

Lenguaje de programación I



Caso

- ✓ La empresa **ciberfarma**, nos pide realizar la siguiente aplicación para el control de sus procesos




CIBERFARMAR - Acceso al Siste...




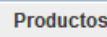
Usuario :

Contraseña:

Bienvenido : null null

 Sistema  Mantenimientos  Reportes

 Clientes

 Productos

Mantenimiento de Productos

Código:

Producto:

Tipo:

Stock: Precio:

Código	Nombre	Categ	Stock	Precio
--------	--------	-------	-------	--------

Reporte de Ventas

Del:

fecha	Nombre	Cantidad
2015/05/01	TiocTan	2
2015/05/02	Multibiotico	5
2015/05/02	Mejoralito	5

Contenido

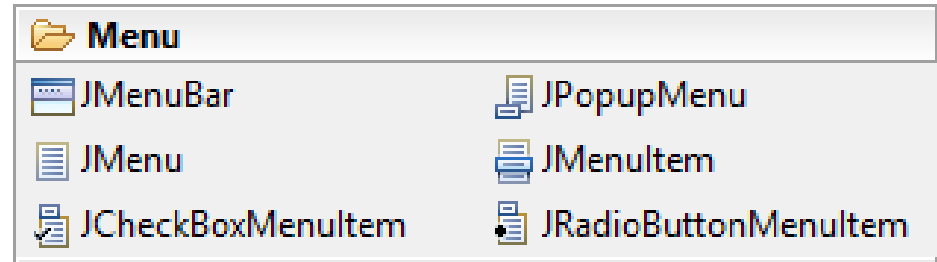
- Diseño de GUI
- JMenu
- JDesktopPane y JInternalFrame
- ✓ JCalendar
- ✓ JTable
- Ejercicios de Aplicación

Logros de la Unidad

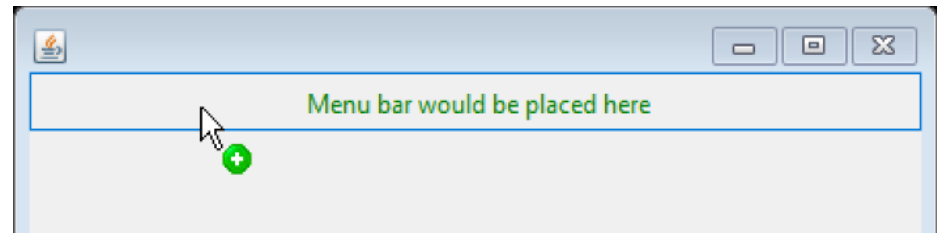
- Crear aplicaciones utilizando de manera individual y combinada las clases JMenu, JMenuBar, JMenuItem, JDesktopPane, JInternalFrame, Jtable, JCalendar.

Diseño del Menú

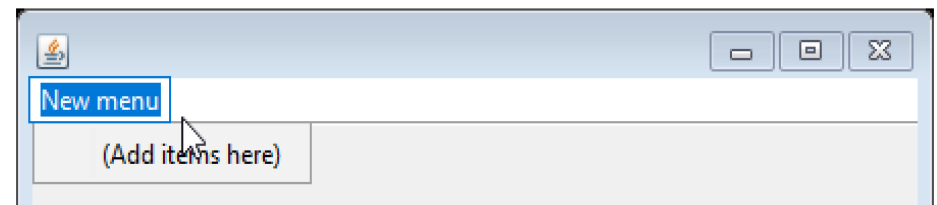
- ✓ Un menú nos permiten **organizar** nuestras opciones por categorías.



- Empezamos por el **JMenuBar**



- Agregamos las opciones de menú (**Jmenu** o **JMenuItem**) y cambiamos sus propiedades como **mnemonic**, **accelerator**, **icon**, etc.

