

Universidad ORT

Ingeniería en Sistemas

Programación I
Obligatorio 2 - Grupo N1A



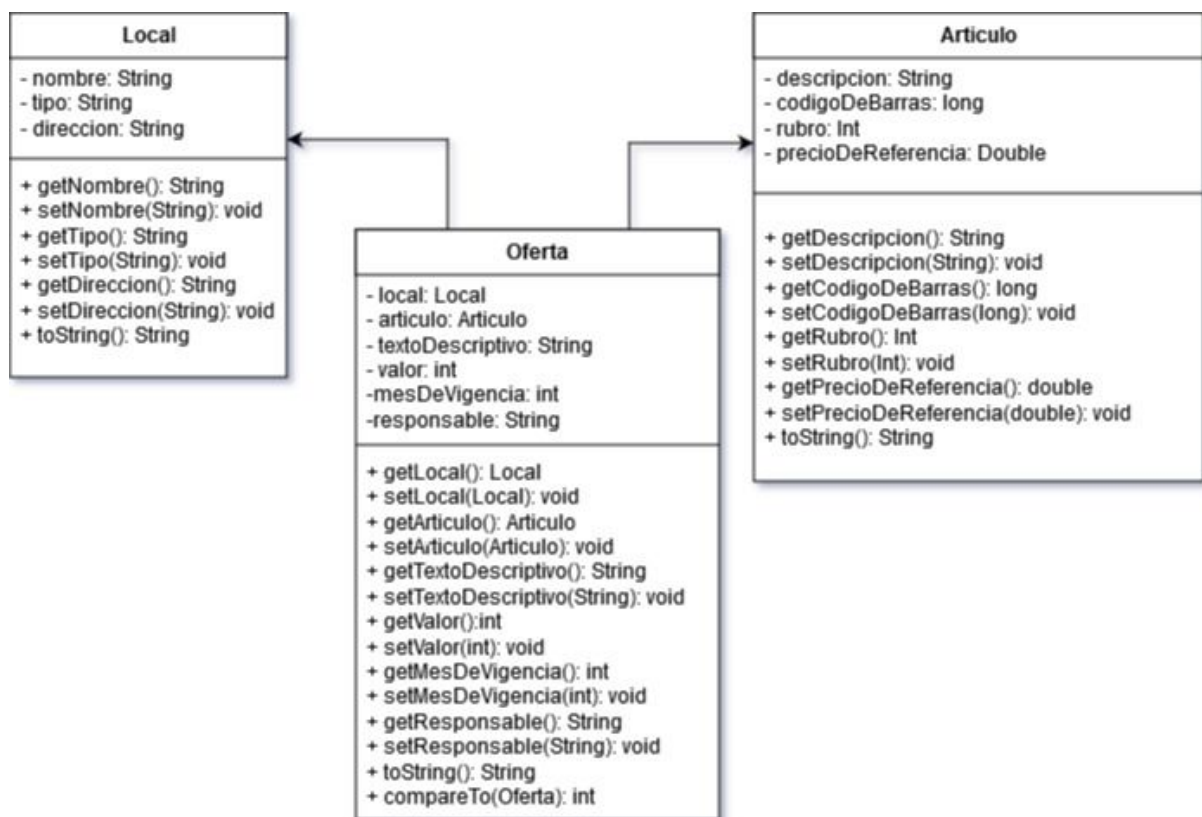
Rodrigo André Santomauro Lema
(199089)

Juliode 2017

Indice

1- Diagrama UML	3
2- Datos de prueba.....	4
3- Listado impreso de clases	7

Diagrama de clases (UML)



Datos de Prueba

(Previamente se ingresa un local y un artículo válido)

Al ingresar el nombre de local:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa un nombre válido de menú (texto): Devoto	Se crea el local correctamente	OK
Se ingresa un nombre repetido de local (texto): Devoto	Mensaje "El nombre Devoto ya existe, ingrese uno nuevo" y pida un nuevo nombre	OK
Se ingresa un nombre válido distinto (texto): Disco	Se crea el local correctamente	OK

Al ingresar el tipo de local:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa una opción válida de menú (número del 1 al 5): 2	Se crea la opción correctamente.	OK
Se ingresa una opción inválida de menú (número entero menor que 1 o mayor que 5): -1	Mensaje "Opción no es correcta. Reintente" y pida un nueva opción.	OK
Se ingresa una opción inválida de menú (letra): a	Mensaje "Opción no es correcta. Reintente" y pida un nueva opción.	Se cae el programa.

