

MANUAL DE USUARIO

SEMA_SERVICIOS

NOVIEMBRE 2018

INDICÉ.

INTRODUCCION	2
REQUERIMIENTOS BÁSICOS.....	3
MANEJO DE LA LIBRERÍA.....	4
AGREGAR LIBRERÍA.....	4
AGREGAR SERVICIO DE RELOJ.	7
AGREGAR SERVICIO DE CLIMA.....	11
AGREGAR SERVICIO DE CLIMA Y RELOJ.	15

INTRODUCCION

¿Qué es SEMA_SERVICIOS?

SEMA_SERVICIOS es una librería para java que poseen una serie de métodos y clases, que nos permite incorporar a nuestros proyectos desarrollados en lenguaje java dos servicios: clima y un reloj, es decir que podemos hacer uso de los métodos, clases y atributos que componen la librería evitando así tener que implementar nosotros mismos esas funcionalidades.

¿Cuál es el objetivo del Manual de Usuarios?

El objetivo que se persigue con el presente manual, es dar a conocer a los usuarios las características y el modo de funcionamiento de la librería SEMA_SERVICIOS.

REQUERIMIENTOS BÁSICOS

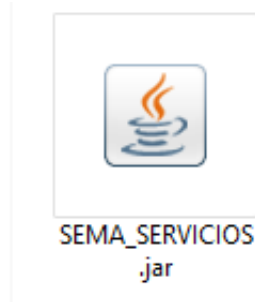
Para la utilización de la librería “SEMA_SERVICIOS”

- Sistema Operativo Windows 10.
- Tener instalado NetBeans IDE 8.2.
- Proyectos desarrollados en Java.

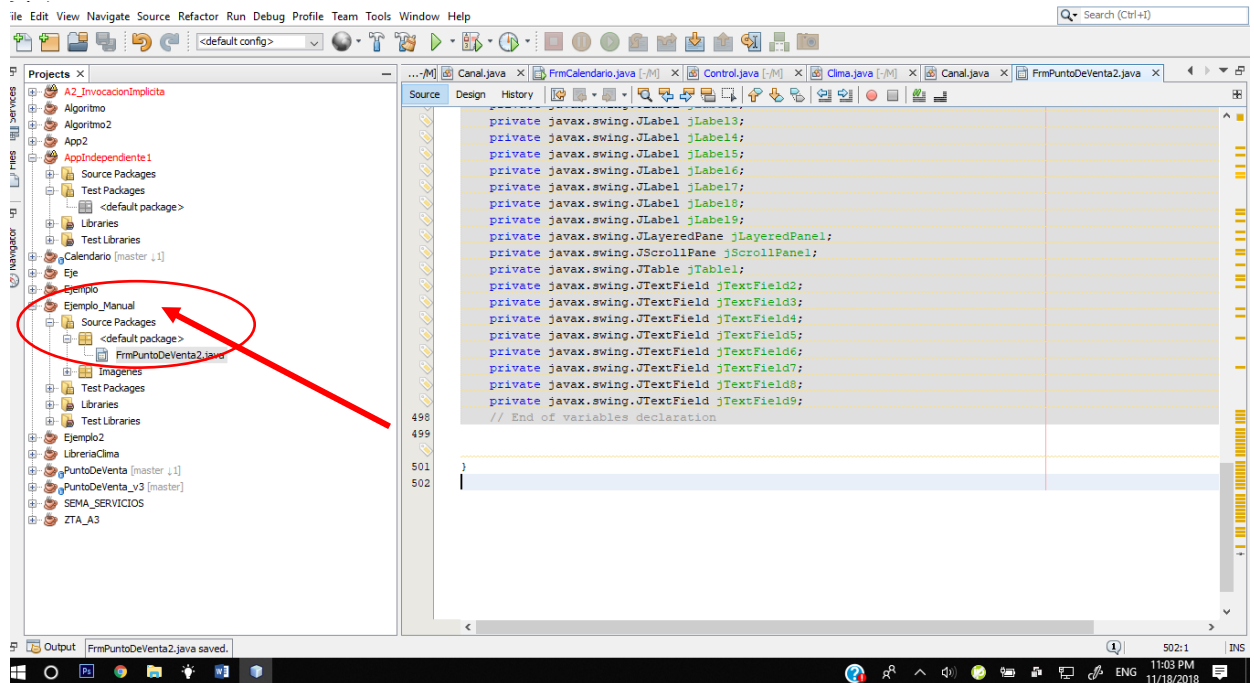
MANEJO DE LA LIBRERÍA

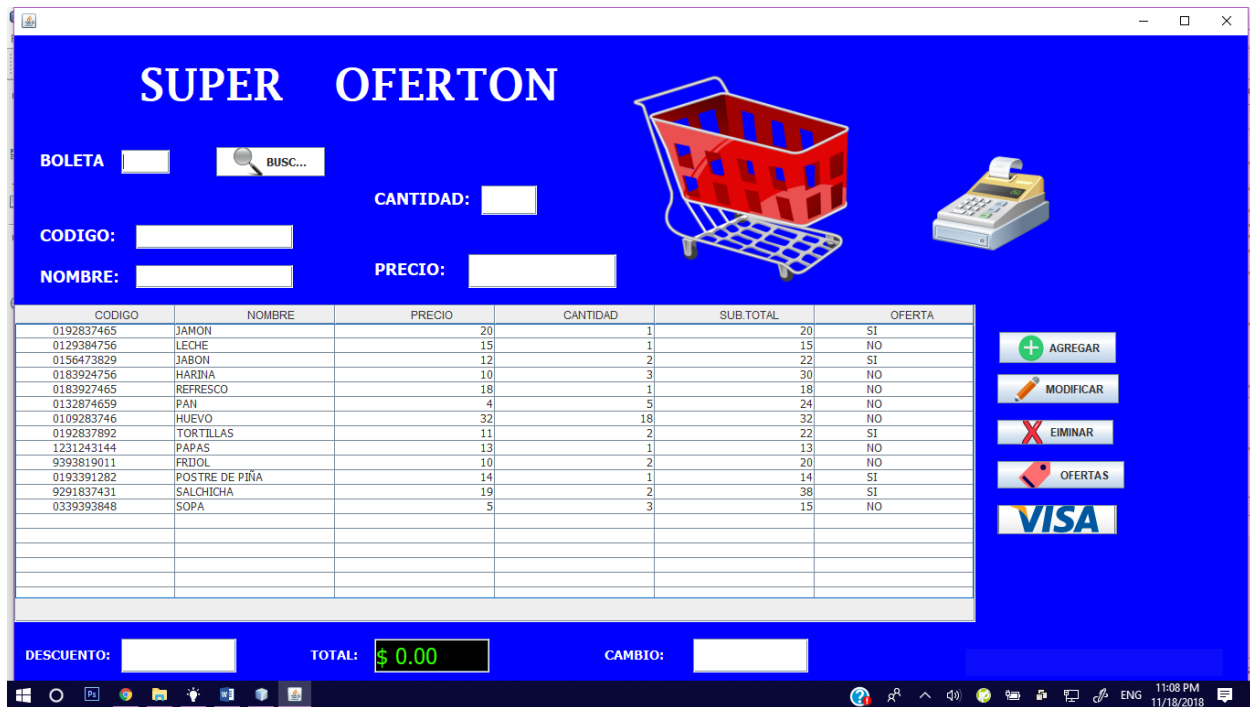
AGREGAR LIBRERÍA.

