

# Base de Datos con ObjectDB y MongoDB

PABLO HORCAJADA GONZÁLEZ  
2ºDAM ISEP CEU

# ÍNDICE

## Contenido

Objetivo de la aplicación .....	3
Especificación de los requisitos .....	4
Diseño de la aplicación.....	5
Login .....	5
Menú de Entidades .....	5
Menú de Acciones.....	5
Nuevo Empleado.....	6
Crear Ordenador .....	6
Añadir Periféricos.....	6
Modificar Empleado.....	7
Modificar Ordenador .....	7
Modificar Periféricos .....	7
Ver Empleado .....	8
Ver Ordenador .....	8
Eliminar Empleado .....	9
Eliminar Ordenador.....	9
ControladorObjectDB.....	10
AñadirEmpleado () .....	10
VerTodosOrdenadores () .....	10
VeEmpleado () .....	10
EliminarPeriferico () .....	11
ModifcarOrdenador () .....	12
ControladorMongoDB.....	13
AñadirPeriférico () .....	13
VertodosEmpleados () .....	14
VerOrdenador ().....	15
EliminarPeriferico () .....	16
EliminarEmpleado ().....	17



# Objetivo de la aplicación

- El Objetivo de la aplicación es realizar la gestión de Empleados, Ordenadores, Periféricos y Dispositivos en una empresa de Informática usando las bases de datos de MongoDB de Objectdb
- Si en algún momento se incorpora alguien o algo nuevo, es de gran utilidad usar la aplicación.
- Si queremos dar de baja algún empleado u objeto, con poner el asiento en el que está dicha entidad, se daría de baja
- Si queremos cambiar o modificar alguna entidad, poniendo el asiento de donde se ubica y el dato que queremos modificar, se cambiaría el antiguo por el nuevo.
- Si queremos ver todos los empleados, periféricos, etc , deberemos elegir marcando al check que aparecerá en la ventana. En cambio, si queremos ver solo un dato en específico, con poner el asiento serviría.

## Especificación de los requisitos

-El requisito fundamental para usar la aplicación, será usar la cuenta de Administrador, con los datos “AdminDam” como nombre de usuario y “2Dam123” como contraseña.

-Los requisitos de la aplicación serían poder modificar, eliminar, ver y añadir una nueva entidad a la base de datos.

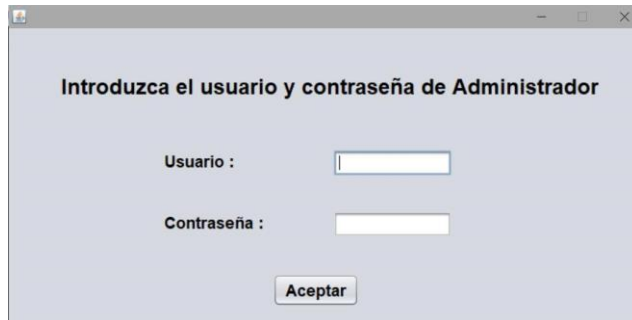
-Deberemos poder realizar dichas acciones en mínimo 2 entidades y mostrando el proceso de operación de dichas acciones visualmente y escribiendo los datos que te pida la ventana en los encontramos

# Diseño de la aplicación

-El diseño de la aplicación consistiría en una serie de ventanas que irás apareciendo y desapareciendo según la ventana que estés.

## Login

-Una básica y simple ventana con un título, varios textos indicando un mensaje, dos cajitas para poder escribir el usuario y la contraseña, y un botón que pasaría a la siguiente ventana si el usuario es el del Administrador.



## Menú de Entidades

-Una básica y simple ventana con un título, 4 botones que indican las entidades y un botón para cerrar el programa.

-Da igual que botón de entidad pulsemos, pasaremos a la siguiente ventana guardando a la entidad seleccionada en el código



## Menú de Acciones

-Una básica y simple ventana con cuatro botones que representan las acciones, y un botón para volver a la ventana anterior.

-Dependiendo de qué entidad elegimos en ventana anterior, al pulsar en una acción pasaremos a la siguiente ventana que corresponderá a la acción elegida para esa entidad.



