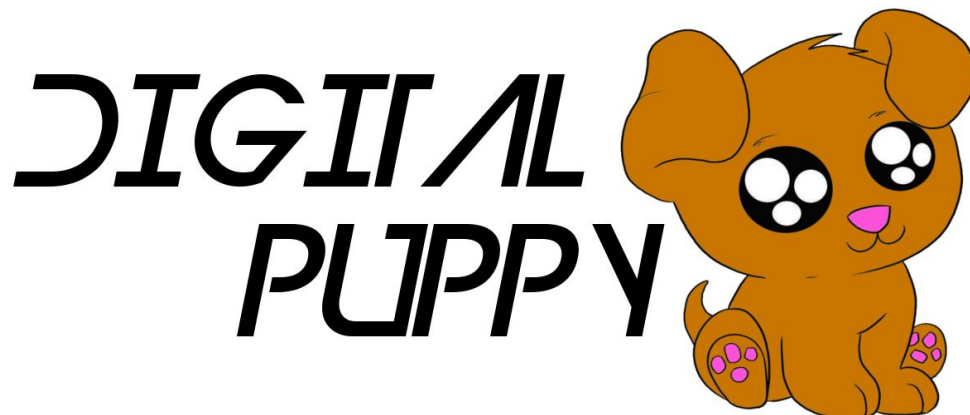




Documentación de la práctica final

- Interacción Persona-Ordenador I -



Pedro-Manuel Gómez-Portillo López

pedromanuel.gomezportillo@alu.uclm.es

Diego Molero Marín

diego.molero@alu.uclm.es

Indice

1. Introducción
2. Instrucciones de instalación
3. Análisis de requisitos
 - 3.1. Diagramas de casos de uso de la Pestaña *Perros*
 - 3.2. Diagrama de casos de uso de la pestaña *Cuentas*
 - 3.3. Diagrama de casos de uso de la pestaña *Perrera*
4. Manual de usuario de la aplicación
 - 4.1. Ventana de login
 - 4.2. Barra de menú superior
 - 4.3. Pestaña *Perros*
 - 4.4. Pestaña *Voluntarios*
 - 4.5. Pestaña *Perdidos*
 - 4.6. Pestaña *Socios*
 - 4.7. Pestaña *Cuentas*
 - 4.8. Pestaña *Perrera*
5. Tecnología y recursos utilizados
6. Justificación del diseño de la GUI en base a lo estudiado en teoría
7. Mockups vs Implementaciones
 - 7.1. Ventana de login
 - 7.2. Pestaña *Perros*
 - 7.3. Pestaña *Voluntarios*
 - 7.4. Pestaña *Perdidos*
 - 7.5. Pestaña *Socios*
 - 7.6. Pestaña *Cuentas*
 - 7.7. Pestaña *Perrera*
8. Webgrafía

1.- Introducción

Esta documentación corresponde a la práctica final de la asignatura de Interacción Persona-Ordenador I. El objeto de la misma es el desarrollo del prototipo de una aplicación, en nuestro caso llamada Digital Puppy, que permita a una protectora de perros manejar su sistema de información.

El prototipo ha sido desarrollada en Java 7 con el IDE Eclipse Mars y el plugin Windows Builder y la herramienta de compartición y versionado de código Git+GitHub.

La aplicación consiste principalmente en varias pestañas, cada una de ellas centrada en una temática y un menú de inicio en el que el usuario se loguea con un par usuario:contraseña válido y elige el idioma que desee. Para definir la estructura de las pestañas se ha seguido la guía de proyecto proporcionada por los profesores de la asignatura (para más detalle consultar Apartado 4).

2.- Instrucciones de instalación

La aplicación no necesita instalación. Un archivo Java Archive, Digital_Puppy.jar, ha sido generado y para ejecutarla solamente es necesario hacer doble click sobre él. Tanto las rutas a las imágenes como a los archivos de persistencia han sido escritas relativamente en el código, de tal forma que es posible ejecutarlo en cualquier máquina. Sin embargo, el **fichero data/** con todas sus subcarpetas **debe estar en el mismo directorio** que el ejecutable JAR para que la persistencia funcione. Dicho directorio es donde el programa guardará y cargará los archivos que utilizará para la persistencia de datos.

3.-Análisis de requisitos

Se debe realizar un programa que ayude a la administración de la protectora de perros a gestionar los diferentes ámbitos que abarca.

