

# Universidad ORT

## Facultad de Ingeniería

Programación I

Obligatorio II – Grupo M1A



*Agustín Verdera Rodríguez (182584)*

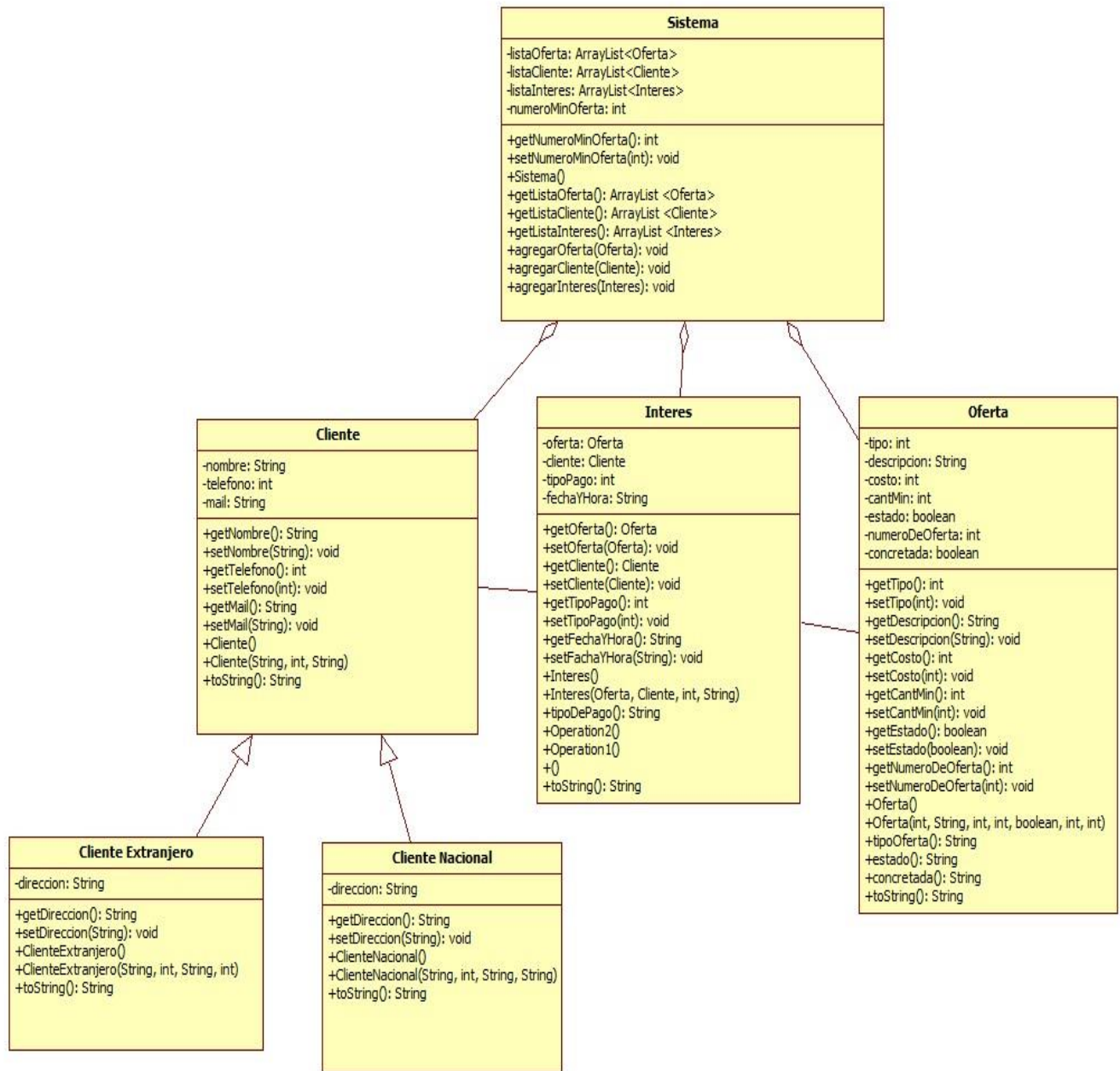
## Índice

Introducción: .....	3
UML: .....	4
Tabla datos de prueba:.....	5
Manual de usuario: .....	13
Código: .....	15
Clase Oferta:.....	15
Clase Cliente:.....	19
Clase Cliente Nacional:.....	20
Clase Cliente Extranjero: .....	21
Clase Interés:.....	22
Clase Sistema:.....	23

