
9 Anexos

Sentencias DDL

Archivo creacionModelo.sql

```
CREATE DATABASE `qcx277` ;
```

```
/*PRODUCTO*/
```

```
CREATE TABLE `qcx277`.`producto` (  
  `#referencia` VARCHAR( 15 ) NOT NULL ,
```

```
`#cod_cat` INT( 3 ) UNSIGNED NOT NULL ,
`#cod_ent` INT( 3 ) UNSIGNED NOT NULL ,
`nombre` TEXT NULL ,
`imagen_url` TEXT NULL ,
`descripcion` TEXT NULL ,
`eliminado` SMALLINT( 1 ) UNSIGNED NOT NULL DEFAULT '0',
PRIMARY KEY ( `#referencia` )
) ENGINE = InnoDB ;
```

```
CREATE TABLE `qcx277`.`categoria` (
`#cod_cat` INT( 3 ) UNSIGNED AUTO_INCREMENT NOT NULL ,
`nombre` TEXT NOT NULL ,
PRIMARY KEY ( `#cod_cat` )
) ENGINE = InnoDB ;
```

```
CREATE TABLE `qcx277`.`marca_edit` (
`#cod_ent` INT( 3 ) UNSIGNED AUTO_INCREMENT NOT NULL ,
`nombre` TEXT NULL ,
PRIMARY KEY ( `#cod_ent` )
) ENGINE = InnoDB ;
```

```
CREATE TABLE `qcx277`.`info_economica` (
`#linea` INT( 8 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`p_pvp` FLOAT( 8 ) NULL ,
`p_pnp` FLOAT( 8 ) NULL ,
`p_pack` FLOAT( 8 ) NULL ,
`cond_pack` INT( 8 ) NULL
) ENGINE = InnoDB ;
```

```
ALTER TABLE `qcx277`.`producto` ADD FOREIGN KEY `#cod_ent` ( `#cod_ent` ) REFERENCES
`qcx277`.`marca_edit` ( `#cod_ent` ) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `qcx277`.`producto` ADD FOREIGN KEY `#cod_cat` ( `#cod_cat` ) REFERENCES
`qcx277`.`categoria` ( `#cod_cat` ) ON DELETE NO ACTION ON UPDATE CASCADE;
```

```
CREATE TABLE `qcx277`.`hist_eco` (
`#referencia` VARCHAR( 15 ) NOT NULL ,
`#linea` INT( 8 ) UNSIGNED NOT NULL ,
`actual` TEXT NULL ,
`stmp_alta` TIMESTAMP NULL ,
`stmp_baja` TIMESTAMP NULL ,
FOREIGN KEY
`#referencia` ( `#referencia` )
REFERENCES `qcx277`.`producto` ( `#referencia` )
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY
`#linea` ( `#linea` )
REFERENCES `qcx277`.`info_economica` ( `#linea` )
ON DELETE NO ACTION
```

```

    ON UPDATE CASCADE,
PRIMARY KEY ( `#referencia` , `#linea` )
) ENGINE = InnoDB ;

CREATE TABLE `qcx277`.`estrella` (
`#pos` INT( 1 ) UNSIGNED NOT NULL ,
`#referencia` VARCHAR( 15 ) NOT NULL ,
`texto_extra` TEXT NULL ,
FOREIGN KEY
`#referencia`(`#referencia`)
REFERENCES `qcx277`.`producto`(`#referencia`)
ON DELETE CASCADE
ON UPDATE CASCADE,
PRIMARY KEY ( `#pos` )
) ENGINE = InnoDB ;

```

```

CREATE TABLE `qcx277`.`libro` (
`#isbn` VARCHAR( 15 ) NOT NULL ,
`#referencia` VARCHAR( 15 ) NOT NULL ,
`sinopsis` TEXT NULL ,
FOREIGN KEY
`#referencia`(`#referencia`)
REFERENCES `qcx277`.`producto`(`#referencia`)
ON DELETE CASCADE
ON UPDATE CASCADE,
PRIMARY KEY ( `#isbn` )
) ENGINE = InnoDB ;

```

```

/*CLIENTE*/

```

```

CREATE TABLE `qcx277`.`localidad` (
`#cod_post` INT( 8 ) UNSIGNED NOT NULL ,
`nombre` VARCHAR( 255 ) NULL DEFAULT NULL ,
PRIMARY KEY ( `#cod_post` )
) ENGINE = InnoDB;

```

```

CREATE TABLE `qcx277`.`empresa` (
`#cif` VARCHAR( 15 ) NOT NULL ,
`nombre` VARCHAR( 255 ) NULL DEFAULT NULL ,
`nombre_contacto` VARCHAR( 255 ) NULL DEFAULT NULL ,
PRIMARY KEY ( `#cif` )
) ENGINE = InnoDB;

```

```

CREATE TABLE `qcx277`.`info_contacto` (
`#movil` VARCHAR( 20 ) NOT NULL ,
`tlf` VARCHAR( 20 ) NULL ,
`fax` VARCHAR( 20 ) NULL ,

```

```
`email` VARCHAR( 255 ) NULL ,
`domicilio` VARCHAR( 255 ) NULL ,
`#cod_post` INT( 8 ) UNSIGNED NOT NULL ,
FOREIGN KEY
  `#cod_post`(`#cod_post`)
REFERENCES `qcx277`.`localidad`(`#cod_post`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
PRIMARY KEY ( `#movil` )
) ENGINE = InnoDB;

CREATE TABLE `qcx277`.`cliente` (
  `#cod_cliente` INT( 10 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `login` VARCHAR( 255 ) NOT NULL ,
  `password` VARCHAR( 10 ) NOT NULL ,
  `eliminado` INT( 1 ) UNSIGNED NOT NULL DEFAULT '0',
  `#movil` VARCHAR( 20 ) NOT NULL ,
FOREIGN KEY
  `#movil`(`#movil`)
REFERENCES `qcx277`.`info_contacto`(`#movil`)
ON DELETE CASCADE
ON UPDATE CASCADE,
PRIMARY KEY ( `#cod_cliente` )
) ENGINE = InnoDB;

CREATE TABLE `qcx277`.`particular` (
  `#dni` VARCHAR( 15 ) NOT NULL ,
  `#cod_cliente` INT( 10 ) UNSIGNED NOT NULL ,
  `nombre` VARCHAR( 255 ) NOT NULL ,
  `#cif` VARCHAR( 15 ) NOT NULL ,
FOREIGN KEY
  `#cod_cliente`(`#cod_cliente`)
REFERENCES `qcx277`.`cliente`(`#cod_cliente`)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY
  `#cif`(`#cif`)
REFERENCES `qcx277`.`empresa`(`#cif`)
ON DELETE NO ACTION
ON UPDATE CASCADE,
PRIMARY KEY ( `#dni` )
) ENGINE = InnoDB;

/*PEDIDO*/

CREATE TABLE `qcx277`.`pedido` (
  `#cod_pedido` INT( 8 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,
```

```

`stmp_1` DATETIME NULL ,
`stmp_2` DATETIME NULL ,
`total` FLOAT( 10 ) NOT NULL ,
`#cod_cliente` INT( 10 ) UNSIGNED NULL,
FOREIGN KEY
  `#cod_cliente`(`#cod_cliente`)
REFERENCES `qcx277`.`cliente`(`#cod_cliente`)
ON DELETE SET NULL
ON UPDATE CASCADE
) ENGINE = InnoDB;

```

```

CREATE TABLE `qcx277`.`comp_pedido` (
`cantidad` INT( 10 ) NOT NULL ,
`importe` FLOAT( 10 ) NOT NULL ,
`#cod_pedido` INT( 8 ) UNSIGNED NOT NULL ,
`#referencia` VARCHAR( 15 ) NOT NULL ,
FOREIGN KEY
  `#cod_pedido`(`#cod_pedido`)
REFERENCES `qcx277`.`pedido`(`#cod_pedido`)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY
  `#referencia`(`#referencia`)
REFERENCES `qcx277`.`producto`(`#referencia`)
ON DELETE CASCADE
ON UPDATE CASCADE,
PRIMARY KEY ( `#cod_pedido` , `#referencia` )
) ENGINE = InnoDB;

```

/*Valores de la tabla categoria*/

```

INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (1, "Bolsas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (2, "Accesorios de envío");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (3, "Embalaje");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (4, "Archivadores palanca");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (5, "Carpetas anillas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (6, "Accesorios archivo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (7, "Indices separadores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (8, "Archivo definitivo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (9, "Carpetas de proyectos");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (10, "Carpetas de gomas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (11, "Carpetería general");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (12, "Archivo suspendido");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (13, "Subcarpetas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (14, "Dossiers");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (15, "Fundas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (16, "Ficheros tarjeteros listines");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (17, "Carpetas de fundas");

