

# Ejercicio colaborativo



**UNIVERSIDAD DE  
CASTILLA-LA MANCHA**

# ¿Cómo hacer los ejercicios?

1)Coger el material del ejercicio X.

Para cada tarea:

2) **ANOTAR EL TIEMPO DE COMIENZO** (HH:MM:SS:)

3) Leer la tarea de modificación “i”.

4) Leer la documentación del sistema.

NO LEAIS EL  
MATERIAL TODAVÍA

ESTO SIGNIFICA QUE NO  
PODEIS OJEAR TODAS LAS  
TAREAS

SOLO SE PUEDE LEER LA DOCUMENTACIÓN DEL  
SISTEMA (CÓDIGO Y MODELOS) MIENTRAS SE ESTÁ  
HACIENDO UNA TAREA Y **NO ANTES!**

5) Resolver la tarea “i” usando los formularios.

6) **ANOTAR EL TIEMPO DE FINALIZACIÓN**.

7) Indica qué información has utilizado para resolver dicha tarea

AÑADID TANTAS FILAS  
COMO SEAN NECESARIAS

# TAREA 1

Anotar el tiempo de comienzo \_\_\_\_\_ : \_\_\_\_\_ :

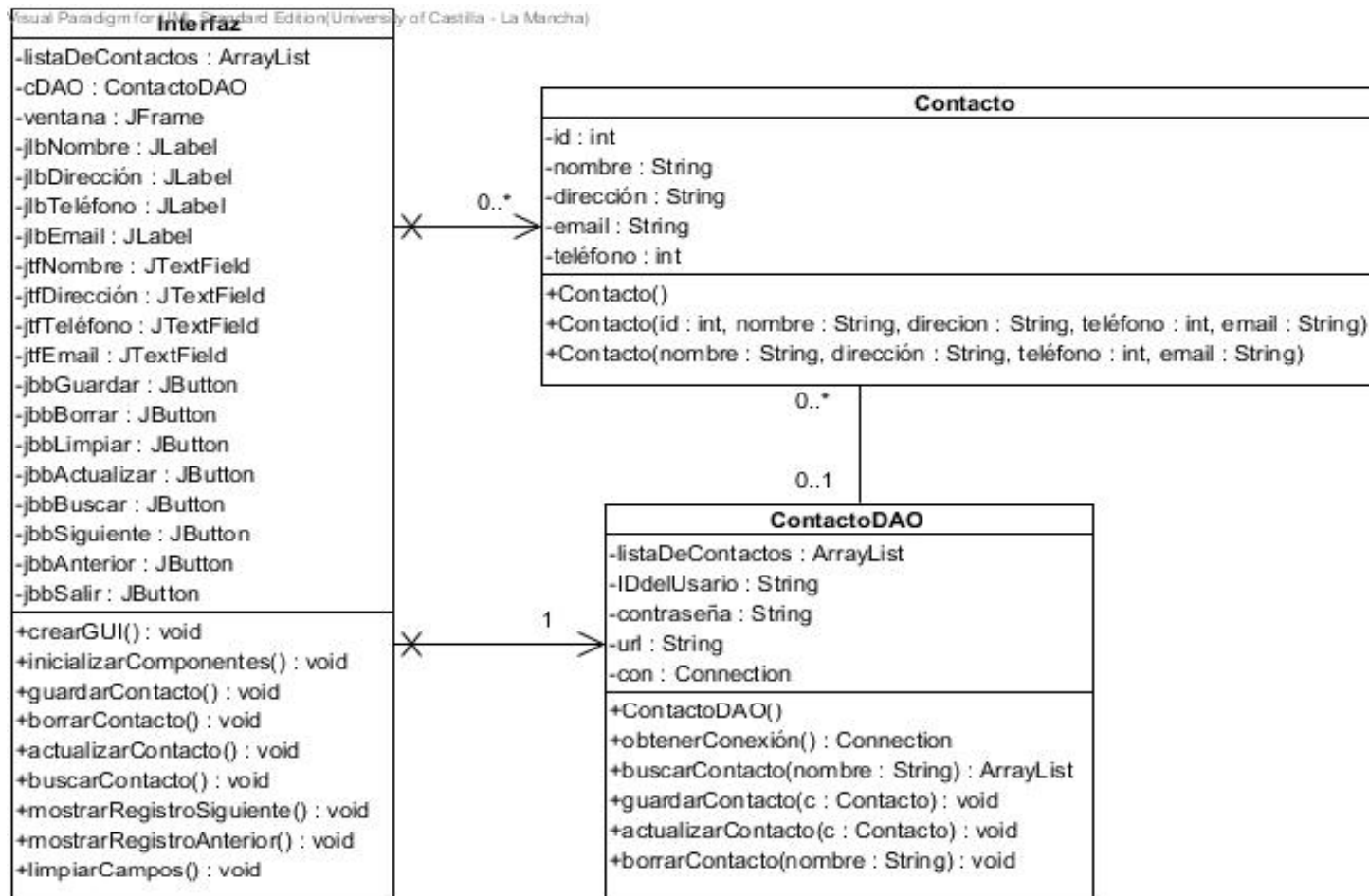
Almacenar la fecha de cuando se registra un contacto.

