

TECNOLÓGICO NACIONAL DE MÉXICO	MANUAL DE PRÁCTICAS
INSTITUTO TECNOLÓGICO DE OAXACA	TÓPICOS AVANZADOS DE PROGRAMACIÓN
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN	DR. ROGELIO LIMÓN CORDERO (2018)

Práctica 9. Desarrollar una aplicación para acceder y manipular a los datos mediante una GUI.

Competencia a desarrollar

Establece conexiones a diferentes orígenes de datos para su manipulación y visualización de información.

Introducción

La creación de aplicaciones de tipo GUI que utilizan datos persistentes implica el uso de otros conocimientos, básicamente principios de SQL. Esto le da una connotación de un trabajo integrador y colaborativo.

Esta práctica se ha diseñado con la intención de crear los cimientos de un proyecto integrador que pueda ser desarrollado en conjunto con los que cursen la asignatura de fundamentos de base de datos. Incluye las habilidades sobre el manejo de aplicaciones GUI para acceder, representar y manejar la información contenida en una base de datos. También se aprovechan experiencias de las prácticas pasadas integrando a la uno y tres en una versión que utiliza base de datos. No está por demás recalcar que es indispensable contar con la aplicación desarrollada en la práctica ocho donde se creó la base de datos y su conexión.

Lo importante aquí es completar el ciclo de lo visto en la práctica ocho vinculándola con una GUI.

Correlación con los temas y aplicación en el contexto

Esta práctica se relaciona con todos los temas de la primera competencia del curso y todos los temas de la competencia de esta cuarta unidad.

El contexto donde se aplica corresponde a software java tipo GUI y uso básico de bases de datos relacionales mediante el manejador de base datos nativo del propio java.

Material y equipo necesario

Equipo de cómputo: Laptop o PC

Software: Aquí se emplea el “IDE” NetBeans pero puedes utilizar cualquier otro.

Java versión del jdk 1.7.0 o posterior

Base de datos creada en la practica ocho.

Código fuente de los proyectos desarrollados en las practicas:

Práctica ocho. Contiene la conexión con la base de datos y el manejo de esta

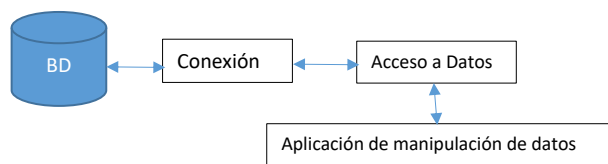
Opcionalmente las practicas uno y tres, pues contienen información y código que puede ser re-utilizado aquí.

Metodología

Se parte de la descripción de la aplicación a desarrollar presentando la GUI que se debe implementar y su funcionalidad. Luego durante el desarrollo de la práctica se expone la forma de implementar la aplicación utilizando como estrategia el aprovechamiento el código realizado en las practicas relacionadas. Se hacen muestras de parte de la implementación y se deja al estudiante que complemente código semejante a las muestras para lograr su participación activa a nivel individual y grupal, además se hacen preguntas explícitas que tiene que ir anotando en su bitácora de seguimiento.

Descripción de la aplicación

Se pretende crear una aplicación tipo GUI que manipule información sobre datos básicos de personas y de la información básica de sus medidas antropométricas. La manipulación consistirá en agregar registros de personas y de sus medidas; recuperación de dicha información y representación de forma diferente y eliminación de registros.



El esquema de abajo muestra una ejecución de la aplicación, en la parte izquierda se aprecian los registros que se tienen capturados sobre información de las personas y las opciones para ingresar nuevos elementos así los controles para la manipulación de los existentes. En la parte derecha se muestra la interfaz para ingresar y conocer la información sobre los registros de medidas de una persona que ha sido seleccionada en el panel de personas, en este ejemplo se muestra que la persona ‘Maria Callas’ ha registrado tres mediciones en diferentes periodos, con la opción ahora de registrar una más o eliminar alguna de las existentes.

