

**Instituto Politécnico Nacional**

**Escuela Superior de Computo**

**Práctica 3**

**Alumno:** Juárez Bouchan José Leonardo

**Grupo:** 2CV2

Introducción.

El uso apropiado de los tipos de datos dispuestos por el lenguaje de programación Java, le da una potencia enorme, esto a pesar de que Java es un lenguaje orientado a objetos. Un valor constante en Java se crea utilizando una representación literal de él. Java utiliza cinco tipos de elementos: enteros, reales en coma flotante, booleanos, caracteres y cadenas, que se pueden poner en cualquier lugar del código fuente de Java. Cada uno de estos literales tiene un tipo correspondiente asociado con él.

Enteros:

byte 8 bits complemento a dos

short 16 bits complemento a dos

int 32 bits complemento a dos

long 64 bits complemento a dos

Por ejemplo: 21 077 0xDC00

Reales en coma flotante:

float 32 bits IEEE 754

double 64 bits IEEE 754

Por ejemplo: 3.14 2e12 3.1E12

Booleanos:

true

false

Caracteres:

Por ejemplo: a \t \u???? [????] es un número unicode

Cadenas:

Por ejemplo: "Esto es una cadena literal"

Variables

Una variable es la unidad básica de almacenamiento de un programa en Lava. Una variable se define mediante la combinación de un identificador, un tipo y un inicializador opcional. Además de que todas las variables tienen un ámbito que define su alcance (scope) y su tiempo de vida. La declaración de una variable se realiza de la siguiente forma:

Tipo identificador = valor, didentificador = valor.......;

Ejemplos de lo anterior se pueden ver a continuación:

int a, b, c; // declara tres enteros tres enteros a, b, y c.

int d = 3, e, f = 5; // declara tres o más enteros, inicializando d y f.

byte z = 22; // inicializa z.

double pi = 3.14159; // declara una aproximación de pi.

char x = 'x'; // La variable x tiene el valor de 'x'.

Matrices

Se pueden declarar en Java matrices de cualquier tipo:

char s[];

int iArray[];

Incluso se pueden construir arreglos de matrices:

int tabla[][] = new int[4][5];

Los límites de los arrays se comprueban en tiempo de ejecución para evitar desbordamientos y la corrupción de memoria.

En Java una matríz es realmente un objeto, porque tiene redefinido el operador []. Tiene una función miembro: length. Se puede utilizar este método para conocer la longitud de cualquier matríz.

int a[][] = new int[10][3];

a.length; /\* 10 \*/

a[0].length; /\* 3 \*/

Para crear un array en Java hay dos métodos básicos. Crear un array vacío:

int lista[] = new int[50];

o se puede crear ya el array con sus valores iniciales:

String nombres[] = {

"Juan","Pepe","Pedro","Maria"

};

Esto que es equivalente a:

String nombres[];

nombres = new String[4];

nombres[0] = new String( "Juan" );

nombres[1] = new String( "Pepe" );

nombres[2] = new String( "Pedro" );

nombres[3] = new String( "Maria" );

No se pueden crear matrices estáticos en tiempo de compilación:

int lista[50]; // generará un error en tiempo de compilación

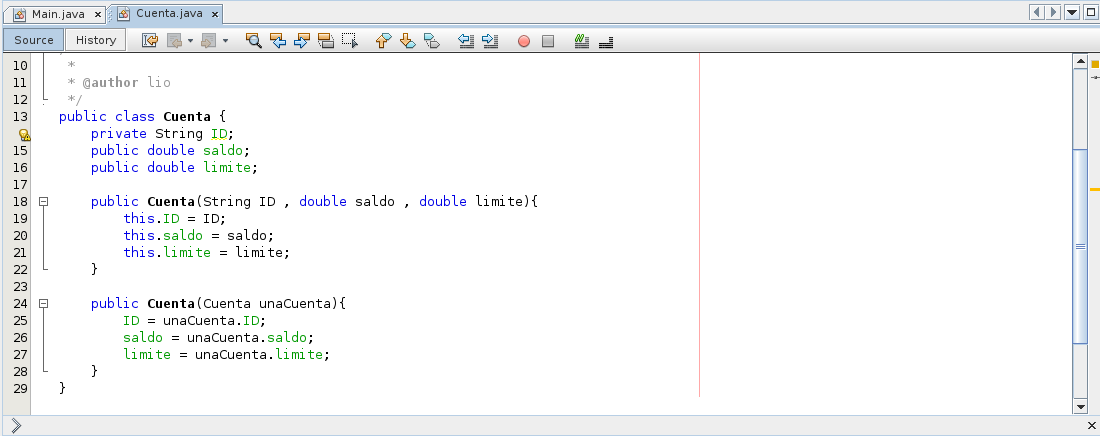
Tampoco se puede rellenar una matriz sin declarar el tamaño con el operador new:

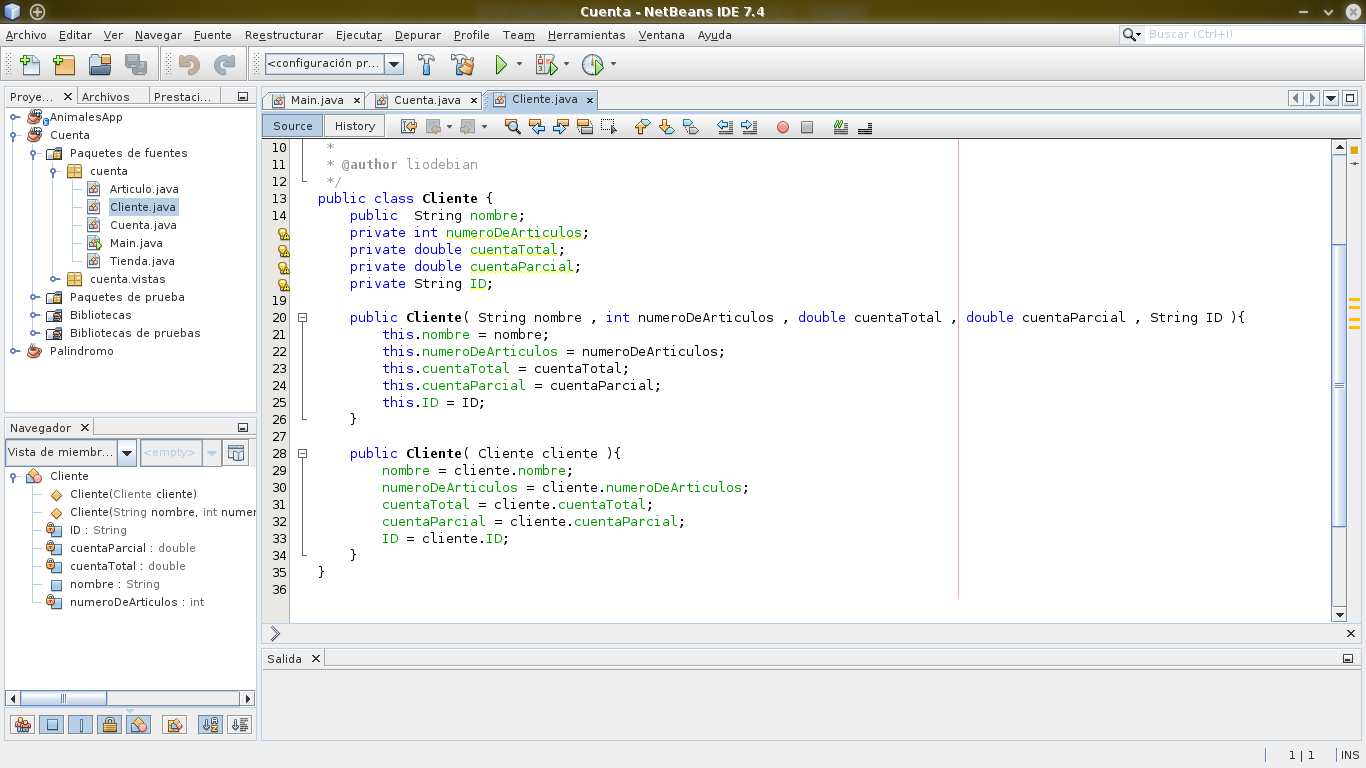
int lista[];