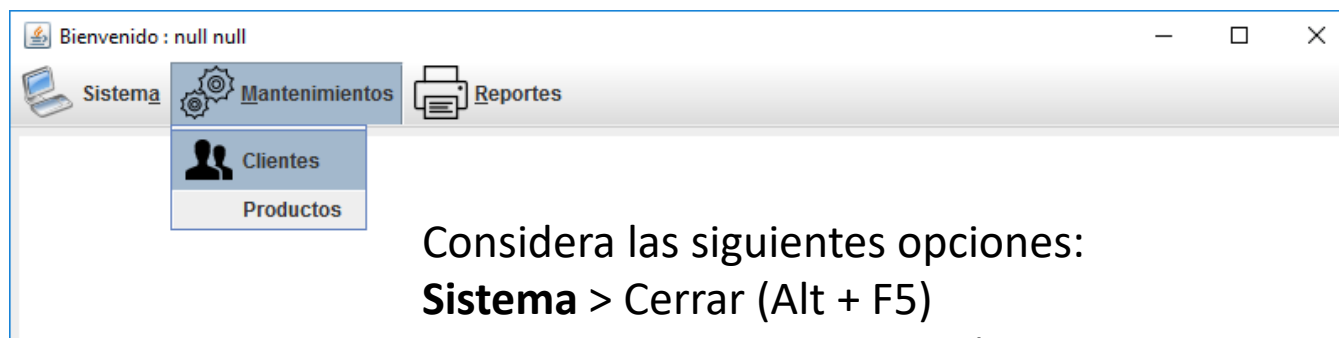


- Podemos agregar imágenes <https://www.iconfinder.com/>
<https://material.io/icons/>

Aplicación



- ✓ Crear el menú para nuestra aplicación




Considera las siguientes opciones:


Sistema > Cerrar (Alt + F5)

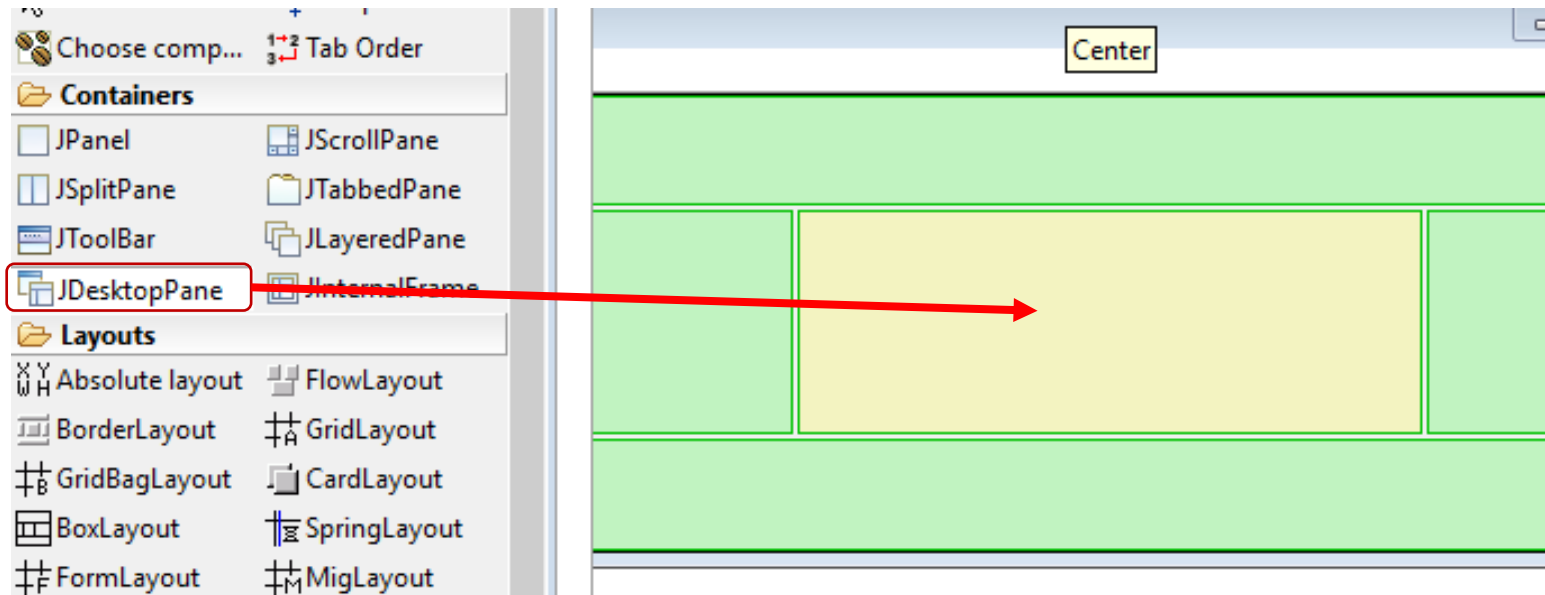
Mantenimientos > Clientes / Productos

Reportes > Clientes / Ventas (Ctrl + T)