## **Introducción:**

Para este obligatorio se desarrolla un programa de una importante empresa que está en plena carrera de renovación. Fuaaaa es una empresa que realiza ofertas de diversos productos y servicios por Internet. Cada oferta está abierta durante un cierto período y requiere un mínimo de clientes para concretarse. Durante su vigencia, los clientes marcan su interés en ella. Si al momento de cerrar la oferta se superó o igualó el mínimo, la oferta se concreta, se notifica a los clientes interesados y se realizan las ventas efectivas.

## UML:



### Tabla datos de prueba:

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
Se ingresa 1.	Ingresa al menú 1) Registro de Oferta.	Ejecuta la acción correspondiente.	OK
Se ingresa 2.	Ingresa al menú 2) Consulta de Ofertas.	Ejecuta la acción correspondiente.	OK
Se ingresa 3.	Ingresa al menú 3) Registro de Cliente Nacional o Extranjero.	Ejecuta la acción correspondiente.	OK
Se ingresa 4	Ingresa al menú 4) Registro de Interés.	Ejecuta la acción correspondiente.	OK
Se ingresa 5	Ingresa al menú 5) Cierre de Oferta.	Ejecuta la acción correspondiente.	OK
Se ingresa 6	Ingresa al menú 6) Consulta de Ofertas.	Ejecuta la acción correspondiente.	OK
Se ingresa 7	Ingresa al menú 7) Consulta de tipo.	Ejecuta la acción correspondiente.	OK
Se ingresa 8	Ingresa al menú 8) Reiniciar numerador de Ofertas.	Ejecuta la acción correspondiente.	OK
Se ingresa 9	Salir del Programa.	Sale del programa.	OK
Se ingresa "a"	ERROR: Debe ingresar un valor numérico.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 0	Ingresar un valor entre 1 y 9.	Ingresar un valor entre 1 y 9.	OK
Se ingresa 10	Ingresar un valor entre 1 y 9.	Ingresar un valor entre 1 y 9.	OK

### Prueba de opción 1 del menú principal (Registro de Oferta)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
Se ingresa 0	Ingresar tipo de oferta: número (1-7)	Ingresar un valor entre 1 y 7.	OK
Se ingresa 8	Ingresar tipo de oferta: número (1-7)	Ingresar un valor entre 1 y 7.	OK
Se ingresa "a"	Ingresar tipo de oferta: número (1-7)	ERROR: Debe ingresar un valor numérico.	OK

Se ingresa 1	Ingresar tipo de oferta: número (1-7)	Ejecuta la acción correspondiente.	OK
<b>Datos ingresados</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Resumen</b>
Se ingresa 7	Ingresar tipo de oferta: número (1-7)	Ejecuta la acción correspondiente.	OK
Se ingresa “una descripción”	Ingresar una descripción.	Ejecuta la acción correspondiente.	OK
Se ingresa 99	Ingresar un valor entre 100 y 5000.	Ingresar un valor entre 100 y 5000.	OK
Se ingresa 5001	Ingresar un valor entre 100 y 5000.	Ingresar un valor entre 100 y 5000.	OK
Se ingresa 5000	Ingresar un valor entre 100 y 5000.	Ejecuta la acción correspondiente.	OK
Se ingresa 100	Ingresar un valor entre 100 y 5000.	Ejecuta la acción correspondiente.	OK
Se ingresa 0	Ingresar cantidad mínima necesaria de interesados para su concreción, número entre 1 y 100.	Ingresar un valor entre 1 y 100.	OK
Se ingresa 101	Ingresar cantidad mínima necesaria de interesados para su concreción, número entre 1 y 100.	Ingresar un valor entre 1 y 100.	OK
Se ingresa “a”	Ingresar cantidad mínima necesaria de interesados para su concreción, número entre 1 y 100.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 1	Ingresar cantidad mínima necesaria de interesados para su concreción, número entre 1 y 100.	Ejecuta la acción correspondiente.	OK
Se ingresa 100	Ingresar cantidad mínima necesaria de interesados para su concreción, número entre 1 y 100.	Ejecuta la acción correspondiente.	OK