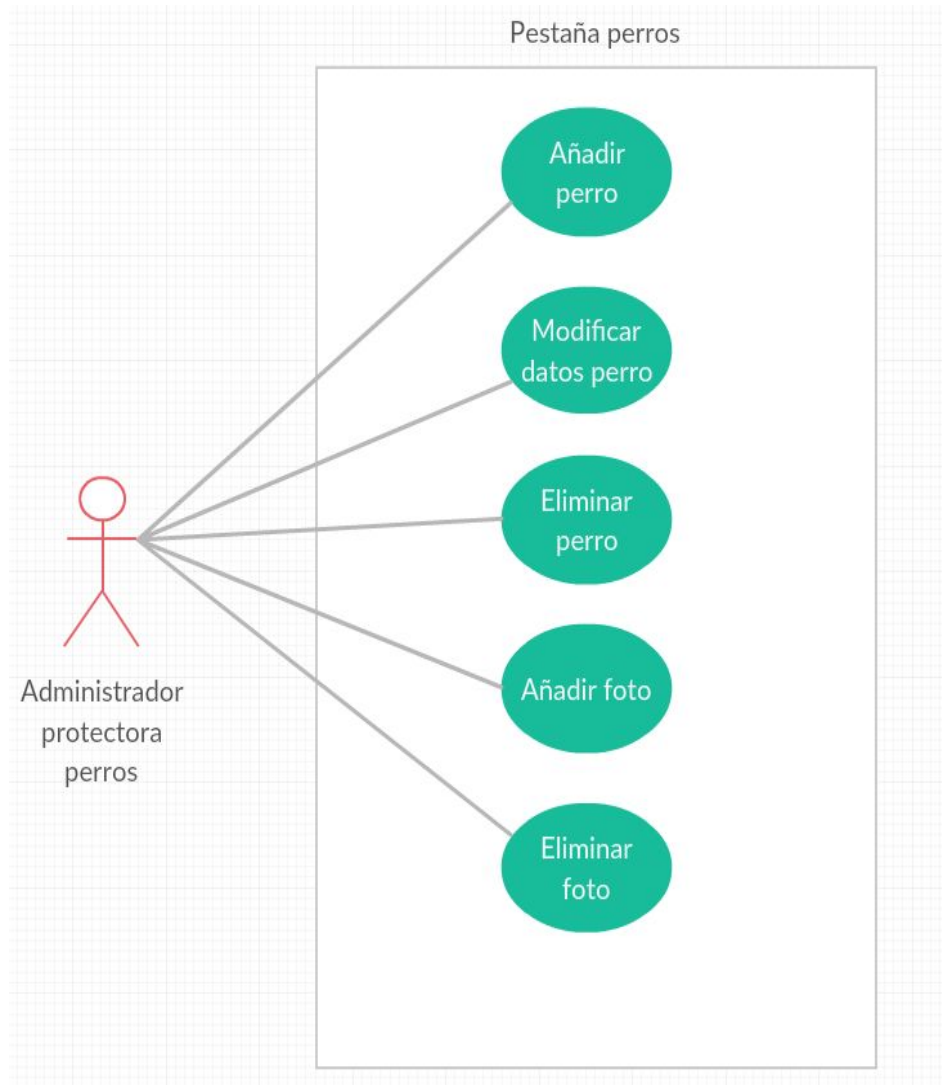
1. Para proteger los datos, el usuario deberá autenticarse, solo así podrá entrar en la aplicación
2. El usuario podrá elegir el idioma que quiera usar la aplicación, entre Español e Inglés
3. El usuario podrá gestionar los siguientes ámbitos
 - a. Voluntarios apuntados
 - b. Perros disponibles en la protectora
 - c. Perros perdidos
 - d. Socios que ayudan a la protectora

Gestionar implica las funciones de:

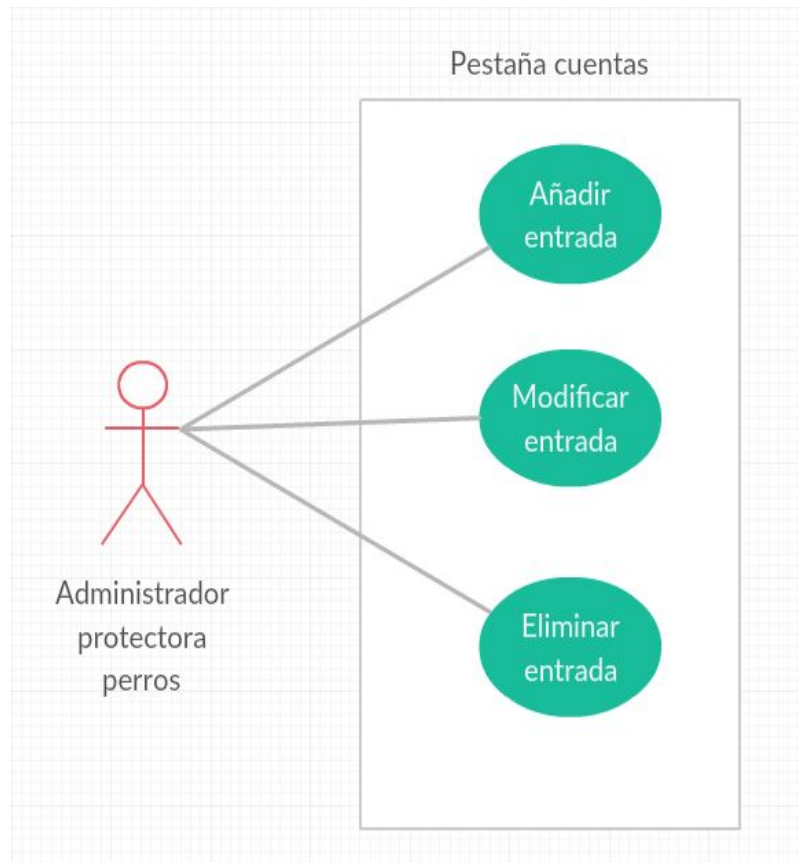
- a. Añadir elemento nuevo
 - b. Modificar sus datos (ej. nombre, edad, raza, tamaño...)
 - c. Eliminar elemento
 - d. Añadir foto
 - e. Eliminar foto
4. El usuario de la aplicación podrá acceder a la pestaña de cuentas donde podrá hacer un balance de los ingresos y los gastos que tiene la protectora, pudiendo añadir la cantidad de la entrada y su razón.
5. El usuario tendrá acceso a el mapa de jaulas de perros donde podrá dibujar para poder administrar la organización de los perros en las jaulas, también podrá añadir texto adicional, formas geométricas y borrar todo lo anterior.