```

```

INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (18, "Blocks");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (19, "Recambios");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (20, "Blocks notas adhesivas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (21, "Agendas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (22, "Talonarios");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (23, "Dibujo Técnico");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (24, "Complementos dibujo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (25, "Manualidades Regalo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (26, "Cintas adhesivas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (27, "Pegamentos");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (28, "Tijeras cutters");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (29, "Grapadoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (30, "Taladradoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (31, "Sellos fechadores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (32, "Elementos sujeción");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (33, "Accesorios");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (34, "Papeleras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (35, "Bandejas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (36, "Cajetines Archivo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (37, "Organizadores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (38, "Portalápices");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (39, "Correctores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (40, "Lápices");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (41, "Portaminas minas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (42, "Bolígrafos");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (43, "Rollers");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (44, "Rotuladores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (45, "Marcadores");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (46, "Plumas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (47, "Recambios Tintas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (48, "Calculadoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (49, "Plastificadoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (50, "Encuadernadoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (51, "Cizallas guillotinas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (52, "Destructoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (53, "Rotulación Electrónica");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (54, "Etiquetas");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (56, "Identificación");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (57, "Almacenamiento y Archivo datos");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (58, "Cartuchos Inkjet");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (59, "Cartuchos Toner Laser");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (61, "Cintas Impresión");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (62, "Cintas y rodillos para calculadoras");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (63, "Mobiliario de Archivo");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (64, "Carros de Carpetas Colgantes");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (65, "Libros");
INSERT INTO `qcx277`.`categoria` ( `#cod_cat` , `nombre` ) VALUES (66, "Papel");

```

/*Valores iniciales de la tabla marca_edit*/

```
INSERT INTO `qcx277`.`marca_edit` ( `#cod_ent` , `nombre` )  
VALUES ('0', '');
```

```
/*Valores iniciales de la tabla producto*/
```

```
INSERT INTO `qcx277`.`producto` ( `#referencia` , `#cod_cat` , `#cod_ent` ,  
`nombre` , `imagen_url` , `descripcion` , `eliminado` ) VALUES  
( '0', '1', '0', NULL , NULL , NULL , '0' );
```

```
/*Valores iniciales de la tabla estrella*/
```

```
INSERT INTO `qcx277`.`estrella` ( `#pos` , `#referencia` , `texto_extra` ) VALUES  
( '1', '0', NULL ),  
( '2', '0', NULL ),  
( '3', '0', NULL );
```

```
/*Valores iniciales de la tabla empresa*/
```

```
INSERT INTO `qcx277`.`empresa` ( `#cif` , `nombre` , `nombre_contacto` )  
VALUES ('0', 'No Empresa', 'No Empresa');
```

```
COMMIT;
```

Archivo creacionCarro.sql

```
CREATE TABLE `qcx277`.`carro` (  
  `referencia` VARCHAR( 15 ) NOT NULL ,  
  `cantidad` INT( 10 ) NOT NULL ,  
  `login` VARCHAR( 50 ) NOT NULL ,  
  `pwd` TEXT NOT NULL ,  
  `n_sesion` INT( 10 ) NOT NULL ,  
  `reg` TEXT NOT NULL ,  
  `tmp` TIMESTAMP NOT NULL ,  
  PRIMARY KEY ( `tmp` )  
) ENGINE = MYISAM;
```

JavaDoc

Knoe

Packages	
<u>knoe</u>	
<u>knoe.soporteBD</u>	
<u>knoe.soporteFTP</u>	
<u>knoe.soporteGrafico</u>	
<u>knoe.soporteGrafico.cliente</u>	
<u>knoe.soporteGrafico.pedido</u>	

knoe.soporteGrafico.producto	
knoe.soporteJava	

Package knoe

Class Summary	
Knoe	Clase principal

Class Knoe

```

java.lang.Object
└─ java.awt.Component
    └─ java.awt.Container
        └─ java.awt.Window
            └─ java.awt.Frame
                └─ javax.swing.JFrame
                    └─ knoe.Knoe

```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
 javax.accessibility.Accessible, javax.swing.RootPaneContainer,
 javax.swing.WindowConstants

```
public class Knoe extends javax.swing.JFrame
```

Field Summary	
java.io.File	imagen

Constructor Summary	
Knoe ()	Constructor de la Clase

Method Summary	
Static void	main (java.lang.String[] args)

Field Detail

imagen

```
public java.io.File imagen
```

Constructor Detail

Knoe

```
public Knoe()
```

Constructor de la Clase

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Package knoe.soporteBD

Class Summary

ConectarQcx277	
--------------------------------	--

Class ConectarQcx277

```
java.lang.Object
```

```
□ knoe.soporteBD.ConectarQcx277
```

```
public class ConectarQcx277 extends java.lang.Object
```

Crea una instancia de la clase. Servirá de apoyo a capas superiores. Es una clase compuesta básicamente por métodos.

Constructor Summary

ConectarQcx277 ()	
-----------------------------------	--

Method Summary

boolean	compruebaExistenciaLibroReferenciaProducto (java.lang.String referencia) Devuelve verdadero o falso si existe o no el libro definido por 'referencia'
int	compruebaExistenciaLineaInfo_economicaProducto (java.lang.String cad1, java.lang.String cad2, java.lang.String cad3, java.lang.String cad4)

	Devuelve la linea si existe o -1 en caso contrario.
boolean	compruebaExistenciaProductosSegunCategoria (int categoria) Comprueba si existe algún producto dado de alta que pertenezca a la categoría pasada como argumento(codigo de categoría).
boolean	conexion () Se usa para crear una conexion con el servidor MySQL en localhost, con la contraseña y base de datos dadas.
java.lang.String	construccionStatementCliente (boolean[] campos) Esta función va a construir el SELECT a ejecutar, en función de los checkBox elegidos en la ventana de dialogo.
java.lang.String	construccionStatementProducto (boolean[] campos, java.lang.String[] modificadores) Esta función va a construir la query a ejecutar, en función de los checkBox elegidos en la ventana de dialogo.
java.lang.String	construccionStatementSelectDetallePedido (java.lang.String codigo_pedido) Crea la statement a ejecutar (Select) sobre la tabla de pedidos.
java.lang.String	construccionStatementSelectForUpdateCliente (java.lang.String codigo_cliente) Crea la clausula SQL SELECT para crear la tabla
java.lang.String	construccionStatementSelectForUpdateProducto (boolean isLibro, java.lang.String referencia) Esta función va a construir la query a ejecutar, en función de los checkBox elegidos en la ventana de dialogo Y en funcion de si es libro o no.
java.lang.String	construccionStatementSelectPedido (boolean flag) Crea la statement a ejecutar (Select).
boolean	deleteFromCliente (java.lang.Object[][] tablaObjetos, int[] numeroFilasEliminar) Elimina un cliente de la tabla cliente si no está involucrado en ningún pedido.
boolean	deleteFromProducto (java.lang.Object[][] tablaObjetos, int[] numerosFilasEliminar) Esta función elimina directamente de la tabla de productos cualquier coincidencia en funcion de la referencia contenida y la elección del usuario, recogida en el array numerosFilasEliminar.
boolean	desconexion () Se utiliza para cerrar una conexion establecida previamente con base de datos.
void	deshacer () En caso de generarse una excepcion durante cualquier proceso de escritura en la base de datos, se intentará realizar una operacion de ROLLBACK