Manipulación de datos de personas

Proporciona los datos

Nombre:

Fecha Nacimiento: Año:

Sexo: ☒ Hombre ☐ Mujer

Botones: Agregar Registro, Inicia valores, Elimina Reg. Seleccionado, Agregar Medidas

No. Idem	Nombre	Fecha Nac.	Sexo
1	Patricio Hardier	1999-01-08	HOMBRE
2	Maria Callas	1999-01-08	MUJER
3	Carlos Fuentes	1999-08-20	HOMBRE
4	Yolanda Aponte	1999-01-15	MUJER
5	Abel	1999-01-12	MUJER
10	Diego Velasco	2019-01-12	HOMBRE
12	Nicola Alonso	1999-01-12	MUJER
13	Isidoro Salavert	1999-01-12	HOMBRE

Manipulación de datos de mediciones para : —Maria Callas—

Proporciona los datos

Estatura(m):

Peso (kg):

Cintura (cm):

Cadera (cm):

Selección de la actividad: MODERADAMENTE ACTIVAS

Botones: Agregar Registro, Inicia valores, Elimina Reg. Seleccionado

Fecha	Estatura(cm)	Peso(kg)	Cintura(cm)	Cadera(cm)	Tipo de Act.
2019-01-14	160	80.5	90	110	2
2019-05-14	160	80.0	85	105	3
2019-01-14	160	78.0	84	105	3

Funcionalidad

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

Panel 'Personas'

Registro de los datos de personas: Se proporciona la información de la persona y se agrega a la base de datos de persona, siempre y cuando cuente con información. Como entrada se tiene como fecha mínima de nacimiento la de una persona de 20 años.

El botón [Agrega Registro], inserta un registro de persona en la base de datos y muestra su información en la tabla de abajo. Nota que su identificación no se captura.

El botón [Inicia valores] hace que los valores de los campos regresen al estado inicial.

El botón [Elimina Reg. Seleccionado], elimina de la base de datos al registro seleccionado , refrescando los valores mostrados en la tabla de abajo.

El botón [agrega medidas] Actúa cuando está seleccionado un registro, dicho registro carga su información al panel de la pestaña "Medidas" a la vez que selecciona los datos de registros sobre mediciones que haya tenido previamente registrados la persona seleccionada.

Panel 'Selección'

Muestra la información sobre medidas registradas de la persona seleccionada y los campos para agregar una nueva medida. En caso de no haberse seleccionado una persona, solo se muestra la información de los campos pero no se puede registrar una medida, puesto que no se tiene designada a qué persona le correspondería.

Los datos a introducir de las medidas incluyen la funcionalidad de la práctica tres, donde se validan los campos numéricos, la estatura tiene límites inferior y superior además de ser cíclica.

El dato sobre 'actividad' es seleccionable y sus datos a mostrar se cargan de la tabla 'TIPOACTIVIDAD' en lugar del arreglo usado en la práctica tres.

El botón [Agregar Registro] agrega los datos de las medidas referidas a la persona previamente seleccionada a la vez que refleja el nuevo valor en la tabla de abajo. Nótese que la fecha se toma del sistema y que aunque aquí no se pide que se valide para evitar que se introduzca dos registros con la misma fecha, extra-práctica es conveniente hacer dicha validación.

El botón [Inicia Valores], hace que los valores de los campos se regresen a valores por omisión.

El botón [Elimina Reg. Seleccionado]. Elimina de la tabla correspondiente de la base de datos al registro seleccionado.

Desarrollo

Utiliza a las clases creadas en la práctica ocho y emplea como referencia el código de las prácticas uno y tres transcribiendo(copiando) y/o adaptando el código requerido.

1. Creación de la estructura básica de la aplicación

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

- 1.1. Crea un nuevo proyecto java, puedes llamarlo practica_09, crea un paquete fuente del mismo nombre.
- 1.2. Copia las clases 'Conexión' y 'ManejoDatos' del proyecto practica_08 a la práctica presente con refactorización ya que se aquí se van a modificar.
- 1.3. En la carpeta 'Libraries' del proyecto, agrega desde las bibliotecas del entorno del IDE de NetBeans las bibliotecas correspondientes del 'Derby'.
- 1.4. En el paquete fuente, crea una clase llamada 'ManipulaDatos' que implemente la interfaz JFrame