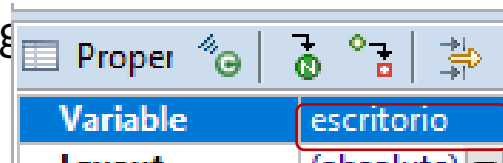
- La opción **Salir**  **Salir Alt-F5** debe cerrar la aplicación

JDesktopPane o escritorio

- ✓ El DesktopPane es un **contenedor**, que nos permite crear un “escritorio” para nuestros proyectos.
- ✓ Usaremos la distribución (Layout) tipo Borde  **BorderLayout** para mantener centrado el escritorio.



- ✓ Asignar **variable** y lo pondremos como **global**.



JInternalFrame

- ✓ Son Ventanas que se abren dentro de un contenedor.
- ✓ Los Internal Frame, necesitan configurar algunas propiedades como

closable	<input checked="" type="checkbox"/> true
closed	<input type="checkbox"/> false
defaultCloseOperation	DISPOSE_ON_CLOSE
enabled	<input checked="" type="checkbox"/> true
foreground	...
frameIcon	...
icon	<input type="checkbox"/> false
iconifiable	<input checked="" type="checkbox"/> true
maximizable	<input checked="" type="checkbox"/> true



- ✓ Para

```
// Instancia el JInternalFrame a visualizar
```

```
FrmRegPro v = new FrmRegPro();
```

```
// lo agrega al escritorio
```

```
escritorio.add(v);
```

```
// lo visualiza
```

```
v.setVisible(true);
```

 el entorno o escritorio. Ej:

- ✓ Para comprobar su funcionalidad, podemos cambiar entre las clases
JFrame a JInternalFrame

Aplicación



✓ Convierte los siguientes InternalFrame:

✓ **FrmRegPro.** La opción **Mantenimiento/Productos** debe abrir dicho Frame

✓ **FrmRptVta.** La opción: **Reportes/Ventas** debe abrir dicho Frame.

The image shows two overlapping application windows from a software interface.

The top window is titled "Mantenimiento de Productos". It contains several input fields: "Código:", "Producto:", "Tipo:" (with a dropdown menu showing "Selecciones tipo"), and "Stock:". There are also "Nuevo" (with a plus icon), "Guardar" (with a floppy disk icon), and "Editar" (with a wrench icon) buttons. Below the input fields is a table with columns "Código", "Nombre", "Categ", "Stock", and "Precio". A "Consultar" button is located below the table.

The bottom window is titled "Reporte de Ventas". It has a "Del:" label followed by a date input field and a calendar icon. A "Reporte" button is to the right. Below this is a table with columns "fecha", "Nombre", and "Cantidad".

fecha	Nombre	Cantidad
2015/05/01	TiocTan	2
2015/05/02	Multibiotico	5
2015/05/02	Mejoralito	5

Gestión de fecha y números



Utilización de Fechas y horas



- La clase `Date` (`java.util`)
- ✓ Crea un objeto **Long** que almacena la fecha y hora como un valor de milisegundos contados desde el 1 de enero de 1970 a las 00:00:00 GMT
- ✓ Actualmente es utilizada como una clase **intermedia** entre objetos tipo `Calendar` , `SimpleDateFormat` y `Locale`.

//Sintaxis

```
Date d = new Date();
```

```
Date fecha = new Date(cantidadLong);
```

//Métodos:

```
getTime(), setTime(objLong), toString()
```

Utilización de Fechas y horas



- La clase `Calendar` (`java.util`)
 - ✓ Clase **abstracta** para el manejo **optimizado** de fechas, horas, así como la utilización de zonas geográficas.