- Para integrar la librería a un proyecto debe identificar donde se encuentra la librería “SEMA_SERVICIOS”.

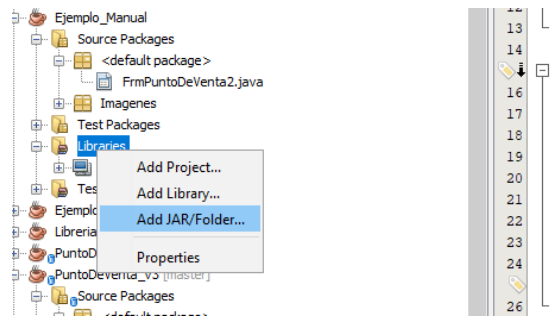


- Identificar el proyecto en el que desea incorporar el servicio, para hacer uso de la librería.

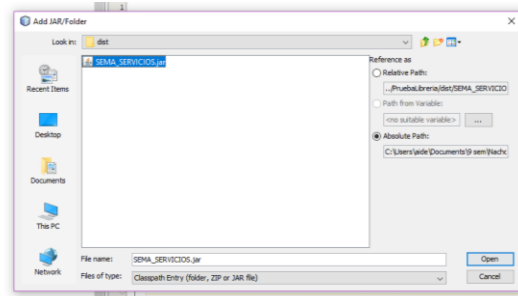




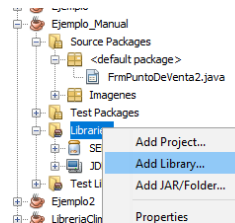
- Agregar Librería al proyecto.
Nos dirigimos a Netbeans IDE donde localizaremos nuestro proyecto; ahora, seleccionaremos la carpeta "Libraries" (Bibliotecas) y damos clic derecho sobre esta, escogiendo seguidamente la opción "Add JAR/Folder..." (Agregar JAR/Carpeta).



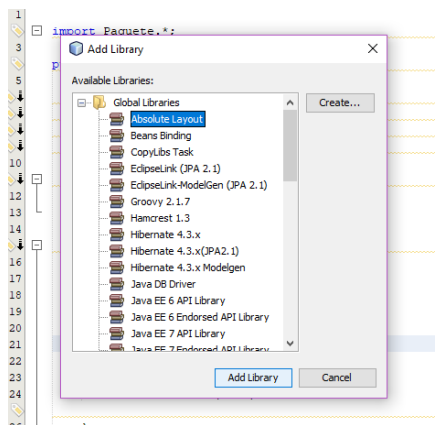
Nos aparecerá una nueva ventana, en la cual vamos a buscar las librerías que queremos y/o necesitamos agregar a nuestro proyecto (seleccione la librería "SEMA_SERVICIOS"), proseguimos a seleccionarla y dar clic en el botón open (abrir).



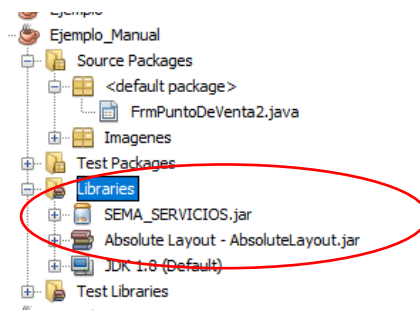
Nos dirigimos nuevamente a la carpeta “Libraries” seleccionamos Add Library (Agregar Librería).



Aparecerá una nueva ventana, en la cual vamos a buscar las librerías de “Absolute Layout”, proseguimos a seleccionarla y dar clic en el botón Add Library (Agregar Librería).

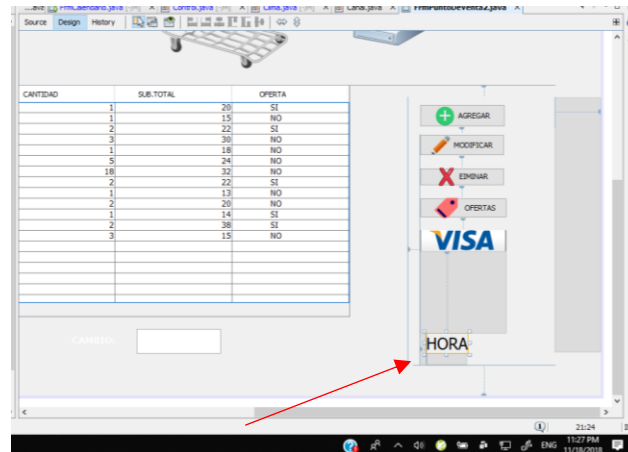


Si nos dirigimos nuevamente a la carpeta "**Libraries**" (Bibliotecas), podremos observar los archivos que hemos agregado en los puntos anteriores.

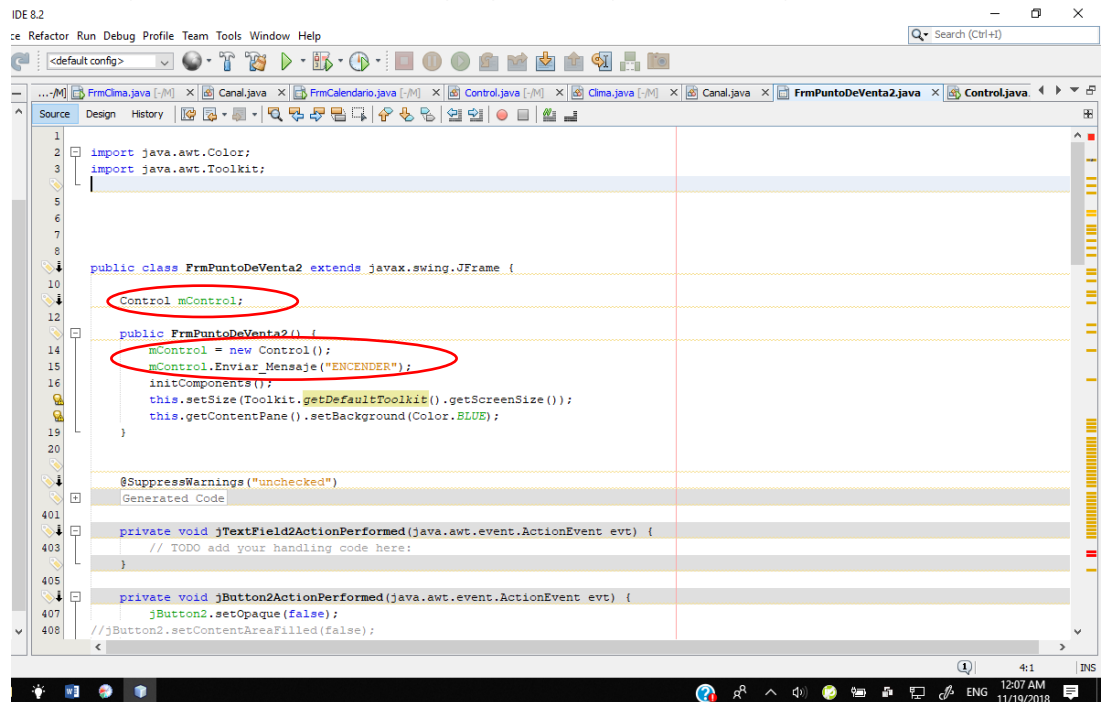


AGREGAR SERVICIO DE RELOJ.

- Para agregar el servicio de reloj a un proyecto desarrollado en java primero asegúrese de haber agregado la librería “SEMA_SERVICIOS” a su proyecto. Identifique el formulario de su proyecto donde desea agregar el servicio de reloj y agregue un Label (etiqueta) al formulario, en la ubicación donde quiere observar la hora. Asegúrese de asignarle un nombre adecuado.