Al ingresar el código de barras de un artículo:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa un código de barras valido (número): 7575327	Se crea el la opción correcta	OK
Se ingresa un codigo repetido (número): 7575327	Mensaje "El código de barras 7575327 ya existe, ingrese uno nuevo"	OK

Al ingresar el rubro de un artículo:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa una opción válida de artículo (número del 1 al 8): 7	Se crea el la opción correcta	OK
Se ingresa una opción inválida de menú (número entero menor que 1 o mayor que 8): -1	Mensaje "Opción no es correcta. Reintente"	OK
Se ingresa una opción inválida de menú (letra): a	Mensaje "Opción no es correcta. Reintente"	El programa se cae

Al ingresar el texto descriptivo para la oferta común:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa una opción válida (texto que contenga "voucher, tarjeta o efectivo"): con voucher	Se crea el la opción correcta	OK
Se ingresa una opción inválida de menú (texto que contenga "voucher, tarjeta o efectivo"): ramon	Mensaje "El texto no contiene las palabras voucher, efectivo, tarjeta, intente nuevamente:"	OK

Menú Principal:

En negrita se encuentra el valor probado.

Descripción de datos a probar	Resultado esperado	Resultado obtenido en la versión entregada
Se ingresa una opción válida de menú (número del 1-4): 2	Ejecuta la opción del menú correspondiente	OK
Se ingresa una opción inválida de menú (número entero menor que 1 o mayor que 4): -1	Mensaje "Opción no es correcta. Reintente"	OK
Se ingresa una opción inválida de menú (letra): a	Mensaje "Opción no es correcta. Reintente"	El programa se cae.
Se ingresa una opción inválida de menú borde(numero menor a 1 o mayor a 4): 0	Mensaje "Opción no es correcta. Reintente"	OK

