

## Documentación

### Librería para manipular datos en tablas de la base de datos

#### Uso

Para usar la librería es necesario importarla a nuestro proyecto en java

```
import DavidL.MantJTableBD;
```

Luego instanciamos la clase MantenimientoNivel1

```
MantJTableBD mnt1;
```

En el constructor colocaremos

```
mnt1 = new MantJTableBD(TbDataCargo);
```

Donde TbDataCargo es el nombre de la tabla que tenemos en nuestro diseño.

Luego le colocamos las dos columnas para mostrarlo en la ejecución del formulario

```
ArrayList Titulos = new ArrayList();  
Titulos.add("Codigo Cargo"); Titulos.add("Nombre de Cargo");  
mnt1.setNombresColumnas(Titulos);
```

Podemos colocar cualquier nombre para las dos columnas del formulario

#### Requisitos:

Tener una tabla para los registros en el diseño con dos columnas

Tener las clases instanciadas de java.sql en el código del formulario:

Connection, ResultSet y Statement

### Componentes:

<i>Metodo</i>	<i>parametros</i>	<i>descripcion</i>
<i>setNombresColumnas()</i>	Titulos : List<String>	Asigna los nombres de las columnas a la tabla descrita en la instancia de la clase
<i>Nuevo()</i>	Textbox[] : Arreglo de JTextField TextoSelecion : JTextField	Limpia los JTextField indicados en el arreglo y coloca el foco al JTextField TextoSelecion
<i>Cancelar()</i>	Frame : JFrame	Realiza comprobación y cerrado del formulario indicado
<i>SeleccionarEnTabla()</i>	Nom : JTextField Cod : JTextField JTable : TablaDeDatos	Coloca los datos seleccionado de la tabla indicada a los JTextField de código y nombre indicados
<i>Guardar()</i>	Nom : JTextField ProcedimientoAlmacenado : String conexion : Connection statement : Statement	Guarda el nombre indicado con el procedimiento indicado, además se tiene que indicar la conexión de tipo Connection de la base de datos y el Statement para procedimientos almacenados.  El procedimiento tiene esta sintaxis: “{call procedimiento(?,?,?,...)}”
<i>Modificar()</i>	Nom : JTextField Cod : JTextField TablaDeDatos : JTable ProcedimientoAlmacenado : String conexion : Connection statement : Statement	Modifica un registro en la base de datos indicando el nombre a actualizar y el código (id), además se tiene que pasar la tabla de datos del formulario (colocando el nombre).  Al igual que el anterior, requiere del procedimiento almacenado, de la conexión de tipo Connection y el Statement para procedimientos almacenados.
<i>BorrarTabla()</i>	Ninguno	No accesible (solo para uso interno)

<i>ActualizarDatos()</i>	conexion : Connection RetornoRs : ResultSet numCampos : int	<p>Obtiene y muestra los datos de la base de datos, indicando la conexión de tipo Connection y un resultset encargado de hacer la consulta, es necesario contar con la tabla en el formulario.</p> <p>numCampos indica cuantos campos se quieren ver, según los campos de la base de datos o la consula.</p>
<i>Buscar()</i>	conexion : Connection rx : ResultSet numCampos : int	<p>Busca un dato en la tabla de la base de datos, indicando la conexión de tipo connection y el resultset encargado de realizar la búsqueda, es necesario contar con la tabla en el formulario.</p> <p>numCampos indica cuantos campos se quieren ver, según los campos de la base de datos o la consula.</p>

**Nota:** La librería no se encarga de realizar directamente las búsquedas y las actualizaciones de los datos, más bien ya se debería contar con los métodos con retorno ResultSet para las consultas.

#### **Además en los métodos:**

ActualizarDatos, guardar, modificar y buscar, es necesario llamarlos dentro de un try-catch, es un requisito indispensable para registrar excepciones, si no se coloca dentro del try, no funcionara.