## Prueba de opción 2 del menú principal (Consulta de Ofertas)

<b>Datos ingresados</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Resumen</b>
Se ingresa 0	Ingresar un tipo de oferta, número (1-7)	Ingresar un valor entre (1 y 7)	OK
Se ingresa 8	Ingresar un tipo de oferta, número (1-7)	Ingresar un valor entre (1 y 7)	OK

Se ingresa 1	Muestra la lista ordenada por costo creciente.	Ejecuta la acción correspondiente.	OK
Se ingresa 7	Muestra la lista ordenada por costo creciente.	Ejecuta la acción correspondiente.	OK
Se ingresa "a"	Ingresar un tipo de oferta, número (1-7)	ERROR: Debe ingresar un valor	OK

### Prueba de opción 3 del menú principal (Registro de Cliente Nacional o Extranjero)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
Se ingresa 0	Desea registrar un cliente 1-Nacional 2-Extranjero.	Ingresar un valor entre 1 y 2.	OK
Se ingresa 3	Desea registrar un cliente 1-Nacional 2-Extranjero.	Ejecuta la acción correspondiente.	OK
Se ingresa 1	Desea registrar un cliente 1-Nacional 2-Extranjero	Ingresar a Registro Cliente Nacional.	OK
Se ingresa "Juan"	Ingresar un nombre.	Ejecuta la acción correspondiente.	OK
Se ingresa "asds"	Ingresar un teléfono.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 0992211	Ingresar un teléfono.	Ingresar un teléfono.	OK
Se ingresa "usuario@hotmail"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ingresar un usuario, formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	OK
Se ingresa "usu@gmail.com.uy"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni	OK

	espacios.	espacios.	
<b>Datos ingresados</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>	<b>Resumen</b>
Se ingresa 000	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	OK
Se ingresa "usuario@gmail.com"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ejecuta la acción correspondiente.	OK
Se ingresa "usuario@yahoo.com"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com la palabra usuario no debe contener @ ni espacios.	Ejecuta la acción correspondiente.	OK
Se ingresa "Cuareim 1451"	Ingresar una dirección.	Ejecuta la acción correspondiente.	OK
Se ingresa 2	Desea registrar un cliente 1-Nacional 2- Extranjero.	Ingresa a Registro Cliente Extranjero.	OK
Se ingresa "Pedro"	Ingresar un nombre.	Ejecuta la acción correspondiente.	OK
Se ingresa "asasd"	Ingresar un teléfono.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 0956611	Ingresar un teléfono.	Ingresa un teléfono.	OK
Se ingresa "usuario@hotmail"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ingresar un usuario; formatos aceptados son : usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	OK



Se ingresa "usu@gmail.com.uy"	Ingresar un usuario, formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ingresar un usuario, formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	OK
Se ingresa 000	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	OK
Se ingresa "usuario@gmail.com"	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ejecuta la acción correspondiente.	OK
Se ingresa " <a href="#">usuario@yahoo.com</a> "	Ingresar un usuario; formatos aceptados son: usuario@gmail.com o usuario@yahoo.com La palabra usuario no debe contener @ ni espacios.	Ejecuta la acción correspondiente.	OK
Se ingresa "tres"	Ingresar código de prefijo de país (3 dígitos)	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 22	Ingresar código de prefijo de país (3 dígitos)	Ingresar código de prefijo de país (3 dígitos)	OK
Se ingresa 3333	Ingresar código de prefijo de país (3 dígitos)	Ingresar código de prefijo de país (3 dígitos)	OK
Se ingresa 111	Ingresar código de prefijo de país (3 dígitos)	Ejecuta la acción correspondiente.	OK

