

EXAMEN	DAW 1	ASIGNATURA: PRG
FICHERO: exaprofinal4	EVALUACIÓN: FINAL	MODELO:4

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 control de acceso. A los tres accesos incorrectos la aplicación terminará automáticamente. Si el usuario y contraseña es correcto se abrirá la ventana VistaPersona, que mostrará los datos del usuario.

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

```
package controlador;
import java.util.ArrayList;
import modelo.Persona;
import vistapantalla.VistaPrincipal;
public class Main {
    public static void main(String[] args) {
        ArrayList<Persona> modelo = new ArrayList<>();
        VistaPrincipal vista = new VistaPrincipal();
        Controlador controlador = new Controlador(vista, modelo);
    }
}
```

El fichero **Bd.java** define el siguiente interface:

```
package vistadatos;
import modelo.Persona;
public interface Bd {
    public void open();
    public void close();
    public void escribir(Persona persona);
    public void limpiar();
}
```

La aplicación deberá cumplir:

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

Figura 1. Vista Principal.

Figura 2. Vista Principal. Con error

Figura 3. Vista Persona. Persona: p1

Figura 4. Vista Persona. Persona: p2

1. MODELO. 1 puntos

Dentro del paquete modelo.

Tendremos la clase Persona y la clase Departamento. Una persona tendrá un departamento y un departamento tendrá muchas personas.

Realizar la clase **Persona.java** en java dentro del paquete modelo, que representa la relación en la que una factura tiene un cliente y un cliente muchas facturas. Ver Figura 5. No poner las funciones set y get.

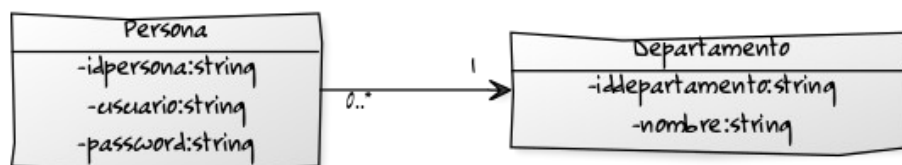


Figura 5. Diagrama de Clases

Los atributos son:

Clase Persona: idpersona, usuario, password.

Clase Departamento: iddepartamento, nombre.

2. VISTA PRINCIPAL. 1 puntos.

Dentro del paquete vista.

Rellenar el código que falta en la clase en java llamada **vistaprincipal.java** que mediante el interfaces gráfica de usuario deberá visualizar como se muestra en la imagen de las Figuras 1 y 2.

```
package vistapantalla;
```

```

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

public class VistaPrincipal extends JFrame {
// Rellenar
}

```

3. CONTROLADOR PRINCIPAL. 3 puntos

Dentro del paquete controlador.

Crear la clase **ControladorPrincipal.java** el cual es llamado por el main.java. Esta clase implementará el interface ActionListener. Rellenar el código que falta.

```

package controlador;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Iterator;
import modelo.Departamento;
import modelo.Persona;
import vistadatos.Bd;
import vistadatos.Db4o;
import vistapantalla.VistaPrincipal;

class ControladorPrincipal implements ActionListener {
    private VistaPrincipal vista;
    private ArrayList<Persona> personas = new ArrayList<>();
    private int intentos = 0;
    private ControladorPersona ControladorPersona;

    ControladorPrincipal(VistaPrincipal vista, ArrayList<Persona>
personas) {
        this.vista = vista;
        this.personas = personas;

        inicializabotones();
        inicializaobjetos();
        grabarPersonas();
        visualizar();
    }

    public void visualizar() {

```

```

        vista.setVisible(true);
    }

    public void inicializabotones() {
// Rellenar. 0.5puntos
// Inicializar los botones de Entrar y Salir de vista
    }

    public void actionPerformed(ActionEvent ae) {
// Rellenar. 0.5puntos
// Si se pulsa el botón de la vista principal entrar ejecutar
    entrar(), si se pulsar el boton salir ejecutar salir,
    }

    public void entrar() {
// Rellenar . 0.5puntos
// Si persona existe ver vista persona sino mostrar mensaje de error. Si la persona no existe, entonces
    si el numero de errores es >3 salirse de la aplicación sino aumentar el contador de intentos y
    mostrar mensaje de error..
    }

    public void salir() {
        vista.dispose();
    }

    public void inicializaobjetos() {
// Rellenar . 0.5puntos
// Crear los siguientes objetos:
// Objeto persona 1: Id:p1, Usuario: admin, password: 123, Grupo:d1
// Objeto persona 2: Id:p2, Usuario: paco, password: 123, Grupo:d1
// Objeto departament 1: id: d1, nombre: Inf
    }

    public Persona buscarPersona(Persona personabuscada) {
// Rellenar . 0.5puntos
// Buscar la persona del parámetro en el arraylist personas o null sino la encuentra.
// Función que se utilizar en la función entrar().
    }

    public void grabarPersonas() {
// Rellenar . 0.5puntos
// Recorre el arrayList y los graba en la base de datos
// orientada a objetos. Utiliza el interface BD.
    }
}

