

M03. Programació bàsica

**UF4 - UF5 - UF6
Programació orientada a objectes**

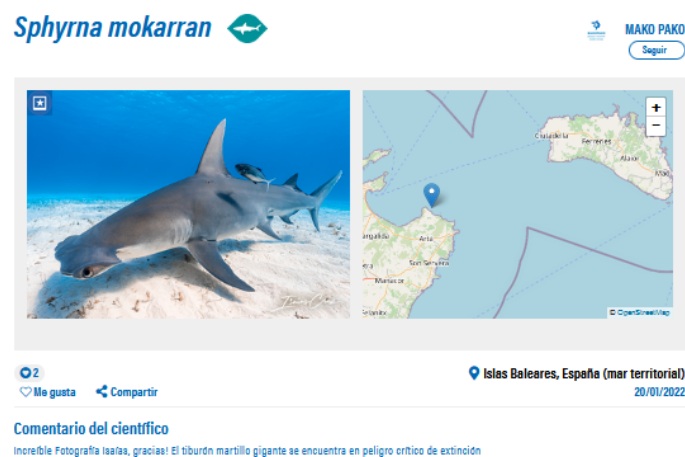
Aplicació per gestionar les observacions marines

Projecte

Catsharks.org és una organització de divulgació científica marina, especialitzada en tiburons. Aquest any han iniciat un projecte, *Observadores del mar*, on els usuaris que han albirat un tauró o una rajada, podran indicar-ho mitjançant un formulari:



Les dades que es volen enregistrar mitjançant l'aplicació (en una base de dades) són les que podem trobar en la seva web (tal i com es veu en la fitxa¹). A més de demanar-nos que l'usuari pugui indicar la mida de l'exemplar, volen preparar l'aplicació per a diferents espècies (posidònia, gavia, nudibrànquis,...), de manera que s'haurà de preparar amb camps comuns i camps específics:



¹ La fitxa amb més detall es pot trobar en aquest enllaç: <https://www.observadoresdelmar.es/Observations/18/126>
La fitxa d'un altre animal es pot trobar en aquest enllaç: <https://www.observadoresdelmar.es/Observations/6/18522>

Especie		Hàbitat
<input type="text" value="Sphyrna mokarran"/>		<input type="text" value="No determinado"/>
Profundidad		
<input type="text" value="8"/> m.		
PROSHARKS (Tiburones y rayas pelágicos)		
<input checked="" type="radio"/> Sí <input type="radio"/> No		
*Tiempo de navegación	*Nubosidad	
<input type="text" value="10 minutos"/>	<input type="text" value="3. (3/9) Tres octas de cielo cubierto"/>	
*Fuerza del viento	*Dirección del viento	
<input type="text" value="0 (<1) Calma"/>	<input type="text" value="N : Norte"/>	
*Hora de llegada a la zona de muestreo	*Número de embarcaciones en la zona	
<input type="text" value="12:30"/>	<input type="text" value="5"/>	
Temperatura superficial	*Tipo de cebo	*Sexo
<input type="text" value="25"/>	<input type="text" value="tunido"/>	<input type="text" value="Hembra"/>
*Parásitos	*Presencia de anzuelos	*Número de anzuelos
<input type="text" value="Si"/>	<input type="text" value="No"/>	<input type="text" value="0"/>

Respecte les dades a mostrar en el formulari, com a mínim, s'han de mostrar tots els atributs possibles i que l'usuari ompli els que consideri. Més eficient seria mostrar únicament els camps segons l'espècie.

Així, l'organització ha definit el producte que vol desenvolupar i ens ha contractat per a portar a terme aquesta tasca.

A més, la pantalla del formulari disposarà d'un botó que en clicar-se, calcularà les següents estadístiques:

- Total d'exemplars albirats (per espècie)
- Profunditat mitjana de l'abirament de cada espècie
- Mida mínima de cada espècie
- Mida màxim de cada espècie

Important!

- Si les dades introduïdes són correctes, l'aplicació haurà de mostrar un missatge informant a l'usuari.
- Si algun dels valors introduïts no és correcte, caldrà indicar-ho a l'usuari.

El projecte consta dels següents punts:

