



**INGENIERÍA EN  
COMPUTACIÓN**  
**PARADIGMAS DE  
PROGRAMACIÓN I**  
**EVALUACIÓN ORDINARIA**



**DOCUMENTACIÓN DEL PROYECTO DE  
DESARROLLO DE UN PROGRAMA QUE ACTUALIZA  
EL SALDO DE UN CLIENTE AL REALIZAR UNA  
COMPRA CON TARJETA BANCARIA.**

Autor: Rodrigo Díaz Salguero

Catedrático: Ing. Alberto Cano García

Tercer Semestre

Ciclo 2024B

Ixtlahuaca, México a 04/12/2024

- Observando las especificaciones del proyecto, se logran identificar los entes sobre los que se va a trabajar que son las tarjetas bancarias, igualmente observando las especificaciones podemos observar cuáles deben ser los atributos de las clases y se pueden deducir los métodos que necesita cada una de estas, de igual manera al observar la construcción y diseño de nuestras clases se observa la necesidad de hacer uso de la herencia para organizar de una mejor manera nuestro diagrama de clases y tener un trabajo más limpio, por lo tanto se creó una super clase de la cuál heredan tres subclases. A continuación, se observa cada una de las clases modelo, la clase vista y la clase de control.
- Superclase CL\_TARJ\_BANCARIA y su seudocódigo

## **CL\_TARJ\_BANCARIA**

```
#saldo_bancario : REAL  
#no_tarjeta : REAL
```

```
set_saldo_bancario(x:REAL)  
set_no_tarjeta(x:CADENA):BOOLEANO  
get_saldo_bancario:REAL  
get_no_tarjeta:CADENA
```

```

CLASE CL_TARJ_BANCARIA
/*
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
*/
INICIO
    SECCIÓN DE ATRIBUTOS
        saldo_bancario : REAL, PROTEGIDO
        no_tarjeta : CADENA, PROTEGIDO
    SECCIÓN DE MÉTODOS

        MÉTODO set_saldo_bancario(x:REAL)
        INICIO
            saldo_bancario <-- x
        FIN MÉTODO set_saldo_bancario

        MÉTODO set_no_tarjeta(x:CADENA) :BOOLEANO
        SECCIÓN DE VARIABLES
            long_tarj : ENTERO
        INICIO
            long_tarj <-- LONGITUD(x)
            SI(long_tarj=16) HACER
                no_tarjeta <-- x
                REGRESAR VERDADERO
            SI NO
                REGRESAR FALSO
        FIN MÉTODO set_no_tarjeta

        MÉTODO get_saldo_bancario:REAL
        INICIO
            REGRESAR saldo_bancario
        FIN MÉTODO get_saldo_bancario

        MÉTODO get_no_tarjeta:CADENA
        INICIO
            REGRESAR no_tarjeta
        FIN MÉTODO get_no_tarjeta

FIN CLASE CL_TARJ_BANCARIA

```

- Subclase CL\_TARJ\_CRED y su seudocódigo

CL_TARJ_CRED
<pre>- lim_cred : REAL - taza_int : REAL</pre>
<pre>set_lim_cred(x:REAL) set_taza_int(x:REAL) get_lim_cred:REAL get_taza_int:REAL calc_monto_total(monto_compra:REAL):REAL comparar_saldo_y_credito(monto_total:REAL):BOOLEANO actualizar_saldo(monto_total:REAL)</pre>

```
CLASE CL_TARJ_CRED DERIVADA DE CL_TARJ_BANCARIA
/*
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
*/
INICIO
    SECCIÓN DE ATRIBUTOS
        lim_cred : REAL, PRIVADO
        taza_int : REAL, PRIVADO
    SECCIÓN DE MÉTODOS

        MÉTODO set_lim_cred(x:REAL)
        INICIO
            lim_cred <-- x
        FIN MÉTODO set_lim_cred

        MÉTODO set_taza_int(x:REAL)
        INICIO
            taza_int <-- x
        FIN MÉTODO set_taza_int

        MÉTODO get_lim_cred:REAL
        INICIO
            REGRESAR lim_cred
        FIN MÉTODO get_lim_cred

        MÉTODO get_taza_int:REAL
        INICIO
            REGRESAR taza_int
        FIN MÉTODO get_taza_int

        //Calcula el monto de la compra con interés
        MÉTODO calc_monto_total(monto_compra:REAL):REAL
```

```

SECCIÓN DE VARIABLES
    monto_total : REAL
INICIO
    monto_total <-- monto_compra + (monto_compra *
taza_int/100)
    REGRESAR monto_total
FIN MÉTODO calc_monto_total

//Calcula un hipotético nuevo saldo y lo compara con el
lim_cred
MÉTODO comparar_saldo_y_credito(monto_total:REAL):BOOLEANO
SECCIÓN DE VARIABLES
    v_comparacion : BOOLEANO
    v_nuevo_saldo : REAL
INICIO
    v_nuevo_saldo <-- saldo_bancario + monto_total
    SI (v_nuevo_saldo <= lim_cred) HACER
        v_comparacion <-- VERDADERO
    SI NO
        v_comparacion <-- FALSO
    FIN (SI)
    REGRESAR v_comparacion
FIN MÉTODO comparar_saldo_y_credito

//Actualiza el saldo bancario
MÉTODO actualizar_saldo(monto_total:REAL)
INICIO
    saldo_bancario <-- saldo_bancario + monto_total
FIN MÉTODO actualizar_saldo

FIN CLASE CL_TARJ_CRED