Anotar el tiempo de finalización \_\_\_\_\_ : \_\_\_\_\_ :

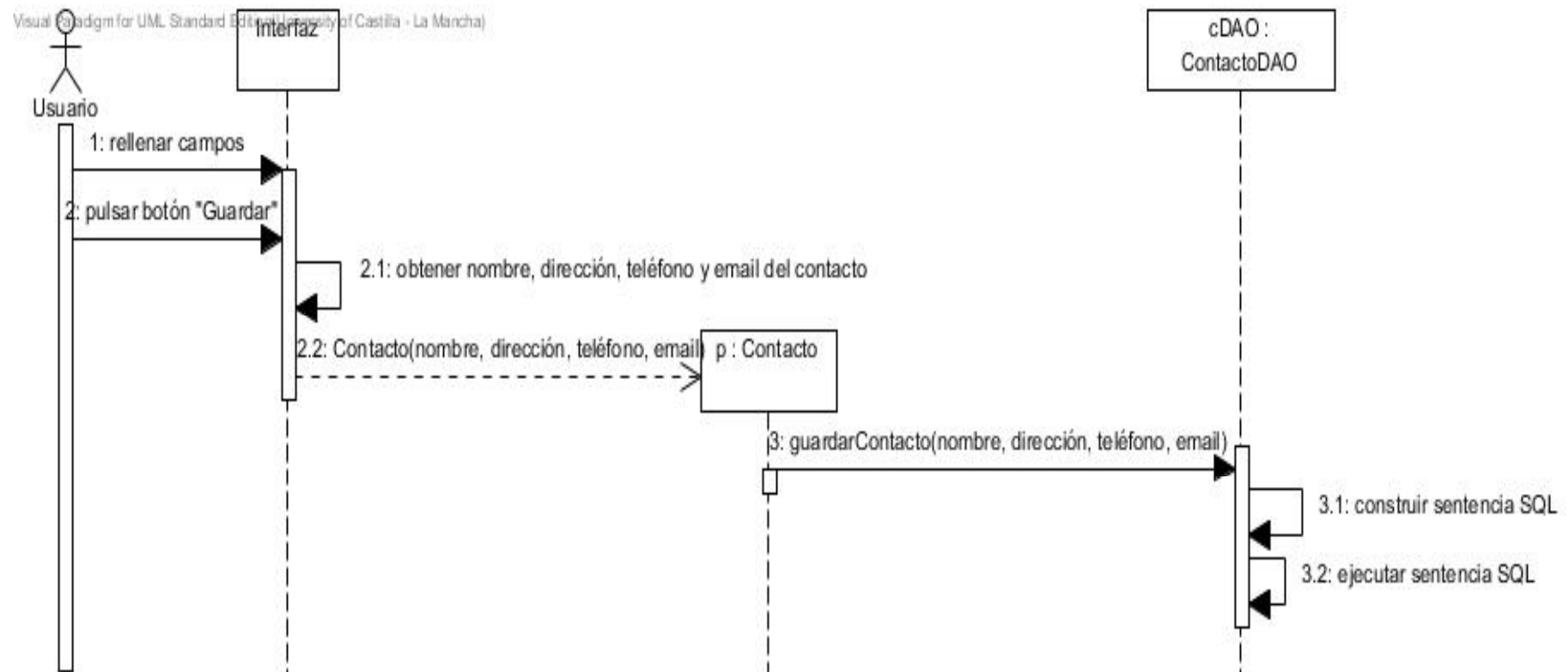
¿Qué artefacto has utilizado para realizar esta tarea?