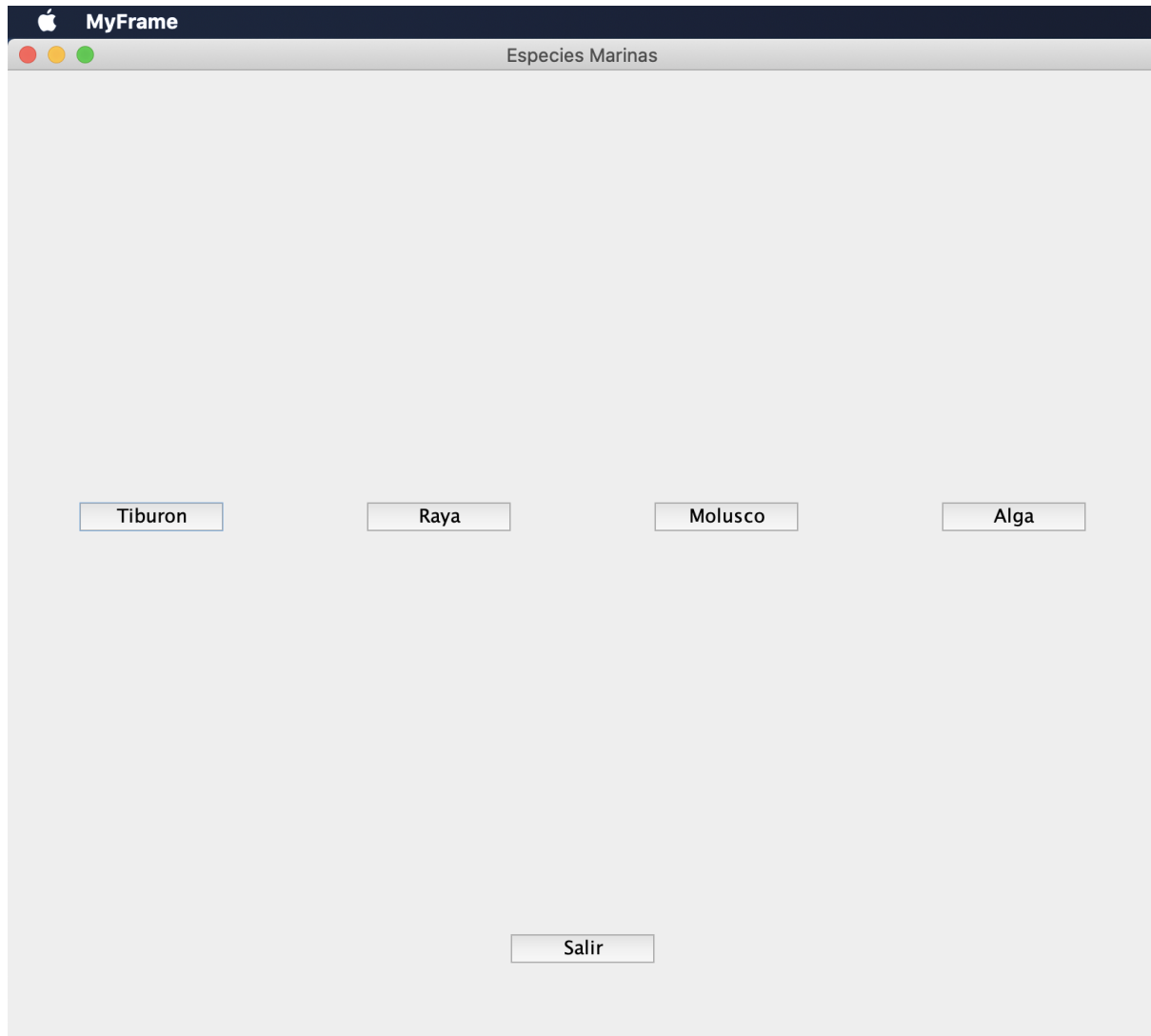
- ~~1. Analitzar tots els requeriments del projecte i plasmar-los en diferents tasques al Jira~~
2. Dissenyar i implementar la interfície gràfica per tal que l'usuari pugui introduir un nou article al sistema
3. Dissenyar e implementar les classes, objectes i interfícies necessàries **seguint la filosofia d'aplicacions modulars de Java**
4. Modelar la base de dades (taules, claus primàries, claus forànies, ...) que reculli les entitats del projecte. Cal afegir al projecte els scripts de creació de la bb (taules, registres, ...).
5. **OPCIONAL:** Calcular les estadístiques* (mostrar per consola i guardar en un fitxer)
6. Documentar l'aplicació i crear el Javadocs
7. **OPCIONAL:** Implementar el document readme de GitHub, explicant el projecte, els seus requeriments, documents associats, enllaç al Jira, ...

* Revisar API de Streams (teoria)

2. Dissenyar i implementar la interfície gràfica per tal que l'usuari pugui introduir un nou article al sistema

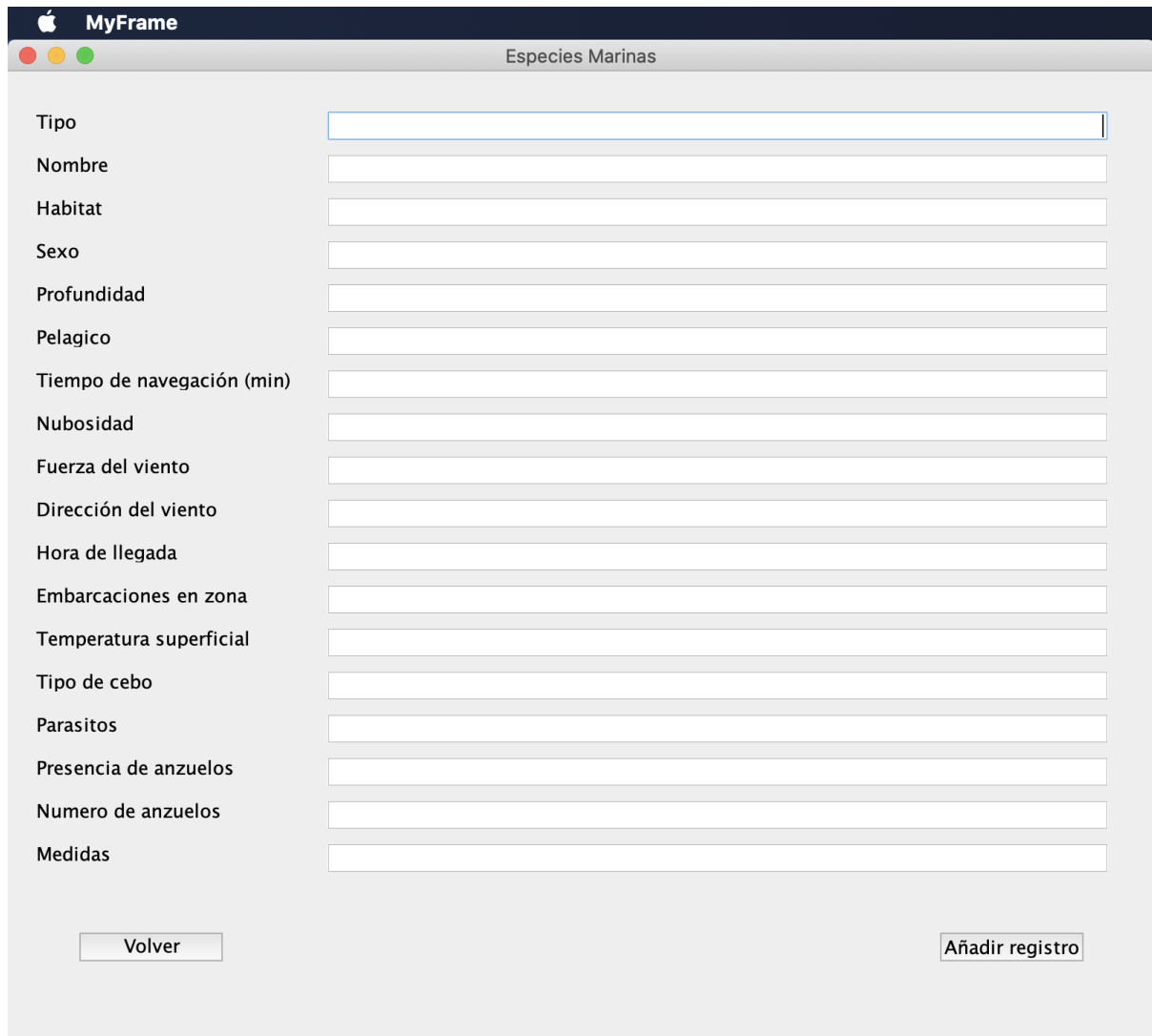
Interfície:

Creamos nuestra interfaz donde podemos elegir que tipo de especie marina queremos añadir en nuestra BBDD. También tenemos un botón para salir del programa.



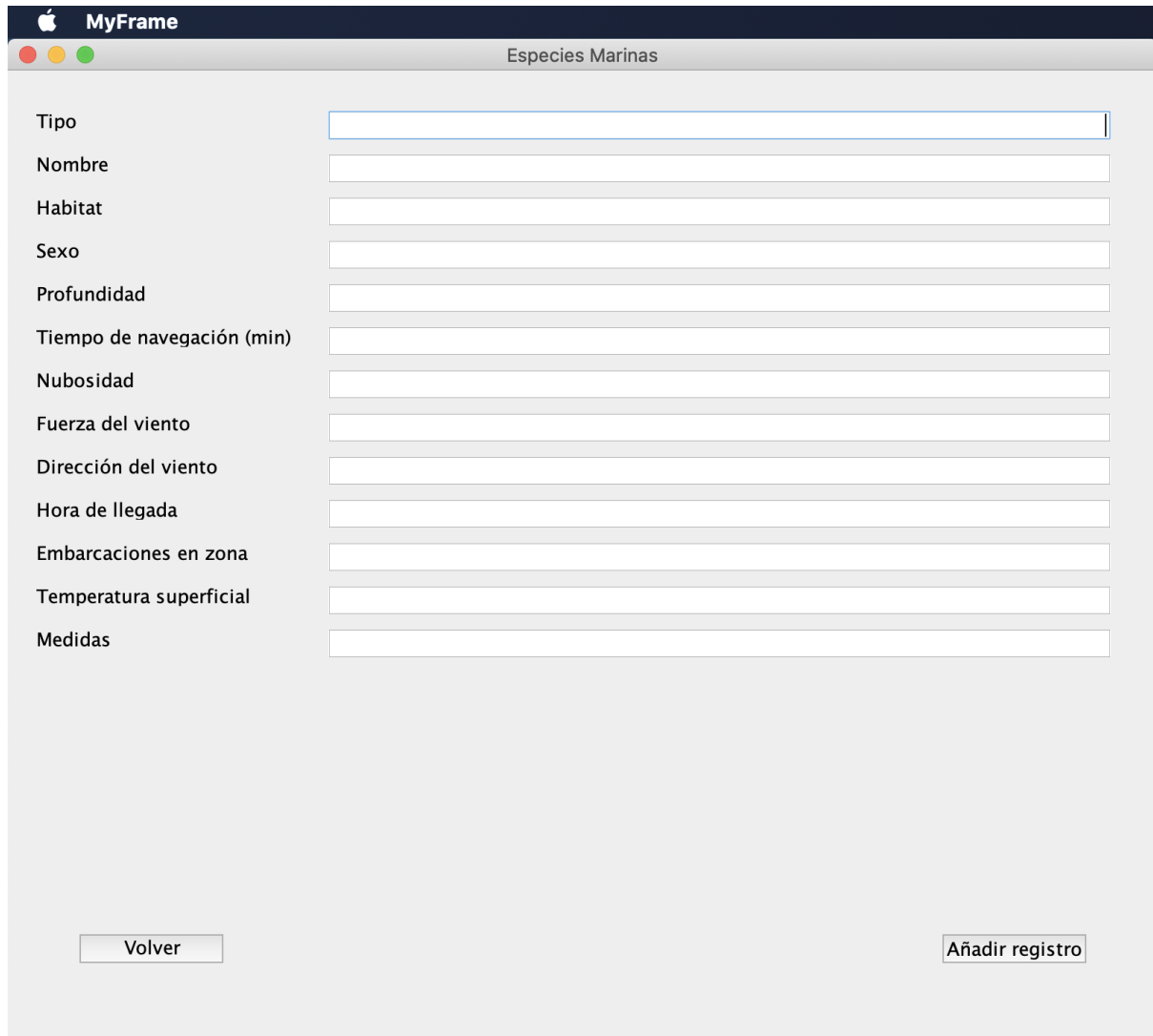
Dependiendo lo que se escoja obtendremos unos campos específicos o otros:

Tiburón

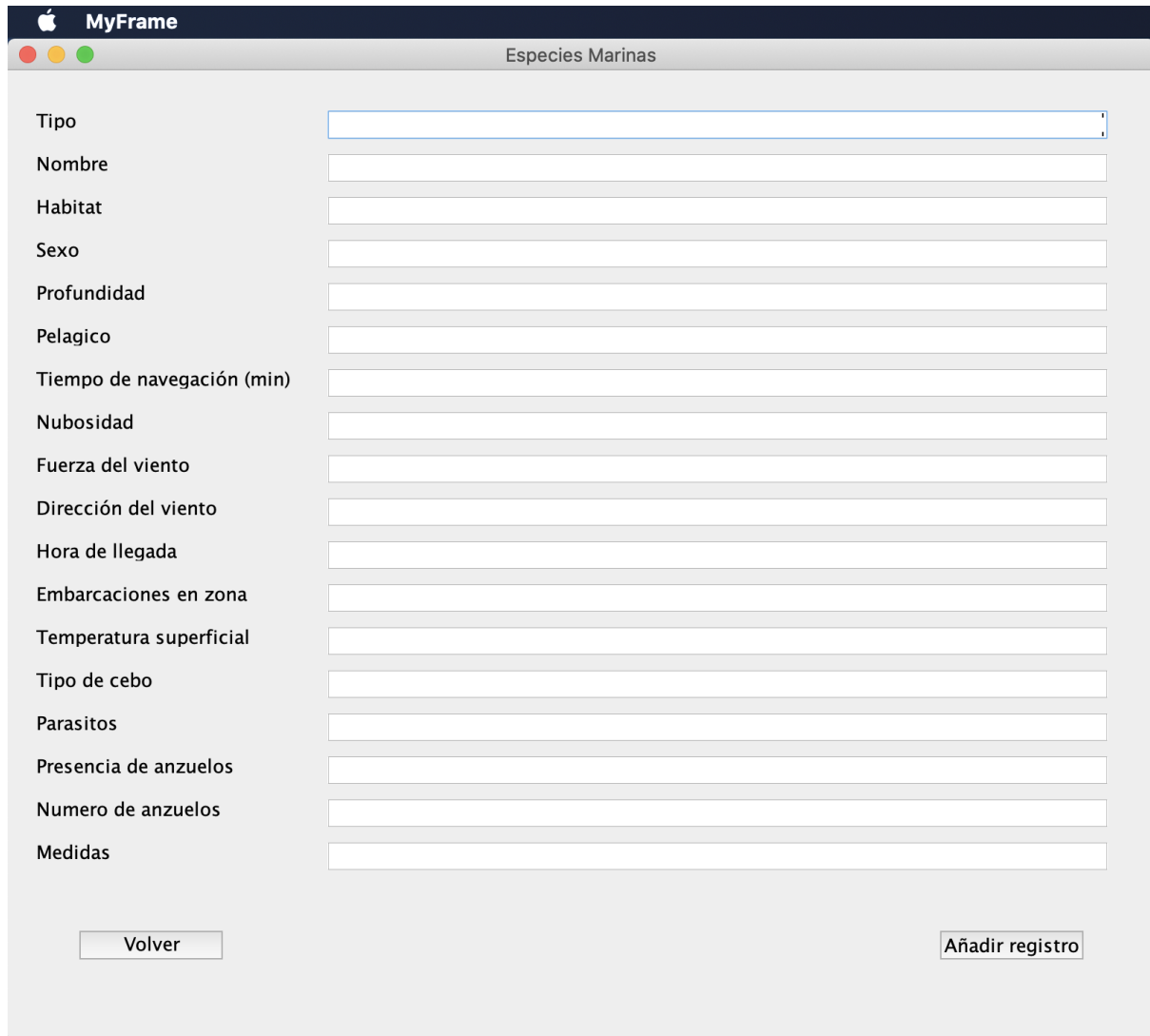


Especies Marinas	
Tipo	<input type="text"/>
Nombre	<input type="text"/>
Habitat	<input type="text"/>
Sexo	<input type="text"/>
Profundidad	<input type="text"/>
Pelagico	<input type="text"/>
Tiempo de navegación (min)	<input type="text"/>
Nubosidad	<input type="text"/>
Fuerza del viento	<input type="text"/>
Dirección del viento	<input type="text"/>
Hora de llegada	<input type="text"/>
Embarcaciones en zona	<input type="text"/>
Temperatura superficial	<input type="text"/>
Tipo de cebo	<input type="text"/>
Parasitos	<input type="text"/>
Presencia de anzuelos	<input type="text"/>
Numero de anzuelos	<input type="text"/>
Medidas	<input type="text"/>

Adicionalmente tenemos el botón de volver para ir hacia atrás.