Listado Impreso de clases

Clase: "Prueba.java"

```
package obligatorio;

import java.util.*;

/**
 *
 * @author Rodrigo Andre Santomauro Lema (199089)
 */
public class Prueba {

    public static void main(String[] args) {
        //INGRESOS
        ArrayList<Local> listaLocales = new ArrayList<>();
        System.out.println("Se debera ingresar al menos un local para
continuar:");
        Local local = Prueba.crearLocal(listaLocales); //Se ingresa un local
        listaLocales.add(local);

        System.out.println("\nSe debera ingresar al menos un articulo para
continuar:");
        ArrayList<Articulo> listaArticulos = new ArrayList<>();
        Articulo articulo = Prueba.crearArticulo(listaArticulos); //Se
ingresa un articulo
        listaArticulos.add(articulo);

        ArrayList<Oferta> listaOfertaComun = new ArrayList<>();

        ArrayList<Oferta> listaOfertaPermanente = new ArrayList<>();

        //        Iterator<Local> itLocal = listaLocales.iterator();
        //        while (itLocal.hasNext()){
        //            Local aux = itLocal.next();
        //            System.out.println(aux.toString());
        //        }

        System.out.println(listaArticulos);
    }
}
```

```
        menuPrincipal(listaLocales, listaArticulos, listaOfertaComun,
listaOfertaPermanente);

    }

    public static Local locales (String unNombre, ArrayList<Local>
listaLocales){
        Local local = new Local();

        for(int i=0; i < listaLocales.size();i++){
            String nombreListaLocal = listaLocales.get(i).getNombre();
            if(unNombre.equals(nombreListaLocal)){
                local = listaLocales.get(i);
            }
            else{
                System.out.println("El local no existe, se debera crear
uno:");
                local = crearLocal(listaLocales);
            }
        }
        return local;
    }

    public static Artículo articulos (Long unCodBar, ArrayList<Articulo>
listaArticulos){
        Artículo articulo = new Artículo();

        for(int i=0; i < listaArticulos.size();i++){
            long codigoDeBarras =
listaArticulos.get(i).getCodigoDeBarras();
            if(unCodBar==codigoDeBarras){
                articulo=listaArticulos.get(i);
            }
            else{
                System.out.println("El articulo no existe se debera
crear uno: ");
                articulo=crearArticulo(listaArticulos);
            }
        }

        return articulo;
    }

    public static void listarLocales(ArrayList<Local> listaLocales){
        for(int i=0; i < listaLocales.size();i++){
            Local local = new Local();
```



```
        local.setNombre(listaLocales.get(i).getNombre());
        local.setTipo(listaLocales.get(i).getTipo());
        local.setDireccion(listaLocales.get(i).getDireccion());

        System.out.println(local.toString());
    }
}

    public static void listarArticulos(ArrayList<Articulo>
listaArticulos){
        for(int i=0; i < listaArticulos.size();i++){
            Articulo articulo = new Articulo();

            articulo.setCodigoDeBarras(listaArticulos.get(i).getCodigoDeBarras());

            articulo.setDescripcion(listaArticulos.get(i).getDescripcion());

            articulo.setPrecioDeReferencia(listaArticulos.get(i).getPrecioDeReferencia
            ());

            articulo.setRubro(listaArticulos.get(i).getRubro());

            System.out.println(articulo.toString());
        }
    }

    public static void menuPrincipal(ArrayList<Local> listaLocales,
ArrayList<Articulo> listaArticulos, ArrayList<Oferta> listaOfertaComun,
ArrayList<Oferta> listaOfertaPermanente){
        Scanner in = new Scanner(System.in);
        int opcion;

        System.out.println("Menu principal, elija una opcion:");

        do {
            System.out.println("1- Registrar un local");
            System.out.println("2- Registrar un artículo");
            System.out.println("3- Registrar una oferta comun");
            System.out.println("4- Registrar una oferta permanente");
            System.out.println("5- Listado de ofertas");
            System.out.println("6- Consulta de articulo");
            System.out.println("7- Estadistica");
            System.out.println("8- Verificar Infraccion");
            System.out.println("9- Terminar");
            opcion = in.nextInt();
            in.nextLine();
        }
```

```
switch(opcion)
{
    case 1:
        Local local = Prueba.crearLocal(listaLocales); //Se
ingresa un local
        listaLocales.add(local);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 2:
        Artículo articulo =
Prueba.crearArticulo(listaArticulos);
        listaArticulos.add(articulo);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 3:
        Oferta ofertaComun =
Prueba.crearOfertaComun(listaLocales,listaArticulos); //Se ingresa una
oferta Comun
        listaOfertaComun.add(ofertaComun);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 4:
        Oferta ofertaPermanente =
Prueba.crearOfertaPermanente(listaLocales,listaArticulos); //Se ingresa
una oferta Permanente
        listaOfertaComun.add(ofertaPermanente);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 5:
        listarOfertas(listaOfertaComun, listaOfertaPermanente);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 6:
        rangoDePrecios(listaOfertaComun, listaOfertaPermanente);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
        break;
    case 7:
        estadistica(listaOfertaComun, listaOfertaPermanente);
        //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
}
```

```
                break;
            case 8:
                verificarInfraccion(listaOfertaComun);
                //menuPrincipal(listaLocales, listaArticulos,
listaOfertaComun, listaOfertaPermanente);
                break;
            case 9:
                break;
            default:
                System.out.println("Opcion no es correcta. Reintente");