## Nuevo Empleado

-Ventana con un título, 4 textos indicando los datos, 4 tablas para escribir los datos indicados para el empleado y dos botones que sirven para volver a la ventana anterior y guardar el nuevo empleado

The screenshot shows a window titled "Nuevo Empleado". It contains four labels with corresponding text input fields: "Nombre :", "Apellidos :", "Departamento :", and "Asiento". At the bottom of the window, there are two buttons: "Volver" and "Guardar".

## Crear Ordenador

-Ventana con un título, textos y cajitas pidiendo el dato, y dos botones para realizar la acción y volver a la ventana anterior.

The screenshot shows a window titled "Crear Ordenador". It contains several labels with corresponding text input fields: "Asiento", "Placa", "Alimentacion", "Disco duro", "Procesador", "Tarjeta Grafica", "Disco duro ssd", "Memoria Ram", "Torre", and "Ventilacion". At the bottom of the window, there are two buttons: "Volver" and "Crear".

## Añadir Periféricos

-Ventana con un título, textos y cajitas pidiendo el dato, y dos botones para realizar la acción y volver a la ventana anterior

The screenshot shows a window titled "Añadir Periféricos". It contains six labels with corresponding text input fields: "Monitor 1", "Monitor 2", "Raton", "Teclado", "Disco portatil", and "Asiento". At the bottom of the window, there are two buttons: "Volver" and "Añadir".

## Modificar Empleado

-Ventana con un título, 4 textos indicando los datos, 4 cajitas para poner datos, dos botones para volver a la ventana anterior o modificar el empleado, y una nota diciendo un texto

The screenshot shows a window titled "Modificar un empleado". It contains four input fields with labels: "Introduce el asiento del empleado :", "Introduce el nuevo nombre :", "Introduce los nuevos apellidos :", and "Introduce el departamento :". Below these fields is a note: "Nota : no hace falta que rellenes todos los nuevos datos, con uno sirve". At the bottom are two buttons: "Volver" and "Modificar".

## Modificar Ordenador

-Ventana con un título, textos y cajitas donde te indica los datos a modificar, y los dos botones para volver a la ventana anterior y modificar la entidad

The screenshot shows a window titled "Modificar Ordenador". It features an "Asiento" input field at the top. Below it, there are three columns of input fields with labels: "Placa", "Tarjeta Grafica", "Procesador" in the first column; "Ventilacion", "Torre", "Memoria Ram" in the second column; and "Alimentacion", "Disco duro", "Disco duro ssd" in the third column. At the bottom are two buttons: "Volver" and "Modificar".

## Modificar Periféricos

-Ventana con un título, textos y cajitas donde te indica los datos a modificar, y los dos botones para volver a la ventana anterior y modificar la entidad

The screenshot shows a window titled "Modificar Perifericos". It has an "Introduce el asiento donde se modificará" input field at the top. Below it, there are three rows of input fields with labels: "Monitor1", "Monitor2", "Raton" in the first row; and "Teclado", "Disco portatil" in the second row. At the bottom are two buttons: "Volver" and "Modificar".



## Ver Empleado

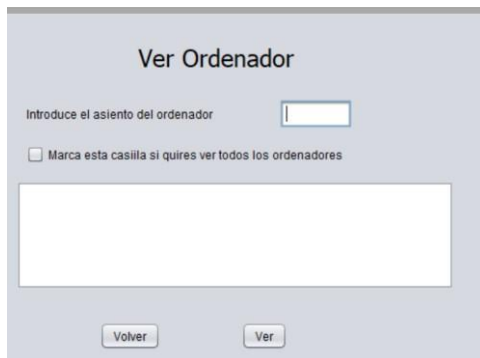
-Ventana con un título, un texto indicando el dato que hay que escribir, una caja para escribir el dato que pide, un check por si queremos ver todos los empleados, y una caja grande donde aparecerá la información que queremos



The screenshot shows a window titled "Ver Empleado". It contains a label "Introduce el asiento del Empleado que quieres ver" followed by a text input field. Below this is a checkbox labeled "Marca esta casilla si quieres ver todos los empleados". A large empty rectangular box is positioned below the checkbox, intended for displaying employee information. At the bottom of the window are two buttons: "Volver" and "Ver".

## Ver Ordenador

-Ventana con un título, un texto indicando el dato que hay que escribir, una caja para escribir el dato que pide, un check por si queremos ver todos los ordenadores, y una caja grande donde aparecerá la información que queremos ver.