- Acceda al código de la aplicación y agregue el siguiente código.



- Agregue la librería “SEMA_SERVICIOS” que contiene el servicio de reloj.


```

import java.awt.Color;
import java.awt.Toolkit;
import SEMA_SERVICIOS.*;

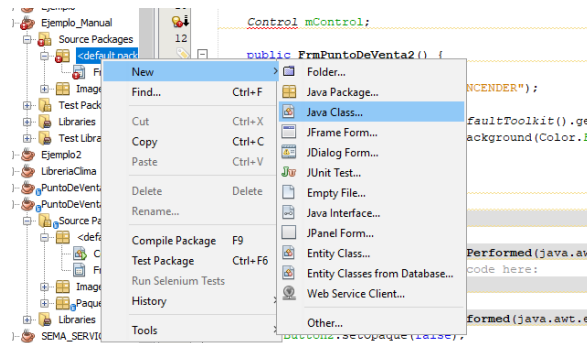
public class FrmPuntoDeVenta2 extends javax.swing.JFrame {

    Control mControl;

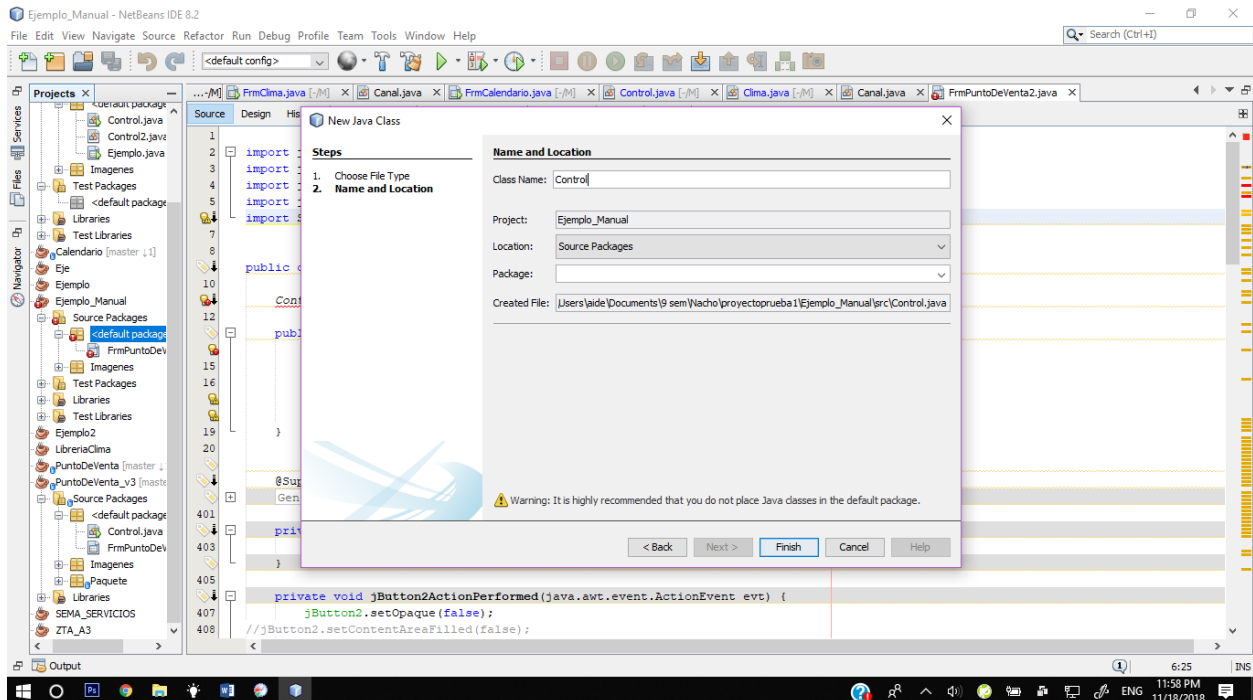
    public FrmPuntoDeVenta2() {
        mControl = new Control();
        mControl.Envia_Mensaje("ENCENDER");
        initComponents();
        this.setSize(Toolkit.getDefaultToolkit().getScreenSize());
        this.getContentPane().setBackground(Color.BLUE);
    }
}

```

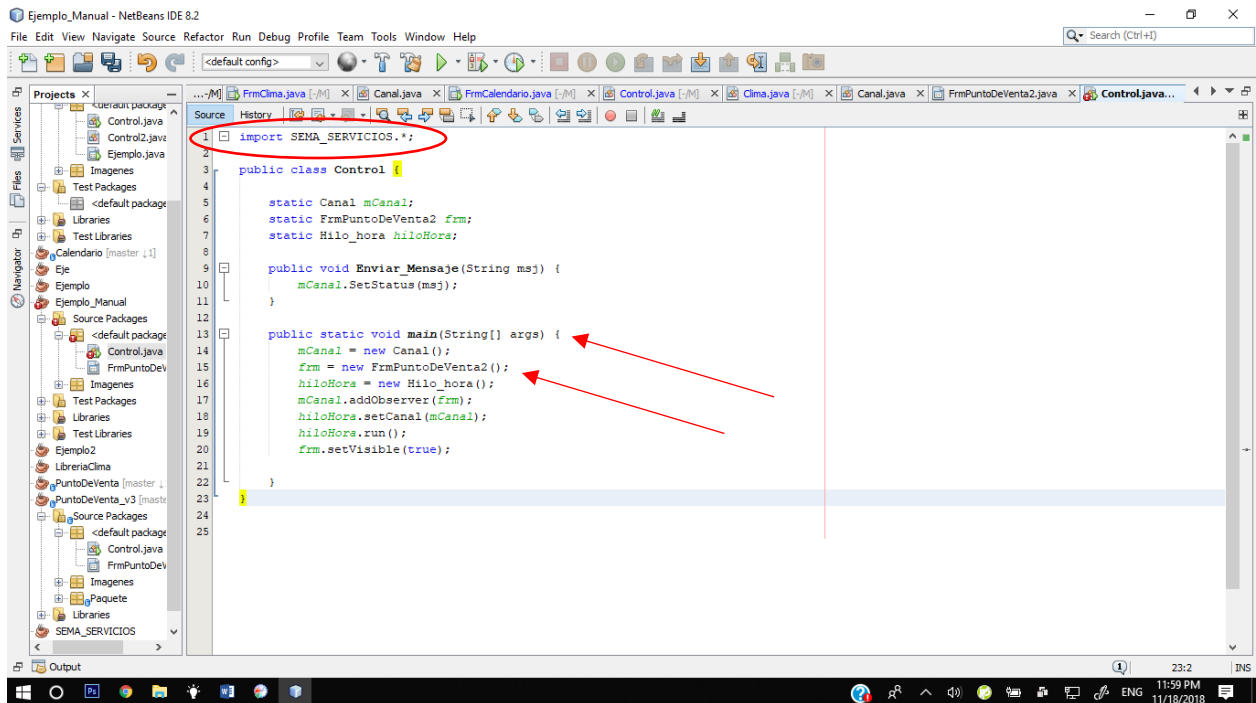
- Para eliminar los errores que marca en la clase control agregue una clase al proyecto.



- Agregue el nombre Control a la clase.