Molusco

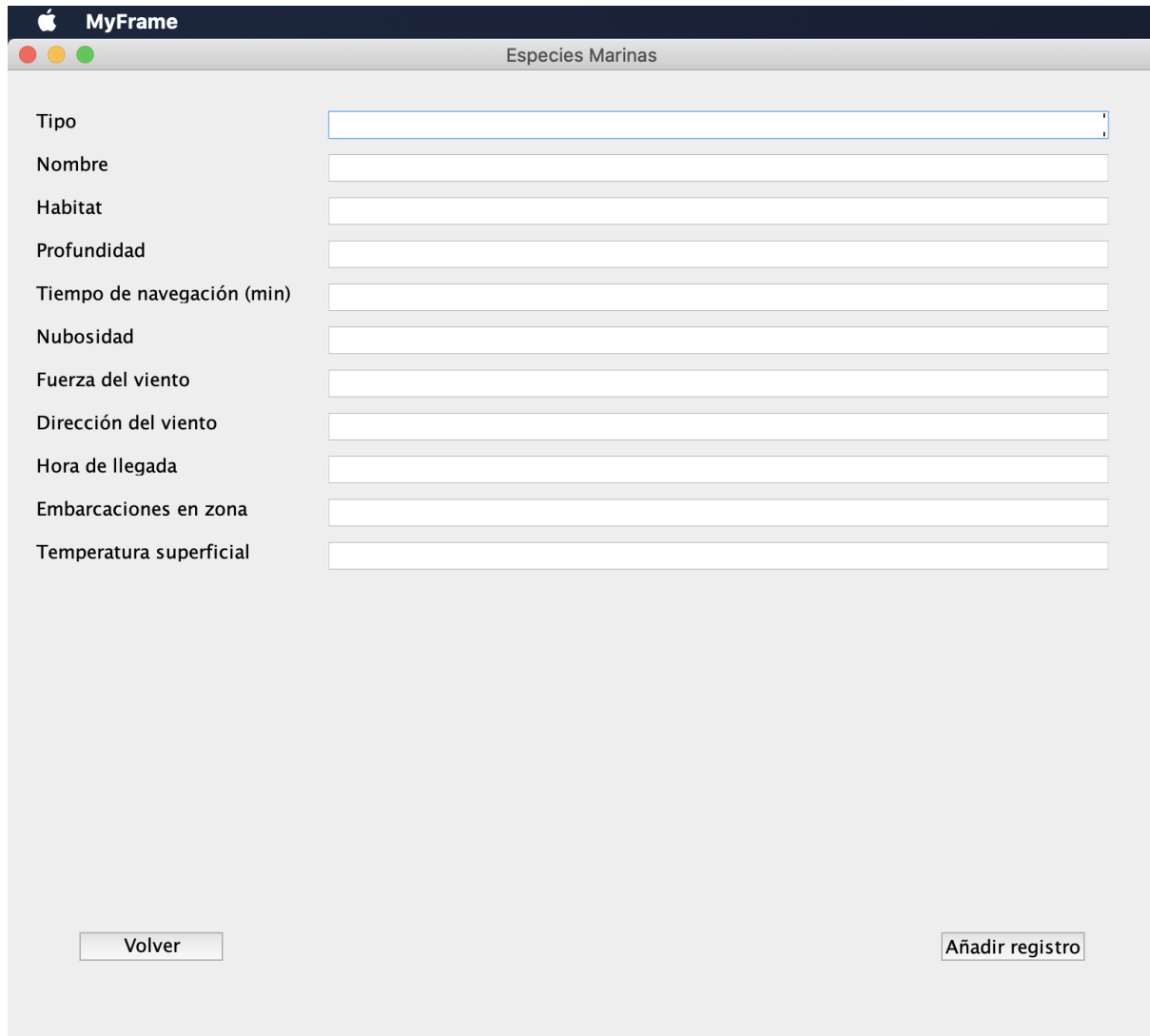
Tipo	<input type="text"/>
Nombre	<input type="text"/>
Habitat	<input type="text"/>
Sexo	<input type="text"/>
Profundidad	<input type="text"/>
Tiempo de navegación (min)	<input type="text"/>
Nubosidad	<input type="text"/>
Fuerza del viento	<input type="text"/>
Dirección del viento	<input type="text"/>
Hora de llegada	<input type="text"/>
Embarcaciones en zona	<input type="text"/>
Temperatura superficial	<input type="text"/>
Medidas	<input type="text"/>

Raya

MyFrame Especies Marinas

Tipo	<input type="text"/>
Nombre	<input type="text"/>
Habitat	<input type="text"/>
Sexo	<input type="text"/>
Profundidad	<input type="text"/>
Pelagico	<input type="text"/>
Tiempo de navegación (min)	<input type="text"/>
Nubosidad	<input type="text"/>
Fuerza del viento	<input type="text"/>
Dirección del viento	<input type="text"/>
Hora de llegada	<input type="text"/>
Embarcaciones en zona	<input type="text"/>
Temperatura superficial	<input type="text"/>
Tipo de cebo	<input type="text"/>
Parasitos	<input type="text"/>
Presencia de anzuelos	<input type="text"/>
Numero de anzuelos	<input type="text"/>
Medidas	<input type="text"/>