The screenshot shows a window titled "Ver Ordenador". It contains a label "Introduce el asiento del ordenador" followed by a text input field. Below this is a checkbox labeled "Marca esta casilla si quieres ver todos los ordenadores". A large empty rectangular box is positioned below the checkbox, intended for displaying computer information. At the bottom of the window are two buttons: "Volver" and "Ver".

## Ver Periféricos

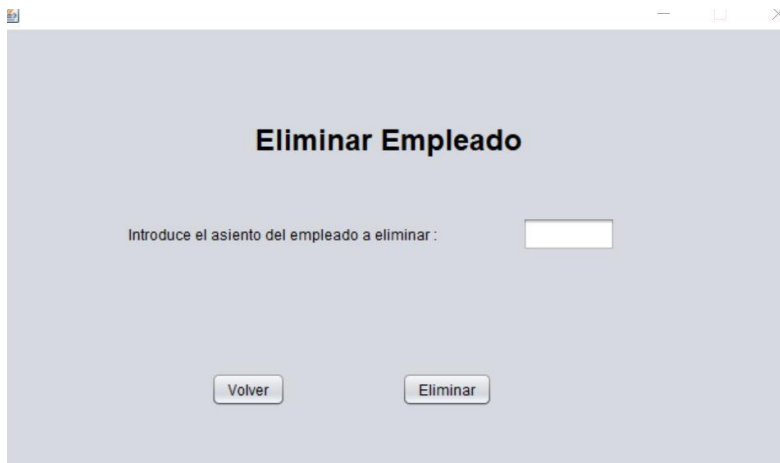
-Ventana con un título, un texto indicando el dato que hay que escribir, una caja para escribir el dato que pide, un check por si queremos ver todos los ordenadores, y una caja grande donde aparecerá la información que queremos ver.



The screenshot shows a window titled "Ver Periféricos". It contains a label "Introduce el Asiento donde se encuentran los periféricos" followed by a text input field. Below this is a checkbox labeled "Marca esta casilla para ver todos los Periféricos". A large empty rectangular box is positioned below the checkbox, intended for displaying peripheral information. At the bottom of the window are two buttons: "Volver" and "Ver".

## Eliminar Empleado

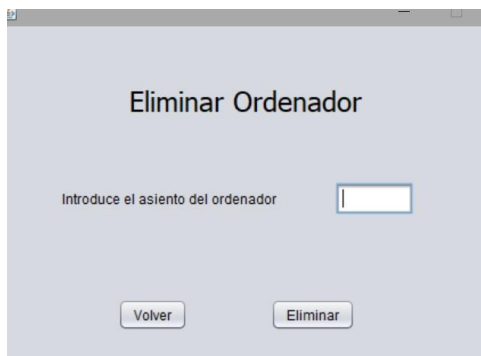
-ventana con un título, dos botones para volver a la ventana anterior o eliminar, un texto indicando el dato, y una cajita para escribir el dato que nos pide.



The screenshot shows a window titled "Eliminar Empleado". Inside, there is a label "Introduce el asiento del empleado a eliminar :" followed by a text input field. At the bottom, there are two buttons: "Volver" and "Eliminar".

## Eliminar Ordenador

-Ventana con un título, dos botones para volver a la ventana anterior y eliminar el ordenador, un texto indicando el dato y una cajita para escribir el dato que nos pide



The screenshot shows a window titled "Eliminar Ordenador". Inside, there is a label "Introduce el asiento del ordenador" followed by a text input field. At the bottom, there are two buttons: "Volver" and "Eliminar".

## Eliminar Periféricos

-Ventana con un título, dos botones para volver a la ventana anterior y eliminar el ordenador, un texto indicando el dato, una cajita para escribir el dato que nos pide y un check por cada dato por si queremos eliminar un periférico en concreto.



The screenshot shows a window titled "Eliminar Periferico". Inside, there is a label "Introduce el asiento para eliminar todos los perifericos de ese asiento" followed by a text input field. Below this, there is a note: "Nota : si solo quieres eliminar un periferico de ese asiento, marca la casilla de ese periferico". Under the note, there are five checkboxes labeled "Monitor1", "Monitor2", "Raton", "Teclado", and "Disco portatil". At the bottom, there are two buttons: "Volver" and "Eliminar".