	de emergencia.
int	<u>generalInsertProducto</u> (java.lang.String[] cads, boolean isBook) Esta funcion se encarga de dar de alta productos en la base de datos.
java.lang.Object[][]	<u>generalSelect</u> (java.lang.String query, boolean[] campos) Esta función crea un array de objetos que son el resultado del select incluido en la cadena "query".
java.lang.Object[][]	<u>generalUpdate</u> (java.lang.String query, boolean[] campos) Esta función crea un array de objetos que son el resultado del select incluido en la cadena "query".
int	<u>generalUpdateCliente</u> (boolean cambioContacto, java.lang.String[] campos, java.lang.String movil) Realiza las modificaciones introducidas por el usuario en el panel de dialogo DialogoUpdateCliente.
int	<u>generalUpdateProducto</u> (boolean cambioPrecios, java.lang.String[] campos) Realiza las modificaciones introducidas por el usuario en el panel de dialogo DialogoUpdate.
boolean	<u>pruebaConexion</u> () Se utiliza inicialmente para testear el estado de las comunicaciones con base de datos.
java.lang.String[]	<u>selectAllWhereCodigoCliente</u> (java.lang.String statement) Se le pasa como argumento la cadena a ejecutar (select all from el gran UNION where cod_cliente)
java.lang.String[]	<u>selectAllWhereReferenciaProducto</u> (java.lang.String statement, boolean isLibro) Se le pasa como argumento la cadena a ejecutar (select all from el gran UNION where referencia) y si es libro o no, para diferenciar entre el número de campos a extraer(12 o 10 respectivamente).
java.lang.String[]	<u>selectCIFFFromEmpresaCliente</u> () Devuelve un array con todos los cif de las empresas.
int[]	<u>selectCodCatFromCategoriaProducto</u> () Devuelve un array con todos los codigos de las categorias dadas de alta.
int	<u>selectCodCatFromCategoriaProducto</u> (java.lang.String cad) Busca el codigo de categoria en funcion del nombre
int[]	<u>selectCodEntFromMarca_editProducto</u> () Devuelve un array con todos los codigos de marcas dadas de alta,

	ordenadas por código de marca.
int	<u>selectCodEntFromMarca_editProducto</u> (java.lang.String cad) Busca el código de marca en función del nombre
java.lang. .String[]	<u>selectCodigoPostalFromLocalidadCliente</u> () Devuelve un array con todos los códigos postales de las localidades existentes.
java.lang. .Object[][]	<u>selectCompPedido</u> (java.lang.String query) Se ejecuta un select incluido en la query (tabla comp_pedido) obtención de detalle de pedido.
java.lang. .String[]	<u>selectNombreFromCategoriaProducto</u> () Devuelve un array con todos los nombres de las categorías dadas de alta.
java.lang. .String[]	<u>selectNombreFromEmpresaCliente</u> () Devuelve un array con todos los nombres de las empresas.
java.lang. .String[]	<u>selectNombreFromLocalidadCliente</u> () Devuelve un array con todos los nombres de las localidades existentes.
java.lang. .String[]	<u>selectNombreFromMarca_editProducto</u> () Devuelve un array con todos los nombres de las marcas existentes.
java.lang. .Object[][]	<u>selectPedido</u> (java.lang.String query, boolean flag) Se ejecuta un select incluido en la query (tabla pedido).
java.lang. .String[]	<u>selectPersonaFromEmpresaCliente</u> () Devuelve un array con los nombres de las personas de contacto de las empresas.
boolean	<u>updateEmpresaCliente</u> (java.lang.String cod_cliente, boolean nuevaEmp, java.lang.String nombreEmp, java.lang.String cifEmp, java.lang.String personaEmp) Actualiza la empresa de cliente.
boolean	<u>updateEstrellaProducto</u> (java.lang.String pos, java.lang.String referencia, java.lang.String texto) Actualiza los campos de los productos estrella en la tabla correspondiente
boolean	<u>updateLocalidadCliente</u> (java.lang.String cod_cliente, boolean nuevaLoc, java.lang.String codPostal, java.lang.String nombreLoc) Actualiza la localidad de cliente.
boolean	<u>updateMarcaProducto</u> (java.lang.String ref, int codEnt) Actualiza el nuevo código de marca en la posición indicada por referencia.
boolean	<u>updatePedido</u> (java.lang.String cod_pedido)

	Sactualiza un pedido, pasándolo a pedidos realizados, añadiendo el valor de la fecha de preparación del pedido en cuestión.
--	---

Constructor Detail

ConectarQcx277

```
public ConectarQcx277()
```

Method Detail

conexion

```
private boolean conexion()
```

Se usa para crear una conexion con el servidor MySQL en localhost, con la contraseña y base de datos dadas. Inicializa la conexion y establece el modo de operación a "SET autocommit=0". Inicia la transaccion en el servidor MySQL con la ejecución de la sentencia "START TRANSACTION".

Returns:

True si se consigue establecer una conexion con la base de datos de MySQL.
False en otro caso.

pruebaConexion

```
public boolean pruebaConexion()
```

Se utiliza inicialmente para testear el estado de las comunicaciones con base de datos.

Returns:

True si exito. False en otro caso.

desconexion

```
private boolean desconexion()
```

Se utiliza para cerrar una conexion establecida previamente con base de datos.

Returns:

True si exito. False en otro caso.

deshacer

```
private void deshacer()  
    throws java.lang.Exception
```

En caso de generarse una excepcion durante cualquier proceso de escritura en la base de

datos, se intentará realizar una operacion de ROLLBACK de emergencia.

Throws:

`java.lang.Exception`

generalSelect

```
public java.lang.Object[][] generalSelect(java.lang.Stringquery,  
                                           boolean[] campos)
```

Esta función crea un array de objetos que son el resultado del select incluido en la cadena "query". Se pasará como parámetro de la tabla a crear.

Parameters:

query - - Sentencia a ejecutar en base de datos.

campos - - conjunto de booleanos que indican los campos a incluir en la sentencia.

Returns:

Object[][] - Resultado del select.

generalUpdate

```
public java.lang.Object[][] generalUpdate(java.lang.Stringquery,  
                                           boolean[] campos)
```

Esta función crea un array de objetos que son el resultado del select incluido en la cadena "query". Sobre dicho array se seleccionará la fila o filas que se quieran eliminar o modificar. Se pasará como parámetro de la tabla a crear.

Parameters:

query - - Sentencia SQL a ejecutar.

campos - - Booleanos que indican los campos sobre los que aplica la sentencia.

Returns:

Esta función crea un array de objetos que son el resultado del select incluido en la cadena "query". Sobre dicho array se seleccionará la fila o filas que se quieran eliminar o modificar. Se pasará como parámetro de la tabla a crear.

construccionStatementSelectDetallePedido

```
public java.lang.String  
construccionStatementSelectDetallePedido(java.lang.Stringcodigo_pedido)
```

Crea la statement a ejecutar (Select) sobre la tabla de pedidos.

Parameters:

codigo_pedido - - El número de pedido.

Returns:

Devuelve la cadena con la sentencia SELECT que se pretende ejecutar.

construccionStatementSelectPedido

```
public java.lang.String construccionStatementSelectPedido(booleanflag)
```

Crea la statement a ejecutar (Select). ORDENADOS POR FECHA.

Parameters:

flag - -Si flag=true => se genera el select para ver los pedidos pendientes. -Si flag=false => se genera el select para ver el historico de pedidos.

Returns:

Sentencia SELECT a ejecutar.

selectPedido

```
public java.lang.Object[][] selectPedido(java.lang.Stringquery,  
                                           booleanflag)
```

Se ejecuta un select incluido en la query (tabla pedido).

Parameters:

query - - Viene, normalmente, de construccionStatementSelectPedido()

flag - -Si flag=true => son cuatro parámetros a recuperar. -Si flag=false => son cinco parámetros a recuperar.

Returns:

Devuelve un array con los pedidos solicitados.

selectCompPedido

```
public java.lang.Object[][] selectCompPedido(java.lang.Stringquery)
```

Se ejecuta un select incluido en la query (tabla comp_pedido) obtención de detalle de pedido.

Parameters:

query - a ejecutar.

Returns:

Devuelve un array con el resultado del SELECT ejecutado.

updatePedido

```
public boolean updatePedido(java.lang.Stringcod_pedido)
```

Sactualiza un pedido, pasándolo a pedidos realizados, añadiendo el valor de la fecha de preparación del pedido en cuestión.

Parameters:

query - a ejecutar.

Returns:

Devuelve true si éxito. False en otro caso.

generalUpdateCliente

```
public int generalUpdateCliente(boolean cambioContacto,  
                                java.lang.String[] campos,  
                                java.lang.String movil)
```

Realiza las modificaciones introducidas por el usuario en el panel de dialogo DialogoUpdateCliente.

Parameters:

cambioContacto - - True si cierto, false si no hay cambios

campos - - Atributos a ser tenidos en cuenta

movil - - numero de movil que caracteriza al cliente

Returns:

0---->cambios realizados 2---->cambios en info_contacto preliminares realizados

-1--->error en el proceso

updateEmpresaCliente

```
public boolean updateEmpresaCliente(java.lang.String cod_cliente,  
                                      boolean nuevaEmp,  
                                      java.lang.String nombreEmp,  
                                      java.lang.String cifEmp,  
                                      java.lang.String personaEmp)
```

Actualiza la empresa de cliente.