Alga

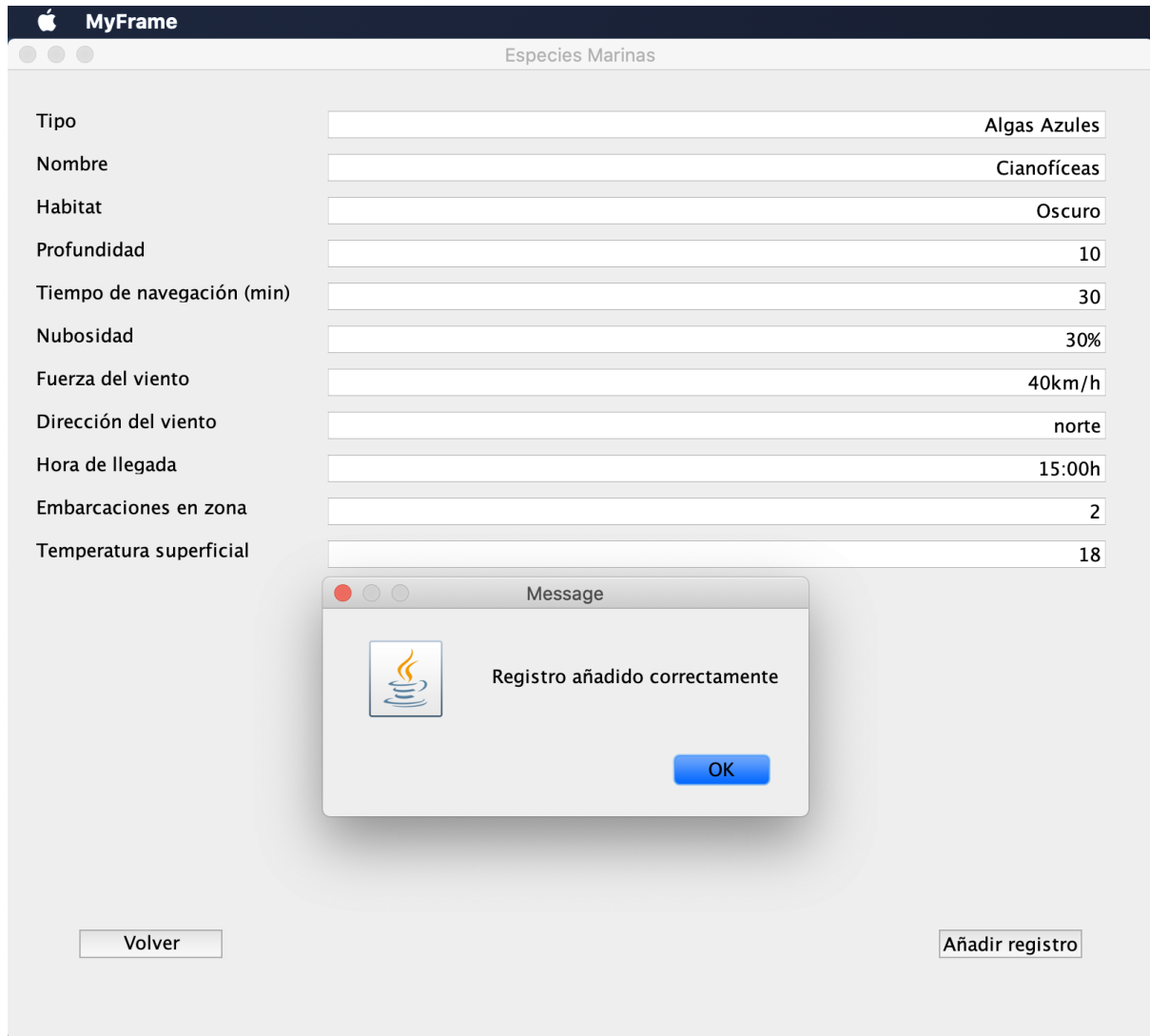


The screenshot shows a web application window titled "MyFrame" with a sub-header "Especies Marinas". The form contains the following fields:

- Tipo
- Nombre
- Habitat
- Profundidad
- Tiempo de navegación (min)
- Nubosidad
- Fuerza del viento
- Dirección del viento
- Hora de llegada
- Embarcaciones en zona
- Temperatura superficial

At the bottom of the form, there are two buttons: "Volver" and "Añadir registro".

Añadiendo un registro “Alga” a BBDD en Cloud:



MyFrame

Especies Marinas

Tipo	Algas Azules
Nombre	Cianofíceas
Habitat	Oscuro
Profundidad	10
Tiempo de navegación (min)	30
Nubosidad	30%
Fuerza del viento	40km/h
Dirección del viento	norte
Hora de llegada	15:00h
Embarcaciones en zona	2
Temperatura superficial	18

Message

Registro añadido correctamente

OK

Volver

Añadir registro

BBDD En PHP MyAdmin:



The screenshot shows the phpMyAdmin interface for a database named 'kev_especiesmarinas'. The table 'alga' is selected. A SQL query is entered in the query box:

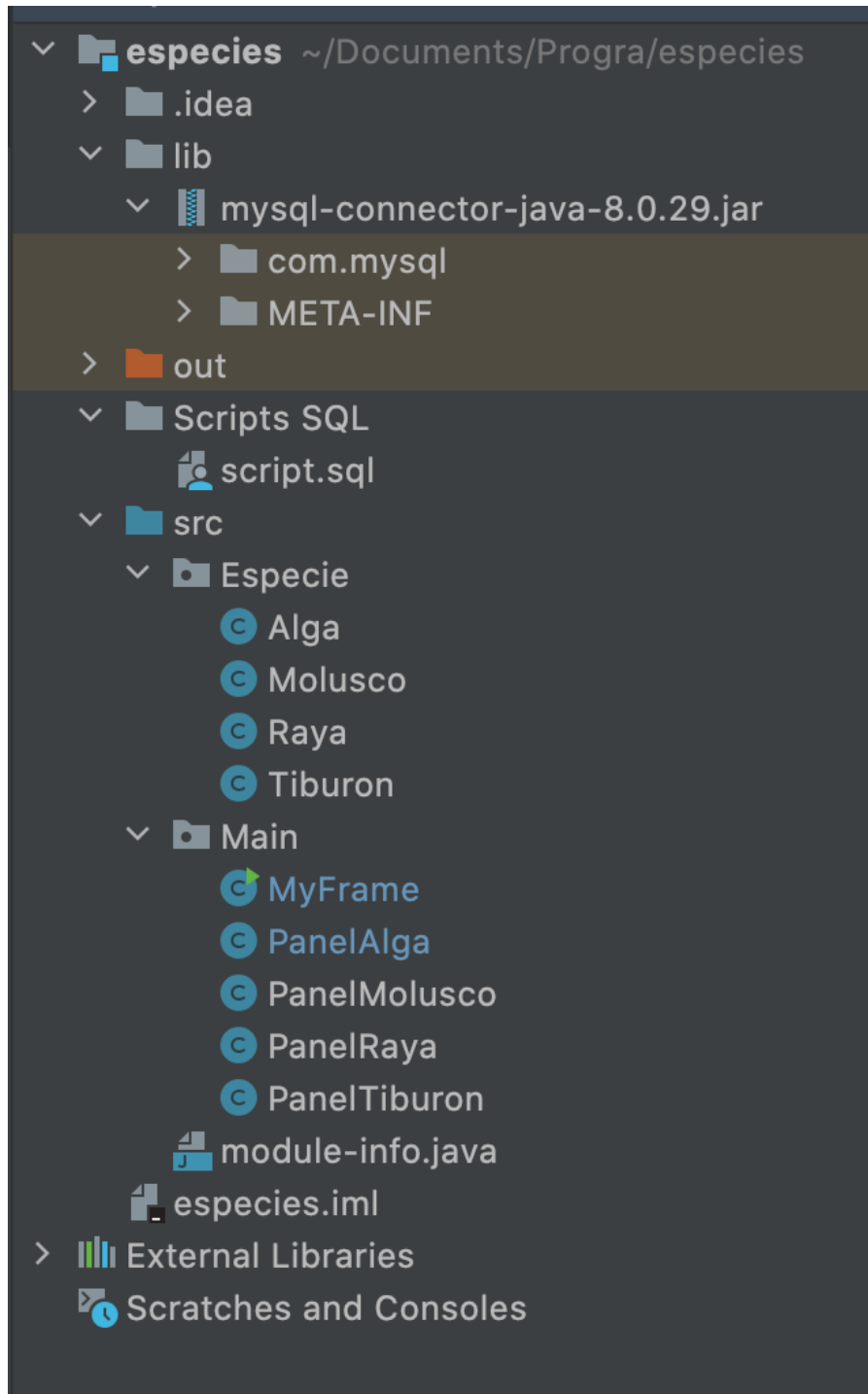
```
SELECT * FROM `alga` WHERE nombre = "Cianoficeas";
```

The results show one row of data:

id	especie	tipo	nombre	habitat	profundidad	tiempo_navegacion	nubosidad	fuerza_viento	direccion_viento	hora_llegada	em
4	alga	Algas Azules	Cianoficeas	Oscuro	10	30	30%	40km/h	norte	15:00h	

3. Dissenyar e implementar les classes, objectes i interfícies necessàries seguint la filosofia d'aplicacions modulars de Java

Arquitectura del proyecto:



4. Dissenyar e implementar les classes, objectes i interfícies necessàries seguint la filosofia d'aplicacions modulars de Java

Se ha procedido a crear la BBDD desde Workbench.

El modelo de tablas es el siguiente:

tiburón	raya	molusco	alga
id INT	id INT	id INT	id INT
especie VARCHAR(45)	especie VARCHAR(45)	especie VARCHAR(45)	especie VARCHAR(45)
tipo VARCHAR(45)	tipo VARCHAR(45)	tipo VARCHAR(45)	tipo VARCHAR(45)
nombre VARCHAR(45)	nombre VARCHAR(45)	nombre VARCHAR(45)	nombre VARCHAR(45)
habitat VARCHAR(45)	habitat VARCHAR(45)	habitat VARCHAR(45)	habitat VARCHAR(45)
sexo VARCHAR(45)	sexo VARCHAR(45)	sexo VARCHAR(45)	profundidad INT
profundidad INT	profundidad INT	profundidad INT	tiempo_navegacion INT
pelagico VARCHAR(45)	pelagico VARCHAR(45)	tiempo_navegacion INT	nubosidad VARCHAR(45)
tiempo_navegacion INT	tiempo_navegacion INT	nubosidad VARCHAR(45)	fuerza_viento VARCHAR(45)
nubosidad VARCHAR(45)	nubosidad VARCHAR(45)	fuerza_viento VARCHAR(45)	direccion_viento VARCHAR(45)
fuerza_viento VARCHAR(45)	fuerza_viento VARCHAR(45)	direccion_viento VARCHAR(45)	hora_llegada VARCHAR(45)
direccion_viento VARCHAR(45)	direccion_viento VARCHAR(45)	hora_llegada VARCHAR(45)	embarcaciones_zona INT
hora_llegada VARCHAR(45)	hora_llegada VARCHAR(45)	embarcaciones_zona INT	temperatura_superf INT
embarcaciones_zona INT	embarcaciones_zona INT	temperatura_superf INT	medidas VARCHAR(45)
temperatura_superf INT	temperatura_superf INT	medidas VARCHAR(45)	
tipo_cebo VARCHAR(45)	tipo_cebo VARCHAR(45)		
parasitos VARCHAR(45)	parasitos VARCHAR(45)		
presencia anzuelos VARCHAR(45)	presencia anzuelos VARCHAR(45)		
num anzuelos INT	num anzuelos INT		
medidas VARCHAR(45)	medidas VARCHAR(45)		
Indexes	Indexes	Indexes	Indexes