// Sintaxis

```
Calendar c = Calendar.getInstance();
```

//Métodos:

```
getTime(), setTime(oLong), toString()  
get(Calendar.Campo), set(Calendar.Campo, valor),  
add(Calendar.Campo, valor), roll(Calendar.campo, valor)
```

Utilización de Fechas y horas



- La clase `SimpleDateFormat` (`java.text`)
- ✓ Clase que permite formatear un objeto `Date`, según una cadena patrón:

<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>

// Sintaxis - `SimpleDateFormat`

`SimpleDateFormat` sdf = new `SimpleDateFormat`(String patrón);

//Métodos:

`format(objDate)`, `parse()`

```
String leerFecha() {  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");  
    return sdf.format(txtFecha.getDate());  
}
```

Método que devuelve una cadena con la fecha según el formato

Configuración regional



- La clase **Locale** (`java.útil`)
 - ✓ Especifica formatos de fechas, números y monedas sobre la base de idiomas

//Sintaxis

```
Locale loc = new Locale(language);
```

```
Locale loc = new Locale(language, country);
```

- Observa el siguiente código:

```
Locale actual = Locale.getDefault();
System.out.println(actual);
Locale[] listaLocales = Locale.getAvailableLocales();
for (Locale locale : listaLocales) {
    System.out.println(
        "Nombre del País : " + locale.getDisplayCountry() + "\n" +
        "Abreviatura : " + locale.getCountry() + "\n" +
        "El lenguaje actual es: " + locale.getLanguage() + "\n" );
}
```

Formatos de Fecha y hora

- La clase abstracta **DateFormat** permite crear objetos que **formatearán** fechas y horas en base a estilos y sensible a la mayoría de Localidades.
- El constructor será:

```
DateFormat dF = DateFormat.getDateInstance(Estilo, Locale)  
// para formatear fechas  
DateFormat tF = DateFormat.getTimeInstance(Estilo, Locale)  
// para formatear tiempo
```

- Para los estilos usamos las constantes: **DateFormat.DEFAULT**, **.SHORT**, **.MEDIUM**, **.LONG** y **.FULL**
- El método **format(objDate)**, devuelve un **String** según el objeto a formatear.

```
String fechaconFormato = dF.format(hoy);
```

Formatos de Número y Moneda

- Java proporciona la clase abstracta **NumberFormat** para **formatear números y monedas** sensible a la Localidad (aunque no soporta todas las posibles definiciones de Localidades).
- El constructor será:

```
NumberFormat nF = NumberFormat.getNumberInstance(Locale) ;  
// para formatear números  
NumberFormat cF = NumberFormat.getCurrencyInstance(Locale) ;  
// para formatear monedas
```

- El método **format(objnum)**, devuelve un **String** según el objeto a formatear.

```
String monedaconFormato = cF.format(monedas) ;
```

JCalendar

- ✓ JCalendar es un **selector** gráfico de fechas. Que a través de sus diversos componentes permite elegir diferentes elementos como años, meses, calendarios, así como exportar los datos seleccionados a objetos **Date** o **Calendar**.



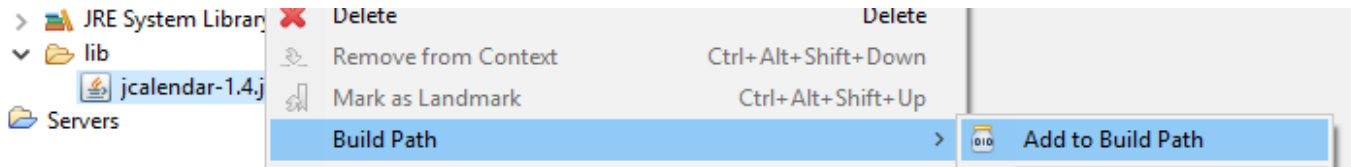
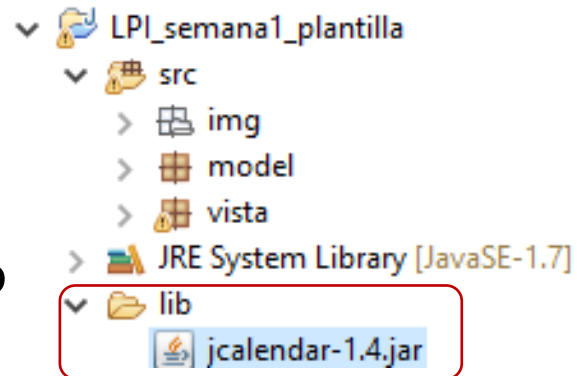
- ✓ El dato ingresado lo leemos como cadena formateada

```
void reporte() {  
    String fecha = leerFecha();  
}  
  
String leerFecha() {  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");  
    return sdf.format(txtFecha.getDate());  
}
```