2. Declaración de atributos y contenedores principales

2.1. Declara los atributos básicos de los datos a capturar

```
// Elementos para leer los datos
//*****
private JTextField nombre;
private JSpinner fechaNac;
private JSpinner estatura;
private JRadioButton hombre;
private JRadioButton mujer;
private JTextField peso;
private JTextField cintura;
private JTextField cadera;
private JComboBox actividad;
//*****
```

2.2. Declara los atributos para control de las acciones

```
private JButton insertaPer; // Activa la accion para registrar una persona
private JButton limpiar; // Activa borrar valores de los campos
private JButton eliminar; // Activa Elimina registro persona
private JButton agregaMed; //Toma datos de persona para agrega medidas
private JButton insertaMed; //Inserta registro a la talaba mediciones
private JButton limpiaMed; // Limpia los campos de mediciones
private JButton eliminaMed; //Elimina registro de mediciones de la persona seleccionada
```

2.3. Declara los atributos que permiten acceder y visualizar los datos de la base de datos

```
private ManejoDatos manejoDatos; //
private ModeloTablaPersona modeloTablaPersona;
private JTable tablaPersona;
private ModeloTablaMedidas modeloTablaMedidas;
private JTable tablaMedidas;
private int idPerSel; // id de persona seleccionada
private JLabel nombrePerSel; //Despliega el nombre de persona seleccionada;
```

2.4. Declaración de los datos que permiten controlar la edición y validación de algunos datos.

```
private final int MIN_EDAD=20; // Edad minima
private final int MAX_EDAD=65; // Edad Maxima
private final double MIN_ESTATURA=1.40; // Estatura Mínima
private final double MAX_ESTATURA=1.95; // Estatura máxima
private final String DA_NOMBRE = "Da tu nombre"; // título predeagregaMeddo en el campo nombre
private final String DR_PESO = "tu peso?"; // título predeagregaMeddo en el campo peso
private boolean validaNombre = false; // Indicacion de validacion de nombre
```

2.5. Crea un constructor y dentro de él especifica los contenedores principales

```
Container base= getContentPane();
base.setLayout(new BorderLayout());
JTabbedPane panelPrincipal= new JTabbedPane();
JPanel panelPersona = new JPanel();
JPanel panelMedidas = new JPanel();
base.add(panelPrincipal);
panelPrincipal.add(panelPersona);
panelPrincipal.add(panelMedidas);
panelPrincipal.setTitleAt(0,"Personas");
panelPrincipal.setTitleAt(1,"Medidas");
```

- 2.6. También dentro del constructor especifica los elementos que formaran el panel principal de persona y las llamadas a métodos que regresan los demás paneles a integran el conjunto de la aplicación

```
JPanel datosN = new JPanel();
JPanel datosC = new JPanel();
JLabel titulo = new JLabel("Manipulación de datos de personas");
titulo.setHorizontalAlignment(SwingConstants.CENTER);
titulo.setFont(new Font("Serif", Font.BOLD, 18));
datosN.add(titulo);
titulo.setBorder(BorderFactory.createEmptyBorder(10,0,10,0));
datosC.add(formarCamposPersona(), BorderLayout.WEST);
panelPersona.add(titulo, BorderLayout.NORTH);
panelPersona.add(datosC, BorderLayout.CENTER);
panelPersona.add(formarPanelResultados(), BorderLayout.SOUTH);
panelMedidas.add(formaCamposMediciones());
```

3. Panel de personas

- 3.1. Contenido del método 'formarCamposPersona()', regresa un panel conteniendo los campos de personas a introducir. Puedes usar el código del diseño de la práctica tres, haciendo una adaptación para llegar a formar esto:

- 3.2. Contenido del método formarPanelResultados(), regresa un panel conteniendo los controles para manipular los registros de las personas y la tabla donde se despliegan sus registros. También puedes usar parte del código de la práctica tres, haciendo las adecuaciones para obtener lo siguiente (La manera de generar la tabla GUI se verá en los pasos siguientes):

Agregar Registro		Inicia valores	Elimina Reg. Seleccionado	Agregar Medidas	
No. Idem	Nombre	Fecha Nac.			Sexo
1	Pablo Hector	1990-01-08			HOMBRE
2	Maria Callas	1970-11-05			MUJER
3	Carlos Fuentes	1950-08-20			HOMBRE
4	Yalitza Aparicio	1980-01-15			MUJER
6	Abel	1988-01-12			MUJER
10	Diego Velasco	2019-01-12			HOMBRE
12	Nidia Alonso	1999-01-12			MUJER
13	Isidro Salavert	1999-01-12			HOMBRE

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

3.3. [Agregar Registros]. Inserción de datos de persona en la tabla correspondiente de la base de datos.

