVACION",MB_OK);
    }
else
    {
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
for(rx=0;rx<n-1;rx++)
{
    if(rqx[rx] != 0.0)
    {
        wr[rx]=rqx[rx+1]/rqx[rx];
    }
}

for(rx=0;rx<20;rx++)
{
    if(wr[rx] != 0.0)
        r[rx]=wr[rx]-1.0;
}
for(rx=0;rx<20;rx++)
{
    if(wr[rx]!=0.0)
    {
        cma=cma+wr[rx];
        mul=mul*wr[rx];
    }
}
for(rx=0;rx<20;rx++)
{
    if(r[rx]!=0.0)
        ret=ret+r[rx];
}
ma=cma/(n-1);
mar=ma-1.0;
mg=pow(mul,1/(n-1));
mgr=mg-1.0;
for(rx=0,rx<n-1;rx++){
    wl=r[rx]-mar;
    mvar=mvar+pow(wl,2);
}

```

```

    }
    var=mvar/(n-1);
    des=sqrt(var);

    sprintf(SizeText," \n Retorno 1    %.4f \t Riqueza 1    %.4f \n Retorno 2    %.4f \t
Riqueza 2    %.4f \n Retorno 3    %.4f \t Riqueza 3    %.4f \n \n          Media Aritmética  Media
Geométrica \n \n Riqueza Relativa  %.4f    %.4f \n Retorno          %.4f    %.4f \n \n \n
Riesgo de retorno de media aritmética \n \n Desviación estándar          %.4f \n Varianza
%.4f ",r[0],wr[0],r[1],wr[1],r[2],wr[2],ma,mg,mar,mgr,des,var);
    MessageBox(HWindow,SizeText,"CALCULOS NIVELES DE RIQUEZA",MB_OK);
}
}
}

```

## Módulo de Pérdidas

```

#include "perdidas.hpp"

extern int fpyg;

//perdidas
extern double epx[7];
extern char epg1[12],epg2[12],epg3[12],epg4[12],epg5[12],epg6[12],epg7[12];
extern double vn,ub,uai2,uai,un;

void TPerdidas::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Análisis Financiero (Estado de
Pérdidas y Ganancias.)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "RESULTADOS OBTENIDOS:");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;

```

```

        strcpy( strLine, "Ventas Netas :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.2f",vn);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Utilidad Bruta Operacional:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.2f",ub);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Utilidades Antes Interés e");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine++;
        strcpy( strLine, "Impuestos:");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.2f",uai2);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Utilidades Antes Impuestos :");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.2f",uai);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "UTILIDAD NETA ");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.2f",un);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine++;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
"InvestMaster para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TPerdidas::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "per"))
    {
        GetDlgItemText(HWindow,E_PX1,Buffer,12);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,E_PX2,Buffer,12);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,E_PX3,Buffer,12);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
    }
}

```

```

        GetDlgItemText(HWindow,E_PX4,Buffer,12);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,E_PX5,Buffer,12);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,E_PX6,Buffer,12);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,E_PX7,Buffer,12);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
    }
    return;
}

void TPerdidas::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "per"))
    {
        GetPrivateProfileString("Datos","dato1","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX1,Buffer);
        GetPrivateProfileString("Datos","dato2","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX2,Buffer);
        GetPrivateProfileString("Datos","dato3","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX3,Buffer);
        GetPrivateProfileString("Datos","dato4","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX4,Buffer);
        GetPrivateProfileString("Datos","dato5","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX5,Buffer);
        GetPrivateProfileString("Datos","dato6","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX6,Buffer);
        GetPrivateProfileString("Datos","dato7","",Buffer,12,szName);
        SetDlgItemText(HWindow,E_PX7,Buffer);
    }
    return;
}

void TPerdidas::ProcCalcular()
{
    int px,i=0;

    for(px=0;px<7;px++)
        epx[px]=0.0;

    GetDlgItemText(HWindow,E_PX1,Buffer,12);
    epx[0]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX2,Buffer,12);
    epx[1]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX3,Buffer,12);
    epx[2]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX4,Buffer,12);
    epx[3]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX5,Buffer,12);
    epx[4]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX6,Buffer,12);
    epx[5]=atof(Buffer);
    GetDlgItemText(HWindow,E_PX7,Buffer,12);
    epx[6]=atof(Buffer);

    for(px=0,px<7,px++)
    {
        if(epx[px] != 0.0)
            i=i+1;
    }

    if (i==0)
    {
        Ipyg=1;
        MessageBox(HWindow,"Debe ingresar datos para calcular", "OBSERVACION",MB_OK);
    }
}

```