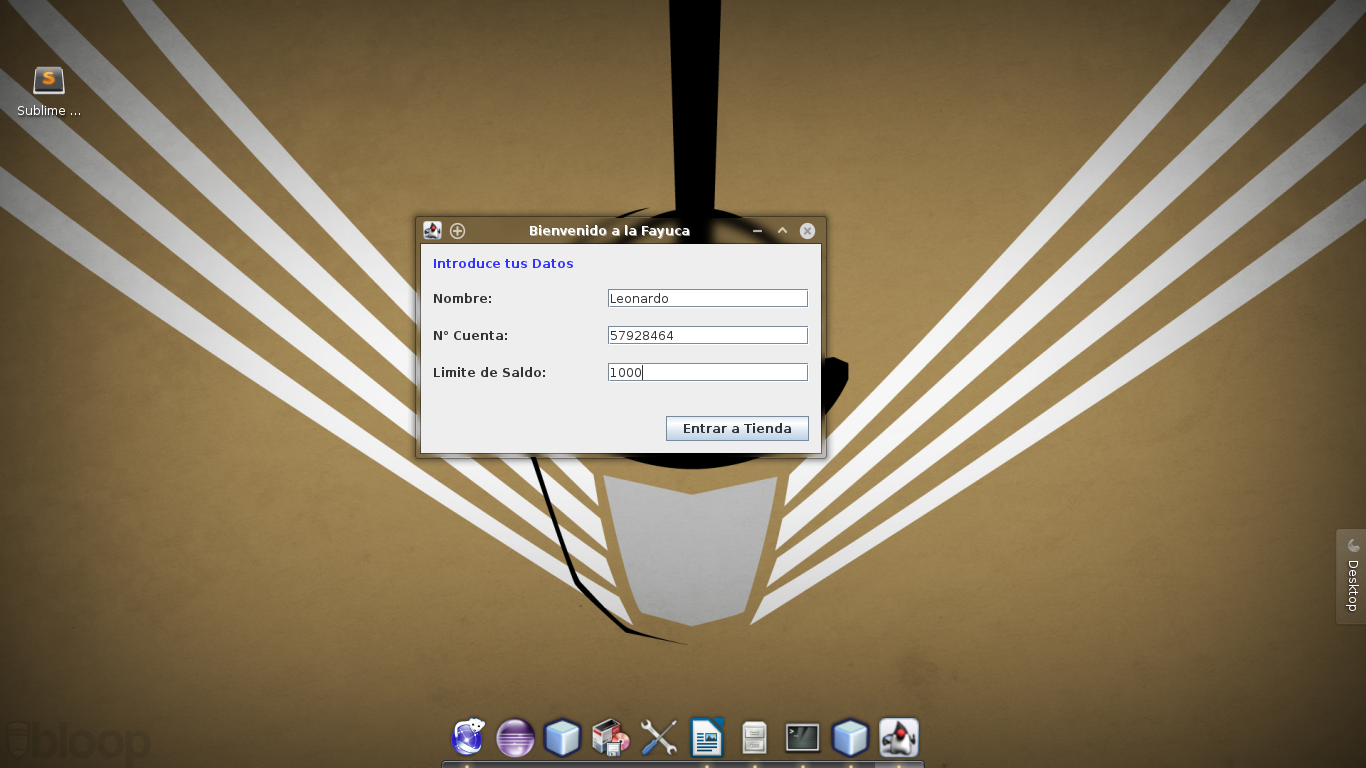
for( int i=0; i < 9; i++ )