La acción que debe ejecutar al dar clic, dentro de su método actionPerformed se muestra a continuación:

```
private class AdminAccion implements ActionListener
{ // Clase utilizado para administración eventos de acción
    @Override
    public void actionPerformed(ActionEvent ae) {
        switch(ae.getActionCommand())
        {
            case "Insertar" :
                if(validarNombre (l nombre.getText())){
                    if(!validarSexo (l sexo.getText())){
                        return;
                    }
                    SimpleDateFormat ff = new SimpleDateFormat("YYYY-MM-dd");
                    String vSexo=ff.format(new Date());
                    // del campo que lee el dato 'sexo'
                    String datos= "INSERT INTO ROOT.PERSONA"
                    + " (" + nombre.getText() + ", " + ff.format(new Date()) + ", " + vSexo + ")";
                    manejoDatos.actualizaDatos(datos);
                    cargaDatosPersona();
                    //Escribe el código para hacer que se actualicen los datos de
                    //la tabla personas
                }
            }
        }
    }
}
```

3.4. El método actualizaDatos(). Tienes que crearlo dentro de la clase ManejoDatos

```
public boolean actualizaDatos(String sql)
{ // Inserta, actualiza o elimina
  boolean res=false;
  try{
      java.sql.Statement st= conexion.createStatement();
      st.executeUpdate(sql);
      res = true;
      System.out.println("Se insertó ");
  }
  catch(Exception e)
  {
      System.err.println("Error al insertar/ACTUALIZAR");
  }
  return res;
}
```

3.5. Método 'cargaDatosPersona()' se encarga de realizar una consulta a la tabla 'PERSONA' mediante el método y el resultado de ésta lo carga a un objeto del modelo la base de datos, como se indica

```
public void cargaDatosPersona()
{ String consultaPersona = // Crea la consulta SQL
  modeloTablaPersona.setDatos(manejoDatos.conexionConsultaPersona(consultaPersona));
}
```

En la clase ManipulaDatos, ya se encuentra definido el atributo modeloTablaPersona utilizado para la tabla GUI que visualiza los datos como se ve enseguida.

Para visualizar los datos de persona se extraen los valores de la tabla Persona y se cargan al modelo de la 'tablaPersona'.

3.6. La creación del modelo para 'tablaPersona'. Crea una clase que se herede de Abstract derivada de 'AbstractTableModel' llámala 'ModeloTablaPersona'. La cual contenga como atributos:

```
private List<Object>[] datos;
private String encabezado[] = new String[] {
    "No.Iden", "Nombre", "Fecha Nac.", "Sexo"};
private Class tipos[] = new Class[] {
    String.class, String.class, String.class, String.class};
```

3.6.1. Sobre carga los métodos:

```

@Override
public Class getColumnClass(int c) { return tipos[c]; }
@Override
public int getRowCount() { return dato.size(); }
@Override
public int getColumnCount() { return tipos.length; }
@Override
public Object getValueAt(int r, int c) { return dato.get(r)[c]; }
@Override
public String getColumnHeader(int col)
{
    return encabezado[col];
}
}

```

3.6.2. Escribe el método que sirva para asignar valores a 'dato':

```

public void setDatos(List <Object []> d)
{
    dato=d;
}

```

3.7. Cargar los valores a la tabla. Al inicio del método formarPanelResultados() se cargan los datos que se visualizarán mediante

```

manejoDatos = new ManejoDatos();
modeloTablaPersona = new ModeloTablaPersona();
cargaDatosPersona();
tablaPersona = new JTable(modeloTablaPersona);

```

3.8. [Inicia Valores]. Hace que los valores de los atributos: nombre, fecha de nacimiento y sexo tomen los valores por omisión. Los cual también está incluido en el código de la práctica tres.