# ControladorObjectDB

-A continuación, mostraré algunos métodos que utilizo (ya que la mayoría son todos iguales, porque todos realizan las 4 acciones)

## AñadirEmpleado ()

-Al pulsar el botón de guardar empleado se ejecutará el método AñadirEmpleado () del controlador de object , que guarda en una variable ,el texto que estaba en las cajitas para meterlo en un objeto “Empleado”, para posteriormente, añadir dicho objeto a la base de datos

```
private void jButtonGuardarActionPerformed(java.awt.event.ActionEvent evt) {  
    if (jTextFieldapellidos.getText().equals("") || jTextFieldnombre.getText().equals("") ||  
        jLabelerror.setVisible(true);  
    }  
    else {  
        controlador.ControladorMongo.AñadirEmpleado();  
        controlador.Controlador.AnadirEmpleado();  
        vista.CrearEmpleado().jLabelerror.setText("Empleado añadido");  
    }  
}  
  
//Añadir un empleado  
public static void AñadirEmpleado() {  
    //Empezamos la transaccion  
    em.getTransaction().begin();  
    String nombre = vista.CrearEmpleado().jTextFieldnombre.getText();  
    String apellidos = vista.CrearEmpleado().jTextFieldapellidos.getText();  
    String departamento = vista.CrearEmpleado().jTextFielddepartamento.getText();  
    int sitio = Integer.parseInt(vista.CrearEmpleado().jTextFieldsiento.getText());  
    //creamos el objeto Empleado con sus datos  
    Empleado empl = new Empleado(nombre, apellidos, departamento, sitio);  
    //añadimos el nuevo empleado  
    em.persist(empl);  
    //La cerramos  
    em.getTransaction().commit();  
    vista.CrearEmpleado().jLabelerror.setText("Empleado añadido");  
}
```

## VerTodosOrdenadores ()

-Dependiendo de si marcamos el check o no, ejecutaremos el método VerTodosOrdenadores () del controlador de object, que muestra con un bucle y el query , todos los ordenadores.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if (!jCheckBox1.isSelected() && jTextFieldasiento.getText().equals("")) {  
        jTextAreaInfoOrdenadores.setText("Debes marcar la casilla o poner un asiento");  
    }  
    else {  
        if (jCheckBox1.isSelected()) {  
            jTextAreaInfoOrdenadores.setText("");  
            controlador.Controlador.VerTodosOrdenador();  
            controlador.ControladorMongo.VertodosOrdenador();  
        }  
        if (!jTextFieldasiento.getText().equals("")) {  
            jTextAreaInfoOrdenadores.setText("");  
            controlador.Controlador.VerOrdenador();  
            controlador.ControladorMongo.VerOrdenador();  
        }  
    }  
}  
  
//Ver todos los ordenadores  
public static void VerTodosOrdenador() {  
    TypedQuery<Ordenador> query = em.createQuery("SELECT ord FROM Ordenador ord", Ordenador.class);  
    List<Ordenador> results = query.getResultList();  
    String datos = "";  
    for (Ordenador ord : results) {  
        datos += ord.toString();  
        vista.VerOrdenador().jTextAreaInfoOrdenadores.setText(datos);  
    }  
}
```

## VeEmpleado ()

-Dependiendo de si escribimos un asiento en la cajita, ejecutaremos el método VerEmpleado () del controlador de object, que muestra con un query el empleado que se sienta en ese asiento

```
private void jButtonVerActionPerformed(java.awt.event.ActionEvent evt) {
    //Si marcamos la casilla de ver todos
    if (!jChecktodos.isSelected() && jTextFieldasiento.getText().equals("")) {
        jLabel4.setText("Debes marcar la casilla o poner un asiento");
    }
    else{
        if (jChecktodos.isSelected()){
            jLabel4.setText("");
            controlador.Controlador.VerTodosEmpleados();
            controlador.ControladorMongo.VertodosEmpleados();
        }
        if (!jTextFieldasiento.getText().equals("")){
            jLabel4.setText("");
            controlador.Controlador.VerEmpleado();
            controlador.ControladorMongo.VerEmpleado();
        }
    }
}
}
```