```

- Subclase CL\_TARJ\_DEB y su seudocódigo

CL_TARJ_DEB
<pre>- p_sobregiro:REAL - capacidad_pago:REAL</pre>
<pre>set_p_sobregiro(x:REAL) set_capacidad_pago(x:REAL) get_p_sobregiro:REAL get_capacidad_pago:REAL calc_capacidad_pago comparar_monto_y_cap_pago(monto_compra:REAL):BOOLEANO actualizar_saldo(monto_compra:REAL)</pre>

```
CLASE CL_TARJ_DEB DERIVADA DE CL_TARJ_BANCARIA
/*
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
*/
INICIO
    SECCIÓN DE ATRIBUTOS
        p_sobregiro : REAL, PRIVADO
        capacidad_pago : REAL, PRIVADO
    SECCIÓN DE MÉTODOS

        MÉTODO set_p_sobregiro(x:REAL)
        INICIO
            p_sobregiro <-- x
        FIN MÉTODO set_p_sobregiro

        MÉTODO set_capacidad_pago(x:REAL)
        INICIO
            capacidad_pago <-- x
        FIN MÉTODO set_capacidad_pago

        MÉTODO get_p_sobregiro:REAL
        INICIO
            REGRESAR p_sobregiro
        FIN MÉTODO get_p_sobregiro

        MÉTODO get_capacidad_pago:REAL
        INICIO
            REGRESAR capacidad_pago
        FIN MÉTODO get_capacidad_pago
```

```

    //Calcula la capacidad de pago con sobregiro
    MÉTODO calc_capacidad_pago
        capacidad_pago <-- saldo_bancario +
(saldo_bancario*p_sobregiro/100)
        FIN MÉTODO calc_capacidad_pago

    //Compara monto de compra con capacidad de pago
    MÉTODO
comparar_monto_y_cap_pago(monto_compra:REAL) :BOOLEANO
    SECCIÓN DE VARIABLES
        v_comparacion : BOOLEANO
    INICIO
        SI (monto_compra <= capacidad_pago) HACER
            v_comparacion <-- VERDADERO
        SI NO
            v_comparacion <-- FALSO
        FIN (SI)
        REGRESAR v_comparacion
    FIN MÉTODO comparar_monto_y_cap_pago

    //Actualiza el saldo bancario
    MÉTODO actualizar_saldo(monto_compra:REAL)
    INICIO
        saldo_bancario <-- saldo_bancario - monto_compra
    FIN MÉTODO actualizar_saldo

FIN CLASE CL_TARJ_DEB

```

- Subclase CL\_TARJ\_PREP y su seudocódigo

### CL\_TARJ\_PREP

```
comparar_monto_y_saldo(monto_compra:REAL):BOOLEANO
actualizar_saldo(monto_compra:REAL)
```

CLASE CL\_TARJ\_PREP DERIVADA DE CL\_TARJ\_BANCARIA  
/\*

NOMBRE: RODRIGO DÍAZ SALGUERO

FECHA: 02/12/2024

ACTUALIZACIÓN: ---

\*/

INICIO

SECCIÓN DE MÉTODOS

MÉTODO comparar\_monto\_y\_saldo(monto\_compra:REAL) :BOOLEANO  
SECCIÓN DE VARIABLES

v\_comparacion : BOOLEANO

INICIO

SI (monto\_compra <= saldo\_bancario) HACER

v\_comparacion <-- VERDADERO

SI NO

v\_comparacion <-- FALSO

FIN (SI)

REGRESAR v\_comparacion

FIN MÉTODO comparar\_monto\_y\_saldo

MÉTODO actualizar\_saldo(monto\_compra:REAL)

INICIO

saldo\_bancario <-- saldo\_bancario - monto\_compra

FIN MÉTODO actualizar\_saldo

FIN CLASE CL\_TARJ\_PREP

- Clase CL\_VISTA y su seudocódigo

CL_VISTA
- mensaje : CADENA - dato : CADENA

set_mensaje(x:CADENA) set_dato(x:CADENA) get_mensaje : CADENA get_dato : CADENA informar(x:ENTERO, y:ENTERO) recibir(x:ENTERO, y:ENTERO) crear_marco(titulo_marco:CADENA)
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
CLASE CL_VISTA
/*
Autor: Rodrigo Díaz Salguero
Fecha: 16/10/2024
Actualización: ---
*/
INICIO
    SECCIÓN DE ATRIBUTOS
        mensaje : CADENA
        dato : CADENA
    SECCIÓN DE MÉTODOS

        MÉTODO set_mensaje(x:CADENA)
        INICIO
            mensaje <-- x
        FIN MÉTODO set_mensaje

        MÉTODO set_dato(x:CADENA)
        INICIO
            dato <-- x
        FIN MÉTODO set_dato

        MÉTODO get_mensaje:CADENA
        INICIO
            REGRESAR mensaje
        FIN MÉTODO get_mensaje

        MÉTODO get_dato:CADENA
        INICIO
            REGRESAR dato
        FIN MÉTODO get_dato
```

```

MÉTODO informar(x:ENTERO,y:ENTERO)
INICIO
    POSICIONAR(x,y)
    ESCRIBIR mensaje
FIN MÉTODO informar

MÉTODO recibir(x:ENTERO,y:ENTERO)
INICIO
    POSICIONAR(x,y)
    LEER(dato)
FIN MÉTODO recibir

MÉTODO crear_marco(titulo_marco:CADENA)
SECCIÓN DE VARIABLES
    v_asteriscos_80 : CADENA
    v_aste_esp_aste : CADENA
INICIO
    //BORRA LA PANTALLA ANTES DE CREAR UN MARCO NUEVO
    BORRAR