3.1.- Diagrama de casos de uso de la pestaña que gestiona los perros

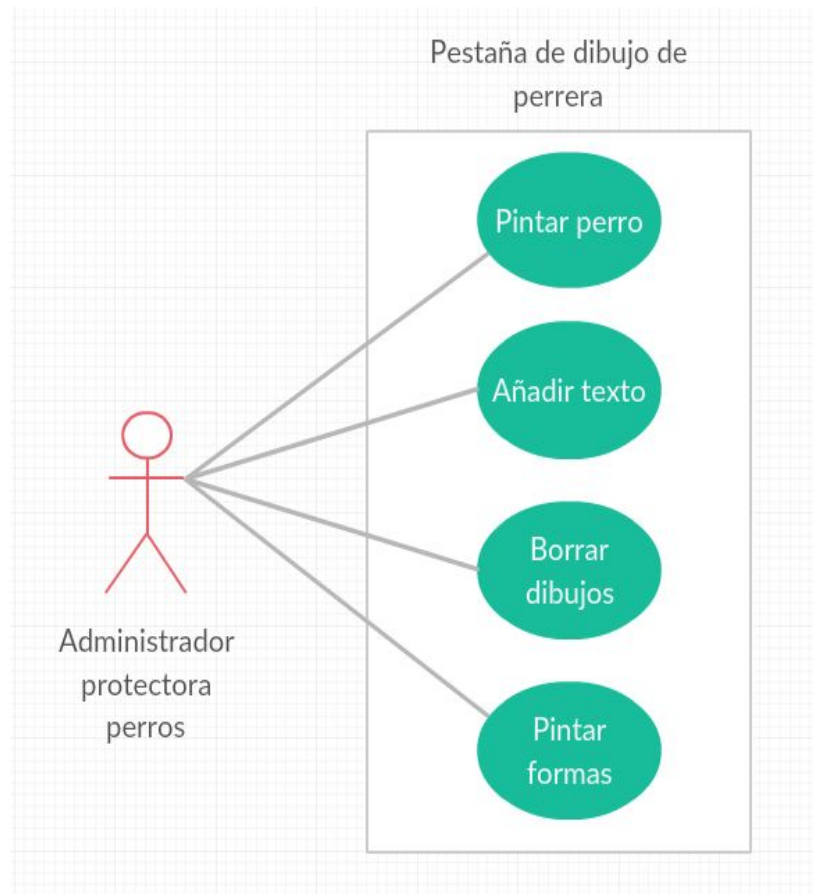
Las pestañas que gestionan los voluntarios, socios, perdidos son similares y resultaría redundante hacer otro diagrama de casos de uso por cada pestaña.



