

## Programación II-Tarea Final (Prototipo Sistema de Préstamos)

- I- Realice los siguientes:
- a) Un Menú con los siguientes opciones:
- 1- Mantenimientos
    - a) De Usuarios
    - b) De Garantía
    - c) De Fiador
    - d) De Cliente
    - e) Préstamos
  - 2- Movimiento
    - a) De Cobros
  - 3- Procesos
    - a) Actualizar Cobros
    - b) Reversar Cobros
  - 4- Consultas
    - a) De Garantías
    - b) De Usuarios
    - c) De Fiador
    - d) De Cliente
    - e) De cliente por nombres
    - f) De Préstamos
    - g) De préstamos por Fecha
    - h) De Préstamos por Fecha de Vencimiento
    - i) De Préstamos con Saldo Pendiente
    - j) De cobro por Fecha
    - k) De cobro por Vencido

Tablas a usar:

Usuarios	
Login_Usuario	String *
Pass_Usuario	String *
Nivel_Acceso	Integer *
Nombre_Usuario	String *
Apellidos_Usuarios	String *
Email_Usuario	String

Garantía	
Id_Garantia	Integer *
Descripcion_Garantia	String *

Fiador	
Id_Fiador	Integer *
Nombre_Fiador	String *
Apellido_Pat_Fiador	String *
Apellido_Mat_Fiador	String *
Dirección_Fiador	String *
Cedula_Fiador	String *
Telef_Fiador	Integer *
Empresa_Fiador	String *
Ocupacion_Fiador	String *
Sueldo_Fiador	Double *
Sexo_Fiador	Boolean *

Cliente
---------

## Programación II-Tarea Final (Prototipo Sistema de Préstamos)

<b>Id_Cte</b>	Integer *
Nombre_Cte	String *
Apellido_Pat_Cte	String *
Apellido_Mat_Cte	String *
<b>Cedula_Cte</b>	String *
Dirección_Cte	String *
Telef_Cte	Integer *
Empresa_Cte	String *
Ocupacion_Cte	String *
Sueldo_Cte	Double *
Sexo_Cte	Boolean *
Id_Cte_Fiador	Integer

Préstamos	
<b>Id_Prestamo</b>	Integer *
Id_Cte_Prestamo	Integer *
Id_Fiador_Prestamo	Integer
Estado_Prestamo	Boolean *
Fecha_Inicio_Prestamo	date *
Plazo_Prestamo	Integer *
Fecha_Final_Prestamo	date *
<b>Tipo_Garantia_Prestamo</b>	Integer *
Monto_Prestamo	Double *
Balance_Prestamo	Double
Tasa_Prestamo	Double *
Cuota_fija_Prestamo	Double
Foto_Garantia	String

Cuota_Prestamo	
Id_Prestamo_Cuota	<b>Id_Prestamo</b>
Num_Cuota	Integer
Id_Cuota_Prestamo_Cte	Id_Cte_Prestamo
Fecha_Cuota	Date
Valor_Cuota	Double
Valor_Amortizacion_Cuota	Double
Valor_Interes_Cuota	Double
Status_Cuota	Boolean

Detalle_Cobro_Prestamo	
<b>Id_Detalle_Cobro</b>	Id_Cobro *
Sec_Cobro	Integer *
Id_Prestamo_Cuota	Id_Prestamo_Cuota *
<b>Sec_Cuota</b>	Num_Cuota *
Valor_Cuota_Cobro	Double *
Valor_Capital	Double *
Valor_Interes	Double *
Valor_Mora	Double *

Cobro_Prestamo	
<b>Id_Cobro</b>	integer *
Fecha_Cobro	Date *
Id_Cliente_Cobro	Id_Cuota_Prestamo_Cte *
Valor_Cobro	Double *
Concepto_Cobro	String *
Status_cobro	Boolean

# Programación II-Tarea Final (Prototipo Sistema de Préstamos)

Nota:

## a) Mantenimiento de Usuarios

### ○ Requisitos:

- i. Se deben validar los atributos obligatorios
- ii. Cree un evento al momento de digitar el Login y el Password donde se valide si existe en el archivo de Usuarios. Si existe permitir ir al menú Principal.
- iii. Si el usuario es Administrador se activa un Botón “Registrarse” donde se permite crear o modificar los datos de los usuarios. El nivel de acceso es (0,1). Cero para el Administrador y uno para los usuarios normales. Los usuarios normales no pueden tener acceso a mantenimiento de Usuarios, ni Generar y reversar cobros
- iv. Luego de validar los atributos presione el botón “Guardar” donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos.

## b) Mantenimiento de Garantía

### ○ Requisitos:

- i. Se deben validar los atributos obligatorios
- ii. Cree un evento al momento de digitar la **Id\_Garantía** donde se valide si existe en el archivo de Garantía. Si existe desplegar el atributo Descripción\_Garantia en la ventana y un mensaje que diga “Modificando”, de lo contrario desplegar un mensaje que diga “Creando” y permita digitar el atributo solicitado.
- iii. Luego de validar los atributos presione el botón “Guardar” donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos.