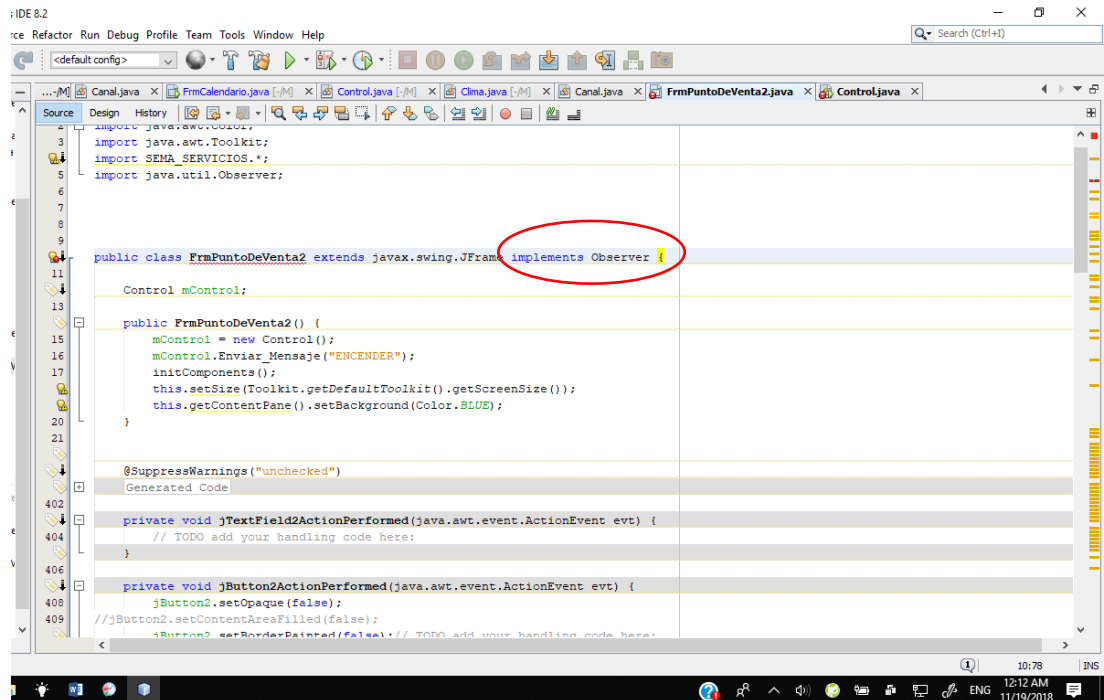


- Dentro de la clase Control agregue el siguiente código.

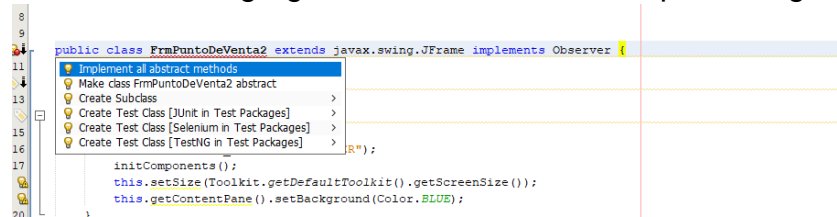


NOTA: Asegúrese de haber importado la librería “SEMA_SERVICIOS”. Además, el método para ejecutar la aplicación e instanciar con el nombre adecuado de su formulario en este ejemplo el nombre es *FrmPuntoDeVenta2*.

- Regrese al código del formulario y agregue el siguiente código “implements Observer”, además de la librería necesaria.



- Para eliminar el error agregue un método abstracto que es sugerido.



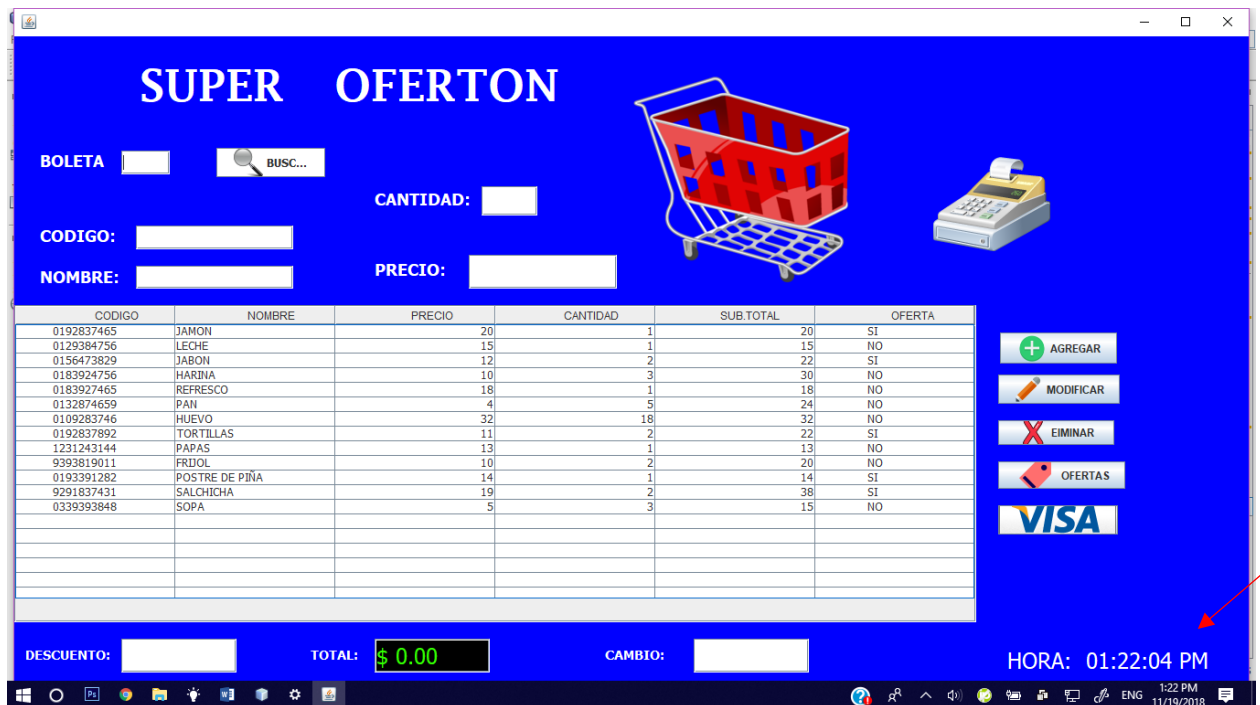
- Al agregar el método abstracto se creara el siguiente método.

```
@Override
public void update(Observable o, Object arg) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates
}
}
```

- Dentro del método agregue el siguiente código (Lreloj es el nombre del Label , agregue el nombre que tenga su Label o etiqueta).

```
@Override
public void update(Observable o, Object arg) {
    Lreloj.setText(arg.toString());
}
```

- Ejecute la aplicación desde la clase control y observe el funcionamiento del servicio reloj.