3.2.- Diagrama de casos de uso de pestaña cuentas



3.3.- Diagrama de casos de uso de pestaña de dibujo



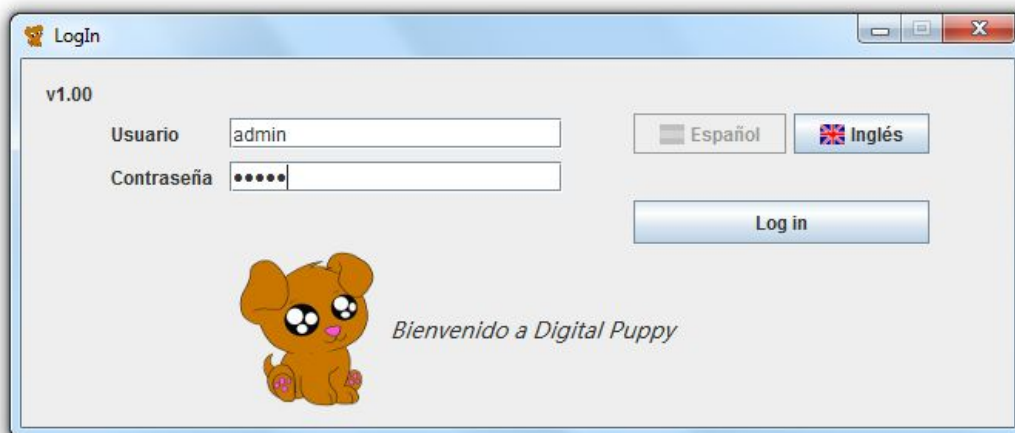
4.- Manual de usuario de la aplicación

Para iniciar la aplicación haremos doble click sobre el archivo Digital_Puppy.jar. Tras esto nos aparecerá un diálogo para que elijamos el idioma del sistema e introduzcamos un par usuario:contraseña válido. El usuario solo podrá loguearse con

admin : admin

Una vez dentro, la aplicación en sí se iniciará mostrando la barra de menú, las diferentes pestañas con las que podemos interactuar y la barra inferior de información, que irá mostrando los eventos más importantes que ocurran.

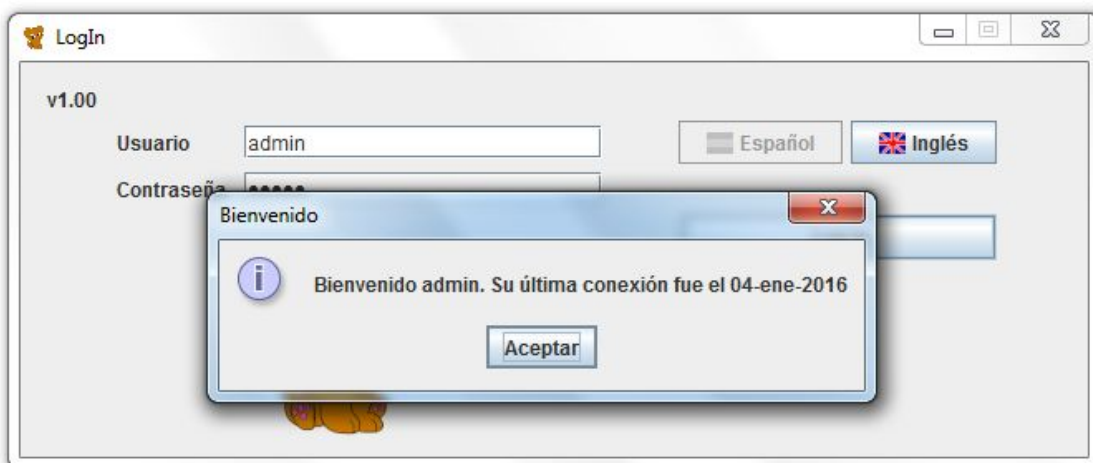
4.1.- Ventana de login



La ventana de login será la primera que veamos al iniciar la aplicación. Desde ella deberemos introducir un par usuario:contraseña válidos y elegir el idioma de la aplicación. Por defecto estará seleccionado el español, pero si queremos bastará con pinchar en el botón “Inglés” para cambiar el idioma de la aplicación. Si introducimos algún dato incorrecto el sistema nos avisará y tendremos que intentarlo otra vez.

