EXAMEN	DAW 1	ASIGNATURA: PRG
FICHERO: exaprofinal5	EVALUACIÓN: FINAL	MODELO:A

NOMBRE		
FECHA		NOTA:

El examen durará 1:30 minutos. Hay que presentar el dni. No se puede salir hasta pasados 15 minutos. La nota saldrá en el aula virtual. La respuesta a las preguntas se realizará en las hojas que se entregan con el logo del ceed. Numerar las hojas ejemplo 1/3, 2/3, 3/3. Se devolverá el enunciado y todas las hojas entregadas grapadas. En el sobre donde se entrega el examen grapado, se deberá poner vuestro dni y nombre.

Se va a realizar un programa en java que tiene como ventana principal un formulario con datos de una examen de un grupo.

Los datos del formulario se han cargado de MYSQL, cuyos datos se han introducidos en los ficheros previamente nada más entrar en la aplicación. Los datos del formulario se grabar en un FICHERO pulsando el botón grabar. El botón salir permite salir de la aplicación.

El fichero **Main.java** se encuentra en el paquete controlador el cual tendrá el main:

```
package controlador;
import datos.Modelo;
import java.io.IOException;
import vista.VistaExamen;

public class Main {
    public static void main(String[] args) throws IOException {
        Modelo modelo = null;
        VistaExamen vista = new VistaExamen();
        Controlador controlador = new Controlador(modelo, vista);
    }
}
```

El fichero **Modelo.java** se encuentra en el paquete datos:

```
package datos;

import modelo.Curso;
import modelo.Examen;

public interface Modelo {

   // Graba en objeto examen en un fichero/mysql
   public void createExamen(Examen examen);

   // Graba en objeto curso en un fichero/mysql
```

```
   public void createCurso(Curso curso);

   // Devuelve el objeto Examen que tengo el valor id
   public Examen readExamen(String id);

   // Devuelve el objeto Curso que tengo el valor id
   public Curso readCurso(String id);

   // Borrar el contenido de los ficheros/tablas
   public void inicializa();
}
```

La aplicación deberá cumplir:

- Diseño Modelo Vista Controlador.
- Encapsulación de los datos.
- Utilizaremos interfaces para el acceso a los datos.

Figura 1. VistaExamen. Al entrar

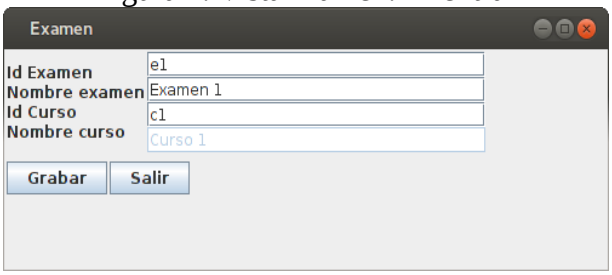
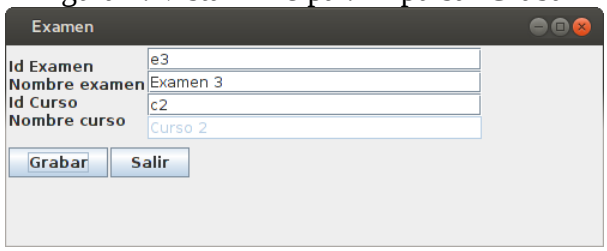


Figura 2. Vista Principal. Al pulsar Grabar



1. MODELO. 1 puntos
Dentro del paquete modelo.

Tendremos la clase Examen y la clase Curso. Un examen tendrá un curso y un curso tendrá muchos exámenes.

Realizar la clase **Examen.java** en java dentro del paquete modelo, que representa la relación en la que una examen y persona. Ver Figura 3. No poner las funciones set y get.

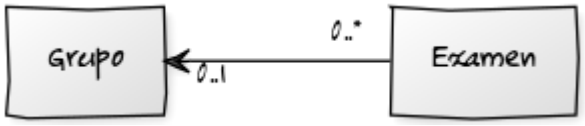


Figura 3. Diagrama de Clases

Los atributos son:

Clase Examen: id, nombre
Clase Grujo: id, nombre.

2. VISTA EXAMEN. 1 puntos.
Dentro del paquete vista.
Relleanr el código que falta en la clase en java llamada **vistaexamen.java** que mediante el interfaces gráfica de usuario deberá visualizar como se muestra en la imagen de las Figuras 1 y 2.

```
package vista;

import java.awt.FlowLayout;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class VistaExamen extends JFrame {

// Rellenar
}
```