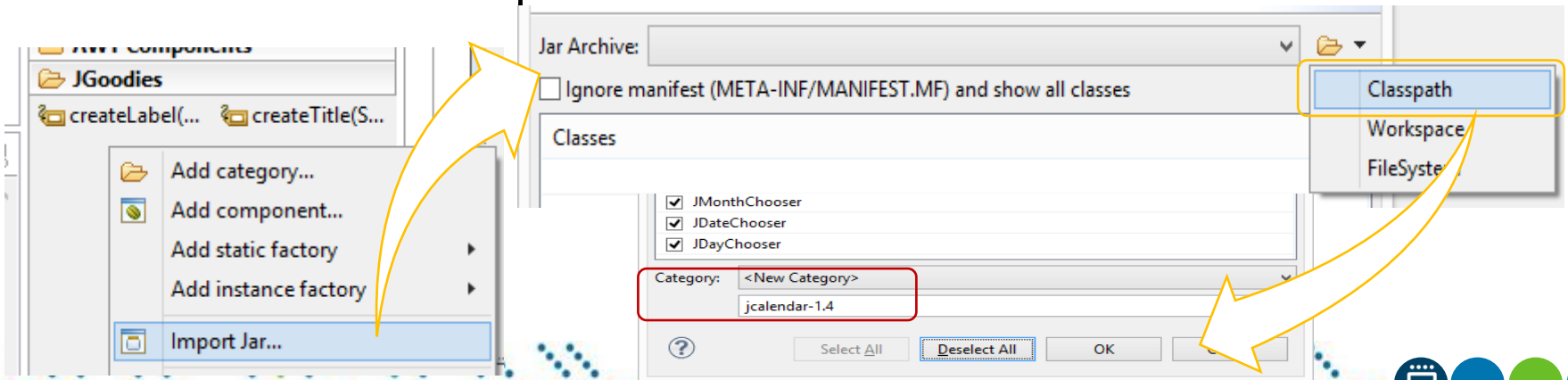

JCalendar

Instalando la librería JCalendar:

- ✓ Descargamos la librería (<https://toedter.com/jcalendar/>) y la copiamos a una carpeta del proyecto
- ✓ Agregamos la librería al Build Path