```

    }
    else
    {
        lpyg=0;

        vn=0.0;
        ub=0.0;
        uai2=0.0;
        uai=0.0;
        un=0.0;

        vn=epx[0]-epx[1];
        ub=vn-epx[2];
        uai2=ub-epx[3]-epx[4];
        uai=uai2-epx[5];
        un=uai-epx[6];
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
        sprintf(SizeText,"%6.2f",vn);
        SetDlgItemText(HWindow,ID_VN,SizeText);

        sprintf(SizeText,"%6.2f",ub);
        SetDlgItemText(HWindow,ID_UB,SizeText);

        sprintf(SizeText,"%6.2f",uai2);
        SetDlgItemText(HWindow,ID_UAI2,SizeText);

        sprintf(SizeText,"%6.2f",uai);
        SetDlgItemText(HWindow,ID_UAI,SizeText);

        sprintf(SizeText,"%6.2f",un);
        SetDlgItemText(HWindow,ID_UN,SizeText);

        //  sprintf(SizeText,"  VENTAS NETAS                %2f \n \n  UTILIDAD BRUTA
        %2f \n  OPERACIONAL \n \n  UTILIDAD ANTES INTERESES          %2f \n  E
        IMPUESTOS \n \n  UTILIDAD ANTES IMPUESTOS          %2f \n \n  UTILIDAD NETA
        %2f ",vn,ub,uai2,uai,un);

        //  MessageBox(HWindow,SizeText,"TOTALES      ESTADO      DE      PERDIDAS      Y
        GANANCIAS",MB_OK);
    }
}

```

## Módulo Retorno

```

#include "retorno.hpp"

void TRetorno::ProclImprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));
}

```

```

CurrentLine++;
strcpy( strLine, "MODULO: Análisis de Retorno de Riqueza.");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "RESULTADOS DE RIQUEZAS.");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Riqueza 1 =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[0]);
TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Riqueza 2 =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[1]);
TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Riqueza 3 =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",r[2]);
TextOut( hPr, 7*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "MEDIA ARITMETICA");
TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

strcpy( strLine, "MEDIA GEOMETRICA");
TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Riqueza Relativa:");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",ma);
TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",mg);
TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "Retorno :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",mar);

```

```

        TextOut( hPr, 9*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%15.4f",mgr);
        TextOut( hPr, 16*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "RIESGO DE RETORNO DE MEDIA
ARITMETICA");
        TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Desviación Estándar =");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.4f",des);
        TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Varianza =");
        TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        sprintf(strLine,"%15.4f",var);
        TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine++;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
                    "InvestMaster", para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
    }

void TRetorno::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "ret"))
    {
        GetDlgItemText(HWindow,R_QX1,Buffer,10);
        WritePrivateProfileString("Datos","dato1",Buffer,szName);
        GetDlgItemText(HWindow,R_QX2,Buffer,10);
        WritePrivateProfileString("Datos","dato2",Buffer,szName);
        GetDlgItemText(HWindow,R_QX3,Buffer,10);
        WritePrivateProfileString("Datos","dato3",Buffer,szName);
        GetDlgItemText(HWindow,R_QX4,Buffer,10);
        WritePrivateProfileString("Datos","dato4",Buffer,szName);
        GetDlgItemText(HWindow,R_QX5,Buffer,10);
        WritePrivateProfileString("Datos","dato5",Buffer,szName);
        GetDlgItemText(HWindow,R_QX6,Buffer,10);
        WritePrivateProfileString("Datos","dato6",Buffer,szName);
        GetDlgItemText(HWindow,R_QX7,Buffer,10);
        WritePrivateProfileString("Datos","dato7",Buffer,szName);
        GetDlgItemText(HWindow,R_QX8,Buffer,10);
        WritePrivateProfileString("Datos","dato8",Buffer,szName);
        GetDlgItemText(HWindow,R_QX9,Buffer,10);
        WritePrivateProfileString("Datos","dato9",Buffer,szName);
        GetDlgItemText(HWindow,R_QX10,Buffer,10);
        WritePrivateProfileString("Datos","dato10",Buffer,szName);
        GetDlgItemText(HWindow,R_QX11,Buffer,10);
        WritePrivateProfileString("Datos","dato11",Buffer,szName);
    }
}

```