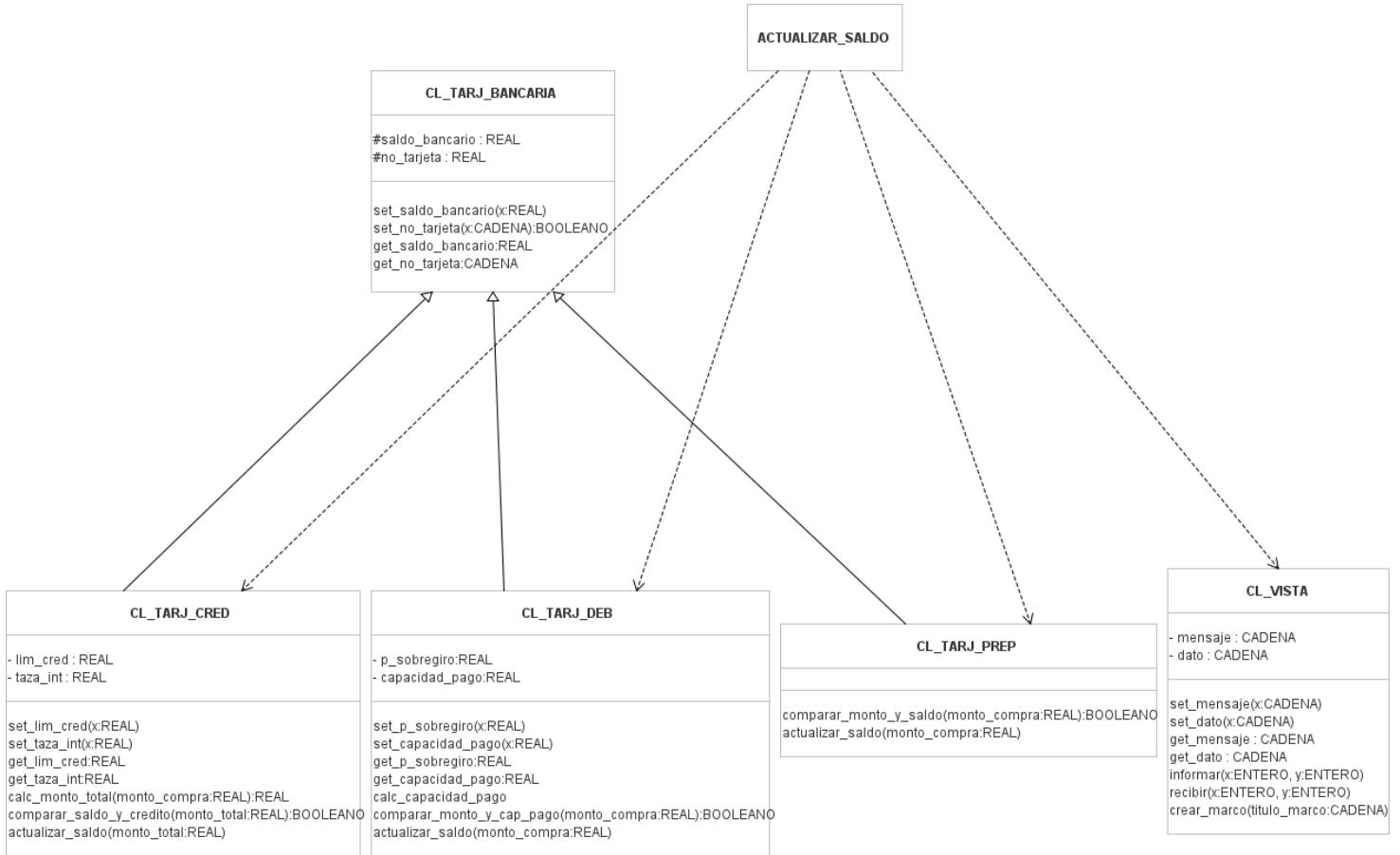
    v_asteriscos_80 <--
*****"*****
    v_aste_esp_aste <-- "*"
    POSICIONAR(1,5)
    ESCRIBIR v_asteriscos_80
    POSICIONAR(1,6)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,7)
    ESCRIBIR titulo_marco
    POSICIONAR(1,8)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,9)
    ESCRIBIR v_asteriscos_80
    POSICIONAR(1,10)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,11)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,12)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,13)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,14)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,15)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,16)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,17)
    ESCRIBIR v_aste_esp_aste
    POSICIONAR(1,18)

```

```
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,19)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,20)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,21)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,22)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,23)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,24)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,25)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,26)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,27)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,28)
ESCRIBIR v_aste_esp_aste
POSICIONAR(1,29)
ESCRIBIR v_asteriscos_80
FIN MÉTODO crear_marco

FIN CLASE CL_VISTA
```

○ Diagrama de clases



- Posterior a nuestro diagrama de clases tenemos nuestro formato de salida.

```

*****
*
*
*                               COMPRAR
*
*
*                               SELECCIONE LA OPCIÓN QUE CORRESPONDA A SU TARJETA BANCARIA
*
*
*                               1.- TARJETA DE CRÉDITO
*
*                               2.- TARJETA DE DÉBITO
*
*                               3.- TARJETA DE PREPAGO
*
*                               4.- SALIR
*
*
*                               DIGITE LA OPCIÓN QUE ELIGIÓ:
*
*
*****
  
```

## [COMPRAR](#)

INGRESE SU NÚMERO DE TARJETA:

## TARJETA DE CRÉDITO

## LÍMITE DE CRÉDITO:

### TAZA DE INTERÉS MENSUAL:

## SALDO ACTUAL:

MONTO DE LA COMPRA:

#### MONTO MÁS INTERÉS:

SU SALDO SE HA ACTUALIZADO CORRECTAMENTE A

[PRESTONA ENTER PARA REGRESAR AL MENU PRINCIPAL](#)

\*\*\*\*\*  
\*  
\* TARJETA DE DÉBITO  
\*  
\*\*\*\*\*  
\*  
\*  
\* PORCENTAJE DE CAPACIDAD DE SOBREGIRO:  
\*  
\*  
\* SALDO ACTUAL:  
\*  
\*  
\* MÁXIMA CAPACIDAD DE PAGO:  
\*  
\*  
\* MONTO DE LA COMPRA:  
\*  
\*  
\* SU SALDO SE HA ACTUALIZADO CORRECTAMENTE A  
\*  
\*  
\*  
\*\*\*\*\*

\*\*\*\*\*  
\*  
\* TARJETA DE PREPAGO  
\*  
\*\*\*\*\*  
\*  
\*  
\*  
\*  
\*  
\*  
\* SALDO ACTUAL:  
\*  
\*  
\*  
\*  
\* MONTO DE COMPRA:  
\*  
\*  
\*  
\* SU SALDO NO SE HA ACTUALIZADO CORRECTAMENTE POR SOBREPASAR EL SALDO  
\*  
\*  
\*  
\*  
\*  
\*  
\*  
\*\*\*\*\*

- El diagrama de secuencias se adjuntará junto con este archivo ya que es muy grande y pierde mucho la calidad de imagen.
- Seudocódigo de la clase de control.