Parameters:

cod_cliente - - codigo de cliente en cuestion.

nuevaEmp - - True si hay alta de nueva empresa.

nombreEmp - - Nombre de la empresa.

cifEmp - - Cif de la empresa.

personaEmp - - Nombre de la persona de contacto de la empresa.

Returns:

Devuelve true si éxito. False en otro caso.

selectPersonaFromEmpresaCliente

```
public java.lang.String[] selectPersonaFromEmpresaCliente()
```

Devuelve un array con los nombres de las personas de contacto de las empresas.

Returns:

Devuelve un array con los nombres.

selectCIFFromEmpresaCliente

```
public java.lang.String[] selectCIFFromEmpresaCliente()
```

Devuelve un array con todos los cif de las empresas.

Returns:

Devuelve un array con todos los cif de las empresas.

selectNombreFromEmpresaCliente

```
public java.lang.String[] selectNombreFromEmpresaCliente()
```

Devuelve un array con todos los nombres de las empresas.

Returns:

Devuelve un array con todos los nombres de las empresas.

updateLocalidadCliente

```
public boolean updateLocalidadCliente(java.lang.Stringcod_cliente,  
                                     booleannuevaLoc,  
                                     java.lang.StringcodPostal,  
                                     java.lang.StringnombreLoc)
```

Actualiza la localidad de cliente.

Parameters:

cod_cliente - - codigo de cliente en cuestion.
nuevaLoc - - True si hay alta de nueva localidad.
codPostal - - Codigo postal de la nueva localidad.
nombreLoc - - Nombre de la localidad.

Returns:

Devuleve true si exito. False en otro caso.

selectNombreFromLocalidadCliente

```
public java.lang.String[] selectNombreFromLocalidadCliente()
```

Devuelve un array con todos los nombres de las localidades existentes.

Returns:

Devuelve un array con todos los nombres de las localidades existentes.

selectCodigoPostalFromLocalidadCliente

```
public java.lang.String[] selectCodigoPostalFromLocalidadCliente()
```

Devuelve un array con todos los codigos postales de las localidades existentes.

Returns:

Devuelve un array con todos los codigos postales de las localidades existentes.

selectAllWhereCodigoCliente

```
public java.lang.String[] selectAllWhereCodigoCliente(java.lang.Stringstatement)
```

Se le pasa como argumento la cadena a ejecutar (select all from el gran UNION where cod_cliente)

Returns:

Devuelve un array con todos los codigos postales de las localidades existentes.

construccionStatementSelectForUpdateCliente

```
public java.lang.String  
construccionStatementSelectForUpdateCliente(java.lang.Stringcodigo_cliente)
```

Crea la clausula SQL SELECT para crear la tabla

Returns:

Devuelve una cadena con el SELECT a ejecutar. Se usará para la creación de la tabla de vista de clientes.

deleteFromCliente

```
public boolean deleteFromCliente(java.lang.Object[][]tablaObjetos,  
int[]numeroFilasEliminar)
```

Elimina un cliente de la tabla cliente si no está involucrado en ningún pedido. En este último caso, actualizará la variable de estado eliminado a 1.

Returns:

Devuelve true si éxito. False en otro caso.

construccionStatementCliente

```
public java.lang.String construccionStatementCliente(boolean[]campos)
```

Esta función va a construir el SELECT a ejecutar, en función de los checkBox elegidos en la ventana de dialogo.

Parameters:

– campos: Protocolo de intercambio de parámetros: campos[0]= CodigoCliente

campos[1]= NombreCliente campos[2]= DNI campos[3]= LOGIN campos[4]=
PASSWORD campos[5]= TelefonoMovil campos[6]= Email campos[7]=
Domicilio campos[8]= Localidad campos[9]=CodigoPostal campos[10]=
TeléfonoFijo campos[11]= Fax campos[12]= CIFEmpresa campos[13]=
NombreEmpresa campos[14]= PersonaContactoEmpresa

Returns:

Devuelve una cadena con el SELECT a ejecutar. Se usará para la creación de la
tabla de vista de clientes.

selectNombreFromMarca_editProducto

```
public java.lang.String[] selectNombreFromMarca_editProducto()
```

Devuelve un array con todos los nombres de las marcas existentes.

Returns:

Devuelve un array con todos los nombres de las marcas existentes.

selectNombreFromCategoriaProducto

```
public java.lang.String[] selectNombreFromCategoriaProducto()
```

Devuelve un array con todos los nombres de las categorias dadas de alta.

Returns:

Devuelve un array con todos los nombres de las categorias dadas de alta.

selectCodCatFromCategoriaProducto

```
public int selectCodCatFromCategoriaProducto(java.lang.Stringcad)
```

Busca el codigo de categoria en funcion del nombre

Parameters:

cad - - nombre de categoria

Returns:

Devuelve el codigo de categoria que corresponde al nombre de categoria dado. -1
si no existe la categoria.

selectCodEntFromMarca_editProducto

```
public int selectCodEntFromMarca_editProducto(java.lang.Stringcad)
```

Busca el codigo de marca en funcion del nombre

Parameters:

cad - - nombre de marca

Returns:

Devuelve el código de categoría que corresponde al nombre de marca dado. -1 si no existe la marca.

selectCodCatFromCategoriaProducto

```
public int[] selectCodCatFromCategoriaProducto()
```

Devuelve un array con todos los códigos de las categorías dadas de alta.

Returns:

Devuelve un array con todos los códigos de las categorías dadas de alta.

selectCodEntFromMarca_editProducto

```
public int[] selectCodEntFromMarca_editProducto()
```

Devuelve un array con todos los códigos de marcas dadas de alta, ordenadas por código de marca.

Returns:

Devuelve un array con todos los códigos de las marcas dadas de alta, ordenadas por código de marca.

compruebaExistenciaLineaInfo_economicaProducto

```
public int compruebaExistenciaLineaInfo_economicaProducto(java.lang.Stringcad1,  
                                                           java.lang.Stringcad2,  
                                                           java.lang.Stringcad3,  
                                                           java.lang.Stringcad4)
```

Devuelve la línea si existe o -1 en caso contrario.

Parameters:

cad1 - - pvp.
cad2 - - Precio del proveedor.
cad3 - - Precio de oferta.
cad4 - - Condiciones de oferta.

Returns:

Devuelve la línea si existe o -1 en caso contrario.

compruebaExistenciaLibroReferenciaProducto

```
public boolean  
compruebaExistenciaLibroReferenciaProducto(java.lang.Stringreferencia)
```

Devuelve verdadero o falso si existe o no el libro definido por 'referencia'

Parameters:

referencia - - referencia a ser buscada en la tabla de productos.

Returns:

Devuelve true si existe. False en otro caso.

selectAllWhereReferenciaProducto

```
public java.lang.String[]  
selectAllWhereReferenciaProducto(java.lang.Stringstatement,  
                                  booleanisLibro)
```

Se le pasa como argumento la cadena a ejecutar (select all from el gran UNION where referencia) y si es libro o no, para diferenciar entre el número de campos a extraer(12 o 10 respectivamente).

Parameters:

statement - - la cadena a ejecutar (select all from Union ... where referencia)
isLibro - - booleano para indicar si es o no libro, ya que los campos a extraer del resultSet seran 12 o 10.

Returns:

Devuelve un array de cadenas con las coincidencias.

compruebaExistenciaProductosSegunCategoria

```
public boolean compruebaExistenciaProductosSegunCategoria(intcategoria)
```

Comprueba si existe algún producto dado de alta que pertenezca a la categoría pasada como argumento(codigo de categoria).

Parameters:

categoria - - entero que caracteriza la categoria seleccionada

Returns:

Devuelve true si existen productos en la categoria dada. False en otro caso.

generalInsertProducto

```
public int generalInsertProducto(java.lang.String[]cads,  
                                  booleanisBook)
```

Esta funcion se encarga de dar de alta productos en la base de datos.