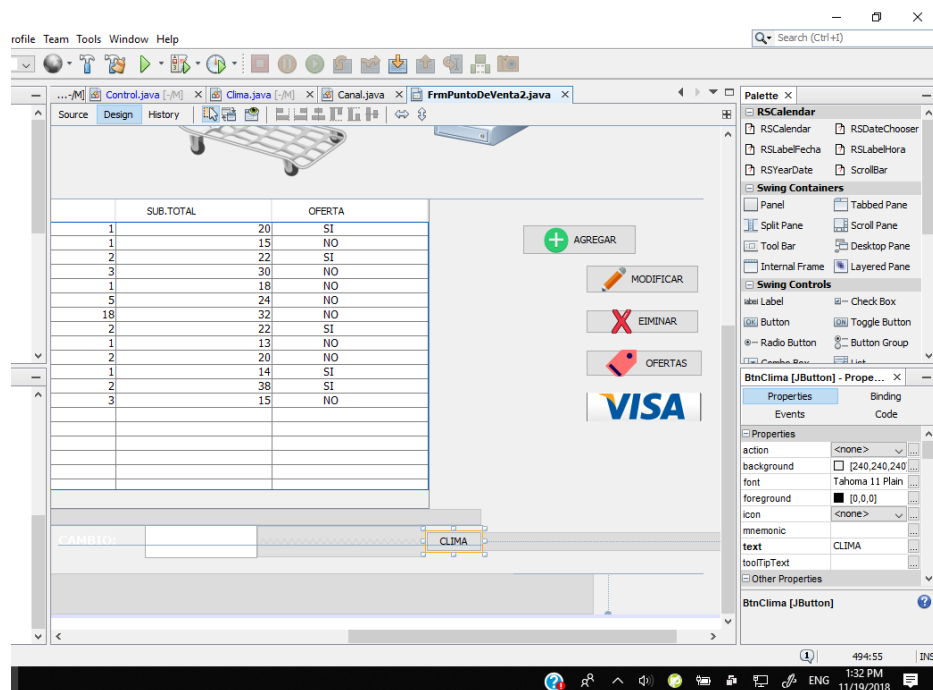


AGREGAR SERVICIO DE CLIMA.

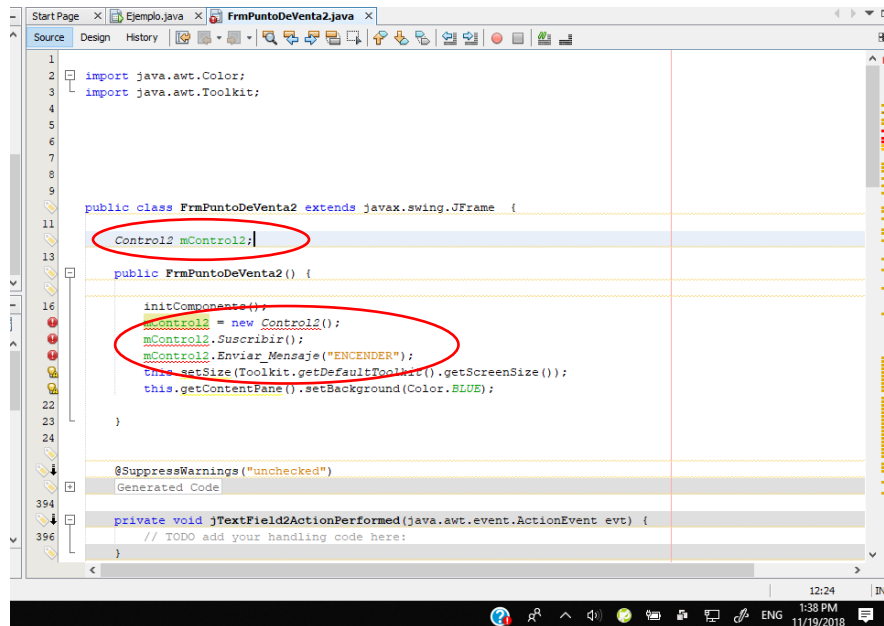
- Para incorporar el servicio de clima que permite consultar el clima primero Identifique el proyecto en el que desea incorporar el servicio e importe la librería "SEMA_SERVICIOS".

CODIGO	NOMBRE	PRECIO	CANTIDAD	SUB.TOTAL	OFERTA
0192837465	JAMON	20	1	20	SI
0129384756	LECHE	15	1	15	NO
0156473829	JABON	12	2	22	SI
0183924756	HARINA	10	3	30	NO
0183927465	REFresco	18	1	18	NO
0132874659	PAN	4	5	24	NO
0109283746	HUEVO	32	18	32	NO
0192837892	TORTILLAS	11	2	22	SI
1231243144	PAPAS	13	1	13	NO
9393819011	FRIJOL	10	2	20	NO
0193391282	POSTRE DE PIÑA	14	1	14	SI
9291837431	SALCHICHA	19	2	38	SI
0339393848	SOPA	5	3	15	NO

- Identifique el formulario donde desea incorporar el servicio de clima y agregue un botón al formulario, recuerde asignarle un nombre adecuado.



- Agregue el siguiente código dentro del código del formulario.



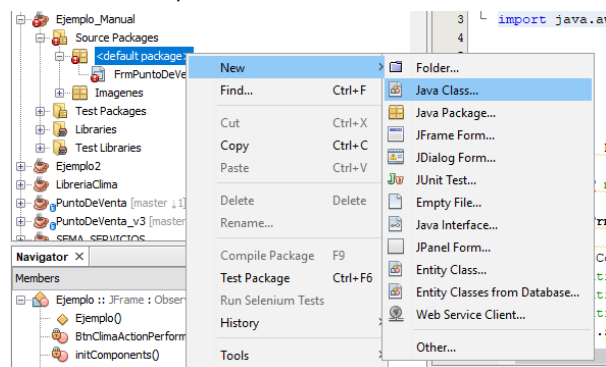
- Agregue el evento de clic al botón que agrego anteriormente, y agregue el siguiente código.

```

private void BtnClimaActionPerformed(java.awt.event.ActionEvent evt) {
    mControl2.Envia_Mensaje("clima");
}

```

- Agregamos una nueva clase, con nombre Control2.



- Agregue dentro de la clase Control2 el siguiente, no olvide agregar la librería "SEMA_SERVICIOS".

```

5
6      Canal mCanal;
7      Clima mClimaClima;
8
9  public Control2() {
10      mCanal = new Canal();
11      mClimaClima = new Clima();
12  }
13
14  public void Suscribirse() {
15      mCanal.addObserver(mClimaClima);
16  }
17
18  public void Enviar_Mensaje(String Mensaje) {
19      mCanal.SetStatus(Mensaje);
20  }

```

- Ejecute su aplicación, para ver el servicio del clima de un clic en el botón que agrego en el formulario (el método para ejecutar su aplicación puede estar dentro del formulario).

[illegible]

AGREGAR SERVICIO DE CLIMA Y RELOJ.

- Puede agregar el servicio de reloj y de clima a un proyecto desarrollado en java primero asegúrese de haber agregado la librería “SEMA_SERVICIOS” a su proyecto. Identifique el formulario de su proyecto donde desea agregar ambos servicios.

The top screenshot shows an IDE with the following code in the editor:

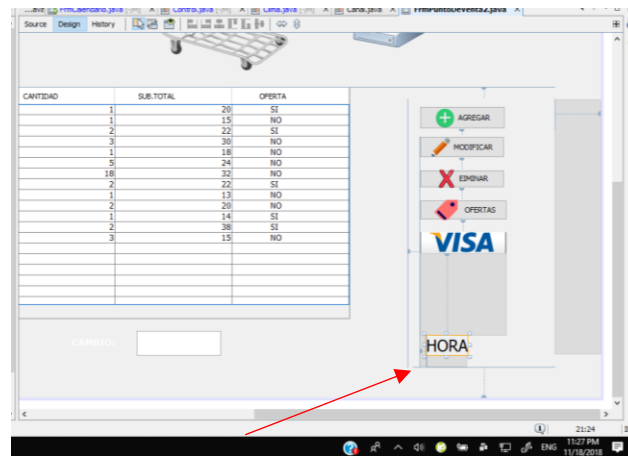
```
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JLayeredPane jLayeredPanel;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JTextField jTextField8;
private javax.swing.JTextField jTextField9;
// End of variables declaration
}

```

The bottom screenshot shows a Java Swing application window titled "SUPER OFERTON". It features a blue background with a shopping cart icon and a receipt printer icon. The interface includes input fields for "BOLETA", "CODIGO", "NOMBRE", "CANTIDAD", and "PRECIO", along with a "BUSC..." button. A table displays product information, and a sidebar contains buttons for "AGREGAR", "MODIFICAR", "ELIMINAR", "OFERTAS", and a "VISA" logo. The bottom status bar shows "DESCUENTO:", "TOTAL: \$ 0.00", and "CAMBIO:".