3. CONTROLADOR. 3 puntos
Dentro del paquete controlador.
Crear la clase **Controlador.java** el cual es llamado por el main.java. Esta clase implentará el interface ActionListener. Rellenar el código que falta.

```
package controlador;
import datos.Modelo;
import datos.ModeloFichero;
import datos.ModeloMysql;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import modelo.Curso;
import modelo.Examen;
import vista.VistaExamen;

class Controlador implements ActionListener {

   Modelo modelo;
   VistaExamen vista;
   Examen examen;
   Curso curso;

   final static String GRABAR = "GRABAR";
   final static String SALIR = "SALIR";

   Controlador(Modelo modelo_, VistaExamen vista_) throws
IOException {
```

```
      modelo = modelo_;
      vista = vista_;

      vista = new VistaExamen();
      vista.setVisible(true);

      fuentededatos();

      modelo.inicializa();
      inicializaModelo();

      examen = ExamenInicial();
      mostrarExamen(examen);
      configuraBotones();

   }

   private Examen ExamenInicial() throws IOException {

      examen = modelo.readExamen("e1");
      String idcurso = examen.getCurso().getId();
      curso = modelo.readCurso(idcurso);
      examen.setCurso(curso);

      return examen;
   }

   private void configuraBotones() {
      /* 3.1 Rellenar. 0.5puntos
Inicializar los botones GRABAR de  y SALIR donde se define la
etiqueta asociada al boton y quien es la clase que escucha los
eventos.
       */

   }

   @Override
   public void actionPerformed(ActionEvent e) {
      /* 3.2 Rellenar. 0.5puntos
       Si se pulsa SALIR hacer salir(), si se pulsa GRABR
       hacer grabar()
       */

   }

   private void salir() {
      System.exit(0);
      vista.dispose();
   }
```

```
   private void grabar() {

      destinodedatos();

      inicializaModelo();
      examen = getExamen();
      modelo.createExamen(examen);
      mostrarCurso(examen);
   }

   private void mostrarExamen(Examen examen) {
      /* 3.3 Rellenar. 0.5puntos
       Muestra en  el formulario VistaExamen el examen
       */


   }

   private Examen getExamen() {

      Examen examen = new Examen();
      Curso curso = new Curso();

      String id = vista.getjtf1().getText();
      String nombre = vista.getjtf2().getText();
      String idcurso = vista.getjtf3().getText();

      curso = modelo.readCurso(idcurso);
      examen = new Examen(id, nombre, curso);
      return examen;
   }

   private void mostrarCurso(Examen examen) {
      vista.getjtf4().setText(examen.getCurso().getNombre());
   }

   private void inicializaModelo() {

      /* 3.4 Rellenar. 0.5puntos
       Crear los objetos examen y curso que se indican y grabarlos
en la fuente de datos.

       En fichero  examen.cvs sería:
       e1;Examen 1;c1
       e2;Examen 2;c2

       En fichero  curso.cvs sería:
       c1;Curso 1
       c2;Curso 2

       */
```

```
   }

   private void fuentededatos() {
      /* 3.5 Rellenar. 0.5puntos
      Definir la fuente de datos sea fichero o mysql
      utilizando el modelo
      */

      modelo = new ModeloFichero();
   }

   private void destinodedatos() {
      /* 3.6 Rellenar. 0.5puntos
      Definir el destino de los datos del formulario VistaExamen
      sea fichero o mysql utilizando el modelo
      */

      modelo = new ModeloMysql();
   }

}
```

4. FICHERO 2 puntos

En el paquete vistabd.

Rellenar lo que falta de la clase que llamaremos **ModeloFichero.java** que implementará las funciones del interface Bd.java, que gestiona la base de datos orientada a objetos db4o.

```
package datos;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.StringTokenizer;
import modelo.Curso;
import modelo.Examen;

/**
 *
 * @author paco
 */
public class ModeloFichero implements Modelo {

   private int id = 0;
   private static String fexamen = "examen.cvs";
   private static String fcurso = "curso.cvs";
```

```
   @Override
   public void createCurso(Curso curso) {
      /* Rellenar.  0.5 puntos
       Graba el objeto curso en el fichero fcurso
       */



   @Override
   public void createExamen(Examen examen) {
   … // No hacer
   }

   @Override
   public Examen readExamen(String id) {

      /* Rellenar.  0.5 puntos
       Lee el objeto examen del fichero fexamen
       Utiliza la función extraeExamen
       */


   @Override
   public Curso readCurso(String idcurso) {
      // No hacerlo
   }

   private Examen extraeExamen(String linea) {
      /* Rellenar.  0.5 puntos
       Extrae el objeto examen de la linea separada por comas
       Utilizado por la función readExamen de ficheros.
       */
   }

   private Curso extraeCurso(String linea) {
      // No hacerlo
   }

   @Override
   public void inicializa() {
      /* Rellenar.  0.5 puntos
       Borra los ficheros
       */
   }

}
```

6. MYSQL 2 punto.

Dentro del paquete Datos. Rellenar los que falta de la clase en java llamada **Mysql.java** que

implementa el interface Modelo.java, que utilizar la base de datos mysql.

```
package datos;
import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import modelo.Curso;
import modelo.Examen;

public class ModeloMysql implements Modelo {

  private final String BD = "exaprgfinal5";
  private final String USER = "alumno";
  private final String PASS = "alumno";
  private Connection con = null;
  private Statement st = null;
  private ResultSet rs = null;

  public void desconectar() {
    try {
      System.out.println("BDR Mysql Connexión cerrada");
      con.close();
    } catch (SQLException ex) {

    }
  }

  public void conectar() {

    try {
      String driver = "com.mysql.jdbc.Driver";
      Class.forName(driver).newInstance();
      String jdbcUrl = "jdbc:mysql://localhost:3306/" + BD;
      con = (Connection) DriverManager.getConnection(jdbcUrl, USER, PASS);
      System.out.println("Conexion establecida con la Base de datos...");

    } catch (SQLException se) {
      se.printStackTrace();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

  @Override
  public void createExamen(Examen examen) {
    /* Rellenar: 1 puntos
```

```
    Graba el examen en la tabla
    */
}


@Override
public Examen readExamen(String id_) {
   /* Rellenar: 1 puntos
    Devolver el examen que se lee de la tabla
    */
}

@Override
public void createCurso(Curso curso) {
   // No hacerlo
}


@Override
public Curso readCurso(String idcurso) {
   // No hacerlo
}

@Override
public void inicializa() {
   /* Rellenar: 1 puntos
    Borra las tablas
    */
   }
```

NOTA: cvs: git clone https://bitbucket.org/1415ceed1/exaprgfinal5