                break;
        }
    }
    while (opcion!=9);

}
```

```
public static Local crearLocal(ArrayList<Local> listaLocales){
    Scanner in = new Scanner(System.in);
    String nombre;
    String tipo="Otro";
    String direccion;
```

```
    boolean correcto = false;
```

```
    do{
        System.out.println("Ingresar nombre del local: ");
        nombre=in.nextLine();
    }while(nombre.length() < 1);
```

```
    int opcion;
    System.out.println("Ingresa el tipo de local: ");
```

```
    do {
        System.out.println("1- Almacen");
        System.out.println("2- Supermercado");
        System.out.println("3- Hipermercado");
        System.out.println("4- Kiosco");
        System.out.println("5- Otro");
        opcion = in.nextInt();
```

```
        switch(opcion)
        {
            case 1:
```

```
        tipo= "1- Almacen";
        correcto=true;
        break;
    case 2:
        tipo= "2- Supermercado";
        correcto=true;
        break;
    case 3:
        tipo= "3- Hipermercado";
        correcto=true;
        break;
    case 4:
        tipo= "4- Kiosco";
        correcto=true;
        break;
    case 5:
        tipo= "5- Otro";
        correcto=true;
        break;
    default:
        System.out.println("Opcion no es correcta.
Reintente");
        break;
    }
}
while (!correcto);
in.nextLine();

do{
    System.out.println("Ingresar la direccion del local: ");
    direccion=in.nextLine();
}while(nombre.length() < 1);

Local local = new Local();
local.setNombre(nombre);
local.setTipo(tipo);
local.setDireccion(direccion);

//Valida si no esta repetido el nombre
correcto=false;
if(listaLocales.size() > 0){
    do{
        for(int i=0; i < listaLocales.size();i++){
            String nombreListaLocal =
listaLocales.get(i).getNombre();
            if(nombre.equals(nombreListaLocal)){
```

```
        System.out.println("El nombre " + nombre + " ya
existe, ingrese uno nuevo");
        do{
            System.out.println("Ingresar nombre del
local: ");
            nombre=in.nextLine();
            i=0;
        }while(nombre.length() < 1);
    }
    else {
        local.setNombre(nombre);
        correcto=true;
    }
    }
    }while(!correcto);
}

return local;
}

public static Artículo crearArticulo(ArrayList<Articulo>
listaArticulos){
    Scanner in = new Scanner(System.in);
    String descripcion;
    int rubro=8;
    long codigoDeBarras;
    double precioDeReferencia;

    do{
        System.out.println("Ingrese la descripcion del articulo: ");
        descripcion = in.nextLine();
    }while(descripcion.length() < 1);

    do{
        System.out.println("Ingrese el codigo de barras del articulo: ");
        codigoDeBarras = in.nextLong();
    }while(codigoDeBarras < 1);

    int opcion;
    System.out.println("Ingrese el rubro del articulo: ");
    do {
        System.out.println("1- Congelado");
        System.out.println("2- Bebida sin alcohol");
        System.out.println("3- Bebida con alcohol");
        System.out.println("4- Fruta/Verdura");
        System.out.println("5- Alimento");
```

```
System.out.println("6- Producto para bebes");
System.out.println("7- Limpieza");
System.out.println("8- Otro");
opcion = in.nextInt();

switch(opcion)
{
    case 1:
        rubro=1;
        break;
    case 2:
        rubro=2;
        break;
    case 3:
        rubro=3;
        break;
    case 4:
        rubro=4;
        break;
    case 5:
        rubro=5;
        break;
    case 6:
        rubro=6;
        break;
    case 7:
        rubro=7;
        break;
    case 8:
        rubro=8;
        break;
    default:
        System.out.println("Opcion no es correcta.
Reintente");
        break;
}

while ((opcion<1) || (opcion>8));

in.nextLine();
do{
    System.out.println("Ingresa el precio de referencia del
articulo: ");
    precioDeReferencia = in.nextInt();
}while(precioDeReferencia < 1);
```

```
Articulo articulo = new Articulo();
articulo.setDescripcion(descripcion);
articulo.setCodigoDeBarras(codigoDeBarras);
articulo.setRubro(rubro);
articulo.setPrecioDeReferencia(precioDeReferencia);

boolean correcto=false;
if(listaArticulos.size() > 0){
    do{
        for(int i=0; i < listaArticulos.size();i++){
            long auxCodBar =
listaArticulos.get(i).getCodigoDeBarras();
            if(codigoDeBarras == auxCodBar){
                System.out.println("El codigo de barras " +
codigoDeBarras + " ya existe, ingrese uno nuevo");
                do{
                    System.out.println("Ingresar nuevo codigo de
barras: ");

                    codigoDeBarras=in.nextLong();
                    i=0;
                }while(codigoDeBarras < 1);
            }
            else {
                articulo.setCodigoDeBarras(codigoDeBarras);
                correcto=true;
            }
        }
    }while(!correcto);
}

return articulo;
}

public static Oferta crearOfertaComun(ArrayList<Local> listaLocales,
ArrayList<Articulo> listaArticulos){
    Scanner in = new Scanner(System.in);
    String nombre = "";
    long codBarras;
    String textoDescriptivo="";
    String textoVoucher = "voucher";
    String textoEfectivo = "efectivo";
    String textoTarjeta = "tarjeta";
    int valor;
```

```
int mesDeVigencia;

boolean correcto = false;
Local local = new Local();
Articulo articulo = new Articulo();

System.out.println("Ingrese el nombre del local comercial de la
lista:");
listarLocales(listaLocales);
nombre = in.nextLine(); //falta contemplar si ingresa uno que no
esta
local = locales(nombre, listaLocales);

System.out.println("Ingrese el codigo de barras del articulo de la
lista: ");
listarArticulos(listaArticulos);
codBarras = in.nextInt(); //falta quee contemplar si ingresa uno que
no esta
articulo = articulos(codBarras, listaArticulos);

System.out.println("Ingrese el texto descriptivo: ");
textoDescriptivo = in.nextLine();
do {
    if(textoDescriptivo.contains(textoEfectivo) ||