## Prueba de opción 4 del menú principal

### (Registro de Registro de Interés)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
(Previo registro de dos clientes) Se ingresa 0	Elija un cliente.	Ingresar un valor entre 1 y 2.	OK
Se ingresa 3	Elija un cliente.	Ingresar un valor entre 1 y 2.	OK
Se ingresa "ajas"	Elija un cliente.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 1	Elija un cliente.	Ejecuta la acción correspondiente.	OK
Se ingresa 2	Elija un cliente.	Ejecuta la acción correspondiente.	OK
(Previo registro de dos ofertas) Se ingresa 0	Elija una oferta.	Ingresar un valor entre 1 y 2.	OK
Se ingresa 3	Elija una oferta.	Ingresar un valor entre 1 y 2.	OK
Se ingresa "una letra"	Elija una oferta.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 1	Elija una oferta.	Ejecuta la acción correspondiente.	OK
Se ingresa 2	Elija una oferta.	Ejecuta la acción correspondiente.	OK
Se ingresa 0	Ingresar tipo de pago número (1-7)	Ingresar un valor entre 1 y 4.	OK
Se ingresa 8	Ingresar tipo de pago número (1-7)	Ingresar un valor entre 1 y 4.	OK
Se ingresa "aa"	Ingresar tipo de pago número (1-7)	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 1	Ingresar tipo de pago número (1-7)	Ejecuta la acción correspondiente.	OK
Se ingresa 7	Ingresar tipo de pago número (1-7)	Ejecuta la acción correspondiente.	OK
Ninguno	Se ha ingresado fecha y hora automáticamente.	Ejecuta la acción correspondiente.	OK

Ninguno	Cada cliente puede marcar interés en una oferta en particular una única vez. Si así lo hace, no se permite que registre de nuevo ese interés con esa misma oferta.	Mensaje: El usuario ya tuvo interés en esta oferta.	OK
---------	--	--	----

## Prueba de opción 5 del menú principal

### (Cierre de Ofertas)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
(Previo registro de interés y oferta)  Se ingresa 0	Muestra las ofertas abiertas, el usuario debe elegir una para poder cerrarse.	Ingresar un valor entre 1 y el largo de la lista.	OK
Se ingresa "aa"	Muestra las ofertas abiertas. El usuario debe elegir una para poder cerrarse.	ERROR: Debe ingresar un valor numérico.	OK
Se ingresa 1	Muestra las ofertas abiertas. El usuario debe elegir una para poder cerrarse.	Ejecuta la acción correspondiente.	OK
Ninguno	Si la cantidad de interesados iguala o supera al mínimo, se concreta y muestra una lista de los clientes que marcaron interés en la oferta.	Ejecuta la acción correspondiente.	OK
Ninguno	Si no se concreta, se dan de baja del sistema todos los registros de interés de esa oferta y se marca como cerrada. Una vez cerrada, no se vuelve a abrir.	Ejecuta la acción correspondiente.	OK

## Prueba de opción 6 del menú principal

### (Consulta de Ofertas)

Resultado esperado	Resultado obtenido	Resumen
Indicar por cada tipo de oferta, el total de ofertas cerradas que fueron concretadas.	Ejecuta la acción correspondiente.	OK
Por cada tipo de oferta, indicar cuánto dinero se generó.	Ejecuta la acción correspondiente.	OK

## Prueba de opción 7 del menú principal

### (Consulta de Tipo)

Resultado esperado	Resultado obtenido	Resumen
Indicar el o los tipos que tienen más ofertas (abiertas o no).	Ejecuta la acción correspondiente.	OK