```

        GetDlgItemText(HWindow,R_QX12,Buffer,10);
        WritePrivateProfileString("Datos", "dato12", Buffer, szName);
        GetDlgItemText(HWindow,R_QX13,Buffer,10);
        WritePrivateProfileString("Datos", "dato13", Buffer, szName);
        GetDlgItemText(HWindow,R_QX14,Buffer,10);
        WritePrivateProfileString("Datos", "dato14", Buffer, szName);
        GetDlgItemText(HWindow,R_QX15,Buffer,10);
        WritePrivateProfileString("Datos", "dato15", Buffer, szName);
        GetDlgItemText(HWindow,R_QX16,Buffer,10);
        WritePrivateProfileString("Datos", "dato16", Buffer, szName);
        GetDlgItemText(HWindow,R_QX17,Buffer,10);
        WritePrivateProfileString("Datos", "dato17", Buffer, szName);
        GetDlgItemText(HWindow,R_QX18,Buffer,10);
        WritePrivateProfileString("Datos", "dato18", Buffer, szName);
        GetDlgItemText(HWindow,R_QX19,Buffer,10);
        WritePrivateProfileString("Datos", "dato19", Buffer, szName);
        GetDlgItemText(HWindow,R_QX20,Buffer,10);
        WritePrivateProfileString("Datos", "dato20", Buffer, szName);
    }
    return;
}

void TRetorno::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "ret"))
    {
        GetPrivateProfileString("Datos", "dato1", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX1,Buffer);
        GetPrivateProfileString("Datos", "dato2", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX2,Buffer);
        GetPrivateProfileString("Datos", "dato3", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX3,Buffer);
        GetPrivateProfileString("Datos", "dato4", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX4,Buffer);
        GetPrivateProfileString("Datos", "dato5", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX5,Buffer);
        GetPrivateProfileString("Datos", "dato6", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX6,Buffer);
        GetPrivateProfileString("Datos", "dato7", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX7,Buffer);
        GetPrivateProfileString("Datos", "dato8", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX8,Buffer);
        GetPrivateProfileString("Datos", "dato9", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX9,Buffer);
        GetPrivateProfileString("Datos", "dato10", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX10,Buffer);
        GetPrivateProfileString("Datos", "dato11", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX11,Buffer);
        GetPrivateProfileString("Datos", "dato12", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX12,Buffer);
        GetPrivateProfileString("Datos", "dato13", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX13,Buffer);
        GetPrivateProfileString("Datos", "dato14", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX14,Buffer);
        GetPrivateProfileString("Datos", "dato15", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX15,Buffer);
        GetPrivateProfileString("Datos", "dato16", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX16,Buffer);
        GetPrivateProfileString("Datos", "dato17", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX17,Buffer);
        GetPrivateProfileString("Datos", "dato18", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX18,Buffer);
        GetPrivateProfileString("Datos", "dato19", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX19,Buffer);
        GetPrivateProfileString("Datos", "dato20", "", Buffer, 10, szName);
        SetDlgItemText(HWindow,R_QX20,Buffer);
    }
    return;
}

```



```

}

void TRetorno::ProcCalcular()
{
    int rx;
    double rqx[20];

    n=ret=wr=wl=ma=mar=mg=mgr=var=des=mu=mvar=med = 0.0;
    for(rx=0,rx<20,rx++)
    {
        rqx[rx]=0.0;
    }