```
//Ver solo un empleado
public static void VerEmpleado(){
    em.getTransaction().begin();
    int asiento = Integer.parseInt(vista.VerEmpleado(jTextFieldasiento.getText()));
    Query query = em.createQuery("SELECT empl FROM Empleado empl WHERE empl.Sitio="+asiento);
    if (!query.equals(true)){
        modelo.Empleado p = (modelo.Empleado) query.getSingleResult();
        if (asiento==p.getSitio()){
            String dato = p.toString();
            vista.VerEmpleado.jTextAreaInfoEmpleados.setText("");
            vista.VerEmpleado.jTextAreaInfoEmpleados.setText(dato);
        }
    }
    else vista.VerEmpleado.jLabel4.setText("Noy nadie sentado en el asiento " + asiento);
    em.getTransaction().commit();
}
```

## EliminarPeriferico ()

-Al pulsar el botón de eliminar (si hemos escrito algo en la cajita) se ejecutará el método EliminarPerifericos () del controlador de object, seleccionando con un query el dato “asiento”. Además, si hemos marcando algún check, solo eliminará ese periférico en concreto.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    if (jTextFieldasiento.getText().equals("")){
        jLabel4.setText("Debes escribir un asiento para eliminar a alguien");
    }
    else{
        controlador.ControladorMongo.EliminarPeriferico();
        controlador.Controlador.EliminarPeriferico();
        vista.EliminarPeriferico.jLabel4.setText("El empleado que se sentaba en el asiento " + this.jTextFli
    }
}
}
```

```
//Eliminar un periferico
public static void EliminarPeriferico(){
    em.getTransaction().begin();
    int asiento = Integer.parseInt(vista.EliminarPeriferico(jTextFieldasiento.getText()));
    Query query = em.createQuery("SELECT per FROM Perifericos per WHERE per.Asiento="+asiento);
    if (!query.equals(true)){
        modelo.Perifericos p = (modelo.Perifericos) query.getSingleResult();
        if (vista.EliminarPeriferico.jCheckBox1.isSelected()){
            //Query query2 = em.createQuery("SELECT per.Monitor1 FROM Perifericos per WHERE per.Asiento="+asiento);
            p.setMonitor1("");
            vista.EliminarPeriferico.jLabel4.setText("El monitor1 del asiento " + asiento + " ha sido eliminado");
        }
        if (vista.EliminarPeriferico.jCheckBox2.isSelected()){
            p.setMonitor2("");
            vista.EliminarPeriferico.jLabel4.setText("El monitor2 del asiento " + asiento + " ha sido eliminado");
        }
        if (vista.EliminarPeriferico.jCheckBox3.isSelected()){
            p.setRaton("");
            vista.EliminarPeriferico.jLabel4.setText("El raton del asiento " + asiento + " ha sido eliminado");
        }
        if (vista.EliminarPeriferico.jCheckBox4.isSelected()){
            p.setTeclado("");
            vista.EliminarPeriferico.jLabel4.setText("El teclado del asiento " + asiento + " ha sido eliminado");
        }
        if (vista.EliminarPeriferico.jCheckBox5.isSelected()){
            p.setDisco_portatil("");
            vista.EliminarPeriferico.jLabel4.setText("El disco portatil del asiento " + asiento + " ha sido eliminado");
        }
        else {
            em.remove(p);
            vista.EliminarPeriferico.jLabel4.setText("Se han borrado los perifericos del asiento " + asiento);
        }
    }
    else vista.EliminarPeriferico.jLabel4.setText("No hay nadie sentado en el asiento " + asiento);
    em.getTransaction().commit();
}
```