3.9. [Elimina Reg. Seleccionado]. Elimina de la tabla 'Persona' de la base de datos al registro seleccionado en la tablaPersona(JTable). El código que debe estar en su método a ejecutar se muestra abajo. Complementa el código que se indica en los comentarios.

```

//Acción del botón eliminar el reg
case "bEliminar" :
    String de="";
    if(rs>0){
        idPersona = Integer.parseInt((String)tablaPersona.getValueAt(rs,0));
        nombrePer = // Toma de la tablaPersona el valor del nombre del registro seleccionado
        int confirmado = JOptionPane.showConfirmDialog(null,"¿Eliminas el registro de " + nombrePer + "?");
        if (JOptionPane.OK_OPTION == confirmado)
        {
            de="DELETE FROM ROOT.PERSONA WHERE IDPERSONA="+idPersona;
            // Usa el método manejoDatos.actualizarDatos();
            // vuelve a cargar los datos de la base de datos al modelo
            // Has que la tablaPersona (JTable) muestre los valores actualizados
        }
        else
            //Has que se deseleccione el registro si no se desea eliminarlo
    }
    break;

```

3.10. [AgregarMedidas]. Para agregar los registros de medidas de las personas en su método de acción agrega el siguiente código completando lo que se indica en los comentarios:

```

case "bAgregarMed" :
    if(tablaPersona.getSelectedRow()==0)
    {
        idPerSel = // obten el valor del idPer del renglon seleccionado de la tablaPersona
        String seleccionMedPer = "SELECT * FROM ROOT.MEDICIONES WHERE IDPERSONA="+idPerSel;
        modeloTablaMedidas.setDatos(manejoDatos.conexionConsultaMedicion(seleccionMedPer));
        // Has que se refresquen los valores de la tablaMedidas
        String nps = //obten el valor del nombre de la persona del renglon seleccionado de la tablaPersona
        nombrePerSel.setText("-----" + nps + "-----");
    }
    break;

```

La clase del elemento 'modeloTablaMedidas' y el método 'conexionConsultaMedicion' se ven en la sección del panel de mediciones.

4. Panel de Mediciones

Este panel semejante al de personas está conformado por dos grupos de paneles con sus respectivos métodos que los integran: `formaCamposMediciones()` y `formarPanelMedidas()`

- 4.1. Método `formaCamposMediciones()`. Regresa un panel muy parecido al de la práctica tres, por lo tanto adapta el código para formarlo, solo agrega los otros campos de captura de datos como estatura, cintura y cadera y una etiqueta que muestre el nombre de la persona que fue seleccionada en el ‘panel de persona’, como se muestra a continuación:

- 4.2. Método `formarPanelMedidas()`. Muestra los controles para manipular los datos y la tabla conteniendo los registros de las mediciones de una persona. Aquí se muestran dos situaciones, uno que corresponde al caso que no se haya seleccionado a una persona o cuando ya se ha seleccionado y no tiene ningún registro previo; el otro caso cuando ya cuenta con registros previos

Agregar Registro					
Inicia valores					
Elimina Reg. Seleccionado					
Fecha	Estatura(cm)	Peso(kg)	Cintura(cm)	Cadera(cm)	Tipo de Act.
2018-01-14	160	80.5	89	110	2
2018-05-14	160	80.0	85	105	3
2019-01-14	160	79.0	84	105	3

Ahora hacen falta crear el modelo de tabla de medidas y el método de recuperación de datos para medidas.

- 4.3. Clase ‘`ModeloTablaMedidas`’ sirve para manejar los valores de `tablaMedidas`(tipo `Table`),

Crea una clase semejante a la de `ModeloTablaPersona`. Completa el código abajo mostrado:

```
public class ModeloTablaMedidas extends AbstractTableModel
{
    private List<Object []> dato;
    private String encabezado[] = new String[] {
        // texto de los encabezados de la tabla
    };
    private Class tipos[] = new Class[] {
        String.class, String.class, String.class, String.class, String.class, String.class
    };
    public void setDatos(List<Object []> d)
    {
        dato=d;
    }
}

@Override
public Class getColumnClass(int c) { return tipos[c]; }
@Override
public int getRowCount() { return dato.size(); }
//Completa los otros metodos.....
```


TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

- 4.4. Método 'conexionConsultaMedicion(consulta)'. Regresa el conjunto de valores de la tabla MEDICIONES de la base de datos.