CODIGO	NOMBRE	PRECIO	CANTIDAD	SUB.TOTAL	OFERTA
0192837465	JAMON	20	1	20	SI
0129384756	LECHE	15	1	15	NO
0156473829	JABON	12	2	22	SI
0183924756	HARINA	10	3	30	NO
0183927465	REFresco	18	1	18	NO
0132874659	PAN	4	5	24	NO
0109283746	HUEVO	32	18	32	NO
0192837892	TORTILLAS	11	2	22	SI
1231243144	PAPAS	13	1	13	NO
9393819011	FRIDOL	10	2	20	NO
0193391282	POSTRE DE PIA	14	1	14	SI
9291837431	SALCHICHA	19	2	38	SI
0339393848	SOPA	5	3	15	NO

- Agregue un Label (etiqueta) a su formulario en la ubicación donde quiere observar la hora. Asegúrese de asignarle un nombre adecuado.



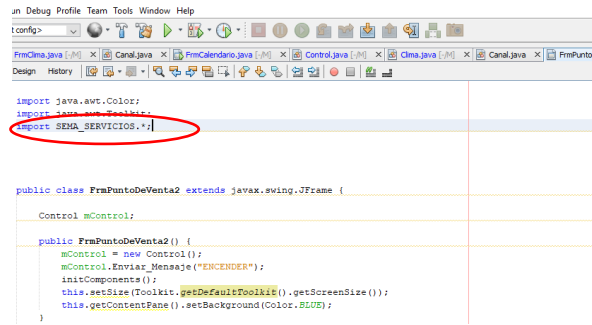
- Acceda al código de la aplicación y agregue el siguiente código.

```

IDE 8.2
Refactor Run Debug Profile Team Tools Window Help
<default config>
FrmClima.java [-M] x Canal.java x FrmCalendario.java [-M] x Control.java [-M] x Cima.java [-M] x Canal.java x FrmPuntoDeVenta2.java x Control.java
1
2 import java.awt.Color;
3 import java.awt.Toolkit;
4
5
6
7
8
9
10 public class FrmPuntoDeVenta2 extends javax.swing.JFrame {
11
12     Control mControl;
13
14     public FrmPuntoDeVenta2() {
15         mControl = new Control();
16         mControl.Envia_Mensaje("ENCENDER");
17         initComponents();
18         this.setSize(Toolkit.getDefaultToolkit().getScreenSize());
19         this.getContentPane().setBackground(Color.BLUE);
20
21     }
22
23     @SuppressWarnings("unchecked")
24     Generated Code
25
26     private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
27         // TODO add your handling code here:
28     }
29
30     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
31         jButton2.setOpaque(false);
32         //jButton2.setContentAreaFilled(false);
33     }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

- No olvide importar la librería que contiene los servicios “SEMA_SERVICIOS”.



```

an Debug Profile Team Tools Window Help
[config]
FrmClima.java [M] X Canal.java X FrmCalendario.java [M] X Control.java [M] X Cima.java [M] X Canal.java X FrmPuntoDe
Design History
import java.awt.Color;
import java.awt.Toolkit;
import SEMA_SERVICIOS.*;

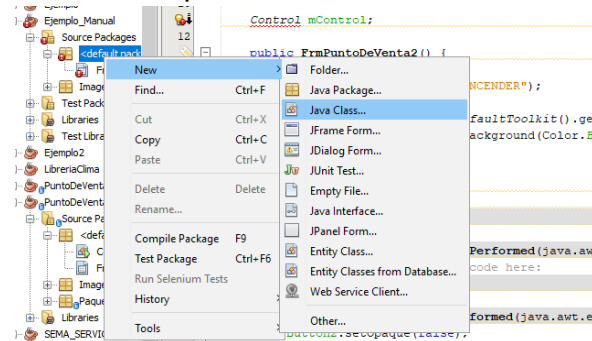
public class FrmPuntoDeVenta2 extends javax.swing.JFrame {

    Control mControl;