```

PROGRAMA ACTUALIZAR_SALDO
/*
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
*/
INICIO
    SECCIÓN DE VARIABLES
        titulo_marco : CADENA
        v_opc : ENTERO
        v_no_tarj : CADENA
        v_valid_tarj : BOOLEANO
        v_lim_cred : REAL
        v_taza_int : REAL
        v_saldo : REAL
        v_monto_c : REAL
        v_monto_tot : REAL
        v_comparacion : BOOLEANO
        v_sobregiro : REAL
        v_cap_pago : REAL
    MÉTODO PRINCIPAL
    INICIO
        interfaz : NUEVO OBJETO DE CL_VISTA
        obj_cred : NUEVO OBJETO DE CL_TARJ_CRED
        obj_deb : NUEVO OBJETO DE CL_TARJ_DEB
        obj_prep : NUEVO OBJETO DE CL_TARJ_PREP

        REPETIR
            titulo_marco <-- "*"
            COMPRAR
        *
            interfaz.crear_marco(titulo_marco)
            interfaz.set_mensaje("SELECCIONE LA OPCIÓN QUE
CORRESPONDA A SU TARJETA BANCARIA")
            interfaz.informar(11,11)
            interfaz.set_mensaje("1.- TARJETA DE CRÉDITO")
            interfaz.informar(25,14)
            interfaz.set_mensaje("2.- TARJETA DE DÉBITO")
            interfaz.informar(25,16)
            interfaz.set_mensaje("3.- TARJETA DE PREPAGO")

```

```

interfaz.informar(25,18)
interfaz.set_mensaje("4.- SALIR")
interfaz.informar(25,20)
interfaz.set_mensaje("DIGITE LA OPCIÓN QUE ELIGIÓ:")
interfaz.informar(17,23)
interfaz.recibir(46,23)
v_opc <-- interfaz.get_dato

SEGÚN v_opc SEA
CASO 1:
    titulo_marco <-- "*"
TARJETA DE CRÉDITO                                     *
interfaz.crear_marco(titulo_marco)
interfaz.set_mensaje("INGRESE SU NÚMERO DE
TARJETA:")
interfaz.informar(9,17)
interfaz.recibir(39,17)
v_no_tarj <-- interfaz.get_dato
v_valid_tarj <--
obj_cred.set_no_tarjeta(v_no_tarj)

SI (v_valid_tarj=VERDADERO) HACER
    interfaz.crear_marco(titulo_marco)
    interfaz.set_mensaje("LIMITE DE CREDITO:")
    interfaz.informar(9,12)
    interfaz.set_mensaje("TAZA DE INTERÉS
MENSUAL:")
    interfaz.informar(9,15)
    interfaz.set_mensaje("SALDO ACTUAL:")
    interfaz.informar(9,18)
    interfaz.set_mensaje("MONTO DE LA
COMPRA:")
    interfaz.informar(9,21)
    interfaz.set_mensaje("MONTO MÁS INTERÉS:")
    interfaz.informar(44,21)
    interfaz.recibir(28,12)
    v_lim_cred <-- interfaz.get_dato
    interfaz.recibir(34,15)
    v_taza_int <-- interfaz.get_dato
    interfaz.recibir(23,18)
    v_saldo <-- interfaz.get_dato
    interfaz.recibir(29,21)
    v_monto_c <-- interfaz.get_dato
    obj_cred.set_lim_cred(v_lim_cred)
    obj_cred.set_taza_int(v_taza_int)
    obj_cred.set saldo_bancario(v_saldo)
    v_monto_tot <--
obj_cred.calc_monto_total(v_monto_c)
    v_comparacion <-
obj_cred.comparar saldo_y_credito(v_monto_tot)
    interfaz.set_mensaje(v_monto_tot)
    informar(63,21)
    SI (v_comparacion=VERDADERO) HACER

```

```

        obj_cred.actualizar_saldo(v_monto_tot)
                                v_saldo <--
obj_cred.get saldo_bancario
                                interfaz.set_mensaje("SU SALDO SE HA
ACTUALIZADO CORRECTAMENTE A $" + v_saldo + " mxn")
                                interfaz.informar(9,24)
                                interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
                                interfaz.informar(9,27)
                                interfaz.recibir(55,27)
SI NO
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("COMPRA
RECHAZADA POR REVASAR EL LÍMITE DE CRÉDITO")
        interfaz.informar(17,18)
        interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
        interfaz.informar(9,27)
        interfaz.recibir(55,27)
FIN(SI)
SI NO
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("EL NÚMERO DE TARJETA
PROPORCIONADO ES INCORRECTO")
        interfaz.informar(17,17)
        interfaz.set_mensaje("FAVOR DE INTENTAR
NUEVAMENTE")
        interfaz.informar(25,18)
        interfaz.set_mensaje("PRESIONA ENTER PARA
REGRESAR AL MENÚ PRINCIPAL")
        interfaz.informar(9,27)
        interfaz.recibir(55,27)
FIN(SI)
CASO 2:
        titulo_marco <-- "*"
TARJETA DE DÉBITO "
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("INGRESE SU NÚMERO DE
TARJETA:")
        interfaz.informar(9,17)
        interfaz.recibir(39,17)
        v_no_tarj <-- interfaz.get_dato
        v_valid_tarj <--
obj_deb.set_no_tarjeta(v_no_tarj)
SI(v_valid_tarj=VERDADERO)
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("PORCENTAJE DE
CAPACIDAD DE SOBREGIRO:")
        interfaz.informar(9,12)
        interfaz.set_mensaje("SALDO ACTUAL:")
        interfaz.informar(9,15)
        interfaz.set_mensaje("MÁXIMA CAPACIDAD DE
PAGO:")

```