    GetDlgItemText(HWindow,R_QX1,Buffer,10);
    rqx[0]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX2,Buffer,10);
    rqx[1]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX3,Buffer,10);
    rqx[2]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX4,Buffer,10);
    rqx[3]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX5,Buffer,10);
    rqx[4]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX6,Buffer,10);
    rqx[5]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX7,Buffer,10);
    rqx[6]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX8,Buffer,10);
    rqx[7]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX9,Buffer,10);
    rqx[8]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX10,Buffer,10);
    rqx[9]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX11,Buffer,10);
    rqx[10]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX12,Buffer,10);
    rqx[11]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX13,Buffer,10);
    rqx[12]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX14,Buffer,10);
    rqx[13]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX15,Buffer,10);
    rqx[14]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX16,Buffer,10);
    rqx[15]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX17,Buffer,10);
    rqx[16]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX18,Buffer,10);
    rqx[17]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX19,Buffer,10);
    rqx[18]=atof(Buffer);
    GetDlgItemText(HWindow,R_QX20,Buffer,10);
    rqx[19]=atof(Buffer);

    //Calculos comunes
    ma=0.0; mar=0.0; mg=0.0; mgr=0.0; var=0.0;des=0.0;
    mu=1.0,ret=0.0,wr=0.0,mvar=0.0;
    for(rx=0,rx<20,rx++)
    {
        if(rqx[rx]!=0.0)
            n=n+1.0;
    }
    if(n==0)
    {
        MessageBox(HWindow,"Debe ingresar datos para
calcular","OBSERVACION",MB_OK);
    }
}

```

```

        EnableWindow(GetDlgItem(HWindow, IDM_IMPRIMIR), FALSE);
    }
    else
    {
        EnableWindow(GetDlgItem(HWindow, IDM_IMPRIMIR), TRUE);

        for(rx=0;rx<20;rx++)
        {
            if(rqx[rx] != 0.0)
                r[rx]=rqx[rx]+1.0;
        }
        for(rx=0;rx<20;rx++)
        {
            if(rqx[rx] != 0.0)
            {
                wr=wr+rqx[rx];
                ret=ret+r[rx];
                mu=mu*r[rx];
                n=n+1.0;
            }
        }
        mar=ret/n;
        ma=mar-1.0;
        mg=pow(mu,1/n);
        mgr=mg-1.0;
        for(rx=0;rx<20;rx++)
        {
            if(rqx[rx] != 0.0)
            {
                wl=rqx[rx]-ma;
                mvar=mvar+pow(wl,2);
            }
        }
        var=mvar/n;
        des=sqrt(var);

        sprintf(SizeText, "\n Riqueza 1   %.4f \n Riqueza 2   %.4f \n Riqueza 3   %.4f \n
\n      Media Aritmética  Media Geométrica \n \n Riqueza Relativa  %.4f   %.4f \n Retorno
%.4f   %.4f \n \n \n Riesgo de retorno de media aritmética \n \n Desviación estándar   %.4f \n
Varianza
          %.4f ", r[0], r[1], r[2], mar, mg, ma, mgr, des, var);
        MessageBox(HWindow, SizeText, "CALCULOS TOTALES DEL RETORNO", MB_OK);
    }
}

```

## Módulo Simple

```

#include "simple.hpp"

void TSimple::ProcGrabar()
{
    if (PedirNombArchivo(TRUE, "sim"))
    {
        GetDlgItemText(HWindow, F_CS1, Buffer, 12);
        WritePrivateProfileString("Datos", "cs1", Buffer, szName);
        GetDlgItemText(HWindow, F_CS2, Buffer, 12);
        WritePrivateProfileString("Datos", "cs2", Buffer, szName);
        GetDlgItemText(HWindow, F_CS3, Buffer, 12);
        WritePrivateProfileString("Datos", "cs3", Buffer, szName);
        GetDlgItemText(HWindow, F_CS4, Buffer, 12);
        WritePrivateProfileString("Datos", "cs4", Buffer, szName);
        /* procesa archivo */
    }

    return;
}

void TSimple::ProcAbrir()
{
    if (PedirNombArchivo(FALSE, "sim"))

```

```

    }
        GetPrivateProfileString("Datos","cs1","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS1,Buffer);
        GetPrivateProfileString("Datos","cs2","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS2,Buffer);
        GetPrivateProfileString("Datos","cs3","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS3,Buffer);
        GetPrivateProfileString("Datos","cs4","",Buffer,12,szName);
        SetDlgItemText(HWindow,F_CS4,Buffer);
    }
    return;
}

void TSimple::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Flujo de Caja Simple");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