```

4. CONTROLADOR PERSONA. 2 puntos

El controlador `ControladorPersona.java` será llamado por controlador principal y cuando el usuario y la contraseña sea correcto. Ver Figuras 3 y 4. Rellenar las partes que falta del código que viene indicado como Rellenar en negrita.

```
package controlador;

import modelo.Persona;
import vistadatos.Bd;
import vistadatos.Mysql;
import vistapantalla.VistaPersona;
import vistapantalla.VistaPrincipal;

class ControladorPersona {

    private Persona persona;
    private VistaPrincipal vista;
    private VistaPersona vistapersona;
    private ControladorPrincipal contralador;

    public ControladorPersona(Persona persona_, VistaPrincipal vista_,
ControladorPrincipal controlador_) {

        vistapersona = new VistaPersona();
        persona = persona_;
        vista = vista_;
        contralador = controlador_;
        inicializabotones();
        rellenapersona();
    }

    public void SalirVistaPersona() {
        vista.setVisible(true);
        vistapersona.setVisible(false);
    }

    public void visualizar() {
        vista.setVisible(false);
        vistapersona.setVisible(true);
    }

    private void inicializabotones() {
// Se inilizar los botones. No rellenar.
    }

    private void rellenapersona() {
```

```
// Rellenar
// Se rellenaran los datos de la persona en el formulario
// VistaPersona.
// Limpiar la base de datos mysql

}

void GrabarVistaPersona() {
// Rellenar
// Limpiar la base de datos mysql
// Escribir insertar la persona en la base de datos que está en el formulario.
}
}
```

5. DB4O 1 puntos

En el paquete vistabd.

Rellenar lo que falta de la clase que llamaremos **Db4o.java** que implementará las funciones del interface Bd.java, que gestiona la base de datos orientada a objetos db4o.

```
package vistadatos;

import com.db4o.Db4oEmbedded;
import com.db4o.ObjectContainer;
import java.io.File;
import modelo.Persona;

public class Db4o implements Bd {

    private String Fichero = "personas.db4o";
    private ObjectContainer bdoo;

    public void open() {
        limpiar();
        bdoo = Db4oEmbedded.openFile(Db4oEmbedded.newConfiguration(),
Fichero);
    }

    public void close() {
        bdoo.close();
    }

    public void escribir(Persona persona) {
// Rellenar
// Graba en la bd.
    }

    public void limpiar() {
// Rellenar

```

```
// Borrar el fichero
}
}
```

6. MYSQL 2 punto.

Dentro del paquete vistabd.

Rellenar lo que falta de la clase en java llamada **Mysql.java** que implementa el interface Bd.java, que utilizar la base de datos mysql.

```
package vistadatos;

import constantes.Constantesbd;
import java.sql.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.Persona;
public class Mysql implements Bd {

    private static final Constantesbd constantes = new Constantesbd();
    private Connection con = null;
    private Statement st = null;
    private ResultSet rs = null;

    @Override
    public void open() {
        try {

            Class.forName(constantes.DRIVER).newInstance();
            con = DriverManager.getConnection(
                constantes.JDBCURL,
                constantes.USU,
                constantes.PASS);
        }
        catch (ClassNotFoundException ex) {}
        catch (SQLException ex) {}
        catch (InstantiationException ex) {}
        catch (IllegalAccessException ex) {}
    }

    public void close() {
        try {
            con.close();
        }
    }
}
```

```
    } catch (SQLException ex) {}  
}  
  
public void limpiar() {  
// Rellenar  
// Borra el contenido de las tablas en mysql.  
}  
  
@Override  
public void escribir(Persona persona) {  
// Rellenar  
// Inserta la persona en la bd  
}
```