## Prueba de opción 8 del menú principal

### (Reiniciar numerador de ofertas)

Datos ingresados	Resultado esperado	Resultado obtenido	Resumen
(Previo registro de dos Ofertas) Se ingresa 3	Se debe validar que el número ingresado supere al de la última oferta registrada. No se requiere que sea consecutivo. Al comienzo del sistema el valor inicial es 1.	Ingresar número mayor a 3.	OK
Se ingresa 4	Se debe validar que el número ingresado supere al de la última oferta registrada. No se requiere que sea consecutivo. Al comienzo del sistema el valor inicial es 1.	Ejecuta la acción correspondiente	OK

## Prueba de opción 9 del menú principal

### (Salir del programa)

Resultado esperado	Resultado obtenido	Resumen
Salir del programa.	Ejecuta la acción correspondiente.	OK

## Manual de usuario:

```
Ingresar valor de la opción deseada:  
1-Registro de Oferta:  
2-Consulta de Ofertas:  
3-Registro de Cliente Nacional o Extranjero:  
4-Registro de Interés:  
5-Cierre de oferta:  
6-Consulta de ofertas:  
7-Consulta de tipo:  
8-Reiniciar numerador de ofertas:  
9-Salir del programa:
```

Aparecerá en el programa el menú con las opciones deseadas por el usuario.

El usuario deberá ingresar el número de la opción deseada a continuación de un enter. El programa es bastante intuitivo y si la persona se confunde de número o hay una opción que no puede registrar el mismo, el programa lo avisa.

**Opción 1 Registrar una oferta:** Cada oferta tiene un tipo (1-turismo, 2-servicio, 3-comida, 4-entretenimiento, 5-belleza, 6-educación, 7-otros), una descripción, un costo (entre 100 y 5000) y la cantidad mínima de interesados (entre 1 y 100). Cuando se registra queda en estado “abierta”. Se guardará en forma automática el número de oferta.

**Opción 2 Consulta de oferta:** Debe ingresar el número dos. Se muestran los tipos de oferta y se debe elegir uno. Se muestra una lista ordenada por costo creciente de las ofertas abiertas de ese tipo. Ejemplo:

```
Ingresar un tipo de oferta
1-Turismo
2-Servicio
3-Comida
4-Entretenimiento
5-Belleza
6-Educacion
7-Otros
4
-----
```

```
Tipo de Oferta:4-Entretenimiento
Descripción:Oferta 2
Costo:555
Cantidad mínima de interesados para su concreción:5
Estado:Abierto
Número de Oferta:2
La oferta no ha sido concretada
*****
-----
```

```
Tipo de Oferta:4-Entretenimiento
Descripción:Oferta 1
Costo:1002
Cantidad mínima de interesados para su concreción:2
Estado:Abierto
Número de Oferta:1
La oferta no ha sido concretada
*****
```

### Opción 3 Registro cliente nacional o extranjero:

Se ingresa el número 3 y aparece la opción 1-Registrar cliente nacional 2-Registrar cliente extranjero según su procedencia.

```
Desea registrar un cliente
1- Nacional
2- Extranjero
```

### Cliente nacional:

Se registra nombre, teléfono y mail (que debe ser de gmail o yahoo). Se valida que el mail tenga formato correcto. Los formatos aceptados son: usuario@gmail.com o usuario@yahoo.com. "usuario" no puede contener @ ni espacios. Se ingresa además la dirección del cliente nacional.

### Cliente extranjero:

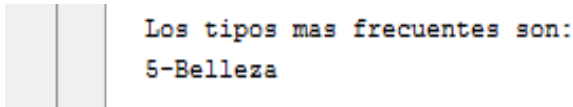
Se registra lo mismo que el cliente nacional a diferencia de código de prefijo que deben ser 3 dígitos.

**4-Opción Registro de interés:** Cuando un cliente encuentra una oferta abierta que le agrada, marca su interés. En el sistema se registra el cliente, la oferta y en forma automática la fecha y hora del sistema. También indica la forma de pago (1- abitab, 2-redpagos, 3-débito bancario, 4-nodefinida). Cada cliente puede marcar interés en una oferta particular una única vez (si el cliente ya había marcado interés en una cierta oferta, no se permite que registre otra vez el interés en esa misma oferta).