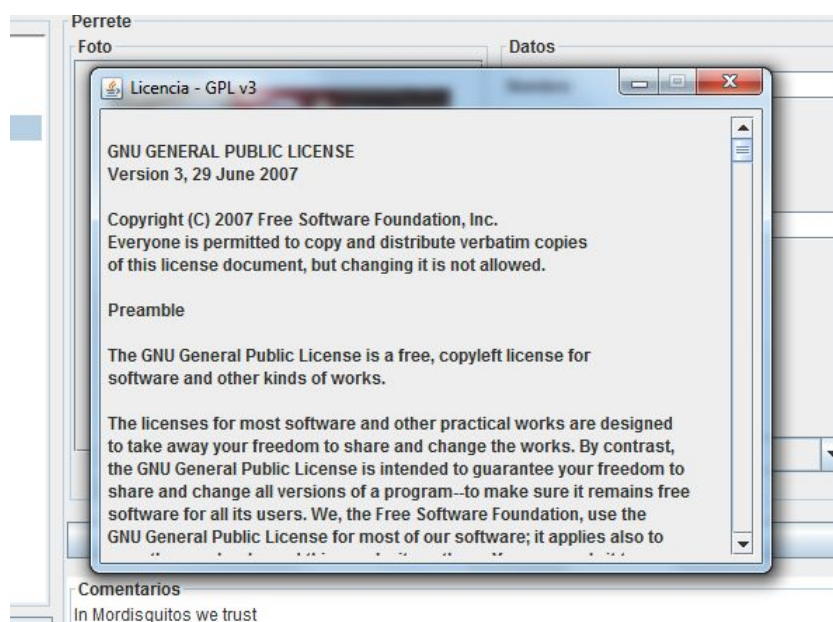
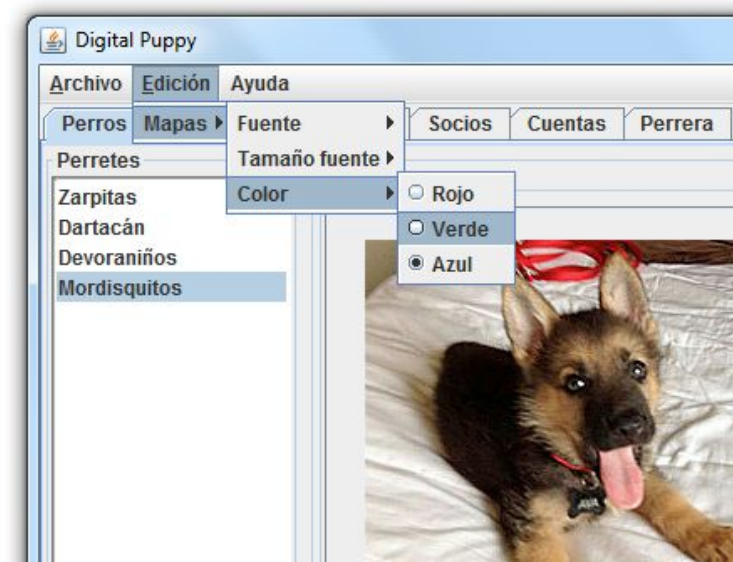


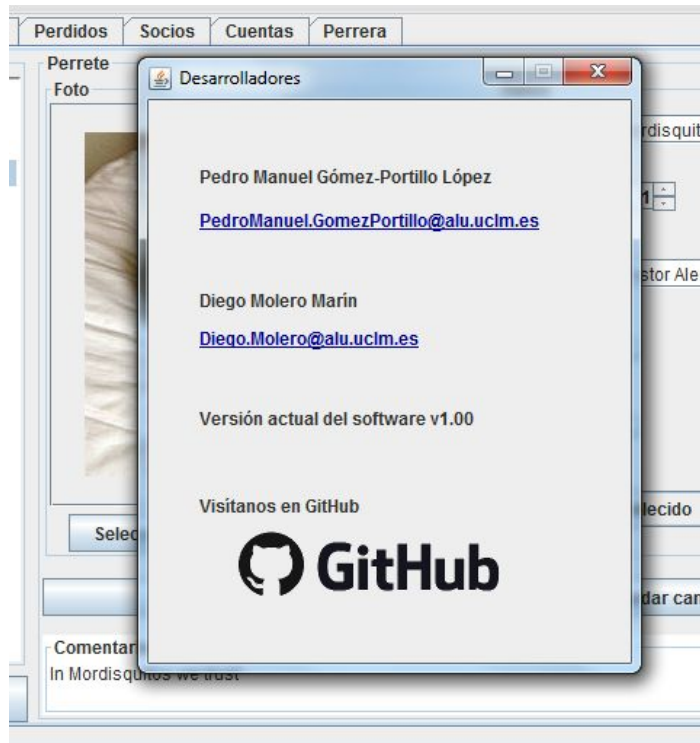
Si todo es correcto, la aplicación nos informará con la fecha de la última conexión que hicimos y se iniciará.