## c) Mantenimiento de Fiador

### ○ Requisitos:

- iv. Se deben validar los atributos obligatorios
- v. Cree un evento al momento de digitar la **Id\_Fiador** donde se valide si existe en el archivo de Fiador. Si existe desplegar los atributos en la ventana y un mensaje que diga “Modificando”, de lo contrario desplegar un mensaje que diga “Creando” y permita digitar el atributo solicitado.
- vi. Luego de validar los atributos presione el botón “Guardar” donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos.

## Programación II-Tarea Final (Prototipo Sistema de Préstamos)

### d) Mantenimiento de Cliente

- Requisitos:
  - vii. Se deben validar los atributos obligatorios
  - viii. Cree un evento al momento de digitar la **Id\_Cte** donde se valide si existe en el archivo de Cliente. Si existe desplegar los atributos en la ventana y un mensaje que diga “Modificando”, de lo contrario desplegar un mensaje que diga “Creando” y permita digitar los atributos solicitados.
  - I. Si se digita el atributo **Id\_Cte\_Fiador** se debe verificar que existe en el archivo de Fiador y desplegar el nombre del fiador.
  - II. Luego de validar los atributos presione el botón “Guardar” donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos.

### e) Mantenimiento de Préstamo

- Requisitos:
  - ix. Se deben validar los atributos obligatorios
  - x. Cree un evento al momento de digitar la **Id\_Prestamo** donde se valide si existe en el archivo de Prestamo. Si existe y el atributo **Estado\_Prestamo** es false desplegar los atributos en la ventana y un mensaje que diga “Modificando” (Si el préstamo tiene cuota paga no se puede modificar), de lo contrario desplegar un mensaje que diga “Creando” y permita digitar los atributos solicitados.
  - III. El atributo **Id\_Cte\_Prestamo** se debe verificar que existe en el archivo de Cliente.
  - IV. Si se digita el atributo **Id\_Fiador\_Prestamo** se debe verificar que existe en el archivo de Fiador y desplegar el nombre del fiador.
  - V. En atributo **Fecha\_Inicio\_Prestamo** se tomará del sistema.
  - VI. El atributo **Plazo\_Prestamo** se digitará en meses y la **Fecha\_Final\_Prestamo** se calculará tomando en cuenta la Fecha de Inicio de Préstamo y el Plazo del Préstamo.
  - VII. El atributo **Tipo\_Garantia\_Prestamo** se debe verificar que existe en el archivo de Garantía.
  - VIII. A la hora de crear el préstamo el **Balance\_Prestamo** es igual al **Monto\_Prestamo**
  - IX. Luego de validar los atributos presione el botón “Guardar” donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos. Además se deben crear las cuotas con todos sus datos en el archivo de **Cuota\_Prestamo** y generar una tabla de amortización en formato PDF

## Programación II-Tarea Final (Prototipo Sistema de Préstamos)

### f) Movimiento de Cobros

#### ○ Requisitos:

- xi. Se deben validar los atributos obligatorios
- xii. Cree un evento al momento de digitar la **Id\_Cobro** donde se valide si existe en el archivo de Cobro\_Prestamo. Si existe y el atributo **Status\_Cobro** es false desplegar los atributos en la ventana y un mensaje que diga "Modificando" , de lo contrario desplegar un mensaje que diga "Creando" y permita digitar los atributos solicitados.
- xiii. En atributo **Fecha\_Cobro** se tomará del sistema.
- X. El atributo **Id\_Cte\_Cobro** se debe verificar que existe en el archivo de Cliente y Cuota\_Prestamo (Se despliegan todas las cuotas vencidas a la fecha con los datos del archivo **Detalle\_Cobro\_Prestamo**).
- XI. Luego de validar los atributos presione el botón "Guardar" donde se ejecuta un evento que permitirá almacenar en un archivo tipo txt dichos atributos. Además el **Status\_Cuota** se le asigna False y generar un recibo de cobro en formato PDF

### g) Procesos:

#### 1- Generar Cobros

#### ○ Requisitos:

- I. Se solicitará el rango de Fechas para actualizar los cobros y se valida que el atributo Status\_Cobro sea igual False para poder actualizar sus registros. Si el Status\_Cobro es True no se actualiza ese registro.
- II. Este proceso se encargará de actualizar los cobros realizados a los diferentes préstamos en ese rango de Fechas.

Si las cuotas quedan saldas el status\_cuota se le asigna true y se actualiza el atributo Balance\_Prestamo en el archivo de Préstamo.

Si el Balance\_Prestamo a la hora de actualizarlo queda en cero el atributo Estado\_Prestamo se le asigna true.

- III. Asignar True al atributo Status\_Cobro en el archivo de Cobro\_Prestamo.

#### 2- Reversar Cobros

#### ○ Requisitos:

- IV. Se solicitará el rango de Fecha para reversar los Cobros. Si en el cobro de esa fecha no existe presentar mensaje de Error.
- V. Este proceso se encargará de actualizar todos los registros correspondientes a dicha fecha de reverso y se debe sumar del Balance\_Prestamo en el archivo Préstamo.