    CurrentLine+=2;
    strcpy( strLine, "Valor Presente:");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    sprintf(strLine,"%8.2f",vp);
    TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Valor Futuro:");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    sprintf(strLine,"%8.2f",vf);
    TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Número de Periodos:");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));

    sprintf(strLine,"%8.2f",np);
    TextOut( hPr, 10*Tab, CurrentLine*LineSpace, strLine,
    strlen(strLine));
}

```

```

        CurrentLine+=2;
        strcpy( strLine, "Tasa de Descuento:");
        TextOut(  hPr,  2*Tab,  CurrentLine*LineSpace,  strLine,
strlen(strLine));

        sprintf(strLine,"%8.2f",r);
        TextOut(  hPr,  10*Tab,  CurrentLine*LineSpace,  strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
        "InvestMaster" para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TSimple::ProcCalcular()
{
    int cs;
    int n=0;

    np=ra=r=n1=n2=e=vpp=vp=vpf=lp=r1=r2 = 0.0;
    for(cs=1;cs<5;cs++)
    {
        fcs[cs]=0.0;
    }

    GetDlgItemText(HWindow,F_CS1,Buffer,12);
    fcs[1]=atof(Buffer);
    GetDlgItemText(HWindow,F_CS2,Buffer,12);
    fcs[2]=atof(Buffer);
    GetDlgItemText(HWindow,F_CS3,Buffer,12);
    fcs[3]=atof(Buffer);
    GetDlgItemText(HWindow,F_CS4,Buffer,12);
    fcs[4]=atof(Buffer);

    for(cs=1;cs<5;cs++)
    {
        if(fcs[cs]!=0.0)
            n=n+1;
    }

    if (n==4)
    {
        MessageBox(HWindow,"Ingreso todos los datos", "OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }

    if(n==0 | n<3)
    {
        MessageBox(HWindow,"Debe ingresar por lo menos 3
datos", "OBSERVACION",MB_OK);
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    }
    else
    {
        EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
    }
}

```





```

        WritePrivateProfileString("Datos","st2",Buffer,szName);
        GetDlgItemText(HWindow,F_IR3,Buffer,8);
        WritePrivateProfileString("Datos","st3",Buffer,szName);
        GetDlgItemText(HWindow,F_IR4,Buffer,8);
        WritePrivateProfileString("Datos","st4",Buffer,szName);
        GetDlgItemText(HWindow,F_IR5,Buffer,8);
        WritePrivateProfileString("Datos","st5",Buffer,szName);
        GetDlgItemText(HWindow,F_IR6,Buffer,8);
        WritePrivateProfileString("Datos","st6",Buffer,szName);
        GetDlgItemText(HWindow,F_IR7,Buffer,8);
        WritePrivateProfileString("Datos","st7",Buffer,szName);
        GetDlgItemText(HWindow,F_IR8,Buffer,8);
        WritePrivateProfileString("Datos","st8",Buffer,szName);
        GetDlgItemText(HWindow,F_IR9,Buffer,8);
        WritePrivateProfileString("Datos","st9",Buffer,szName);
        GetDlgItemText(HWindow,F_IR10,Buffer,8);
        WritePrivateProfileString("Datos","st10",Buffer,szName);
        GetDlgItemText(HWindow,F_IR11,Buffer,8);
        WritePrivateProfileString("Datos","st11",Buffer,szName);
    }
    return;
}

void TStock::ProcAbrir()
{
    if( PedirNombArchivo( FALSE,"sto" ))
    {
        GetPrivateProfileString("Datos","st1","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR1,Buffer);
        GetPrivateProfileString("Datos","st2","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR2,Buffer);
        GetPrivateProfileString("Datos","st3","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR3,Buffer);
        GetPrivateProfileString("Datos","st4","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR4,Buffer);
        GetPrivateProfileString("Datos","st5","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR5,Buffer);
        GetPrivateProfileString("Datos","st6","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR6,Buffer);
        GetPrivateProfileString("Datos","st7","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR7,Buffer);
        GetPrivateProfileString("Datos","st8","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR8,Buffer);
        GetPrivateProfileString("Datos","st9","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR9,Buffer);
        GetPrivateProfileString("Datos","st10","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR10,Buffer);
        GetPrivateProfileString("Datos","st11","",Buffer,8,szName);
        SetDlgItemText(HWindow,F_IR11,Buffer);
    }
    return;
}

void TStock::ProcImprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();
    // GetDlgItemText( HWindow, ID_CONTROL, strLine, 120);
}

```