```

                interfaz.informar(9,18)
                interfaz.set_mensaje("MONTO DE LA
COMPRA:")
                interfaz.informar(9,21)
                interfaz.recibir(47,12)
                v_sobregiro <-- interfaz.get_dato
                interfaz.recibir(23,15)
                v_saldo <-- interfaz.get_dato
                obj_deb.set_p_sobregiro(v_sobregiro)
                obj_deb.set saldo_bancario(v_saldo)
                obj_deb.calc_capacidad_pago
                v_cap_pago <-- obj_deb.get_cap_pago
                interfaz.set_mensaje("$"+v_cap_pago)
                interfaz.informar(35,18)
                interfaz.recibir(29,21)
                v_monto_c <-- interfaz.get_dato
                v_comparacion <--
obj_deb.comparar_monto_y_cap_pago(v_monto_c)
SI (v_comparacion=VERDADERO)
    - obj_deb.actualizar_saldo(v_monto_c)
    v_saldo <--
obj_deb.get saldo_bancario
    interfaz.set_mensaje("SU SALDO SE HA
ACTUALIZADO CORRECTAMENTE A $" + v_saldo + " mxn")
    interfaz.informar(9,24)
    interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
    interfaz.informar(9,27)
    interfaz.recibir(55,27)
SI NO
    interfaz.crear_marco(titulo_marco)
    interfaz.set_mensaje("COMPRA
RECHAZADA POR REVASAR LA CAPACIDAD DE PAGO")
    interfaz.informar(17,18)
    interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
    interfaz.informar(9,27)
    interfaz.recibir(55,27)
FIN(SI)
SI NO
    interfaz.crear_marco(titulo_marco)
    interfaz.set_mensaje("EL NÚMERO DE TARJETA
PROPORCIONADO ES INCORRECTO")
    interfaz.informar(17,17)
    interfaz.set_mensaje("FAVOR DE INTENTAR
NUEVAMENTE")
    interfaz.informar(25,18)
    interfaz.set_mensaje("PRESIONA ENTER PARA
REGRESAR AL MENÚ PRINCIPAL")
    interfaz.informar(9,27)
    interfaz.recibir(55,27)
FIN(SI)
CASO 3:

```

```

        titulo_marco <-- "*"
    TARJETA DE PREPAGO                                *"
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("INGRESE SU NÚMERO DE
TARJETA:")
        interfaz.informar(9,17)
        interfaz.recibir(39,17)
        v_no_tarj <-- interfaz.get_dato
        v_valid_tarj <--
obj_prep.set_no_tarjeta(v_no_tarj)
        SI(v_valid_tarj=VERDADERO)
            interfaz.crear_marco(titulo_marco)
            interfaz.set_mensaje("SALDO ACTUAL:")
            interfaz.informar(9,14)
            interfaz.set_mensaje("MONTO DE COMPRA:")
            interfaz.informar(9,18)
            interfaz.recibir(23,14)
            v_saldo <-- interfaz.get_dato
            interfaz.recibir(26,18)
            v_monto_c <-- interfaz.get_dato
            obj_prep.set_saldo_bancario(v_saldo)
            v_comparacion <-
obj_prep.comparar_monto_y_saldo(v_monto_c)
        SI(v_comparacion=VERDADERO)
            obj_prep.actualizar_saldo(v_monto_c)
            v_saldo <-
obj_prep.get_saldo_bancario
            interfaz.set_mensaje("SU SALDO SE HA
ACTUALIZADO CORRECTAMENTE A $" +v_saldo+" mxn")
            interfaz.informar(9,24)
            interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
            interfaz.informar(9,27)
            interfaz.recibir(55,27)
        SI NO
            interfaz.crear_marco(titulo_marco)
            interfaz.set_mensaje("COMPRA
RECHAZADA POR SALDO INSUFICIENTE")
            interfaz.informar(17,18)
            interfaz.set_mensaje("PRESIONA ENTER
PARA REGRESAR AL MENÚ PRINCIPAL")
            interfaz.informar(9,27)
            interfaz.recibir(55,27)
        FIN(SI)
        SI NO
            interfaz.crear_marco(titulo_marco)
            interfaz.set_mensaje("EL NÚMERO DE TARJETA
PROPORCIONADO ES INCORRECTO")
            interfaz.informar(17,17)
            interfaz.set_mensaje("FAVOR DE INTENTAR
NUEVAMENTE")
            interfaz.informar(25,18)
            interfaz.set_mensaje("PRESIONA ENTER PARA
REGRESAR AL MENÚ PRINCIPAL")

```

```
        interfaz.informar(9,27)
        interfaz.recibir(55,27)
    FIN(SI)
CASO 4:

        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("EJECUCIÓN DEL PROGRAMA
FINALIZADA")
        interfaz.informar(23,19)
OTRO:
        interfaz.crear_marco(titulo_marco)
        interfaz.set_mensaje("LA OPCION SELECCIONADA NO
EXISTE, PRESIONA ENTER PARA CONTINUAR")
        interfaz.informar(5,17)
        interfaz.recibir(68,17)
    FIN(SEGÚN)
HASTA(v_opc=4)

FIN MÉTODO PRINCIPAL
FIN PROGRAMA ACTUALIZAR_SALDO
```

- Codificación en PASCAL

- U\_CL\_TARJ\_BANCARIA

```
unit U_CL_TARJ_BANCARIA;

{$mode objfpc}{$H+}

{
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
}

interface

type
  CL_TARJ_BANCARIA = class
protected
  saldo_bancario: Real;
  no_tarjeta: String;
public
  procedure set_saldo_bancario(x: Real);
  function set_no_tarjeta(x: String): Boolean;
  function get_saldo_bancario: Real;
  function get_no_tarjeta: String;
end;

implementation

procedure CL_TARJ_BANCARIA.set_saldo_bancario(x: Real);
begin
  saldo_bancario := x;
end;

function CL_TARJ_BANCARIA.set_no_tarjeta(x: String): Boolean;
var
  long_tarj: Integer;
begin
  long_tarj := Length(x);
  if (long_tarj = 16) then
  begin
    no_tarjeta := x;
    Result := True;
  end
  else
    Result := False;
end;

function CL_TARJ_BANCARIA.get_saldo_bancario: Real;
begin
  Result := saldo_bancario;
end;
```