**5- Opción Cierre de oferta:** Se muestran las ofertas abiertas y debe elegir el número de oferta para que pase a estar “cerrada”. Se le informa si se concretará o no. En caso de que la cantidad de interesados iguale o supere al mínimo, se concreta y se muestra una lista de los clientes que manifestaron su interés en la oferta. Si no se concreta, se dan de baja del sistema todos los registros de interés de esa oferta y se marca como cerrada. Una vez cerrada, no se vuelve a abrir.

**6 - Opción Consulta de ofertas:** Muestra por pantalla cada tipo de oferta, el total de ofertas cerradas que fueron concretadas. Por cada tipo, muestra además cuánto dinero se generó.

**7-Opción Consulta de tipo:** Muestra por pantalla el o los tipos que tiene más ofertas (abiertas o no). Ejemplo se registran 3 ofertas de tipo entretenimiento y se ingresa a la opción 7:



```
Los tipos mas frecuentes son:
5-Belleza
```

**8-Opción Reiniciar numerador de ofertas:** Se valida que el número a ingresar supere al de la última oferta registrada. Y se ingresa el número al numerador de ofertas.

**9-Salir del Programa:** Al ingresar el número nueve el programa se cierra.

## Código:

### Clase Oferta:

```
package obligatorio;

public class Oferta implements Comparable {
    //Atributos

    private int tipo;
    private String descripcion;
    private int costo;
    private int cantMin;
    private boolean estado;
    private int numeroDeOferta;
    private boolean concretada;

    //Metodos de acceso y modificacion
    public int getTipo() {
        return this.tipo;
    }

    public void setTipo(int unTipo) {
        tipo = unTipo;
    }
}
```

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

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

public int getCosto() {
    return this.costo;
}

public void setCosto(int unCosto) {
    costo = unCosto;
}

public int getCantMin() {
    return this.cantMin;
}

public void setCantMin(int unaCantMin) {
    cantMin = unaCantMin;
}

public boolean getEstado() {
    return this.estado;
}

public void setEstado(boolean unEstado) {
    estado = unEstado;
}

public int getNumeroDeOferta(){
    return this.numeroDeOferta;
}

public void setNumeroDeOferta(int unNumeroDeOferta){
    numeroDeOferta = unNumeroDeOferta;
}

public boolean getConcretada(){
    return this.concretada;
}

public void setConcretada(boolean unConcretada){
    this.concretada = unConcretada;
}

//Constructor sin parámetros
public Oferta() {
    setTipo(0);
}
```



```
setDescripcion("Sin descripción");
setCosto(0);
setCantMin(0);
setEstado(false);
setNumeroDeOferta(1);
setConcretada(false);
}
```

//Constructor con parámetros

```
public Oferta(int unTipo, String unaDescripcion, int unCosto, int unaCantMin,
    boolean unEstado , int unNumeroDeOferta , boolean unConcretada) {
    setTipo(unTipo);
    setDescripcion(unaDescripcion);
    setCosto(unCosto);
    setCantMin(unaCantMin);
    setEstado(unEstado);
    setNumeroDeOferta(unNumeroDeOferta);
    setConcretada(unConcretada);
}
```

```
public String tipoOferta() {
```

```
    String retorno = "Ningun tipo corresponde";
```

```
    switch (this.tipo) {
        case 1: {
            retorno = "1-Turismo";
            break;
        }
        case 2: {
            retorno = "2-Servicio";
            break;
        }
        case 3: {
            retorno = "3-Comida";
            break;
        }
        case 4: {
            retorno = "4-Entretenimiento";
            break;
        }
        case 5: {
            retorno = "5-Belleza";
            break;
        }
        case 6: {
            retorno = "6-Educación";
            break;
        }
        case 7: {
            retorno = "7-Otros";
            break;
        }
    }
```