4.2.- Barra de menú superior

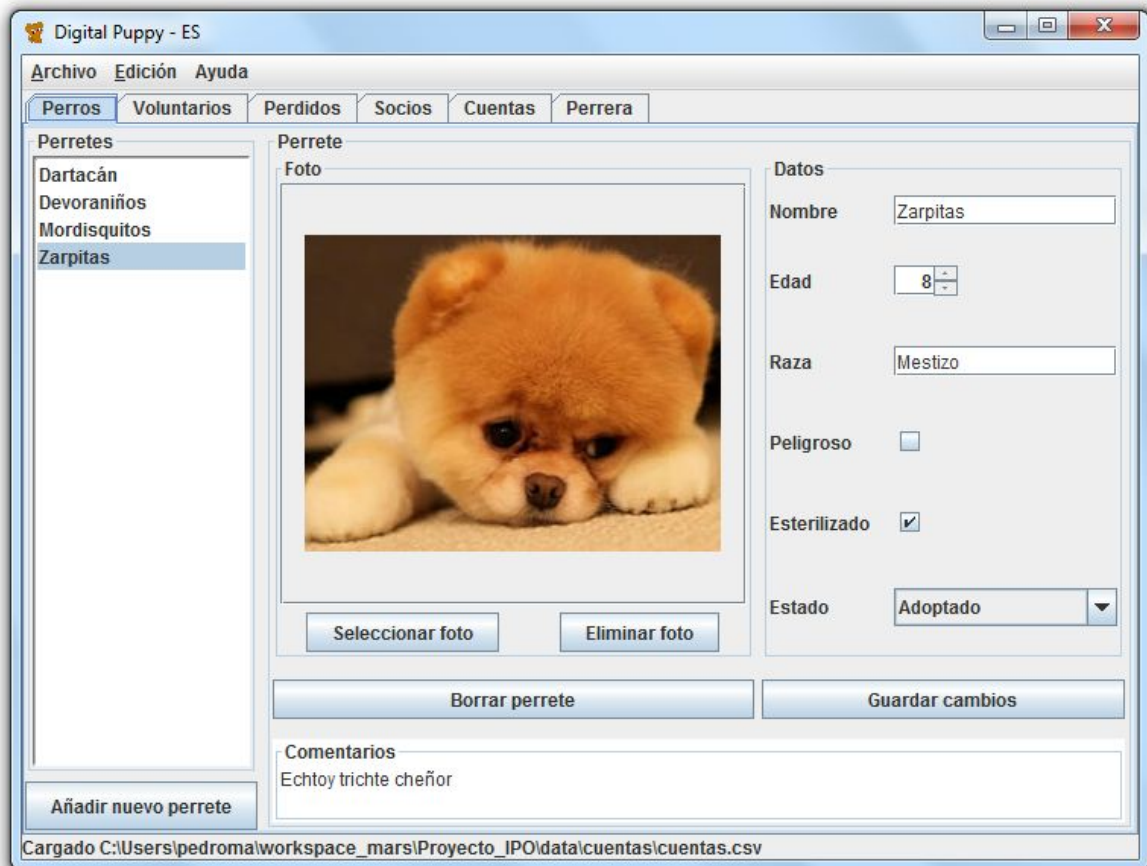
En el menú superior nos encontramos tres opciones, Archivo, que contiene Archivo>Salir, permitiéndonos cerrar la aplicación, Edición, que nos permitirá cambiar el tipo y tamaño de la fuente que aparece en el mapa de la perrera y el color tanto de la letra como de los dibujables que añadamos, y Ayuda, que contiene Ayuda>Acerca de..., que abrirá un diálogo con información sobre los dos integrantes del equipo de desarrollo de este prototipo y Ayuda>Licencia, que carga y muestra el archivo de licencia GPL v3 que adopta esta aplicación.





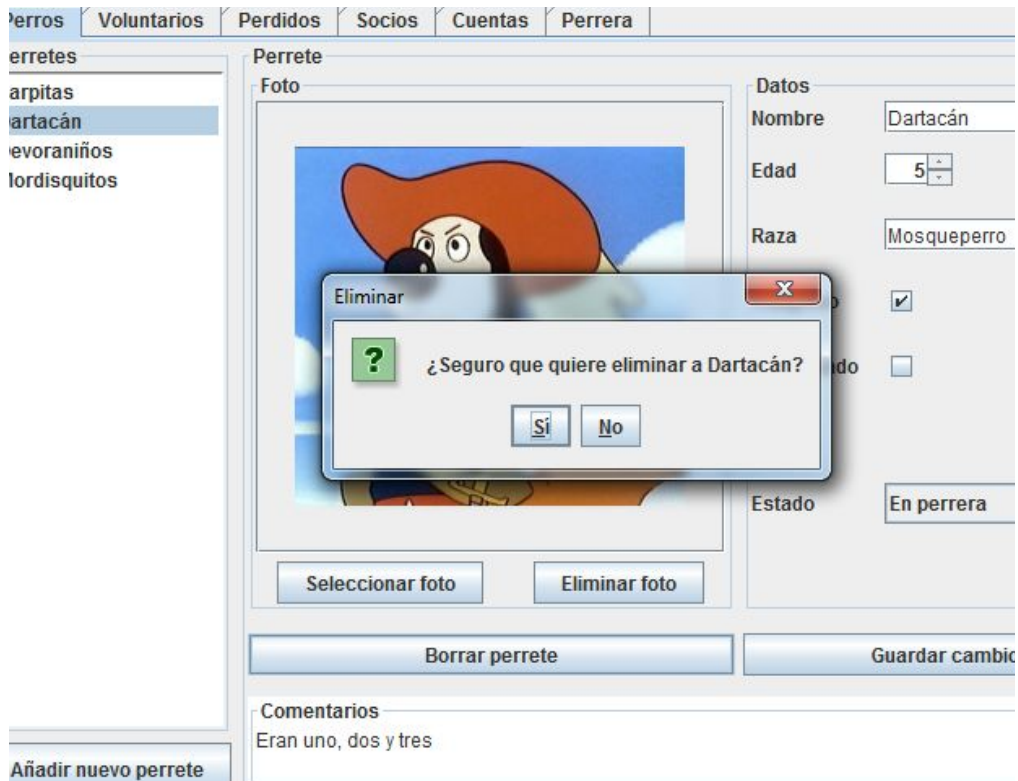
Pinchando en los correos de los desarrolladores se abrirá un link estilo *mailto:*, y hacer lo mismo en el logo de GitHub abrirá la [página web del proyecto](#), donde pueden verse los distintos commits y versiones del desarrollo.