Parameters:

cads - - Campos a insertar en tablas: cads[0]=jTextField1.getText(); referencia
cads[1]=jTextField2.getText(); nombre cads[2]=jTextField3.getText();
descripcion cads[3] imagenUrl cads[4]=jComboBox3.getText(); categoria
cads[5]=jTextField6.getText(); pvp cads[6]=jTextField7.getText(); pnp
cads[7]=jTextField8.getText(); precio oferta cads[8]=jTextField9.getText();
condiciones oferta cads[9]=jComboBox4.getSelectedItem().toString(); marca_edit

```
cads[10]=jTextField4.getText(); isbn cads[11]=jTextField11.getText(); sinópsis  
isBook - - Booleano para especificar si el producto a insertar es libro o no.
```

Returns:

a continuación el protocolo de devolución del error/ack PROTOCOLO DE
ERRORES_ACKS Excepcion de ejecucion 0 Inserción libro y alta de editorialOK
1 Insercion libroOK 2 Insercion productoOK 3 Entrada duplicada para referencia
7 Entrada duplicada para isbn 8 No se establece conex. con bbdd 10 Insercion
producto y alta ok 11

deleteFromProducto

```
public boolean deleteFromProducto (java.lang.Object[][] tablaObjetos,  
int[] numerosFilasEliminar)
```

Esta función elimina directamente de la tabla de productos cualquier coincidencia en
funcion de la referencia contenida y la elección del usuario, recogida en el array
numerosFilasEliminar. Este array contiene los índices de filas de la tabla a borrar
seleccionados por el usuario. Debemos notar que para conservar la integridad de la base
de datos hemos de borrar solo los productos no implicados en pedidos con fechas
pasadas. Para ello, definimos dos tipos de eliminación: ----Eliminación real: borrado de
la base de datos (prods no implicados). ----Ocultación: se actualiza el valor de atributo
"eliminado" a "1" ya que "0" implica "no eliminado."

Parameters:

tablaObjetos - - Elementos de la tabla de seleccion.
numerosFilasEliminar - - Filas seleccionadas de la tabla para la
eliminacion.

Returns:

True si exito. False en otro caso.

construccionStatementProducto

```
public java.lang.String construccionStatementProducto (boolean[] campos,  
java.lang.String[] modifica  
dores)
```

Esta función va a construir la query a ejecutar, en función de los checkBox elegidos en
la ventana de dialogo.

Parameters:

campos - - Campos de seleccion a incluir en la sentencia a construir. Protocolo
de intercambio de parámetros: campos[0]=jCheckBox3.isSelected() referencia
campos[1]=jCheckBox4.isSelected() nombre campos[2]=jCheckBox5.isSelected()
descripcion campos[3]=jCheckBox6.isSelected() url_imagen
campos[4]=jCheckBox8.isSelected() pvp campos[5]=jCheckBox9.isSelected()
precio_pack campos[6]=jCheckBox10.isSelected() cond_oferta
campos[7]=jCheckBox7.isSelected() precio_pnp
campos[8]=jCheckBox13.isSelected() categoria

```
campos[9]=jCheckBox11.isSelected() marca  
campos[10]=jCheckBox12.isSelected() isbn  
campos[11]=jCheckBox14.isSelected() sinopsis  
modificadores - - Criterios extra a tener en cuenta en la construccion de la  
sentencia: modificadores[0]= Categoria de producto. modificadores[1]= Criterio  
ordenacion.
```

construccionStatementSelectForUpdateProducto

```
public java.lang.String  
construccionStatementSelectForUpdateProducto(booleanisLibro,  
trstringreferencia)
```

Esta función va a construir la query a ejecutar, en función de los checkBox elegidos en la ventana de dialogo Y en funcion de si es libro o no.

Parameters:

isLibro - - Booleano para indicar si es libro o no.
referencia - - Referencia de producto.

Returns:

Devuelve el SELECT a ejecutar de los checkBox elegidos en la ventana de dialogo Y en funcion de si es libro o no.

updateMarcaProducto

```
public boolean updateMarcaProducto(java.lang.Stringref,  
intcodEnt)
```

Actualiza el nuevo código de marca en la posicion indicada por referencia.

Parameters:

ref - - Referencia del producto
codEnt - - Codigo de marca

Returns:

True si exito. False en caso contrario.

generalUpdateProducto

```
public int generalUpdateProducto(booleancambioPrecios,  
java.lang.String[]campos)
```

Realiza las modificaciones introducidas por el usuario en el panel de dialogo DialogoUpdate.

Parameters:

cambioPrecios - - Booleano que indica si es necesario actualizar los precios del producto.

campos - - Protocolo de intercambio de campos: campos[0]//nombre de producto
campos[1]//descripcion de producto campos[2]//pvp campos[3]//pnp
campos[4]//cond_oferta campos[5]//p_oferta campos[6]//ISBN
campos[7]//sinopsis del libro campos[8]//url de imagen campos[9]//referencia

Returns:

Devuelve los siguientes códigos de error: resultado=0---->no hay cambios.
resultado=1---->cambios en producto realizados. resultado=4---->modificacines
sobre libro realizadas. resultado=-1--->error en el proceso.

updateEstrellaProducto

```
public boolean updateEstrellaProducto (java.lang.Stringpos,  
                                       java.lang.Stringreferencia,  
                                       java.lang.Stringtexto)
```

Actualiza los campos de los productos estrella en la tabla correspondiente

Parameters:

pos - - Esta cadena contiene la posicion en la tabla estrella que debe ocupar la
referencia de producto dada.
referencia - - Referencia del producto.
texto - - Texto extra a mostrar en la página web.

Returns:

True si exito. False en caso contrario.

Package knoe.soporteFTP

Class Summary

ClienteFTP	
----------------------------	--

Class *ClienteFTP*

```
java.lang.Object  
└─knoe.soporteFTP.ClienteFTP
```

```
public class ClienteFTP extends java.lang.Object
```

Constructor Summary

ClienteFTP ()	
-------------------------------	--

Method Summary

v	subirImagen (java.io.File imagen)
oid	Sube una imagen mediante protocolo FTP al servidor instalado en localhost.

Constructor Detail

ClienteFTP

```
public ClienteFTP()
```

Method Detail

subirImagen

```
public void subirImagen(java.io.File imagen)  
    throws java.lang.Exception
```

Sube una imagen mediante protocolo FTP al servidor instalado en localhost.

Parameters:

imagen - - Instancia de clase File que contiene la imagen a subir.

Throws:

java.lang.Exception

Package knoe.soporteGrafico

Class Summary

DialogoACK	
DialogoTabla	

Class DialogoACK

```
java.lang.Object  
└─ java.awt.Component  
    └─ java.awt.Container  
        └─ java.awt.Window  
            └─ java.awt.Frame  
                └─ javax.swing.JFrame  
                    └─ knoe.soporteGrafico.DialogoACK
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoACK extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Constructor Summary

[**DialogoACK**](#)()

Creates new form DialogoACK

[**DialogoACK**](#)(int ack)

Este constructor lanza los distintos mensajes de error en función del entero que se le pasa como parámetro.

Method Summary

static void	main (java.lang.String[] args)
----------------	---

Constructor Detail

DialogoACK

```
public DialogoACK()
```

Creates new form DialogoACK

DialogoACK

```
public DialogoACK(intack)
```

Este constructor lanza los distintos mensajes de error en función del entero que se le pasa como parámetro.

Parameters:

ack - - Codigo de ack.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Class *DialogoTabla*

```
java.lang.Object
└─ java.awt.Component
    └─ java.awt.Container
        └─ java.awt.Window
            └─ java.awt.Frame
                └─ javax.swing.JFrame
                    └─ knoe.soporteGrafico.DialogoTabla
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoTabla extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

int	categoria
javax.swing. .JTable	jTable1

Constructor Summary

[**DialogoTabla**](#) ()

[**DialogoTabla**](#) (java.lang.String[] nombColumnas, java.lang.Object[]
[] datos)
Constructor avanzado.

[**DialogoTabla**](#) (java.lang.String[] nombColumnas, java.lang.Object[]
[] datos, boolean flag)
Constructor avanzado.

[**DialogoTabla**](#) (java.lang.String[] nombColumnas, java.lang.Object[]
[] datos, boolean flag, int cat)
Constructor avanzado.

[**DialogoTabla**](#) (java.lang.String[] nombColumnas, java.lang.Object[]
[] datos, int cat)
Constructor avanzado.

DialogoTabla (java.lang.String[] nombColumnas, java.lang.Object[] [] datos, int cat, int flag) Constructor avanzado.
--

Method Summary

static void	main (java.lang.String[] args)
void	setDialogoModel (javax.swing.table.DefaultTableModel modeloTabla) Establece el modelo de tabla al modelo por defecto.
void	setDialogoModel (ModeloTabla modeloTabla) Establece el modelo de tabla al modelo ModeloTabla, clase implementada en el paquete knoe.soporteJava.

Field Detail

jTable1

```
public javax.swing.JTable jTable1
```

categoria

```
public int categoria
```

Constructor Detail

DialogoTabla

```
public DialogoTabla()
```

DialogoTabla

```
public DialogoTabla (java.lang.String[] nombColumnas,  
                     java.lang.Object[] [] datos)
```

Constructor avanzado.