- ☐ Código fuente
- ☐ Diagrama de clase
- ☐ Diagrama de secuencia

# DIAGRAMAS (I)



# DIAGRAMAS (II)



# CÓDIGO (I)

Interfaz.java

```
1 import java.awt.Container;
2 import java.awt.GridBagConstraints;
3 import java.awt.GridBagLayout;
4 import java.awt.Insets;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.util.ArrayList;
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JTextField;
13
14 public class Interfaz implements ActionListener{
15
16     ArrayList listaDeContactos;
17     ContactoDAO cDAO;
18     JFrame ventana;
19     JLabel jlbNombre, jlbDireccion, jlbTelefono, jlbEmail;
20     JTextField jtfNombre, jtfDireccion, jtfTelefono, jtfEmail;
21     JButton jbbGuardar, jbnBorrar, jbnLimpiar, jbnActualizar, jbnBuscar,
        jbnSiguiente, jbnAnterior, jbnSalir;
22     String nombre, direccion, email;
23     int telefono;
24     int numeroDeRegistro; // Se usa para navegar utilizando los botones >> y <<
25     Container panel;
26
27     public static void main(String args[]){
28         new Interfaz();
29     }
```

# CÓDIGO (II)

Contacto.java

```
1 public class Contacto{
2     private String nombre,direccion,email;
3     private int id, telefono;
4
5     public Contacto(){
6         nombre = "";
7         direccion = "";
8         email = "";
9         id = 0;
10        telefono = 0;
11    }
12
13    public Contacto(int id, String n, String d, int t, String e){
14        this.id = id;
15        this.nombre = n;
16        this.direccion = d;
17        this.telefono = t;
18        this.email = e;
19    }
20
21    public Contacto(String n, String d, int t, String e){
22        this.nombre = n;
23        this.direccion = d;
24        this.telefono = t;
25        this.email = e;
26    }
27
28    public void setId(int i){
29        id = i;
30    }
31    --
```

# CÓDIGO (II)

ContactoDAO.java

```
1 import java.util.*;
2 import java.sql.*;
3
4 public class ContactoDAO{
5
6     private ArrayList listaDeContactos;
7     private String IDdelUsuario = "scott";
8     private String contraseña = "tiger";
9     static String url = "jdbc:odbc:bob";
10    private Connection con;
11
12    public ContactoDAO(){
13        listaDeContactos = new ArrayList();
14        obtenerConexion(); //Crea una conexión con la Base de datos
15    }
16
17    public Connection obtenerConexion(){
18        try {
19            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
20
21        } catch (java.lang.ClassNotFoundException e) {
22            System.err.print("ClassNotFoundException: ");
23            System.err.println(e.getMessage());
24        }
25        try {
26            con = DriverManager.getConnection(url, IDdelUsuario, contraseña);
27        } catch (SQLException ex) {
28            System.err.println("SQLException: " + ex.getMessage());
29        }
30
31        return con;
32    }
```



Si es una clase								
Tarea	Acción (A=Añadir B=Borrar M=Modificar)	Nombre	Métodos (excluir setters y getters)				Atributos (tipo:nombre)	Detalles/ Información adicional
			Nombre	Tipo de retorno	Parámetros (tipo:nombre)	Código JAVA (o pseudocódigo)		

Si es un método							
Tarea	Acción (A=Añadir B=Borrar M=Modificar)	En la clase	Nombre	Tipo de retorno	Parámetros (tipo:nombre)	Código JAVA (o pseudocódigo)	Detalles/Info. adicional

Si es un atributo					
Tarea	Acción (A=Añadir B=Borrar M=Modificar)	Nombre	En la clase	Tipo de datos	Detalles/Información adicional

Si es un trozo de código (una(s) línea(s))					
Tarea	Acción (A=Añadir B=Borrar M=Modificar)	En la clase	En la línea	Código JAVA (o pseudocódigo)	Detalles/Información adicional

# TAREA 2

**Anotar el tiempo de comienzo** \_\_\_\_\_ : \_\_\_\_\_ :

**A partir de ahora, no queremos almacenar los emails de los contactos.**

**Anotar el tiempo de finalización** \_\_\_\_\_ : \_\_\_\_\_ :

**¿Qué artefacto has utilizado para realizar esta tarea?**

- ☐ Código fuente
- ☐ Diagrama de clase
- ☐ Diagrama de secuencia