4.3.- Pestaña Perros

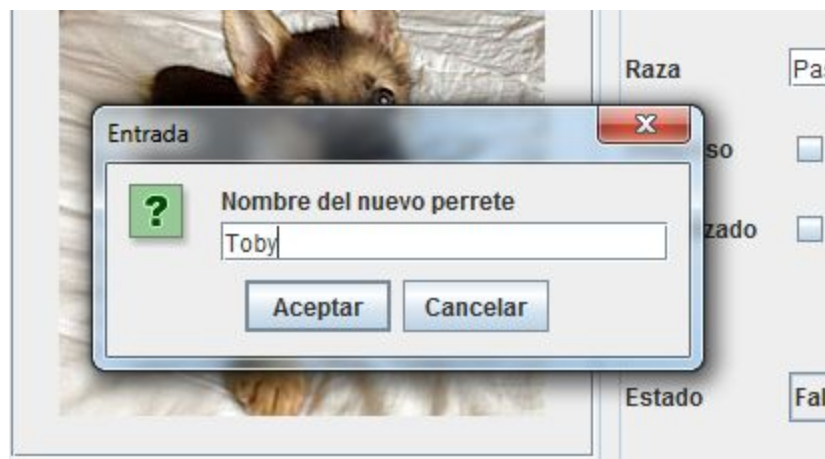


Esta pestaña está destinada al manejo de los perros que se encuentran en la protectora. Se puede dividir principalmente en dos secciones; la lista con todos los cánidos de la perrera y la zona con la información del animal seleccionado. Para ver la información de cada perro habrá que pinchar en su nombre en la lista y se cargará su información en la sección de la derecha. En esta podremos simplemente verlo o realizar cambios, como cambiar su edad, su foto o su estado. Para hacer que estos sean persistentes pinchamos en el botón "Guardar Cambios" y "Aceptar" en el diálogo que aparecerá preguntándonos si estamos seguros.

Para borrar un perrete permanentemente del sistema seleccionamos su nombre en la lista y pinchamos en "Borrar perrete"; el sistema nos preguntará si realmente queremos borrarlo, y tras clicar en *SÍ* lo hará definitivo. En caso de darle a *NO*, la operación se cancelará.



Para añadir un nuevo integrante al sistema hacemos click en el botón “Añadir nuevo perrete”, bajo la lista de perros. Nos preguntará por su nombre, y una vez introducido podremos rellenar sus datos.



Importante: recordar guardar los cambios antes de salir.

4.4.- Pestaña *Voluntarios*

Digital Puppy - ES

Archivo Edición Ayuda


Perros Voluntarios Perdidos Socios Cuentas Perrera

Voluntarios

- Alexelcapo
- Rodolfo Chikilicuatre
- SnoopDogg

Voluntario

Foto



Seleccionar foto Eliminar foto

Datos

Nombre Alexelcapo

Edad 34

Direccion Las Palmas

Telefono 433-551-234

NIF 59182374-F

Veterinari@ ☒

Borrar voluntario Guardar cambios

Comentarios

Sube gameplays de "the binding of isaac"

Añadir nuevo voluntario

Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\voluntarios\Alexelcapo.voluntario

La pestaña voluntarios permite al usuario gestionar las personas que ayudan en la protectora añadiendo los datos básicos de la persona. El teléfono y el DNI tienen una plantilla para que el usuario de la aplicación no pueda equivocarse en la introducción de datos.

4.5.- Pestaña *Perdidos*

Digital Puppy - ES

Archivo Edición Ayuda

Perros Voluntarios **Perdidos** Socios Cuentas Perrera


Perdidos

Darth Dog

Yodita

Perdido

Foto



Seleccionar foto Eliminar foto

Borrar perdido

Datos

Nombre Yodita

Edad 20

Raza Jedi

Telefono 000-000-001

Fecha Perdida 22-ene-2016

Tamaño

Comentarios

Desapareció en el planeta Dagobah mientras luchaba contra L

Añadir nuevo perdido

Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\perdidos\Yodita.perdido

La pestaña perdidos ayuda a gestionar la pérdida de perros recogiendo los datos esenciales para la búsqueda del perro como son la fecha de pérdida, el teléfono de contacto, la foto del perro, o los comentarios que puedan añadir, según el caso, como podrían ser el lugar de pérdida, cómo se perdió, dónde podría encontrarse o alguna marca física característica del perro que permita su identificación. Por defecto, si el usuario no ha especificado la foto del perro, el sistema cargará la de una interrogación.