## ModificarOrdenador ()

-Al pulsar el botón de Modificar (si hemos escrito el asiento) se ejecutará el método ModificarOrdenador () del controlador de object . 1º busca “find” el Ordenador que tenga el dato de “asiento” y dependiendo si hemos escrito algo o no en las cajitas, modificará el dato antiguo por el nuevo que hemos escrito

```
private void jButtonCrearActionPerformed(java.awt.event.ActionEvent evt) {  
    if (jTextFieldasiento.getText().equals("")) {  
        jLabel12.setText("Para modificar tienes que escribir el asiento");  
    }  
    else {  
        controlador.Controlador.ModificarOrdenador();  
        controlador.ControladorMongo.ModificarOrdenador();  
        this.jLabel12.setText("El ordenador en el asiento " + this.jTextFieldasient  
    }  
}
```

```
//Modificar un Ordenador  
public static void ModificarOrdenador() {  
    em.getTransaction().begin();  
    int asiento = Integer.parseInt((vista.ModificarOrdenador.jTextFieldasiento.getText()));  
    Ordenador o = em.find(Ordenador.class, asiento);  
    if (!vista.ModificarOrdenador.jTextFieldasiento.getText().equals("")) {  
        if (!vista.ModificarOrdenador.jTextFieldplaca.getText().equals("")) {  
            o.setPlaca(vista.ModificarOrdenador.jTextFieldplaca.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFieldprocesador.getText().equals("")) {  
            o.setProcesador(vista.ModificarOrdenador.jTextFieldprocesador.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFielddram.getText().equals("")) {  
            o.setMemoria_ram(vista.ModificarOrdenador.jTextFielddram.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFieldalimentacion.getText().equals("")) {  
            o.setAlimentacion(vista.ModificarOrdenador.jTextFieldalimentacion.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFieldtarjeta.getText().equals("")) {  
            o.setTarjeta_grafica(vista.ModificarOrdenador.jTextFieldtarjeta.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextField torre.getText().equals("")) {  
            o.setCaja(vista.ModificarOrdenador.jTextField torre.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFielddisco.getText().equals("")) {  
            o.setDisco_duro(vista.ModificarOrdenador.jTextFielddisco.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFieldssd.getText().equals("")) {  
            o.setDisco_ssd(vista.ModificarOrdenador.jTextFieldssd.getText());  
        }  
        if (!vista.ModificarOrdenador.jTextFieldventilacion.getText().equals("")) {  
            o.setVentilacion(vista.ModificarOrdenador.jTextFieldventilacion.getText());  
        }  
    }  
    vista.ModificarOrdenador.jLabel12.setText("Ordenador Modificado");  
    em.getTransaction().commit();  
}  
else vista.ModificarOrdenador.jLabel12.setText("Debes escribir el asiento para modificar");  
}
```



# ControladorMongoDB

--A continuación, mostraré algunos métodos que utilizo (ya que la mayoría son todos iguales, porque todos realizan las 4 acciones)

-Para este controlador, debemos conectar el cliente de Mongo con NetBeans, conectarnos a nuestra base de datos "basededatos", a las colecciones de las entidades y crear los objetos cursores para localizar datos.

```
static MongoClient cliente = MongoClient.create();
static MongoDB database = cliente.getDatabase("basededatos");
static MongoCollection coleccionempl = database.getCollection("Empleado");
static MongoCollection coleccionord = database.getCollection("Ordenador");
static MongoCollection coleccionper = database.getCollection("Perifericos");
static MongoCursor cursor = coleccionempl.find().iterator();
static MongoCursor cursor2 = coleccionord.find().iterator();
static MongoCursor cursor3 = coleccionper.find().iterator();
```

## AñadirPeriférico ()

-Si pulsamos el botón de añadir en la ventana de añadir Periféricos y hemos escrito en todas las cajitas, ejecutaremos el método Añadirperifericos () desde el controlador de mongo que añadirá en un documento los datos que contienen las cajitas junto al nombre de los campos para posteriormente insertar dicho documento en la colección correspondiente.

```
private void AñadirActionPerformed(java.awt.event.ActionEvent evt) {
    if (jTextFieldasiento.getText().equals("") || jTextFielddisco.getText().equals("") || jTextFieldmonitor1.getText().equals("") || jTextFieldmonitor2.getText().equals("") || jTextFieldraton.getText().equals("") || jTextFieldteclado.getText().equals("")){
        jLabelerror.setText("No puedes dejar ningún campo vacio");
    }
    else{
        controlador.ControladorMongo.AñadirPeriferico();
        controlador.Controlador.AñadirPeriferico();
        vista.CrearPeriferico(jLabelerror.setText("Perifericos creados"));
    }
}
```

```
//Añadir un Periferico
public static void AñadirPeriferico(){
    Document perifericos = new Document("Monitor1", vista.CrearPeriferico.jTextFieldmonitor1.getText())
        .append("Monitor2", vista.CrearPeriferico.jTextFieldmonitor2.getText())
        .append("Raton", vista.CrearPeriferico.jTextFieldraton.getText())
        .append("Teclado", vista.CrearPeriferico.jTextFieldteclado.getText())
        .append("Disco portatil", vista.CrearPeriferico.jTextfielddisco1.getText())
        .append("Asiento", Integer.parseInt(vista.CrearPeriferico.jTextFieldasiento.getText()));
    coleccionper.insertOne(perifericos);
    vista.CrearPeriferico(jLabelerror.setText("Perifericos insertados"));
}
```