Script(Disponible en el documento Script.sql)

-- MySQL Script generated by MySQL Workbench

-- Mon May 30 14:34:55 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema kev_especiesmarinas

DROP SCHEMA IF EXISTS `kev_especiesmarinas` ;

-- Schema kev_especiesmarinas

*CREATE SCHEMA IF NOT EXISTS `kev_especiesmarinas` DEFAULT CHARACTER SET utf8 ;
USE `kev_especiesmarinas` ;*

-- Table `kev_especiesmarinas`.`tiburon`

*CREATE TABLE IF NOT EXISTS `kev_especiesmarinas`.`tiburon` (
 `id` INT NOT NULL AUTO_INCREMENT,
 `especie` VARCHAR(45) NULL,
 `tipo` VARCHAR(45) NULL,
 `nombre` VARCHAR(45) NULL,
 `habitat` VARCHAR(45) NULL,
 `sexo` VARCHAR(45) NULL,
 `profundidad` INT NULL,
 `pelagico` VARCHAR(45) NULL,
 `tiempo_navegacion` INT NULL,
 `nubosidad` VARCHAR(45) NULL,*

```
`fuerza_viento` VARCHAR(45) NULL,  
`direccion_viento` VARCHAR(45) NULL,  
`hora_llegada` VARCHAR(45) NULL,  
`embarcaciones_zona` INT NULL,  
`temperatura_superf` INT NULL,  
`tipo_cebo` VARCHAR(45) NULL,  
`parasitos` VARCHAR(45) NULL,  
`presencia_anzuelos` VARCHAR(45) NULL,  
`num_anzuelos` INT NULL,  
`medidas` VARCHAR(45) NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `kev_especiesmarinas`.`raya`  
-----
```

```
CREATE TABLE IF NOT EXISTS `kev_especiesmarinas`.`raya` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `especie` VARCHAR(45) NULL,  
  `tipo` VARCHAR(45) NULL,  
  `nombre` VARCHAR(45) NULL,  
  `habitat` VARCHAR(45) NULL,  
  `sexo` VARCHAR(45) NULL,  
  `profundidad` INT NULL,
```

```
`pelagico` VARCHAR(45) NULL,  
`tiempo_navegacion` INT NULL,  
`nubosidad` VARCHAR(45) NULL,  
`fuerza_viento` VARCHAR(45) NULL,  
`direccion_viento` VARCHAR(45) NULL,  
`hora_llegada` VARCHAR(45) NULL,  
`embarcaciones_zona` INT NULL,  
`temperatura_superf` INT NULL,  
`tipo_cebo` VARCHAR(45) NULL,  
`parasitos` VARCHAR(45) NULL,  
`presencia_anzuelos` VARCHAR(45) NULL,  
`num_anzuelos` INT NULL,  
`medidas` VARCHAR(45) NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `kev_especiesmarinas`.`molusco`  
-----
```

```
CREATE TABLE IF NOT EXISTS `kev_especiesmarinas`.`molusco` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `especie` VARCHAR(45) NULL,  
  `tipo` VARCHAR(45) NULL,  
  `nombre` VARCHAR(45) NULL,
```



```
`habitat` VARCHAR(45) NULL,  
`sexo` VARCHAR(45) NULL,  
`profundidad` INT NULL,  
`tiempo_navegacion` INT NULL,  
`nubosidad` VARCHAR(45) NULL,  
`fuerza_viento` VARCHAR(45) NULL,  
`direccion_viento` VARCHAR(45) NULL,  
`hora_llegada` VARCHAR(45) NULL,  
`embarcaciones_zona` INT NULL,  
`temperatura_superf` INT NULL,  
`medidas` VARCHAR(45) NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `kev_especiesmarinas`.`alga`  
-----
```

```
CREATE TABLE IF NOT EXISTS `kev_especiesmarinas`.`alga` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `especie` VARCHAR(45) NULL,  
  `tipo` VARCHAR(45) NULL,  
  `nombre` VARCHAR(45) NULL,  
  `habitat` VARCHAR(45) NULL,  
  `profundidad` INT NULL,
```

```
`tiempo_navegacion` INT NULL,  
`nubosidad` VARCHAR(45) NULL,  
`fuerza_viento` VARCHAR(45) NULL,  
`direccion_viento` VARCHAR(45) NULL,  
`hora_llegada` VARCHAR(45) NULL,  
`embarcaciones_zona` INT NULL,  
`temperatura_superf` INT NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

6. Documentar l'aplicació i crear el Javadocs*Código comentado y con Javadocs*

```
package Main;

/**
 * Api para añadir nuevos registros de especies marinas
 * en BBDD de Cloud Mysql
 * @autor Kevin Añazco Honores
 *
 */

import javax.swing.*;
import java.awt.event.ActionEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.event.ActionListener;
import java.sql.SQLException;

import static java.awt.EventQueue.invokeLater;

public class MyFrame extends JFrame implements ActionListener{

    /**
     * Definición de parametros para Interfaz
     * Luego los llamaremos para trabajar con ellos
     */

    //Definimos un Panel
    private JPanel contenedor;

    //Definimos los botones para la elección de especie
    private JButton jbtTiburón;
    private JButton jbtRaya;
    private JButton jbtMolusco;
```