```

    }
}
return retorno;
}

public String estado() {
    String estado = "";
    if (this.getEstado()) {
        estado = "Abierto";
    } else {
        estado = "Cerrado";
    }
    return estado;
}

public String concretada() {
    String concretada = "";
    if (this.getConcretada()) {
        concretada = "La oferta ha sido concretada";
    } else {
        concretada = "La oferta no ha sido concretada";
    }
    return concretada;
}

@Override
public int compareTo(Object o) {
    return this.getCosto() - ((Oferta) o).getCosto();
}

@Override
public String toString() {

    return "-----\n"
        + "\nTipo de Oferta:" + this.tipoOferta() + "\nDescripción:"
        + this.getDescripcion() + "\nCosto:" + this.getCosto()
        + "\nCantidad mínima de interesados para su concreción:"
        + this.getCantMin() + "\nEstado:" + this.estado() + "\nNúmero de Oferta:"
        + this.getNumeroDeOferta() + "\n" + this.concretada()
        +
        "\n*****",
    }
}

```

## Clase Cliente:

```
package obligatorio;
```

```
public abstract class Cliente {
```

```
    //Atributos
```

```
    private String nombre;
```

```
    private int telefono;
```

```
    private String mail;
```

```
    //Metodos de acceso y modificacion
```

```
    public String getNombre() {
```

```
        return this.nombre;
```

```
    }
```

```
    public void setNombre(String unNombre) {
```

```
        nombre = unNombre;
```

```
    }
```

```
    public int getTelefono() {
```

```
        return this.telefono;
```

```
    }
```

```
    public void setTelefono(int unTelefono) {
```

```
        telefono = unTelefono;
```

```
    }
```

```
    public String getMail() {
```

```
        return this.mail;
```

```
    }
```

```
    public void setMail(String unMail) {
```

```
        mail = unMail;
```

```
    }
```

```
    //Constructor sin parametros
```

```
    public Cliente(){
```

```
        setNombre("Sin nombre");
```

```
        setTelefono(0);
```

```
        setMail("Sin mail");
```

```
    }
```

```
    //Constructor con parametros
```

```
    public Cliente(String unNombre, int unTelefono,String unMail){
```

```
        setNombre(unNombre);
```

```
        setTelefono(unTelefono);
```

```
        setMail(unMail);
```

```
    }
```

```

@Override
public String toString(){
    return "-----\n" +
        "\nNombre cliente:" + this.getNombre() + "\nTeléfono:" + this.getTelefono() +
        "\nMail:" + this.getMail()
        ;
    }
}

```

## Clase Cliente Nacional:

package obligatorio;

```

public class ClienteNacional extends Cliente {
    //Atributos

    private String direccion;

    //Metodos de acceso y modificacion
    public String getDireccion() {
        return direccion;
    }

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

    //Constructor sin parametros
    public ClienteNacional() {
        super.setNombre("Sin nombre");
        super.setTelefono(0);
        super.setMail("Sin mail");
        this.setDireccion("Sin direccion");
    }

    //Constructor con parametros
    public ClienteNacional(String unNombre, int unTelefono, String unMail, String unaDireccion)
    {
        super.setNombre(unNombre);
        super.setTelefono(unTelefono);
        super.setMail(unMail);
        this.setDireccion(unaDireccion);
    }

    @Override
    public String toString() {

```

```

        return "Cliente Nacional : " + super.toString()
            + "Dirección: " + this.getDireccion();
    }
}

```

## Clase Cliente Extranjero:

package obligatorio;

```

public class ClienteExtranjero extends Cliente {

    //Atributos
    private int codigoDePrefijo;

    //Metodos de acceso y modificacion
    public int getCodigoDePrefijo() {
        return codigoDePrefijo;
    }

    public void setCodigoDePrefijo(int unCodigoDePrefijo) {
        codigoDePrefijo = unCodigoDePrefijo;
    }