```

function CL_TARJ_BANCARIA.get_no_tarjeta: String;
begin
  Result := no_tarjeta;
end;

end.

```

### o U\_CL\_TARJ\_CRED

```

unit U_CL_TARJ_CRED;

{$mode objfpc}{$H+}

{
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
}

interface

uses
  U_CL_TARJ_BANCARIA;

type
  CL_TARJ_CRED = class(CL_TARJ_BANCARIA)
  private
    lim_cred: Real;
    taza_int: Real;
  public
    procedure set_lim_cred(x: Real);
    procedure set_taza_int(x: Real);
    function get_lim_cred: Real;
    function get_taza_int: Real;
    function calc_monto_total(monto_compra: Real): Real;
    function comparar saldo_y_credito(monto_total: Real): Boolean;
    procedure actualizar_saldo(monto_total: Real);
  end;

implementation

procedure CL_TARJ_CRED.set_lim_cred(x: Real);
begin
  lim_cred := x;
end;

procedure CL_TARJ_CRED.set_taza_int(x: Real);
begin
  taza_int := x;
end;

function CL_TARJ_CRED.get_lim_cred: Real;

```

```

begin
  Result := lim_cred;
end;

function CL_TARJ_CRED.get_taza_int: Real;
begin
  Result := taza_int;
end;

function CL_TARJ_CRED.calc_monto_total(monto_compra: Real): Real;
var
  monto_total: Real;
begin
  monto_total := monto_compra + (monto_compra * taza_int / 100);
  Result := monto_total;
end;

function CL_TARJ_CRED.comparar_saldo_y_credito(monto_total: Real): Boolean;
var
  v_comparacion: Boolean;
  v_nuevo_saldo: Real;
begin
  v_nuevo_saldo := saldo_bancario + monto_total;
  if (v_nuevo_saldo <= lim_cred) then
    v_comparacion := True
  else
    v_comparacion := False;
  Result := v_comparacion;
end;

procedure CL_TARJ_CRED.actualizar_saldo(monto_total: Real);
begin
  saldo_bancario := saldo_bancario + monto_total;
end;

end.

```

### o U\_CL\_TARJ\_DEB

```

unit U_CL_TARJ_DEB;

{$mode objfpc}{$H+}

{
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
}

interface

```

```

uses
  U_CL_TARJ_BANCARIA;

type
  CL_TARJ_DEB = class(CL_TARJ_BANCARIA)
  private
    p_sobregiro: Real;
    capacidad_pago: Real;
  public
    procedure set_p_sobregiro(x: Real);
    procedure set_capacidad_pago(x: Real);
    function get_p_sobregiro: Real;
    function get_capacidad_pago: Real;
    procedure calc_capacidad_pago;
    function comparar_monto_y_cap_pago(monto_compra: Real): Boolean;
    procedure actualizar_saldo(monto_compra: Real);
  end;

implementation

procedure CL_TARJ_DEB.set_p_sobregiro(x: Real);
begin
  p_sobregiro := x;
end;

procedure CL_TARJ_DEB.set_capacidad_pago(x: Real);
begin
  capacidad_pago := x;
end;

function CL_TARJ_DEB.get_p_sobregiro: Real;
begin
  Result := p_sobregiro;
end;

function CL_TARJ_DEB.get_capacidad_pago: Real;
begin
  Result := capacidad_pago;
end;

procedure CL_TARJ_DEB.calc_capacidad_pago;
begin
  capacidad_pago := saldo_bancario + (saldo_bancario * p_sobregiro / 100);
end;

function CL_TARJ_DEB.comparar_monto_y_cap_pago(monto_compra: Real): Boolean;
var
  v_comparacion: Boolean;
begin
  if (monto_compra <= capacidad_pago) then
    v_comparacion := True
  else

```

```

    v_comparacion := False;
    Result := v_comparacion;
end;

procedure CL_TARJ_DEB.actualizar_saldo(monto_compra: Real);
begin
    saldo_bancario := saldo_bancario - monto_compra;
end;

end.

```

### o U\_CL\_TARJ\_PREP

```

unit U_CL_TARJ_PREP;

{$mode objfpc}{$H+}

{
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
}

interface

uses
  U_CL_TARJ_BANCARIA;

type
  CL_TARJ_PREP = class(CL_TARJ_BANCARIA)
  public
    function comparar_monto_y_saldo(monto_compra: Real): Boolean;
    procedure actualizar_saldo(monto_compra: Real);
  end;

implementation

function CL_TARJ_PREP.comparar_monto_y_saldo(monto_compra: Real): Boolean;
var
  v_comparacion: Boolean;
begin
  if (monto_compra <= saldo_bancario) then
    v_comparacion := True
  else
    v_comparacion := False;
  Result := v_comparacion;
end;

procedure CL_TARJ_PREP.actualizar_saldo(monto_compra: Real);
begin
  saldo_bancario := saldo_bancario - monto_compra;
end;

```

## o U\_CL\_VISTA

```
unit U_CL_VISTA;

{$mode objfpc}{$H+}

{
  Autor: Rodrigo Díaz Salguero
  Fecha: 16/10/2024
  Actualización: ---
}

interface

uses
  Crt;

type
  CL_VISTA = class
  private
    mensaje: String;
    dato: String;
  public
    procedure set_mensaje(x: String);
    procedure set_dato(x: String);
    function get_mensaje: String;
    function get_dato: String;
    procedure informar(x, y: Integer);
    procedure recibir(x, y: Integer);
    procedure crear_marco(titulo_marco: String);
  end;

implementation

procedure CL_VISTA.set_mensaje(x: String);
begin
  mensaje := x;
end;

procedure CL_VISTA.set_dato(x: String);
begin
  dato := x;
end;

function CL_VISTA.get_mensaje: String;
begin
  Result := mensaje;
end;

function CL_VISTA.get_dato: String;
begin
  Result := dato;
```

```
end;

procedure CL_VISTA.informar(x, y: Integer);
begin
  GotoXY(x, y);
  WriteLn(mensaje);
end;

procedure CL_VISTA.recibir(x, y: Integer);
begin
  GotoXY(x, y);
  ReadLn(dato);
end;

procedure CL_VISTA.crear_marco(titulo_marco: String);
var
  v_asteriscos_80: String;
  v_aste_esp_aste: String;
begin
  // BORRAR LA PANTALLA ANTES DE CREAR UN MARCO NUEVO
  ClrScr; //BORRAR

  v_asteriscos_80 := ****
  *****';
  v_aste_esp_aste := '*';