Parameters:

nombColumnas - - Array de String que contiene la cabecera de la tabla.

datos - - Tabla de Objects con los datos a mostrar en la tabla.

DialogoTabla

```
public DialogoTabla(java.lang.String[]nombColumnas,  
                    java.lang.Object[][]datos,  
                    booleanflag)
```

Constructor avanzado. Este es usado para actualizar o eliminar clientes en función de "flag"

Parameters:

nombColumnas - - Array de String que contiene la cabecera de la tabla.
datos - - Tabla de Objects con los datos a mostrar en la tabla.
flag - - Si true, modificacion de cliente. Si false, eliminacion de cliente.

DialogoTabla

```
public DialogoTabla(java.lang.String[]nombColumnas,  
                    java.lang.Object[][]datos,  
                    booleanflag,  
                    intcat)
```

Constructor avanzado. Constructor de eliminacion de productos en función de "flag"

Parameters:

nombColumnas - - Array de String que contiene la cabecera de la tabla.
datos - - Tabla de Objects con los datos a mostrar en la tabla.
flag - - Se usa para diferenciar de otros constructores. Sin uso.
cat - - Categoria del producto.

DialogoTabla

```
public DialogoTabla(java.lang.String[]nombColumnas,  
                    java.lang.Object[][]datos,  
                    intcat)
```

Constructor avanzado. Constructor de modificacion de productos.

Parameters:

nombColumnas - - Array de String que contiene la cabecera de la tabla.
datos - - Tabla de Objects con los datos a mostrar en la tabla.
cat - - Categoria del producto.

DialogoTabla

```
public DialogoTabla(java.lang.String[]nombColumnas,  
                    java.lang.Object[][]datos,  
                    intcat,  
                    intflag)
```

Constructor avanzado. Constructor para la edicion de productos estrella.

Parameters:

nombColumnas - - Array de String que contiene la cabecera de la tabla.

datos - - Tabla de Objects con los datos a mostrar en la tabla.

cat - - Categoria del producto.

flag - - Se usa para diferenciar de otros constructores. Sin uso.

Method Detail

setDialogoModel

```
public void setDialogoModel(javax.swing.table.DefaultTableModelmodeloTabla)
```

Establece el modelo de tabla al modelo por defecto.

setDialogoModel

```
public void setDialogoModel(ModeloTablamodeloTabla)
```

Establece el modelo de tabla al modelo ModeloTabla, clase implementada en el paquete knoe.soporteJava.

Parameters:

modeloTabla - - Modelo de tabla al que se desea migrar.

See Also:

[ModeloTabla](#)

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Package knoe.soporteGrafico.cliente

Class Summary

DialogoEmpresa	
DialogoLocalidad	
DialogoUpdateCliente	

Class DialogoEmpresa

java.lang.Object

□ java.awt.Component

□ java.awt.Container

```

    □ java.awt.Window
      □ java.awt.Frame
        □ javax.swing.JFrame
          □ knoe.soporteGrafico.cliente.DialogoEmpresa
  
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
 javax.accessibility.Accessible, javax.swing.RootPaneContainer,
 javax.swing.WindowConstants

```
public class DialogoEmpresa extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

java.lang .String[]	arrayPersonas
java.lang .String	cod_cli

Constructor Summary

[**DialogoEmpresa**](#)()

Creates new form DialogoEmpresa

[**DialogoEmpresa**](#)(java.lang.String cod_cliente)

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Summary

static void	main (java.lang.String[] args)
----------------	---

Field Detail

cod_cli

```
public java.lang.String cod_cli
```

arrayPersonas

```
public java.lang.String[] arrayPersonas
```

Constructor Detail

DialogoEmpresa

```
public DialogoEmpresa()
```

Creates new form DialogoEmpresa

DialogoEmpresa

```
public DialogoEmpresa(java.lang.Stringcod_cliente)
```

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Class DialogoLocalidad

```
java.lang.Object
└─ java.awt.Component
    └─ java.awt.Container
        └─ java.awt.Window
            └─ java.awt.Frame
                └─ javax.swing.JFrame
                    └─ knoe.soporteGrafico.cliente.DialogoLocalidad
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoLocalidad extends javax.swing.JFrame
```

See Also:

Field Summary

java.lang.String	<code>cod_cli</code>
java.lang.String	<code>loc</code>

Constructor Summary

[`DialogoLocalidad\(\)`](#)

Creates new form DialogoLocalidad

[`DialogoLocalidad\(java.lang.String cod\_cliente\)`](#)

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Summary

static void	<code>main</code> (java.lang.String[] args)
-------------	---

Field Detail

`loc`

public java.lang.String `loc`

`cod_cli`

public java.lang.String `cod_cli`

Constructor Detail

`DialogoLocalidad`

public `DialogoLocalidad()`

Creates new form DialogoLocalidad

DialogoLocalidad

```
public DialogoLocalidad(java.lang.Stringcod_cliente)
```

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Class DialogoUpdateCliente

```
java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Window
│   │   │   ├── java.awt.Frame
│   │   │   │   ├── javax.swing.JFrame
│   │   │   │   └── knoe.soporteGrafico.cliente.DialogoUpdateCliente
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoUpdateClientextends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

java.lang.String	<u>cif</u>
java.lang.String	<u>cod_cli</u>

java.lang.String	<u>cod_postal</u>
java.lang.String	<u>dni</u>
java.lang.String	<u>domicilio</u>
java.lang.String	<u>email</u>
java.lang.String	<u>fax</u>
java.lang.String	<u>fijo</u>
java.lang.String	<u>localidad</u>
java.lang.String	<u>login</u>
java.lang.String	<u>movil</u>
java.lang.String	<u>nombCli</u>
java.lang.String	<u>nombreEmpresa</u>
java.lang.String	<u>password</u>
java.lang.String	<u>personaContacto</u>

Constructor Summary

[DialogoUpdateCliente](#) ()

Creates new form DialogoUpdate

[DialogoUpdateCliente](#) (java.lang.String cod_cliente)

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Summary

static c void	<u>main</u> (java.lang.String[] args)
---------------------	---

Field Detail

cod_cli

public java.lang.String **cod_cli**

nombCli

public java.lang.String **nombCli**

dni

public java.lang.String **dni**

login

public java.lang.String **login**

password

public java.lang.String **password**

movil

public java.lang.String **movil**

fijo

public java.lang.String **fijo**

fax

public java.lang.String **fax**

email

```
public java.lang.String email
```

domicilio

```
public java.lang.String domicilio
```

cod_postal

```
public java.lang.String cod_postal
```

localidad

```
public java.lang.String localidad
```

cif

```
public java.lang.String cif
```

nombreEmpresa

```
public java.lang.String nombreEmpresa
```

personaContacto

```
public java.lang.String personaContacto
```

Constructor Detail

DialogoUpdateCliente

```
public DialogoUpdateCliente()
```

Creates new form DialogoUpdate

DialogoUpdateCliente

```
public DialogoUpdateCliente(java.lang.Stringcod_cliente)
```

Este constructor inicializa la variable código de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Package knoe.soporteGrafico.pedido

Class Summary

DialogoPedido	
-------------------------------	--

Class DialogoPedido

```
java.lang.Object
└─ java.awt.Component
    └─ java.awt.Container
        └─ java.awt.Window
            └─ java.awt.Frame
                └─ javax.swing.JFrame
                    └─ knoe.soporteGrafico.pedido.DialogoPedido
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoPedido extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

javax.swing.JTable	jTable1
--------------------	-------------------------

Constructor Summary

[DialogoPedido\(\)](#)

Creates new form DialogoPedido

[DialogoPedido](#)(boolean tipo, java.lang.Object[][] datos)

Constructor para la vision del conjunto de pedidos pendientes o para la vision de pedidos realizados.

[DialogoPedido](#)(java.lang.String cod_pedido, boolean flag)

Este constructor se usa para mostrar el desglose en productos de un pedido.

Method Summary

static void	<u>main</u> (java.lang.String[] args)
----------------	---

Field Detail

jTable1

public javax.swing.JTable **jTable1**

Constructor Detail

DialogoPedido

public **DialogoPedido**()

Creates new form DialogoPedido

DialogoPedido

public **DialogoPedido**(boolean tipo,
java.lang.Object[][] datos)

Constructor para la vision del conjunto de pedidos pendientes o para la vision de pedidos realizados.