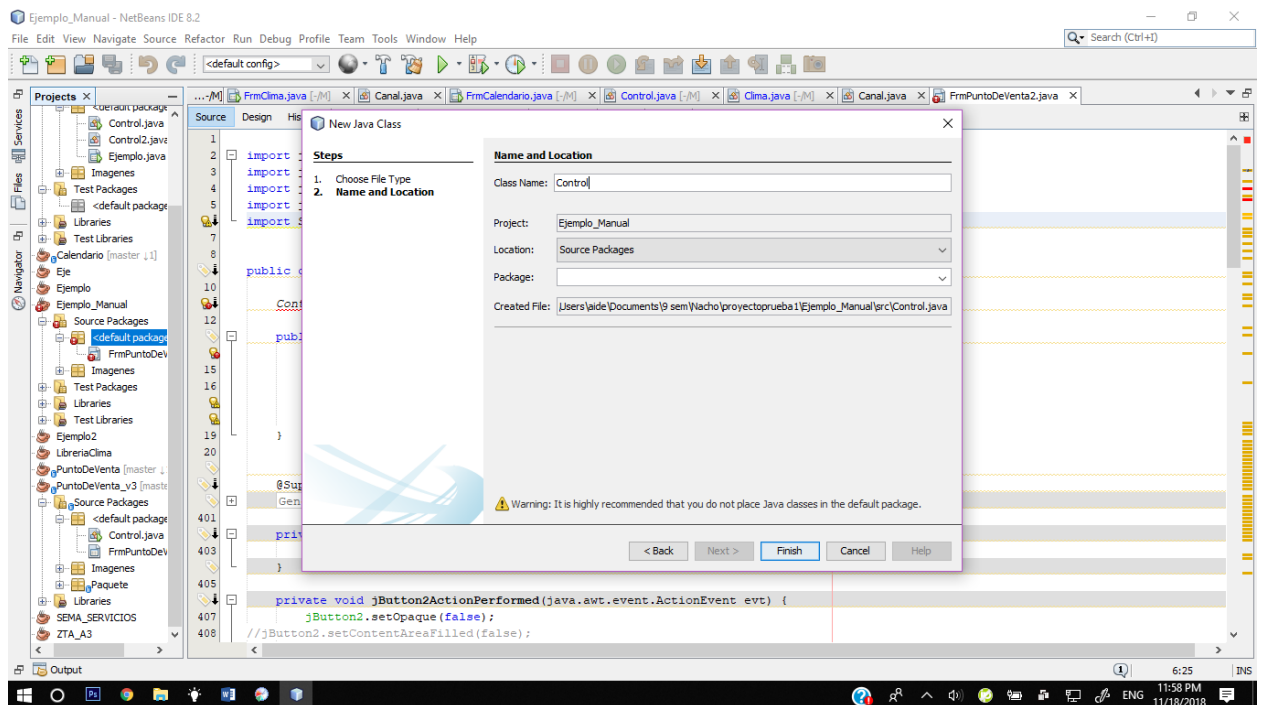
    public FrmPuntoDeVenta2() {
        mControl = new Control();
        mControl.Envia_Mensaje("ENCENDER");
        initComponents();
        this.setSize(Toolkit.getDefaultToolkit().getScreenSize());
        this.getContentPane().setBackground(Color.BLUE);
    }

```

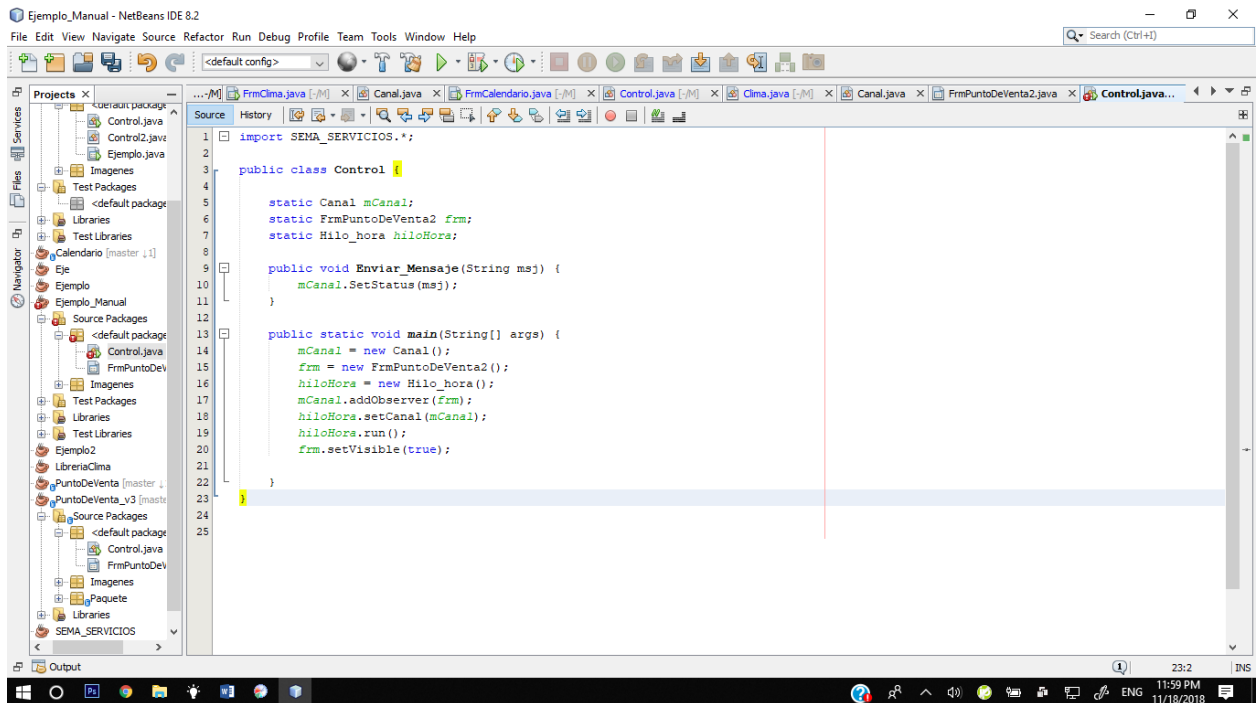
- Para eliminar los errores que marca en la clase control, agregue una clase.



- Agregue el nombre Control a la clase.

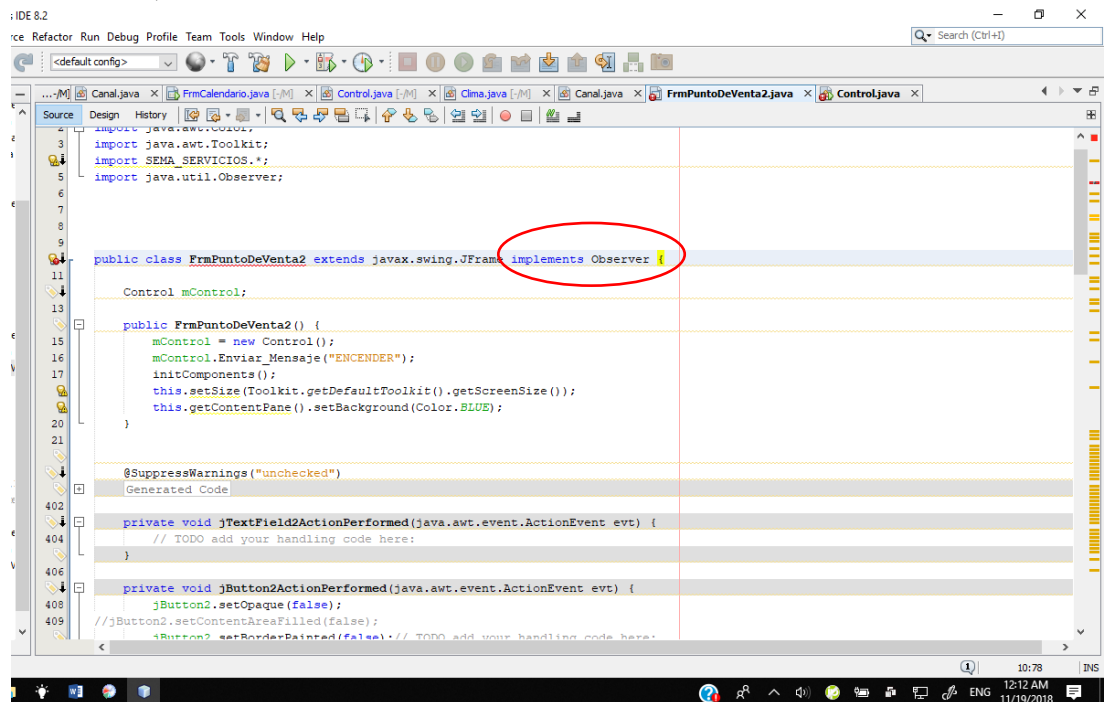


- Agregue el siguiente código dentro de la clase control.

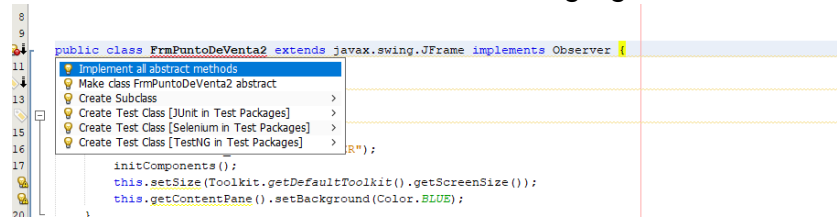


Nota: Asegúrese de haber importado la librería “SEMA_SERVICIOS”. Además, el método para ejecutar la aplicación e instanciar con el nombre adecuado de su formulario en este ejemplo el nombre es FrmPuntoDeventa2.

- Regrese al código del formulario y agregue el siguiente código “implements Observer”, además de la librería necesaria.



- Para eliminar el error marcado es necesario agregar un método abstracto.



- Al agregar el método abstracto se creará un método.

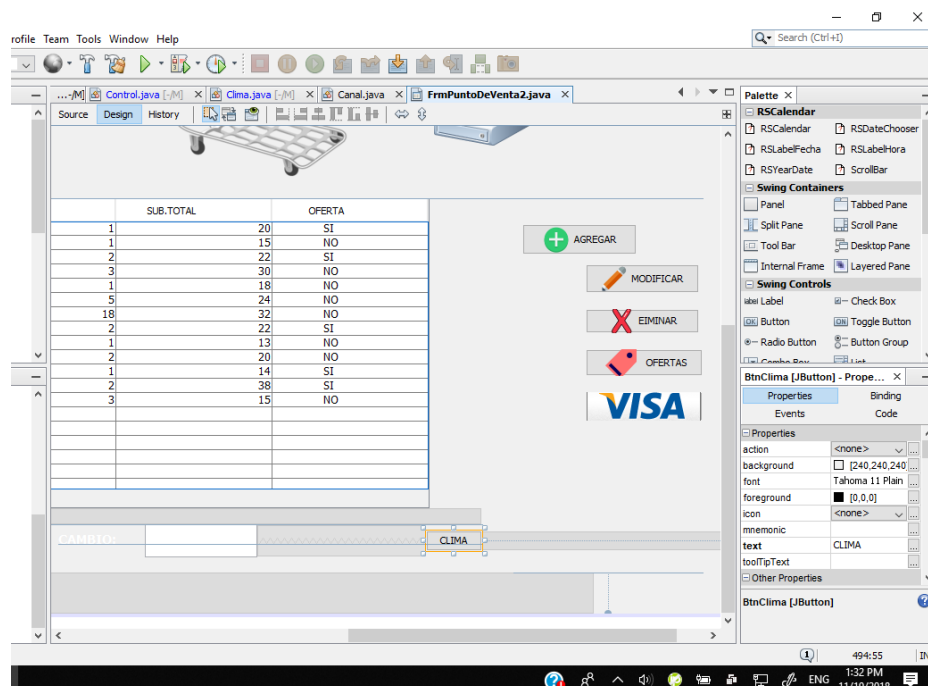


- Agregue el siguiente código dentro del método (Lreloj es el nombre del Label , agregue el nombre que tenga su Label o etiqueta).

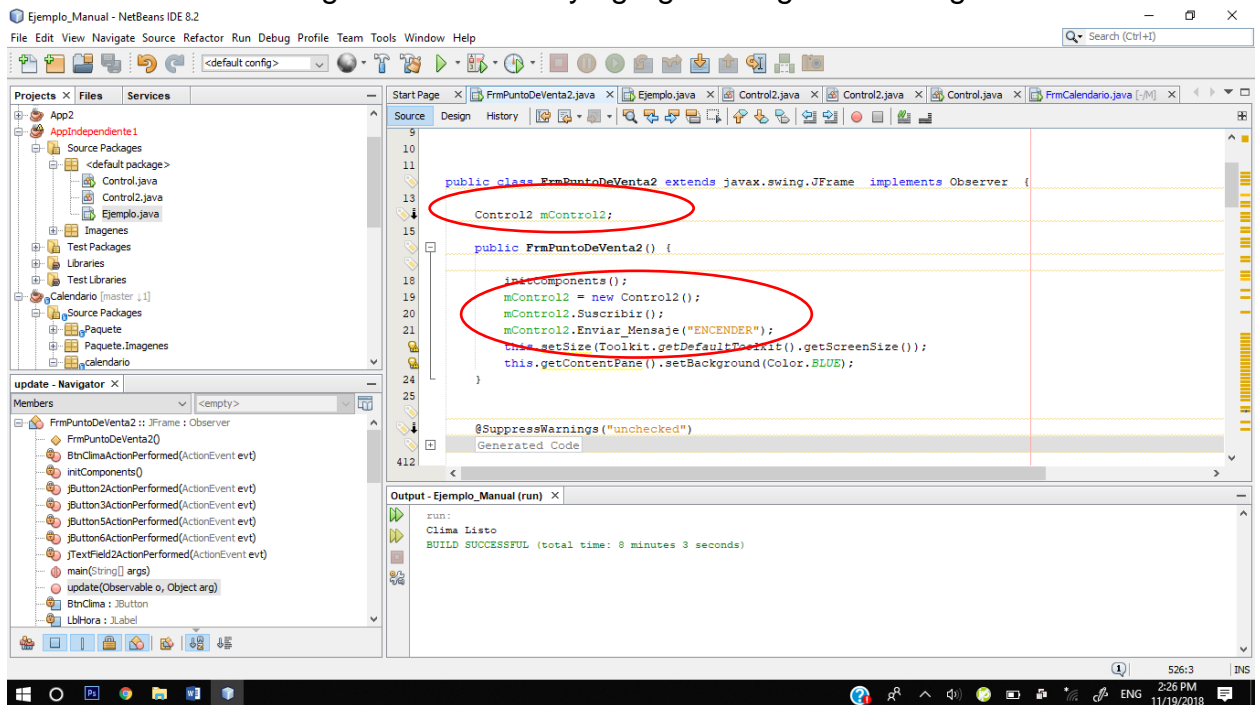
```
@Override