  GotoXY(1, 5); WriteLn(v_asteriscos_80);
  GotoXY(1, 6); WriteLn(v_aste_esp_aste);
  GotoXY(1, 7); WriteLn(titulo_marco);
  GotoXY(1, 8); WriteLn(v_aste_esp_aste);
  GotoXY(1, 9); WriteLn(v_asteriscos_80);

  GotoXY(1, 10); WriteLn(v_aste_esp_aste);
  GotoXY(1, 11); WriteLn(v_aste_esp_aste);
  GotoXY(1, 12); WriteLn(v_aste_esp_aste);
  GotoXY(1, 13); WriteLn(v_aste_esp_aste);
  GotoXY(1, 14); WriteLn(v_aste_esp_aste);
  GotoXY(1, 15); WriteLn(v_aste_esp_aste);
  GotoXY(1, 16); WriteLn(v_aste_esp_aste);
  GotoXY(1, 17); WriteLn(v_aste_esp_aste);
  GotoXY(1, 18); WriteLn(v_aste_esp_aste);
  GotoXY(1, 19); WriteLn(v_aste_esp_aste);
  GotoXY(1, 20); WriteLn(v_aste_esp_aste);
  GotoXY(1, 21); WriteLn(v_aste_esp_aste);
  GotoXY(1, 22); WriteLn(v_aste_esp_aste);
  GotoXY(1, 23); WriteLn(v_aste_esp_aste);
  GotoXY(1, 24); WriteLn(v_aste_esp_aste);
  GotoXY(1, 25); WriteLn(v_aste_esp_aste);
  GotoXY(1, 26); WriteLn(v_aste_esp_aste);
  GotoXY(1, 27); WriteLn(v_aste_esp_aste);
  GotoXY(1, 28); WriteLn(v_aste_esp_aste);
  GotoXY(1, 29); WriteLn(v_asteriscos_80);
```

```
end;
end.
```

## o P\_ACTUALIZAR\_SALDO

```
program P_ACTUALIZAR_SALDO;
{$mode objfpc}{$H+}

{
NOMBRE: RODRIGO DÍAZ SALGUERO
FECHA: 02/12/2024
ACTUALIZACIÓN: ---
}

uses
  crt, U_CL_VISTA, U_CL_TARJ_CRED, U_CL_TARJ_DEB, U_CL_TARJ_PREP,
SysUtils;

var
  titulo_marco: string;
  v_opc: integer;
  v_no_tarj: string;
  v_valid_tarj: boolean;
  v_lim_cred, v_taza_int, v_saldo, v_monto_c, v_monto_tot,
  v_sobregiro, v_cap_pago: real;
  v_comparacion: boolean;

  interfaz: U_CL_VISTA;
  obj_cred: U_CL_TARJ_CRED;
  obj_deb: U_CL_TARJ_DEB;
  obj_prep: U_CL_TARJ_PREP;

begin
  interfaz := U_CL_VISTA.Create;
  obj_cred := U_CL_TARJ_CRED.Create;
  obj_deb := U_CL_TARJ_DEB.Create;
  obj_prep := U_CL_TARJ_PREP.Create;

repeat
  titulo_marco := '*'                                     COMPRAR
*';
  interfaz.crear_marco(titulo_marco);
  interfaz.set_mensaje('SELECCIONE LA OPCIÓN QUE CORRESPONDA A SU
TARJETA BANCARIA');
  interfaz.informar(11, 11);
  interfaz.set_mensaje('1.- TARJETA DE CRÉDITO');
  interfaz.informar(25, 14);
  interfaz.set_mensaje('2.- TARJETA DE DÉBITO');
  interfaz.informar(25, 16);
  interfaz.set_mensaje('3.- TARJETA DE PREPAGO');
  interfaz.informar(25, 18);
  interfaz.set_mensaje('4.- SALIR');
  interfaz.informar(25, 20);
```

```

interfaz.set_mensaje('DIGITE LA OPCIÓN QUE ELIGIÓ:');
interfaz.informar(17, 23);
interfaz.recibir(46, 23);
v_opc := StrToInt(interfaz.get_dato);

case v_opc of
  1: begin
    titulo_marco := '*' *';
CRÉDITO                                     TARJETA DE
                                              '*';
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('INGRESE SU NÚMERO DE TARJETA:');
    interfaz.informar(9, 17);
    interfaz.recibir(39, 17);
    v_no_tarj := interfaz.get_dato;
    v_valid_tarj := obj_cred.set_no_tarjeta(v_no_tarj);

    if v_valid_tarj then begin
      interfaz.crear_marco(titulo_marco);
      interfaz.set_mensaje('LIMITE DE CREDITO:');
      interfaz.informar(9, 12);
      interfaz.set_mensaje('TAZA DE INTERÉS MENSUAL:');
      interfaz.informar(9, 15);
      interfaz.set_mensaje('SALDO ACTUAL:');
      interfaz.informar(9, 18);
      interfaz.set_mensaje('MONTO DE LA COMPRA:');
      interfaz.informar(9, 21);
      interfaz.set_mensaje('MONTO MÁS INTERÉS:');
      interfaz.informar(44, 21);
      interfaz.recibir(28, 12);
      v_lim_cred := StrToFloat(interfaz.get_dato);
      interfaz.recibir(34, 15);
      v_taza_int := StrToFloat(interfaz.get_dato);
      interfaz.recibir(23, 18);
      v_saldo := StrToFloat(interfaz.get_dato);
      interfaz.recibir(29, 21);
      v_monto_c := StrToFloat(interfaz.get_dato);

      obj_cred.set_lim_cred(v_lim_cred);
      obj_cred.set_taza_int(v_taza_int);
      obj_cred.set_saldo_bancario(v_saldo);
      v_monto_tot := obj_cred.calc_monto_total(v_monto_c);
      v_comparacion :=

obj_cred.comparar_saldo_y_credito(v_monto_tot);
      interfaz.set_mensaje(FloatToStr(v_monto_tot));
      interfaz.informar(63, 21);

      if v_comparacion then begin
        obj_cred.actualizar_saldo(v_monto_tot);
        v_saldo := obj_cred.get_saldo_bancario;
        interfaz.set_mensaje('SU SALDO SE HA ACTUALIZADO
CORRECTAMENTE A $' + FloatToStr(v_saldo) + ' mxn');
        interfaz.informar(9, 24);
        interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');

```