Parameters:

tipo - - si true=> se inicia una tabla que lee donde el timestamp de ejecucion no existe (pedido pendiente). si false=> se inicia una tabla que lee donde el timestamp de ejecucion existe (histórico de pedidos).

DialogoPedido

public **DialogoPedido**(java.lang.String cod_pedido,
boolean flag)

Este constructor se usa para mostrar el desglose en productos de un pedido.

Parameters:

cod_pedido - - codigo de pedido.

flag - - Boolean que se usa para distinguir entre tablas de pedidos realizados.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Package knoe.soporteGrafico.producto

Class Summary

DialogoEstrella	
DialogoMarca	
DialogoUpdate	

Class DialogoEstrella

java.lang.Object

□ java.awt.Component

□ java.awt.Container

□ java.awt.Window

□ java.awt.Frame

□ javax.swing.JFrame

□ knoe.soporteGrafico.producto.DialogoEstrella

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,

javax.accessibility.Accessible, javax.swing.RootPaneContainer,

javax.swing.WindowConstants

```
public class DialogoEstrella extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

java.lang. .String	referencia
-----------------------	----------------------------

Constructor Summary

[`DialogoEstrella\(\)`](#)

Creates new form DialogoEstrella

[`DialogoEstrella\(java.lang.String ref\)`](#)

Este constructor inicializa la variable referencia, con lo que la instancia devuelta está unívocamente ligada a un solo producto.

Method Summary

static void	<code>main</code> (java.lang.String[] args)
----------------	---

Field Detail

referencia

public java.lang.String **referencia**

Constructor Detail

DialogoEstrella

public **DialogoEstrella**()

Creates new form DialogoEstrella

DialogoEstrella

public **DialogoEstrella**(java.lang.Stringref)

Este constructor inicializa la variable referencia, con lo que la instancia devuelta está unívocamente ligada a un solo producto.

Method Detail

main

public static void **main**(java.lang.String[]args)

Parameters:

args - the command line arguments

Class DialogoMarca

```
java.lang.Object
└─ java.awt.Component
    └─ java.awt.Container
        └─ java.awt.Window
            └─ java.awt.Frame
                └─ javax.swing.JFrame
                    └─ knoe.soporteGrafico.producto.DialogoMarca
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoMarca extends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

java.lang .String	ref
----------------------	---------------------

Constructor Summary

[DialogoMarca](#) ()

Creates new form DialogoCategoria

[DialogoMarca](#) (java.lang.String current, java.lang.String referencia)

Este constructor inicializa la variable referencia, con lo que la instancia devuelta está unívocamente ligada a un solo producto.

Method Summary

static void	main (java.lang.String[] args)
----------------	--

Field Detail

ref

```
public java.lang.String ref
```

Constructor Detail

DialogoMarca

```
public DialogoMarca()
```

Creates new form DialogoCategoria

DialogoMarca

```
public DialogoMarca(java.lang.Stringcurrent,  
                    java.lang.Stringreferencia)
```

Este constructor inicializa la variable referencia, con lo que la instancia devuelta está unívocamente ligada a un solo producto.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Class DialogoUpdate

```
java.lang.Object  
└─ java.awt.Component  
    └─ java.awt.Container  
        └─ java.awt.Window  
            └─ java.awt.Frame  
                └─ javax.swing.JFrame  
                    └─ knoe.soporteGrafico.producto.DialogoUpdate
```

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.RootPaneContainer,
javax.swing.WindowConstants

```
public class DialogoUpdateextends javax.swing.JFrame
```

See Also:

[Serialized Form](#)

Field Summary

<code>java.lang.String</code>	<u>cond_pack</u>
<code>java.lang.String</code>	<u>pnp</u>
<code>java.lang.String</code>	<u>ppack</u>
<code>java.lang.String</code>	<u>pvp</u>
<code>java.lang.String</code>	<u>ref</u>

Constructor Summary

[DialogoUpdate](#)()

Creates new form DialogoUpdate

[DialogoUpdate](#)(`java.lang.String` referencia)

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está unívocamente ligada a dicho cliente.

Method Summary

<code>static void</code>	<u>main</u> (<code>java.lang.String[]</code> args)
--------------------------	---

Field Detail

ref

`public java.lang.String ref`

pvp

`public java.lang.String pvp`

pnp

`public java.lang.String pnp`

ppack

```
public java.lang.String ppack
```

cond_pack

```
public java.lang.String cond_pack
```

Constructor Detail

DialogoUpdate

```
public DialogoUpdate()
```

Creates new form DialogoUpdate

DialogoUpdate

```
public DialogoUpdate(java.lang.Stringreferencia)
```

Este constructor inicializa la variable codigo de cliente, con lo que la instancia devuelta está univocamente ligada a dicho cliente.

Method Detail

main

```
public static void main(java.lang.String[]args)
```

Parameters:

args - the command line arguments

Package knoe.soporteJava

Class Summary

FiltroFile	
ModeloTabla	
SoporteArray	

Class FiltroFile

```
java.lang.Object
```

```
  □ javax.swing.filechooser.FileFilter
```

```
    □ knoe.soporteJava.FiltroFile
```

```
public class FiltroFile extends javax.swing.filechooser.FileFilter
```

Constructor Summary

[FiltroFile](#)()

Creates a file filter.

[FiltroFile](#)(java.lang.String extension)

Creates a file filter that accepts files with the given extension.

[FiltroFile](#)(java.lang.String[] filters)

Creates a file filter from the given string array.

[FiltroFile](#)(java.lang.String[] filters, java.lang.String description)

Creates a file filter from the given string array and description.

[FiltroFile](#)(java.lang.String extension, java.lang.String description)

Creates a file filter that accepts the given file type.

Method Summary

boolean **[accept](#)**(java.io.File f)

Return true if this file should be shown in the directory pane, false if it shouldn't.

void **[addExtension](#)**(java.lang.String extension)

Adds a filetype "dot" extension to filter against.

java.lang.
.String **[getDescription](#)**()

Returns the human readable description of this filter.

java.lang.
.String **[getExtension](#)**(java.io.File f)

Return the extension portion of the file's name .

boolean **[isExtensionListInDescription](#)**()

Returns whether the extension list (.jpg, .gif, etc) should show up in the human readable description.

void **[setDescription](#)**(java.lang.String description)

Sets the human readable description of this filter.

void **[setExtensionListInDescription](#)**(boolean b)

Determines whether the extension list (.jpg, .gif, etc) should show up in the human readable description.

Constructor Detail

FiltroFile

```
public FiltroFile()
```

Creates a file filter. If no filters are added, then all files are accepted.

See Also:

[addExtension\(java.lang.String\)](#)

FiltroFile

```
public FiltroFile(java.lang.String extension)
```

Creates a file filter that accepts files with the given extension. Example: new ExampleFileFilter("jpg");

See Also:

[addExtension\(java.lang.String\)](#)

FiltroFile

```
public FiltroFile(java.lang.String extension,  
                  java.lang.String description)
```

Creates a file filter that accepts the given file type. Example: new ExampleFileFilter("jpg", "JPEG Image Images"); Note that the "." before the extension is not needed. If provided, it will be ignored.

See Also:

[addExtension\(java.lang.String\)](#)

FiltroFile

```
public FiltroFile(java.lang.String[] filters)
```

Creates a file filter from the given string array. Example: new ExampleFileFilter(String {"gif", "jpg"}); Note that the "." before the extension is not needed and will be ignored.

See Also:

[addExtension\(java.lang.String\)](#)

FiltroFile

```
public FiltroFile(java.lang.String[] filters,  
                  java.lang.String description)
```

Creates a file filter from the given string array and description. Example: new ExampleFileFilter(String {"gif", "jpg"}, "Gif and JPG Images"); Note that the "." before the extension is not needed and will be ignored.

See Also:

[addExtension\(java.lang.String\)](#)

Method Detail

accept

```
public boolean accept(java.io.Filef)
```

Return true if this file should be shown in the directory pane, false if it shouldn't. Files that begin with "." are ignored.

Specified by:

accept in class javax.swing.filechooser.FileFilter

See Also:

[getExtension\(java.io.File\)](#), FileFilter#accepts

getExtension

```
public java.lang.String getExtension(java.io.Filef)
```

Return the extension portion of the file's name .