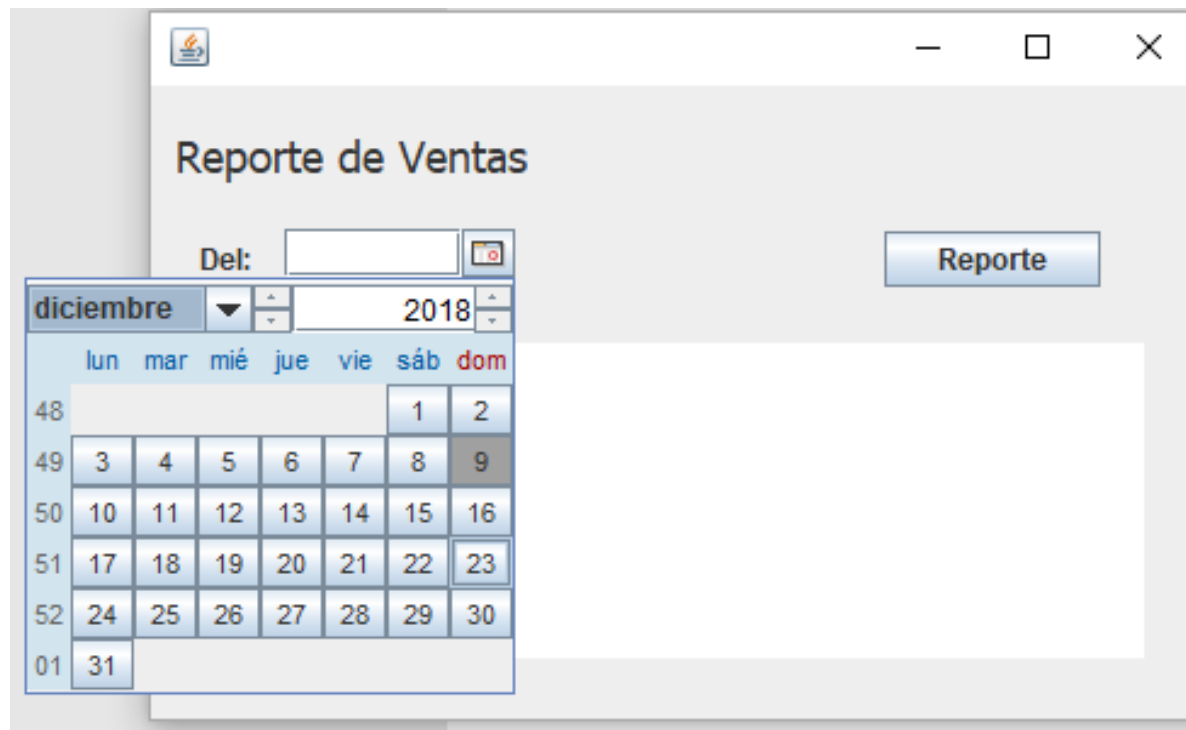
- ✓ Instalamos los componentes:



Aplicación



- ✓ Modifica el formulario **FrmRptVta** para usar jCalendar .
- ✓ Al pulsar el botón **Reporte** se debe mostrar la fecha seleccionada



Actividad Propuesta



- ✓ Diseña la siguiente GUI.
- ✓ Muestra en una etiqueta la fecha actual usando el formato indicado al cargar la aplicación.
- ✓ La fecha de reserva se obtiene mediante un JCalendar
- ✓ Obtener la cantidad de días de reserva
- ✓ Obtener y mostrar la fecha final
- ✓ Asumiendo que el costo diario de reserva es de S/. 50, mostrar el monto final con el formato correspondiente:

Modulo de Reserva de Local

Fecha actual: dd/mm/yyyy

Fecha Reserva: 15/04/2018

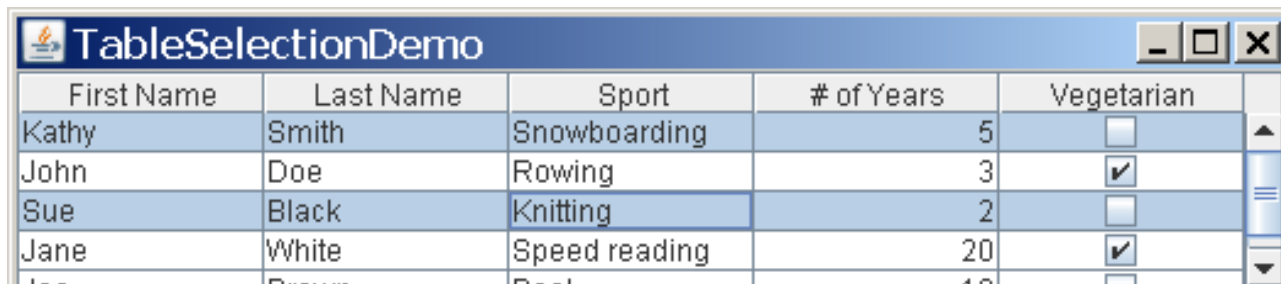
Días tolerancia: 5

Realiza Reserva

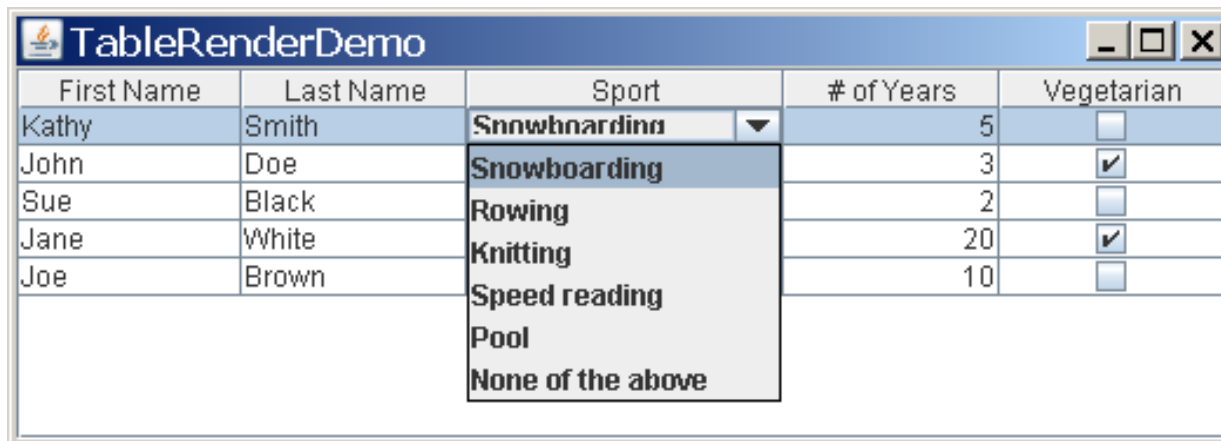
Su reserva termina el: **20/04/2018**
El total a pagar es : **S/250,00**

JTable

- ✓ Es un **componente** que permite mostrar una [tabla](#) con determinados elementos expresados en su Modelo.



First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Rowing	3	<input checked="" type="checkbox"/>
Sue	Black	Knitting	2	<input type="checkbox"/>
Jane	White	Speed reading	20	<input checked="" type="checkbox"/>
Joe	Brown	Pool	10	<input type="checkbox"/>

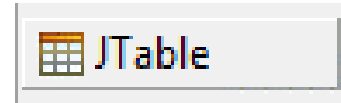


First Name	Last Name	Sport	# of Years	Vegetarian
Kathy	Smith	Snowboarding	5	<input type="checkbox"/>
John	Doe	Snowboarding	3	<input checked="" type="checkbox"/>
Sue	Black	Rowing	2	<input type="checkbox"/>
Jane	White	Knitting	20	<input checked="" type="checkbox"/>
Joe	Brown	Speed reading	10	<input type="checkbox"/>

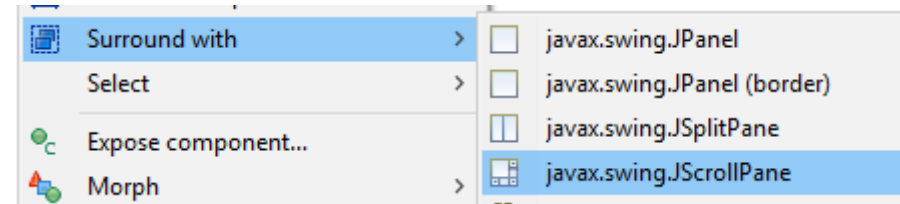
JTable

✓ Para ello:

- Dibujar el área con el componente, asignar su **nombre de variable** (Ej. `tblSalida`).



- Rodear con un JScrollPane (Barra de desplazamiento)



- Instanciar un **modelo** para definir las columnas y filas

```
// variable global
```

```
DefaultTableModel modelo = new DefaultTableModel();
```

- Asociar la tabla con el modelo

```
// constructor
```

```
tblSalida = new JTable();
```

```
tblSalida.setModel(modelo);
```

JTable

- ✓ Para las **columnas**, se agregan los **títulos** a utilizar

```
modelo.addColumn("Nombre de la columna");
```

```
tblSalida.setModel(modelo);  
modelo.addColumn("Fecha Venta");  
modelo.addColumn("Producto");  
modelo.addColumn("Cantidad");
```