```

        interfaz.informar(9, 27);
        interfaz.recibir(55, 27);
    end else begin
        interfaz.crear_marco(titulo_marco);
        interfaz.set_mensaje('COMPRA RECHAZADA POR REVASAR EL
LÍMITE DE CRÉDITO');
        interfaz.informar(17, 18);
        interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
        interfaz.informar(9, 27);
        interfaz.recibir(55, 27);
    end;
    end else begin
        interfaz.crear_marco(titulo_marco);
        interfaz.set_mensaje('EL NÚMERO DE TARJETA PROPORCIONADO ES
INCORRECTO');
        interfaz.informar(17, 17);
        interfaz.set_mensaje('FAVOR DE INTENTAR NUEVAMENTE');
        interfaz.informar(25, 18);
        interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
        interfaz.informar(9, 27);
        interfaz.recibir(55, 27);
    end;
end;
2: begin
    titulo_marco := '*'                                     TARJETA DE
DÉBITO                                         '*';
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('INGRESE SU NÚMERO DE TARJETA:');
    interfaz.informar(9, 17);
    interfaz.recibir(39, 17);
    v_no_tarj := interfaz.get_dato;
    v_valid_tarj := obj_deb.set_no_tarjeta(v_no_tarj);

    if v_valid_tarj = True then
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('PORCENTAJE DE CAPACIDAD DE
SOBREGIRO:');
    interfaz.informar(9, 12);
    interfaz.set_mensaje('SALDO ACTUAL:');
    interfaz.informar(9, 15);
    interfaz.set_mensaje('MÁXIMA CAPACIDAD DE PAGO:');
    interfaz.informar(9, 18);
    interfaz.set_mensaje('MONTO DE LA COMPRA:');
    interfaz.informar(9, 21);
    interfaz.recibir(47, 12);
    v_sobregiro := interfaz.get_dato;
    interfaz.recibir(23, 15);
    v_saldo := interfaz.get_dato;
    obj_deb.set_p_sobregiro(v_sobregiro);
    obj_deb.set saldo_bancario(v_saldo);
    obj_deb.calc_capacidad_pago;

```

```

v_cap_pago := obj_deb.get_cap_pago;
interfaz.set_mensaje('$' + FloatToStr(v_cap_pago));
interfaz.informar(35, 18);
interfaz.recibir(29, 21);
v_monto_c := interfaz.get_dato;
v_comparacion :=
obj_deb.comparar_monto_y_cap_pago(v_monto_c);

if v_comparacion = True then
begin
    obj_deb.actualizar_saldo(v_monto_c);
    v_saldo := obj_deb.get_saldo_bancario;
    interfaz.set_mensaje('SU SALDO SE HA ACTUALIZADO
CORRECTAMENTE A $' + FloatToStr(v_saldo) + ' mxn');
    interfaz.informar(9, 24);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end
else
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('COMPRA RECHAZADA POR REBASAR LA
CAPACIDAD DE PAGO');
    interfaz.informar(17, 18);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end;
end
else
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('EL NÚMERO DE TARJETA PROPORCIONADO ES
INCORRECTO');
    interfaz.informar(17, 17);
    interfaz.set_mensaje('FAVOR DE INTENTAR NUEVAMENTE');
    interfaz.informar(25, 18);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end;
end;

3: begin
    titulo_marco := '*'                                     TARJETA DE
PREPAGO                                         '*';
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('INGRESE SU NÚMERO DE TARJETA:');
    interfaz.informar(9, 17);
    interfaz.recibir(39, 17);

```

```

v_no_tarj := interfaz.get_dato;
v_valid_tarj := obj_prep.set_no_tarjeta(v_no_tarj);

if v_valid_tarj = True then
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('SALDO ACTUAL:');
    interfaz.informar(9, 14);
    interfaz.set_mensaje('MONTO DE COMPRA:');
    interfaz.informar(9, 18);
    interfaz.recibir(23, 14);
    v_saldo := interfaz.get_dato;
    interfaz.recibir(26, 18);
    v_monto_c := interfaz.get_dato;
    obj_prep.set saldo_bancario(v_saldo);
    v_comparacion := obj_prep.comparar_monto_y_saldo(v_monto_c);

    if v_comparacion = True then
begin
    obj_prep.actualizar_saldo(v_monto_c);
    v_saldo := obj_prep.get saldo_bancario;
    interfaz.set_mensaje('SU SALDO SE HA ACTUALIZADO
CORRECTAMENTE A $' + FloatToStr(v_saldo) + ' mxn');
    interfaz.informar(9, 24);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end
else
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('COMPRA RECHAZADA POR SALDO
INSUFICIENTE');
    interfaz.informar(17, 18);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end;
end
else
begin
    interfaz.crear_marco(titulo_marco);
    interfaz.set_mensaje('EL NÚMERO DE TARJETA PROPORCIONADO ES
INCORRECTO');
    interfaz.informar(17, 17);
    interfaz.set_mensaje('FAVOR DE INTENTAR NUEVAMENTE');
    interfaz.informar(25, 18);
    interfaz.set_mensaje('PRESIONA ENTER PARA REGRESAR AL MENÚ
PRINCIPAL');
    interfaz.informar(9, 27);
    interfaz.recibir(55, 27);
end;

```

```
    end;
  end;
until v_opc = 4;
end.
```