public void update(Observable o, Object arg) {
    Lreloj.setText(arg.toString());
}
```

El servicio de reloj ya está listo las siguientes indicaciones son para agregar el servicio de clima.

- Agregue un botón al formulario del proyecto donde quiere consultar el clima. Recuerde ponerle el nombre adecuado.



- Acceda al código del formulario y agregue el siguiente código.



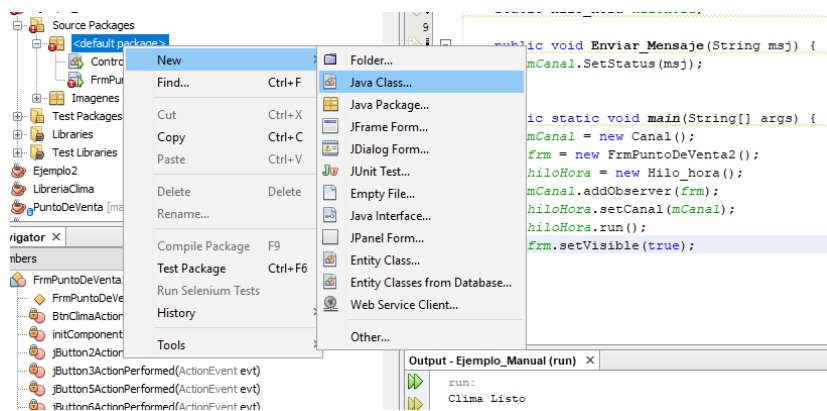
- Agregue el evento de clic al botón, y agregue el siguiente código.

```

private void BtnClimaActionPerformed(java.awt.event.ActionEvent evt) {
    mControl2.Envia_Mensaje("clima");
}

```

- Agregue una nueva clase, con nombre Control2.



- Agregue el siguiente código en la clase Control2, no olvide importar la librería “SEMA_SERVICIOS”.

```

1
2 import SEMA_SERVICIOS.*;
3
4 public class Control2 {
5
6     Canal mCanal;
7     Clima mClaseClima;
8
9     public Control2() {
10         mCanal = new Canal();
11         mClaseClima = new Clima();
12     }
13
14     public void Suscribirse() {
15         mCanal.addObserver(mClaseClima);
16     }
17
18     public void Enviar_Mensaje(String Mensaje) {
19         mCanal.SetStatus(Mensaje);
20     }
21 }

```

- Ejecute la aplicación desde la clase control. Podrá observar el horario y consultar el clima cada vez que de un clic en el botón.