```

strcpy( strLine, "InvestMaster Para Windows.(Versión 1.0)");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine++;
strcpy( strLine, "MODULO: Valoración de Stock");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3);

MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+8);
MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+9);

CurrentLine+=2;
strcpy( strLine, "RESULTADOS OBTENIDOS:");
TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Dividendo 1 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",fir[1]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Dividendo 2 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",fir[2]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Dividendo 3 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",fir[3]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Dividendo 4 :");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",fir[4]);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Precio Corriente =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",pc);
TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

CurrentLine+=2;
strcpy( strLine, "Tasa de descuento =");
TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

sprintf(strLine,"%15.4f",k);

```



```

strlen(strLine));
    TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
    CurrentLine+=2;
    strcpy( strLine, "Tasa de crecimiento =");
    TextOut( hPr, 2*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
    sprintf(strLine,"%15.4f",g);
    TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2),
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+2),
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3),
    LineTo(hPr,21*Tab, (CurrentLine+1)*LineSpace+3),

    Pr.EndPage();
    Pr.EndDoc();
    DeleteDC(hPr);
    MessageBox( HWindow, "Impresión finalizada con éxito ",
    "InvestMaster" para
Windows 1.0", MB_OK | MB_ICONINFORMATION );
}

void TStock::ProcCalcular()
{
int ir,n=0;

    pc=pcpre=den=elev=k=g=con=cosa=0.00;
    for(ir=1;ir<12;ir++)
        fir[ir]=0.0;

        GetDlgItemText(HWindow,F_IR1,Buffer,8);
        fir[1]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR2,Buffer,8);
        fir[2]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR3,Buffer,8);
        fir[3]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR4,Buffer,8);
        fir[4]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR5,Buffer,8);
        fir[5]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR6,Buffer,8);
        fir[6]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR7,Buffer,8);
        fir[7]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR8,Buffer,8);
        fir[8]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR9,Buffer,8);
        fir[9]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR10,Buffer,8);
        fir[10]=atof(Buffer);
        GetDlgItemText(HWindow,F_IR11,Buffer,8);
        fir[11]=atof(Buffer);

    for(ir=1;ir<12;ir++)
    {
        if(fir[ir]!=0.0)
            n=n+1;
    }
    if(n==0 | n<6 | | (fir[10]==0.0 && fir[11]==0.0))
    {
        MessageBox(HWindow,"Debe ingresar datos para calcular","OBSERVACION",MB_OK);
    }
}

```



```

        EnableWindow(GetDlgItem(HWindow, IDM_IMPRIMIR), FALSE);
    }
    else
    {
        EnableWindow(GetDlgItem(HWindow, IDM_IMPRIMIR), TRUE);
        if (fir[10]!=0.0 && fir[11]!=0.0)
        {
            g=fir[11], k=fir[10];
            for(ir=1,ir<n-2,ir++)
            {
                elev=pow((1.0+k),ir);
                pcpre=pcpre+fir[ir]/elev;
            }
            cosa=pow((1.0+k),n-3);
            den=(k-g)*cosa;
            con=fir[n-2]/den;
            pc=pcpre+con;
            sprintf(SizeText,"El Precio Corriente es %.4f",pc);
            MessageBox(HWindow,SizeText,"CALCULOS",MB_OK);
        }
    }
}
}
}

```

## Módulo Tasa

```
#include "tasa.hpp"
```

```
void TTasa::ProcGrabar()
```

```
{
    if( PedirNombArchivo( TRUE, "tas" ))
    {
        GetDlgItemText(HWindow,A_NU1,Buffer,10);
        WritePrivateProfileString("Datos","anu1",Buffer,szName);
        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        WritePrivateProfileString("Datos","anu2",Buffer,szName);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        WritePrivateProfileString("Datos","anu3",Buffer,szName);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        WritePrivateProfileString("Datos","anu4",Buffer,szName);
    }
    return;
}

```

```
void TTasa::ProcAbrir()
```

```
{
    if( PedirNombArchivo( FALSE, "tas" ))
    {
        GetPrivateProfileString("Datos","anu1","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU1,Buffer);
        GetPrivateProfileString("Datos","anu2","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU2,Buffer);
        GetPrivateProfileString("Datos","anu3","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU3,Buffer);
        GetPrivateProfileString("Datos","anu4","",Buffer,10,szName);
        SetDlgItemText(HWindow,A_NU4,Buffer);
    }
}

```