See Also:

[getExtension\(java.io.File\)](#),
FileFilter.accept(java.io.File)

addExtension

```
public void addExtension(java.lang.Stringextension)
```

Adds a filetype "dot" extension to filter against. For example: the following code will create a filter that filters out all files except those that end in ".jpg" and ".tif":
ExampleFileFilter filter = new ExampleFileFilter(); filter.addExtension("jpg");
filter.addExtension("tif"); Note that the "." before the extension is not needed and will be ignored.

getDescription

```
public java.lang.String getDescription()
```

Returns the human readable description of this filter. For example: "JPEG and GIF Image Files (*.jpg, *.gif)"

Specified by:

getDescription in class javax.swing.filechooser.FileFilter

See Also:

```
setDescription, setExtensionListInDescription,  
isExtensionListInDescription,  
FileFilter.getDescription()
```

setDescription

```
public void setDescription(java.lang.String description)
```

Sets the human readable description of this filter. For example:
filter.setDescription("Gif and JPG Images");

See Also:

```
setDescription, setExtensionListInDescription,  
isExtensionListInDescription
```

setExtensionListInDescription

```
public void setExtensionListInDescription(boolean b)
```

Determines whether the extension list (.jpg, .gif, etc) should show up in the human readable description. Only relevant if a description was provided in the constructor or using setDescription();

See Also:

```
getDescription, setDescription,  
isExtensionListInDescription
```

isExtensionListInDescription

```
public boolean isExtensionListInDescription()
```

Returns whether the extension list (.jpg, .gif, etc) should show up in the human readable description. Only relevant if a description was provided in the constructor or using setDescription();

See Also:

```
getDescription, setDescription,  
setExtensionListInDescription
```

Class ModeloTabla

```
java.lang.Object  
└─ javax.swing.table.AbstractTableModel  
    └─ knoe.soporteJava.ModeloTabla
```

All Implemented Interfaces:

java.io.Serializable, javax.swing.table.TableModel

```
public class ModeloTabla extends javax.swing.table.AbstractTableModel
```

See Also:

[Serialized Form](#)

Constructor Summary

ModeloTabla (java.lang.String[] col, java.lang.Object[][] dat) Este constructor inicializa los campo internos referentes a Cabecera de tabla y datos.

Method Summary

java.lang .Class	getColumnClass (int c)
int	getColumnCount ()
java.lang .String	getColumnName (int col)
int	getRowCount ()
java.lang .Object	getValueAt (int row, int col)
boolean	isCellEditable (int row, int col)
void	setValueAt (java.lang.Object value, int row, int col)

Constructor Detail

ModeloTabla

```
public ModeloTabla(java.lang.String[]col,  
                   java.lang.Object[][]dat)
```

Este constructor inicializa los campo internos referentes a Cabecera de tabla y datos.

Parameters:

col - - Array de Strings, cabecera de la tabla.

dat - - Tabla de Object, datos de la tabla.

Method Detail

getColumnCount

```
public int getColumnCount()
```

getRowCount

```
public int getRowCount()
```

getColumnName

```
public java.lang.String getColumnName(int col)
```

Specified by:

getColumnName in interface javax.swing.table.TableModel

Overrides:

getColumnName in class javax.swing.table.AbstractTableModel

getValueAt

```
public java.lang.Object getValueAt(int row,
                                   int col)
```

setValueAt

```
public void setValueAt(java.lang.Object value,
                       int row,
                       int col)
```

Specified by:

setValueAt in interface javax.swing.table.TableModel

Overrides:

setValueAt in class javax.swing.table.AbstractTableModel

getColumnClass

```
public java.lang.Class getColumnClass(int col)
```

Specified by:

getColumnClass in interface javax.swing.table.TableModel

Overrides:

getColumnClass in class
javax.swing.table.AbstractTableModel

isCellEditable

```
public boolean isCellEditable(introw,  
                               intcol)
```

Specified by:

isCellEditable in interface javax.swing.table.TableModel

Overrides:

isCellEditable in class
javax.swing.table.AbstractTableModel

Class SoporteArray

```
java.lang.Object  
└─ knoe.soporteJava.SoporteArray
```

```
public class SoporteArray extends java.lang.Object
```

Constructor Summary

<u>SoporteArray()</u>	
---------------------------------------	--

Method Summary

java.lang. .String[]	<u>arrayStringParaTableModel_UpdateCliente</u> (boolean[] campos, boolean flag) Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear.
java.lang. .String[]	<u>arrayStringParaTableModel_UpdateProducto</u> (boolean[] campos, boolean flag) Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear.
java.lang. .String[]	<u>arrayStringParaTableModelCliente</u> (boolean[] campos) Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear.
java.lang. .String[]	<u>arrayStringParaTableModelProducto</u> (boolean[] campos)

Constructor Detail

SoporteArray

```
public SoporteArray()
```

Method Detail

arrayStringParaTableModel_UpdateCliente

```
public java.lang.String[]  
arrayStringParaTableModel_UpdateCliente(boolean[] campos,  
                                           boolean flag)
```

Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear. En función de los checkBox elegidos en la ventana de dialogo. Añade una columna al inicio para seleccionar la fila a eliminar o modificar.

Parameters:

campos - - Protocolo de intercambio de parámetros: campos[0] referencia
campos[1] nombre campos[2] descripcion campos[3] url_imagen campos[4] pvp
campos[5] precio_pack campos[6] cond_oferta campos[7] precio_pnp campos[8]
categoria campos[9] marca campos[10] isbn campos[11] sinopsis
flag - - sirve para elegir el nombre de la columna 1: true ---> "modificar"
false---> "eliminar"

Returns:

Devuelve un array de objetos que contiene los campos de la tabla jTable a crear.

arrayStringParaTableModelCliente

```
public java.lang.String[] arrayStringParaTableModelCliente(boolean[] campos)
```

Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear. En función de los checkBox elegidos en la ventana de dialogo.

Parameters:

campos - - Protocolo de intercambio de parámetros: campos[0]= CodigoCliente
campos[1]= NombreCliente campos[2]= DNI campos[3]= LOGIN campos[4]=
PASSWORD campos[5]= TelefonoMovil campos[6]= Email campos[7]=
Domicilio campos[8]= Localidad campos[9]= CodigoPostal campos[10]=
TeléfonoFijo campos[11]= Fax campos[12]= CIFEmpresa campos[13]=
NombreEmpresa campos[14]= PersonaContactoEmpresa

Returns:

Devuelve un array de String que será la cabecera de una tabla.

arrayStringParaTableModel_UpdateProducto

```
public java.lang.String[]  
arrayStringParaTableModel_UpdateProducto(boolean[] campos,  
                                           boolean flag)
```

Esta función va a devolver un array de objetos que contiene los campos de la tabla jTable a crear. En función de los checkBox elegidos en la ventana de dialogo. Añade

una columna al inicio para seleccionar la fila a eliminar o modificar.

Parameters:

`campos` - - Protocolo de intercambio de parámetros: `campos[0]` referencia
`campos[1]` nombre `campos[2]` descripcion `campos[3]` url_imagen `campos[4]` pvp
`campos[5]` precio_pack `campos[6]` cond_oferta `campos[7]` precio_pnp `campos[8]`
categoria `campos[9]` marca `campos[10]` isbn `campos[11]` sinopsis
`flag` - - sirve para elegir el nombre de la columna 1: true ---> "modificar"
false---> "eliminar"

Returns:

Devuelve un array de objetos que contiene los campos de la cabecera de una tabla.

arrayStringParaTableModelProducto

```
public java.lang.String[] arrayStringParaTableModelProducto(boolean[] campos)
```

Bibliografía

Java

- Patrick Naughton. MANUAL DE JAVA. Osborne-McGraw Hill, 1996.
- Patrick Naughton y Herbert Schildt. Java. MANUAL DE REFERENCIA. Osborne-McGraw Hill.
- Javadocs en Sun: <http://download.oracle.com/javase/6/docs/api/>

Bases de datos

- Apuntes de la asignatura Bases de Datos de 5º curso de Ingeniería de Telecomunicaciones, Universidad de Sevilla.
- Conectores/drivers bases de datos
 - Sitio oficial de MySQL:
<http://dev.mysql.com/doc/refman/5.0/es/connectors.html>
 - Manual en sitio javadabbadoo.org:
<http://www.javadabbadoo.org/apendice/bds/jdbc/tiposDrivers/index.html>
 - Apuntes de la asignatura de la Universidad de Valencia:
<http://informatica.uv.es/it3guia/LP/teoria/apuntes/cuatr2/Tema8.pdf>
- MySQL
 - Manual de referencia del sitio oficial:
<http://dev.mysql.com/doc/refman/5.0/es/index.html>

PHP

- Till Gerken and Tobias Ratschiller. Creación de Aplicaciones Web Con PHP 4. Paperback, Mayo 2001.
- Sitio oficial: <http://www.php.net/>