    //Constructor sin parametros
    public ClienteExtranjero() {
        super.setNombre("Sin nombre");
        super.setTelefono(0);
        super.setMail("Sin mail");
        setCodigoDePrefijo(0);
    }

    //Constructores con parametros
    public ClienteExtranjero(String unNombre, int unTelefono, String unMail, int
unCodigoDePrefijo) {
        super.setNombre(unNombre);
        super.setTelefono(unTelefono);
        super.setMail(unMail);
        setCodigoDePrefijo(unCodigoDePrefijo);
    }

    @Override
    public String toString() {
        return "Cliente Extranjero: " + super.toString()

```

```

        + "Código de prefijo de país: " + this.getCodigoDePrefijo());
    }
}

```

## Clase Interés:

package obligatorio;

```

public class Interes {
    //Atributos

    private Oferta oferta;
    private Cliente cliente;
    private int tipoPago;
    private String fechaYHora;

    //metodos de acceso y modificacion
    public Oferta getOferta() {
        return oferta;
    }

    public void setOferta(Oferta unaOferta) {
        oferta = unaOferta;
    }

    public Cliente getCliente(){
        return cliente;
    }

    public void setCliente(Cliente unCliente){
        cliente = unCliente;
    }

    public int getTipoPago() {
        return tipoPago;
    }

    public void setTipoPago(int unTipoPago) {
        tipoPago = unTipoPago;
    }

    public String getFechaYHora() {
        return fechaYHora;
    }

    public void setFechaYHora(String unaFechaYHora) {
        fechaYHora = unaFechaYHora;
    }

    public Interes() {
        setOferta(new Oferta());
    }
}

```

```

        setCliente(new ClienteNacional());
        setTipoPago(0);
        setFechaYHora("sin fecha");
    }

    public Interes(Oferta unaOferta, Cliente unCliente, int unTipoDePago, String unaFechaYHora)
    {

        setOferta(unaOferta);
        setCliente(unCliente);
        setTipoPago(unTipoDePago);
        setFechaYHora(unaFechaYHora);
    }

    public String tipoDePago(int unTipoPago) {
        String[] tipoDePago = {" ", "1-Abitab", "2-Redpagos", "3-Debito bancario", "4-No
definida"};
        String retorno = tipoDePago[unTipoPago];
        return retorno;
    }

    @Override
    public String toString() {
        return "El cliente \n:" + this.getCliente() + "\nTiene interes en la oferta \n:" +
oferta.toString()
        + "\nEl cliente pagara por: " + tipoDePago(this.getTipoPago()) + "\nFecha y Hora del
sistema:\n" + this.getFechaYHora();

    }
}

```

## Clase Sistema:

```

package obligatorio;

import java.util.*;

public class Sistema {
    //Atributos
    private ArrayList <Oferta> listaOferta ;
    private ArrayList <Cliente> listaCliente;
    private ArrayList <Interes> listaInteres ;
    private int numeroMinOferta;

    //Metodos de acceso y modificacion
    public ArrayList<Oferta> getListaOferta(){
        return listaOferta;
    }
    public ArrayList<Cliente> getListaCliente(){

```

```

        return listaCliente;
    }
    public ArrayList<Interes> getListInteres(){
        return listaInteres;
    }
    public void agregarOferta(Oferta unaOferta){
        listaOferta.add(unaOferta);
    }
    public void agregarCliente(Cliente unCliente){
        listaCliente.add(unCliente);
    }
    public void agregarInteres(Interes unInteres){
        listaInteres.add(unInteres);
    }
    public int getNumeroMinOferta(){
        return this.numeroMinOferta;
    }
    public void setNumeroMinOferta(int unNumeroMinDeOfertas){
        numeroMinOferta = unNumeroMinDeOfertas;
    }
}

//Constructor sin parametros
public Sistema(){
    listaOferta = new ArrayList<Oferta>();
    listaCliente = new ArrayList<Cliente>();
    listaInteres = new ArrayList<Interes>();
    this.setNumeroMinOferta(0);
}
}

```