```
void TTasa::ProcCalcular()
```

```

{
    int a,z=0;
    num=den=fraccion=term=ntc=dtc=tasa=d=c=elev=0.0;

    for(a=1;a<5;a++)
        ta[a]=0.0;

        GetDlgItemText(HWindow,A_NU1,Buffer,10);
        ta[1]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU2,Buffer,10);
        ta[2]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU3,Buffer,10);
        ta[3]=atof(Buffer);
        GetDlgItemText(HWindow,A_NU4,Buffer,10);
        ta[4]=atof(Buffer);

    for(a=1;a<5;a++)
    {
        if(ta[a]!=0.0)
            z=z+1;
    }

    if (z==0 || z<3)
    //{
        MessageBox(HWindow,"Debe ingresar datos para calcular","OBSERVACION",MB_OK);
        // EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), FALSE);
    // }
    else
    {
        // EnableWindow(GetDlgItem(HWindow,IDM_IMPRIMIR), TRUE);
        d=ta[4];
        c=ta[3];
        if ((d-c==0.0) || (d<c))
        {
            MessageBeep(-1);
            MessageBox(HWindow,"El tiempo de vencimiento período largo\n no puede ser
menor o igual al \n tiempo de vencimiento período corto","MENSAJE DE
ERROR",MB_OK|MB_ICONHAND);
        }
        else
        {
            num=1.0+ta[2]; den=1.0+ta[1];
            ntc=pow(num,d);
            dtc=pow(den,c);
            elev=d-c;
            fraccion= ntc/dtc;
            term=pow(fraccion,1/elev);
            tasa=term-1.0;

            sprintf(SizeText," \n La tasa de crecimiento del mes %.0f al mes %.0f es %.4f
\n",c,d,tasa);
            MessageBox(HWindow,SizeText," CALCULOS DE TASA DE
CRECIMIENTO ",MB_OK);
        }
    }
}

```

```

}

void TTasa::Proclmprimir()
{
    HANDLE hLibrary;
    HDC hPr;
    int CurrentLine= 5;
    int LinesPerPage;
    int LineSpace;
    int Tab;

    TPrint Pr;
    hPr= Pr.GetDC();
    LineSpace= Pr.GetLineSpace();
    Tab= Pr.GetTab();
    Pr.StartDoc();
    Pr.StartPage();

    strcpy( strLine, "InvestMaster Para Windows.(Versión
1.0)");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, "MODULO: Tasa de Interés");
    TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
    LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+2);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
    LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+3);

    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+8);
    LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+8);
    MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+9);
    LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+9);
    CurrentLine+=2;
    strcpy( strLine, "Bono de Periodo");
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    strcpy( strLine, "Bono de periodo");
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine++;
    strcpy( strLine, " Corto");
    TextOut( hPr, 11*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    strcpy( strLine, " Largo");
    TextOut( hPr, 17*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

    CurrentLine+=2;
    strcpy( strLine, "Retorno al vencimiento :");

```

```

        TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%2.4f",ta[1]);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%2.4f",ta[2]);
        TextOut( hPr, 18*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Tiempo de vencimiento en meses ");
        TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%3.0f",c);
        TextOut( hPr, 12*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%3.0f",d);
        TextOut( hPr, 18*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        strcpy( strLine, "Tasa de crecimiento:");
        TextOut( hPr, Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));
        sprintf(strLine,"%2.4f",tasa);
        TextOut( hPr, 18*Tab, CurrentLine*LineSpace, strLine,
strlen(strLine));

        CurrentLine+=2;
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+2);
        LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+2);
        MoveTo(hPr,Tab, (CurrentLine+1)*LineSpace+3);
        LineTo(hPr,23*Tab, (CurrentLine+1)*LineSpace+3);

        Pr.EndPage();
        Pr.EndDoc();
        DeleteDC(hPr);
        MessageBox( HWindow, "Impresión finalizada con éxito.",
                    "InvestMaster para Windows 1.0", MB_OK |
MB_ICONINFORMATION );
    }

```