## VertodosEmpleados ()

-Si pulsamos en el botón de ver y hemos marcado la casilla que ver todos, ejecutaremos el método VertodosEmpleados () desde el controlador de Mongo que con un bucle y con un cursor, recorre el contenido de los documentos, guardando los datos de los campos en un String que se pondrá como texto en la cajita grande de la ventana Ver Empleados

```
private void jButtonVerActionPerformed(java.awt.event.ActionEvent evt) {  
    //Si marcamos la casilla de ver todos  
    if (!jChecktodos.isSelected() && jTextFieldasiento.getText().equals("")) {  
        jLabel4.setText("Debes marcar la casilla o poner un asiento");  
    }  
    else{  
        if (jChecktodos.isSelected()){  
            jLabel4.setText("");  
            controlador.Controlador.VerTodosEmpleados();  
            controlador.ControladorMongo.VertodosEmpleados();  
        }  
        if (!jTextFieldasiento.getText().equals("")){  
            jLabel4.setText("");  
            controlador.Controlador.VerEmpleado();  
            controlador.ControladorMongo.VerEmpleado();  
        }  
    }  
}
```

```
public static void VertodosEmpleados() {  
    String datos = "";  
    vista.VerEmpleado.jTextAreaInfoEmpleados.setText("");  
    while(cursor.hasNext()){  
        Document empleado =(Document) cursor.next();  
        datos += "Nombre: " + empleado.get("Nombre");  
        datos += "; Apellidos: " + empleado.get("Apellidos");  
        datos += "; Departamento: " + empleado.get("Departamento");  
        datos += "; Asiento: " + empleado.get("Sitio");  
        datos += '\n';  
    }  
    vista.VerEmpleado.jTextAreaInfoEmpleados.setText(vista.VerEmpleado.jT  
}
```

## VerOrdenador ()

-Si al pulsar el botón de ver hemos escrito un asiento en la cajita, nos ejecutará el método VerOrdenador () del controlador de mongo que mirará dentro de la colección correspondiente con el cursor y un bucle, si alguna entidad tiene en el campo asiento el asiento que le hemos pasado, y entonces guardará en un String los datos de la entidad para luego imprimirlos

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if (!jCheckBox1.isSelected() && jTextFieldasiento.getText().equals(""))  
        jTextAreaInfoOrdenadores.setText("Debes marcar la casilla o poner  
    }  
    else {  
        if (jCheckBox1.isSelected()) {  
            jTextAreaInfoOrdenadores.setText("");  
            controlador.Controlador.VerTodosOrdenador();  
            controlador.ControladorMongo.VertodosOrdenador();  
        }  
        if (!jTextFieldasiento.getText().equals("")) {  
            jTextAreaInfoOrdenadores.setText("");  
            controlador.Controlador.VerOrdenador();  
            controlador.ControladorMongo.VerOrdenador();  
        }  
    }  
}
```