(textoDescriptivo.contains(textoTarjeta)) ||
textoDescriptivo.contains(textoVoucher) ){
        correcto=true;
    }
    else{
        System.out.println("El texto no contiene las palabras
voucher, efectivo, tarjeta, intente nuevamente: ");
        textoDescriptivo = in.nextLine();
    }
}while (!correcto);

do{
    System.out.println("Ingrese el valor de la oferta: ");
    valor = in.nextInt();
}while(valor < 1);

System.out.println("Ingrese el mes de vigencia: ");
mesDeVigencia = in.nextInt();
while ((mesDeVigencia<1)||(mesDeVigencia>12))
{
    System.out.println("La opcion es invalida, ingresar
nuevamente.");
}
```



```
        mesDeVigencia = in.nextInt();
    }

    Oferta oferta = new Oferta();
    oferta.setLocal(local);
    oferta.setArticulo(articulo);
    oferta.setTextoDescriptivo(textoDescriptivo);
    oferta.setValor(valor);
    oferta.setMesDeVigencia(mesDeVigencia);

    return oferta;
}

public static Oferta crearOfertaPermanente(ArrayList<Local>
listaLocales, ArrayList<Articulo> listaArticulos){
    Scanner in = new Scanner(System.in);
    String nombre;
    long codBarras;
    int valor;
    String responsable;
    Local local = new Local();
    Articulo articulo = new Articulo();

    System.out.println("Ingrese el nombre del local comercial de la
lista:");
    listarLocales(listaLocales);
    nombre = in.nextLine(); //Hay que contemplar si ingresa uno que no
esta
    local = locales(nombre, listaLocales);

    System.out.println("Ingrese el codigo de barras del articulo de la
lista: ");
    listarArticulos(listaArticulos);
    codBarras = in.nextInt(); //Hay que contemplar si ingresa uno que
no esta
    articulo = articulos(codBarras, listaArticulos);

    do{
        System.out.println("Ingrese el valor de la oferta: ");
        valor = in.nextInt();
    }while(valor < 1);

    in.nextLine();

    do{
        System.out.println("Ingrese el responsable de la oferta: ");
```

```
        responsable = in.nextLine();
    }while(responsable.length() < 1);

    Oferta oferta = new Oferta();
    oferta.setLocal(local);
    oferta.setArticulo(articulo);
    oferta.setValor(valor);
    oferta.setResponsable(responsable);

    return oferta;
}

public static void listarOfertas(ArrayList<Oferta> listaOfertaComun,
ArrayList<Oferta> listaOfertaPermanente){
    ArrayList<Oferta> todasLasOfertas = new ArrayList<>();

    System.out.println("Ofertas comunes:");
    for(int i=0; i < listaOfertaComun.size();i++){
        todasLasOfertas.add(listaOfertaComun.get(i));
    }

    System.out.println("Ofertas permanentes:");
    for(int i=0; i < listaOfertaPermanente.size();i++){
        todasLasOfertas.add(listaOfertaPermanente.get(i));
    }

    Collections.sort(todasLasOfertas);
    System.out.println("Listado de las ofertas");
    System.out.println(todasLasOfertas);
}

public static void rangoDePrecios(ArrayList<Oferta>
listaOfertaComun, ArrayList<Oferta> listaOfertaPermanente){
    Scanner in = new Scanner(System.in);
    long codBarras;

    ArrayList<Oferta> todasLasOfertas = new ArrayList<>();
    for(int i=0; i < listaOfertaComun.size();i++){
        todasLasOfertas.add(listaOfertaComun.get(i));
    }

    for(int i=0; i < listaOfertaPermanente.size();i++){
        todasLasOfertas.add(listaOfertaPermanente.get(i));
    }
}
```

```
        System.out.println("Ingrese un articulo de la siguiente lista de  
codigos de barra: ");  
        for(int i=0; i < listaOfertaComun.size();i++){  
            codBarras =  
listaOfertaComun.get(i).getArticulo().getCodigoDeBarras();  
            System.out.println(codBarras);  
        }  
        for(int i=0; i < listaOfertaPermanente.size();i++){  
            codBarras =  
listaOfertaPermanente.get(i).getArticulo().getCodigoDeBarras();  
            System.out.println(codBarras);  
        }  
  
        codBarras=in.nextLong();  
  
        encontrarArticulos(codBarras, todasLasOfertas);  
    }  
  
    public static void encontrarArticulos(long unCodigoDeBarras,  
ArrayList<Oferta> todasLasOfertas){  
        int maximo, minimo;  
        maximo = todasLasOfertas.get(0).getValor();  
        minimo = todasLasOfertas.get(0).getValor();  
        for(int i=0; i < todasLasOfertas.size(); i++){  
            if(maximo < todasLasOfertas.get(i).getValor()){  
                maximo = todasLasOfertas.get(i).getValor();  
            }  
            if(minimo > todasLasOfertas.get(i).getValor()){  
                minimo = todasLasOfertas.get(i).getValor();  
            }  
        }  
    }  
  
    System.out.println("El rango es $" + minimo + " - $" + maximo);  
    }  
  
    public static void estadistica(ArrayList<Oferta> listaOfertaComun,  
ArrayList<Oferta> listaOfertaPermanente){  
        int contador=0;  
        int maximo=0;  
        int auxRubro=0;  
  
        ArrayList<Oferta> todasLasOfertas = new ArrayList<>();  
        for(int i=0; i < listaOfertaComun.size();i++){  
            todasLasOfertas.add(listaOfertaComun.get(i));  
        }  
    }
```

```
for(int i=0; i < listaOfertaPermanente.size();i++){
    todasLasOfertas.add(listaOfertaPermanente.get(i));
}

for(int rubro=1; rubro <=8; rubro++){
    for(int i=0; i < todasLasOfertas.size();i++){
        if (rubro ==
todasLasOfertas.get(i).getArticulo().getRubro()){
            contador++;
        }
    }
    System.out.println("Para el rubro " + rubro + " existen " +
contador + " ofertas");
    if(maximo<contador){
        maximo=contador;
        auxRubro=rubro;
    }
    contador=0;
}
System.out.println("El rubro con mas ofertas es " + auxRubro);
}

public static void verificarInfraccion(ArrayList<Oferta>
listaOfertaComun){
    for(int i = 0; i < listaOfertaComun.size();i++){
        if((listaOfertaComun.get(i).getValor() /2) >
listaOfertaComun.get(i).getArticulo().getPrecioDeReferencia()){
            System.out.println("LOCAL EN INFRACCION:\n" +
listaOfertaComun.get(i).getLocal().toString());
        }
    }
}
}
```