En la clase ManejoDatos crea el método completando los datos indicados en los comentarios

```
public List<Object []> conexionConsultaMedicion(String sql)
{
    //Regresa los registros de las medidas en una lista
    List<Object []> datos= new ArrayList<Object []>();
    DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
    try{
        Statement ps = conexion.createStatement();
        ResultSet rs = ps.executeQuery(sql);
        while(rs.next())
        {
            String dat[] = new String[5];
            //Estructura del registro medicin pasado como cadena
            dat[0]=fecha.format((Date)rs.getDate(2)); //Fecha Registro
            dat[1]= String.valueOf((Integer)rs.getInt(3)); //estatura
            dat[2]= //peso
            dat[3]= // Cintura
            dat[4]= //Cadera
            dat[5]= //Tipo Actividad
            datos.add(dat);
        }
    } catch (Exception e)
    {
        System.err.println("Error al conexion consultar mediciones "+e);
    }
    return datos;
}
```

- 4.5. Método cargaDatosMedidas. Regresa los registros de las mediciones que tiene la persona indicada por su 'idPer'. Este método se llama en tres situaciones: cuando se selecciona una persona en el panel de persona; después de haber registrado una nueva medición y después de eliminar un registro. Completa el código indicado en los comentarios.

```
public void cargaDatosMedidas(int idPer)
{
    //La consulta regresa todos los registros de mediciones de la persona idPer
    String consultaMedidas = //Escribe la consulta SQL que debe hacerse
    modeloTablaMedidas.setDatos(manejoDatos.conexionConsultaMedicion(consultaMedidas));
}
```

- 4.6. [Agrega Registro] Inserta en la tabla MEDICIONES de la base de datos un registro, el código correspondiente en su método de acción de este botón se muestra a continuación. Completa lo que hace falta indicado en los comentarios.

```
case "bInsertarMed" :
{
    if(tablaPersona.getSelectedRow()>0)
    {
        Calendar fecha = Calendar.getInstance();
        SimpleDateFormat ff = new SimpleDateFormat("YYYY-MM-dd");
        Long estatura = ((Double)estatura.getValue()).longValue()*100;
        int idAct= actividad.getSelectedIndex()+1;
        if(peso.getText().isEmpty() && cintura.getText().isEmpty() && cadera.getText().isEmpty())
        {
            Integer peso = Integer.parseInt(peso.getText());
            Integer cintura = Integer.parseInt(cintura.getText());
            Integer cadera = Integer.parseInt(cadera.getText());
            System.out.println("Insercion de "+idPerSel+" Estatura "+estatura);
            String medidas = "INSERT INTO ROOT.MEDICIONES"
            + "(fecha,estatura,peso,cintura,cadera,idtipoact,idpersona) values('"+
            // Completa el código respecto a los valores de los campos que deben insertarse
            //
            manejoDatos.actualizaDatos(medidas);
            cargaDatosMedidas(idPerSel);
            //Refresca los valores de tablaPersona
        }
    }
    break;
}
```

- 4.7. Te corresponde hacer el código para limpiar los campos y eliminar

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

5. Realiza una clase prueba y construye un plan de pruebas, guarda los resultados de las ejecuciones para diferentes casos

Sugerencias didácticas

Documentar el código

Llevar una bitácora de seguimiento.

Crear plan de pruebas

Trabajo en equipos

Compartir experiencias de esta práctica mediante un foro de tu grupo

Reporte del estudiante

Entrega del código de la aplicación y la base de datos implementada con datos

Entrega de imágenes de pruebas de ejecución

Entrega de la bitácora de seguimiento

Bibliografía

1. <https://netbeans.org/kb/docs/ide/mysql.html#configuring>. Último acceso 10/01/19
2. <https://docs.oracle.com/javase/tutorial/index.html> Último acceso 10/01/19
3. https://www.tamps.cinvestav.mx/~visosa/clases/bd/Uso_JavaJDBC.pdf Último acceso 10/01/19
4. Deitel y Deitel. Como programar en Java. Prentice Hall. Séptima Edición, 2008