```
//Ver un ordenador  
public static void VerOrdenador() {  
    boolean ver = false;  
    String dato = "";  
    int asiento = Integer.parseInt(vista.VerOrdenador.jTextFieldasiento.getText());  
    vista.VerOrdenador.jTextAreaInfoOrdenadores.setText("");  
    while(cursor2.hasNext()){  
        Document ordenador =(Document) cursor2.next();  
        if(ordenador.get("Asiento").equals(asiento)){  
            dato += "Placa: " + ordenador.get("Placa");  
            dato += ";   Procesador: " + ordenador.get("Procesador");  
            dato += ";   Tarjeta_grafica: " + ordenador.get("Tarjeta_grafica");  
            dato += ";   Memoria_ram: " + ordenador.get("Memoria_ram");  
            dato += ";   Alimentacion: " + ordenador.get("Alimentacion");  
            dato += ";   Disco_duro: " + ordenador.get("Disco_duro");  
            dato += ";   Disco_ssd: " + ordenador.get("Disco_ssd");  
            dato += ";   Ventilacion: " + ordenador.get("Ventilacion");  
            dato += ";   Caja: " + ordenador.get("Caja");  
            dato += ";   Asiento: " + ordenador.get("Asiento");  
            vista.VerOrdenador.jTextAreaInfoOrdenadores.setText(dato);  
            ver = true;  
        }  
    }  
    if(!ver){  
        vista.VerOrdenador.jTextAreaInfoOrdenadores.setText("Noy nadie sentado en el asiento  
    }  
}
```



## EliminarPeriferico ()

-Si pulsamos el botón de modificar y hemos escrito algún dato nuevo, ejecutaremos el método ModificarPeriferico () del controlador de mongo que mirará dentro de los documentos, con un bucle y un cursor, si el asiento introducido coincide con alguien que se sienta en ese asiento, entonces actualizaremos dicho campo con los nuevos datos en la colección correspondiente.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    if(this.jTextFielddasiento.getText().equals("")){  
        this.jLabelerror.setText("Para modificar tienes que escribir el asiento");  
    }  
    else {  
        controlador.Controlador.ModificarPeriferico();  
        controlador.ControladorMongo.ModificarPeriferico();  
        this.jLabelerror.setText("Los perifericos del asiento " + this.jTextFielddasiento.getText() +  
    }  
}
```

```
//Modificar Periferico  
public static void ModificarPeriferico(){  
    int asiento = Integer.parseInt(vista.ModificarPerifericos.jTextFielddasiento.getText());  
    while(cursor3.hasNext()){  
        Document perifericos =(Document) cursor3.next();  
        if(perifericos.get("Asiento").equals(asiento)){  
            if (!vista.ModificarPerifericos.jTextFielddmlq.getText().equals("")){  
                coleccionper.updateOne(eq("Asiento",asiento), combine(set("Monitor1",vista.ModificarPerifericos.jTextFielddmlq.getText())));  
            }  
            if(!vista.ModificarPerifericos.jTextFielddm2.getText().equals("")){  
                coleccionper.updateOne(eq("Asiento",asiento), combine(set("Monitor2",vista.ModificarPerifericos.jTextFielddm2.getText())));  
            }  
            if (!vista.ModificarPerifericos.jTextFielddraton.getText().equals("")){  
                coleccionper.updateOne(eq("Asiento",asiento), combine(set("Raton",vista.ModificarPerifericos.jTextFielddraton.getText())));  
            }  
            if(!vista.ModificarPerifericos.jTextFielddteclado.getText().equals("")){  
                coleccionper.updateOne(eq("Asiento",asiento), combine(set("Teclado",vista.ModificarPerifericos.jTextFielddteclado.getText())));  
            }  
            if(!vista.ModificarPerifericos.jTextFieldddisco.getText().equals("")){  
                coleccionper.updateOne(eq("Asiento",asiento), combine(set("Disco portatil",vista.ModificarPerifericos.jTextFieldddisco.getText())));  
            }  
            vista.ModificarPerifericos.jLabelerror.setText("Perifericos modificados");  
        }  
    }  
    vista.ModificarPerifericos.jLabelerror.setText("No hay perifericos en el asiento " + asiento);  
}
```

## EliminarEmpleado ()

-Si pulsamos el botón de eliminar y hemos escrito el un asiento, ejecutaremos el método EliminarEmpleado () del controlador de mongo que eliminará la entidad que se sienta en ese sitio usando un cursor y un documento.

```
private void jButtonEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    if (jTextFieldAsiento.getText().equals("")) {  
        jLabel3.setText("Debes escribir un asiento para eliminar a alguien");  
    }  
    else {  
        controlador.ControladorMongo.EliminarEmpleado();  
        controlador.Controlador.EliminarEmpleado();  
        vista.EliminarEmpleado.jLabel3.setText("El empleado que se sentaba en  
    }  
}
```

```
//Eliminar un empleado  
public static void EliminarEmpleado() {  
    boolean eliminar = false;  
    int asiento = Integer.parseInt(vista.EliminarEmpleado.jTextFieldAsiento.getText());  
    while(cursor.hasNext()) {  
        Document empleado = (Document) cursor.next();  
        if(empleado.get("Sitio").equals(asiento)) {  
            coleccionempl.deleteOne(eq("Sitio", asiento));  
            eliminar = true;  
            vista.EliminarEmpleado.jLabel3.setText("El empleado que se sentaba  
        }  
    }  
    if(!eliminar) {  
        vista.EliminarEmpleado.jLabel3.setText("Noy nadie sentado en el asiento  
    }  
}
```