Clase: "Local.java"

```
package obligatorio;

import java.util.*;
/**
 *
 * @author Rodrigo Andre Santomauro Lema (199089)
 */
public class Local {
    private String nombre;
    private String tipo;
    private String direccion;

    public String getNombre(){
        return this.nombre;
    }

    public void setNombre(String unNombre){
        this.nombre = unNombre;
    }

    public String getTipo(){
        return this.tipo;
    }

    public void setTipo(String unTipo){
        this.tipo = unTipo;
    }

    public String getDireccion(){
        return this.direccion;
    }

    public void setDireccion(String unaDireccion){
        this.direccion = unaDireccion;
    }

    @Override
```

```
    public String toString(){
        return "Nombre: " + this.nombre + "\nTipo: " + this.tipo +
"\nDireccion: " + this.direccion + "\n";
    }
}
```

Clase: "Articulo.java"

```
package obligatorio;

/**
 *
 * @author Rodrigo Andre Santomauro Lema (199089)
 */
public class Articulo {
    private String descripcion;
    private long codigoDeBarras;
    private int rubro;
    private double precioDeReferencia;

    public String getDescripcion(){
        return this.descripcion;
    }

    public void setDescripcion(String unaDescripcion){
        this.descripcion = unaDescripcion;
    }

    public long getCodigoDeBarras(){
        return this.codigoDeBarras;
    }

    public void setCodigoDeBarras(long unCodigoDeBarras){
        this.codigoDeBarras = unCodigoDeBarras;
    }

    public int getRubro(){
        return this.rubro;
    }

    public void setRubro(int unRubro){
        this.rubro = unRubro;
    }

    public double getPrecioDeReferencia(){
```

```
    return this.precioDeReferencia;
}

    public void setPrecioDeReferencia(double unPrecioDeReferencia){
        this.precioDeReferencia = unPrecioDeReferencia;
    }
    @Override
    public String toString(){
        return "Descripcion: " + this.descripcion + "\nCodigo de Barras: " +
this.codigoDeBarras + "\nRubro: " + this.rubro + "\nPrecio de referencia:
" + this.precioDeReferencia + "\n";
    }
}
```