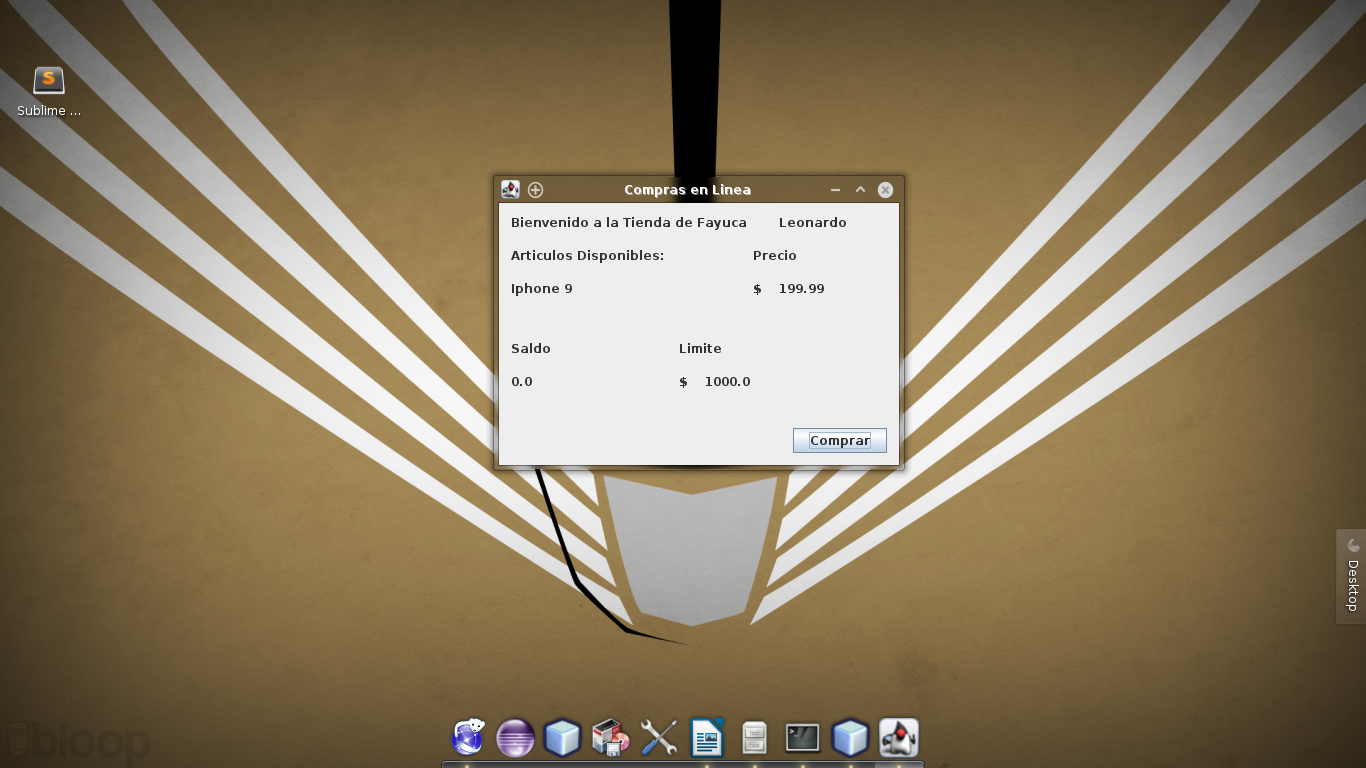
lista[i] = i;

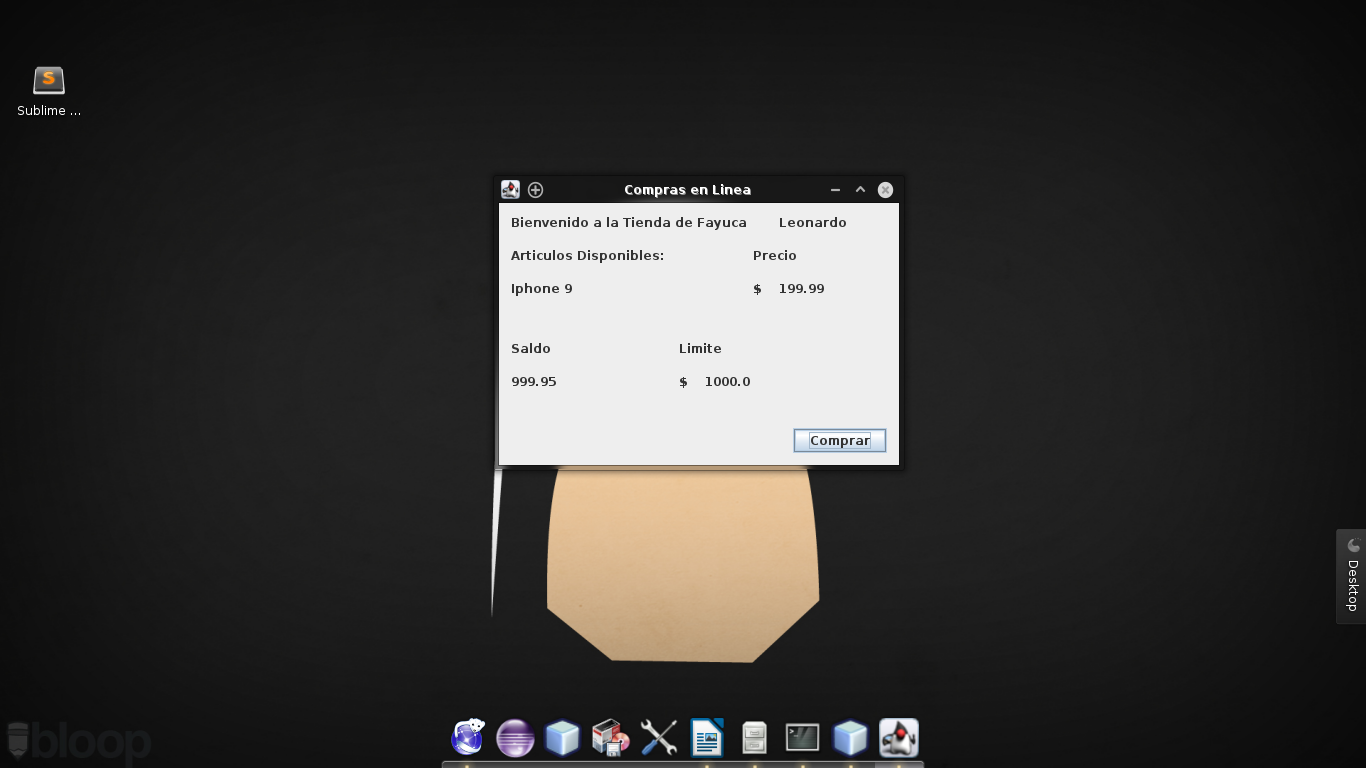
Desarrollo:

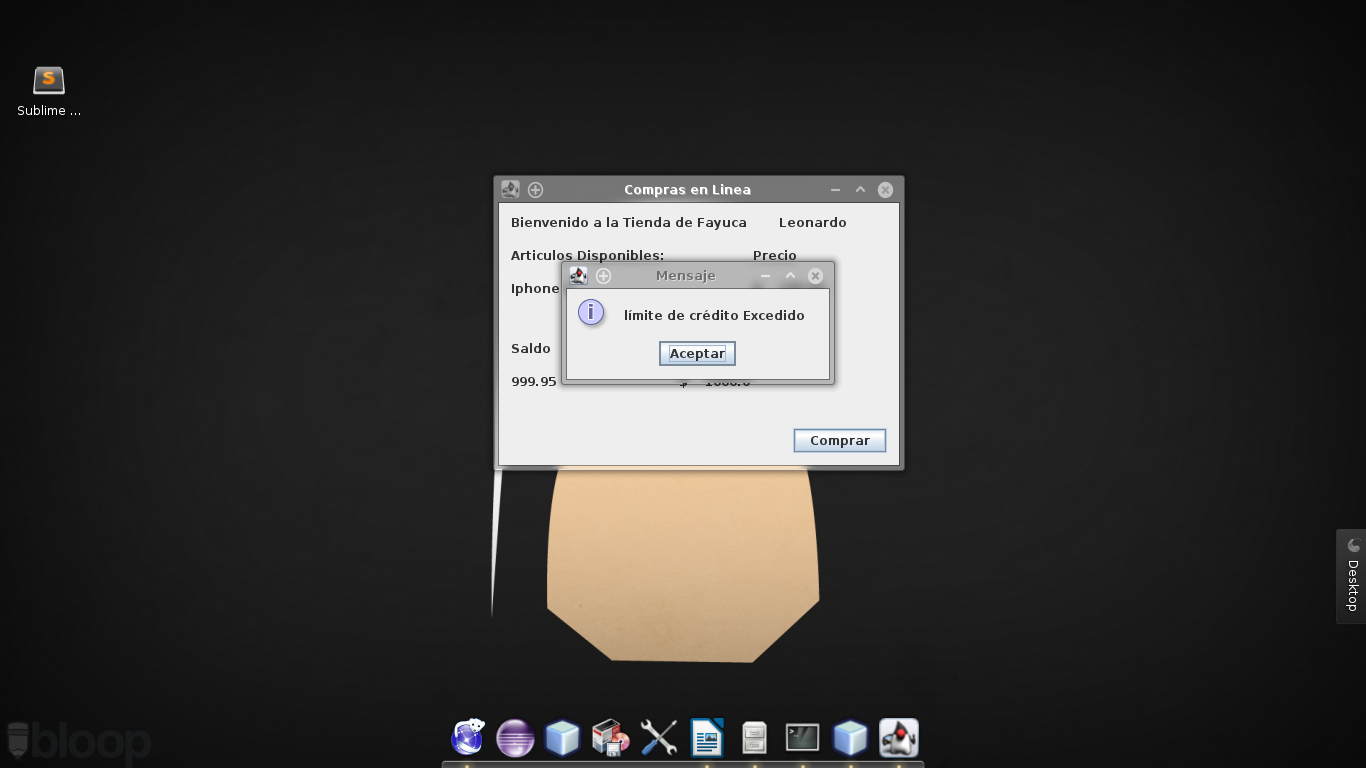


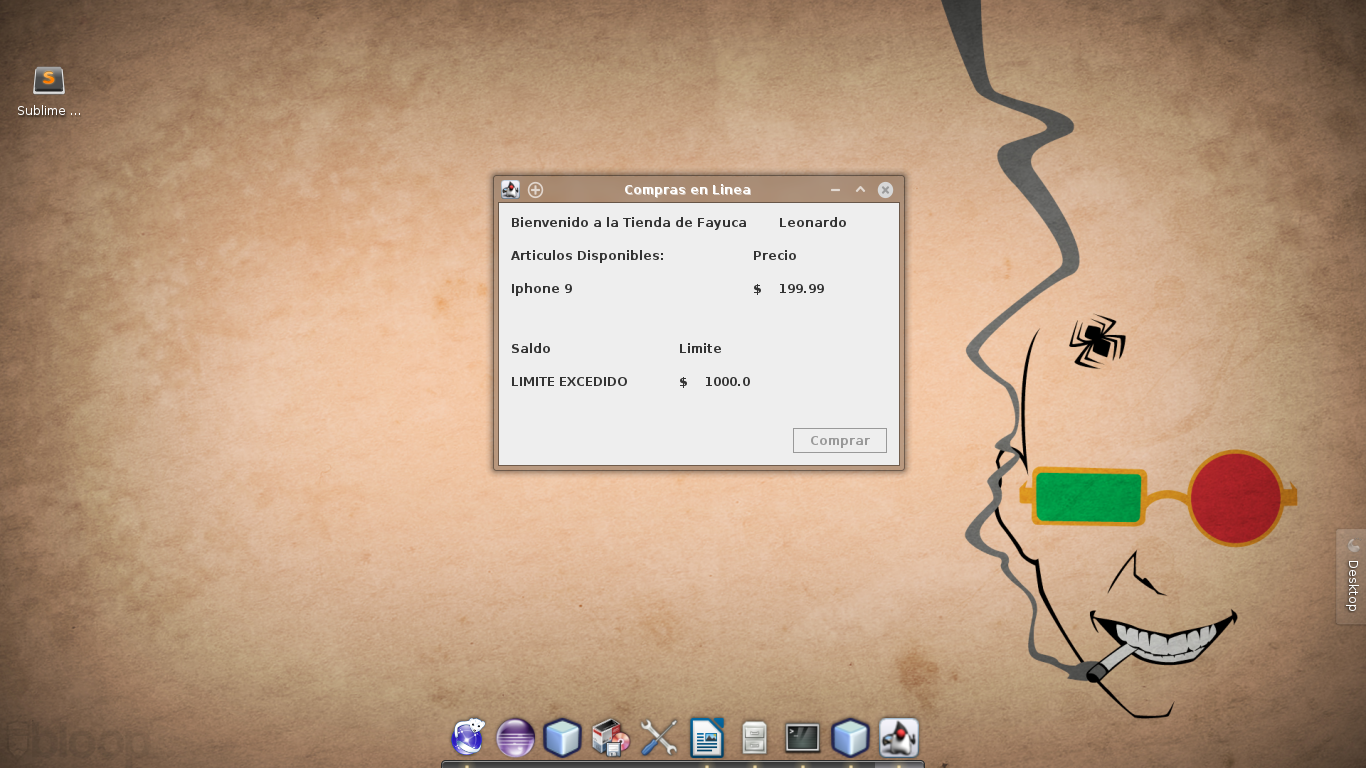












Creé una aplicación en java con Java swing, me aventuré a hacerla así ya que se me hizo buena idea hacer algo más vistoso me guso mucho por que aprendí mucho sobre las propiedades de los Frames, de los Labels, y de los botones, cuando el saldo pasa del limite de crédito muestra una alerta al usuario que dice que su limite ha sido excedido el usuario da aceptar y se desabilita el botón para que ya no pueda comprar más.