Fecha Venta	Producto	Cantidad

- ✓ Para las **filas**, se trabaja en el método del reporte:
- ✓ Se crea un arreglo de objetos con la información a colocar :

```
Object datos[] = {dato1, dato2, etc};
```

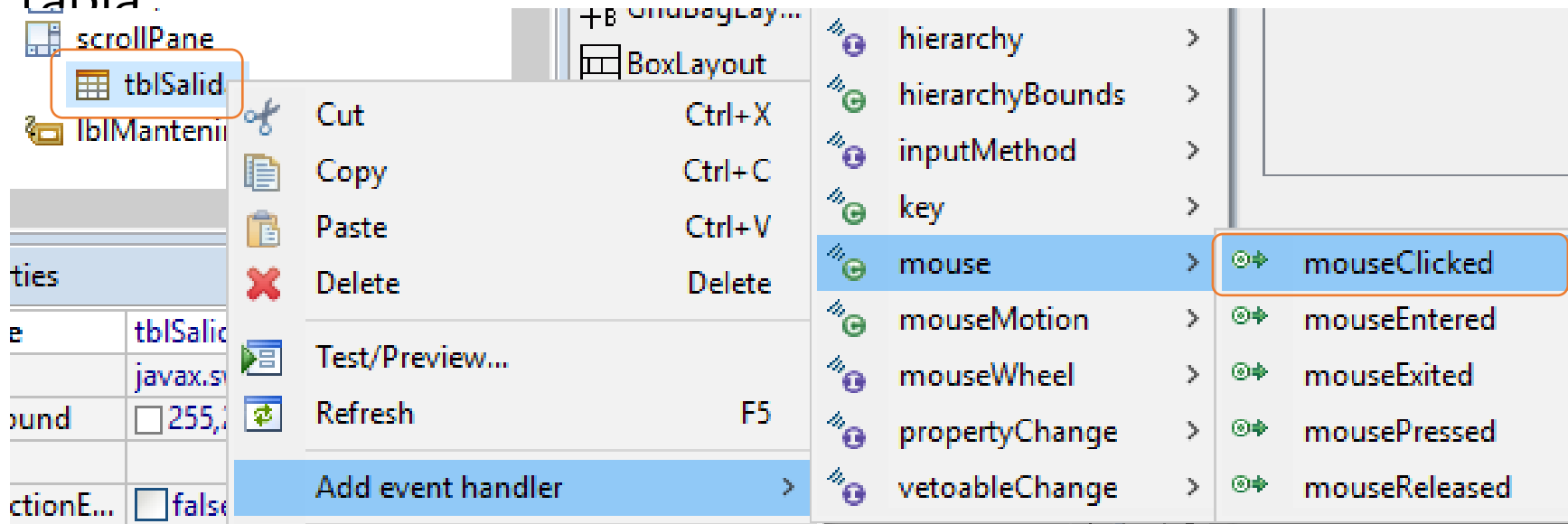
- ✓ Se **agrega** el arreglo de datos al modelo.

```
modelo.addRow(datos);
```

```
Object datos[] = {u.getCodigo(), u.getNombre(),  
                  u.getApellido()};  
modelo.addRow(datos);
```

JTable

- ✓ Podemos usar **eventos** en la tabla, para enviar los datos de la tabla.



- ✓ Se captura la fila seleccionada y se envían los datos

```
int fila = tblSalida.getSelectedRow();  
txtCodigo.setText(tblSalida.getValueAt(fila, 0).toString());
```

Referencias

- ✓ <http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>
- ✓ <https://toedter.com/jcalendar/>
- ✓ <https://docs.oracle.com/javase/tutorial/uiswing/components/table.html>
- ✓ <http://chuwiki.chuidiang.org/index.php?title=JTable>
- ✓ <http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Locale.html>

GRACIAS



SEDE MIRAFLORES

Calle Díez Canseco Cdra 2 / Pasaje Tello
Miraflores – Lima
Teléfono: 633-5555

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE SAN MIGUEL

Av. Federico Gallese 847
San Miguel – Lima
Teléfono: 632-4900

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benvides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535