Clase: "Oferta.java"

```
package obligatorio;

/**
 *
 * @author Rodrigo Andre Santomauro Lema (199089)
 */
public class Oferta implements Comparable<Oferta> {
    private Local local;
    private Artículo articulo;
    private String textoDescriptivo;
    private int valor;
    private int mesDeVigencia;
    private String responsable;

    public Local getLocal(){
        return local;
    }

    public void setLocal(Local unLocal){
        this.local = unLocal;
    }

    public Artículo getArticulo(){
        return articulo;
    }

    public void setArticulo(Artículo unArticulo){
        this.articulo = unArticulo;
    }

    public String getTextoDescriptivo(){
        return textoDescriptivo;
    }

    public void setTextoDescriptivo(String unTextoDescriptivo){
        this.textoDescriptivo = unTextoDescriptivo;
    }
}
```



```
public int getValor(){
return valor;
}

public void setValor(int unValor){
this.valor = unValor;
}

public int getMesDeVigencia(){
return mesDeVigencia;
}

public void setMesDeVigencia(int unMesDeVigencia){
this.mesDeVigencia = unMesDeVigencia;
}

public String getResponsable(){
return this.responsable;
}

public void setResponsable(String unResponsable){
this.responsable = unResponsable;
}

@Override
public String toString(){
return "El local es: " + this.local + "\nEl articulo es: " +
this.articulo + "\nDescripcion: " + this.textoDescriptivo + "\nSu valor
es: " + this.valor + "\nSu mes de vigencia es: " + this.mesDeVigencia +
"\n";
}

@Override
public int compareTo(Oferta valor) {
return this.getValor() - this.valor;
}
}
```