4.6.- Pestaña Socios

Digital Puppy - ES

Archivo Edición Ayuda


Perros Voluntarios Perdidos **Socios** Cuentas Perrera

Socios

FeverGuy
OldFlow
SwagBoy

Socio

Foto



Seleccionar foto Eliminar foto

Datos

Nombre SwagBoy

Edad 20

Email Diego.Molero@alu.uclm.es

Telefono ***-***-***

Datos bancarios 0983DSAN92384F

Cuota mensual 500€

Borrar socio Guardar cambios

Comentarios

El miedo lleva a la ira, la ira lleva al odio, el odio lleva al sufrimiento

Añadir nuevo socio

Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\socios\SwagBoy.socio

Esta pestaña permite gestionar los socios. El usuario podrá anotar los datos bancarios y seleccionar la cuota mensual que el socio elija para ayudar a la protectora.

4.7.- Pestaña Cuentas

Archivo Edición Ayuda

Perros Voluntarios Perdidos Socios **Cuentas** Perrera

Histórico

Razón	Cantidad (€)	Fecha (d/M/A)
Ingreso	100	25-dic-2015
Robo	-100	26-dic-2015
Donativos	50	31-dic-2015
Comida	-10.5	01-dic-2016
Lotería	100	06-ene-2016
Vacunas	-40	03-ene-2016
Veterinario	-15	03-ene-2016

Gestionar dinero

Añadir

Razón

Cantidad

Añadir entrada

Borrar

Borrar entrada

Guardar

Guardar cambios

Balance actual

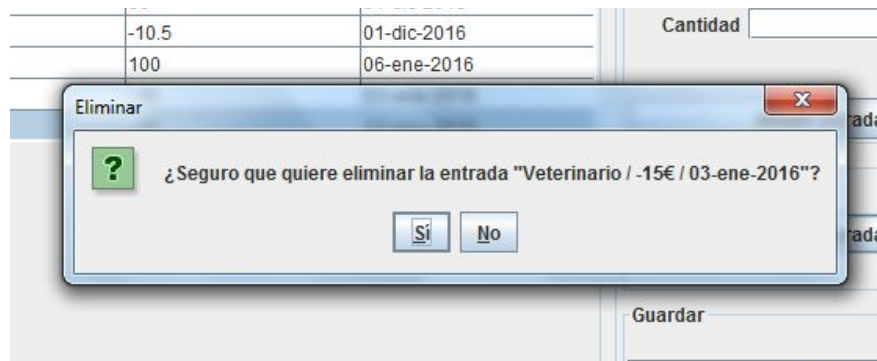
84.5 €

Información

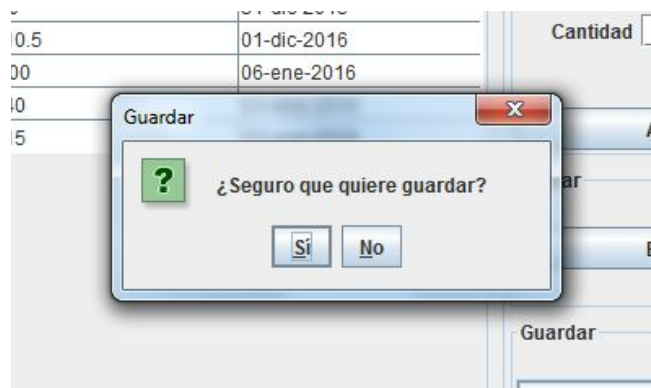
Esta pestaña se centra en el manejo de la contabilidad de la protectora. De nuevo, la pestaña está dividida en dos partes; a la izquierda tenemos el histórico, que nos muestra las tuplas Razón, Cantidad y Fecha de todos los movimientos económicos que se han registrado en la aplicación, así como calcula el balance total de la perrera, que se muestra abajo a la derecha.

En la parte derecha se encuentran las herramientas para gestionar los registros. Podemos añadir una entrada al histórico especificando una razón y una cantidad válidas (si no lo son el sistema nos avisará), ya que la aplicación asignará automáticamente como fecha el día en el que se añada la entrada, aparte de actualizar el balance total con la nueva cantidad.

Además, si seleccionamos un registro del histórico podemos borrarlo del sistema pulsando el botón “Borrar entrada” y “Sí” en el diálogo que aparecerá preguntándonos si estamos seguros de nuestra decisión.

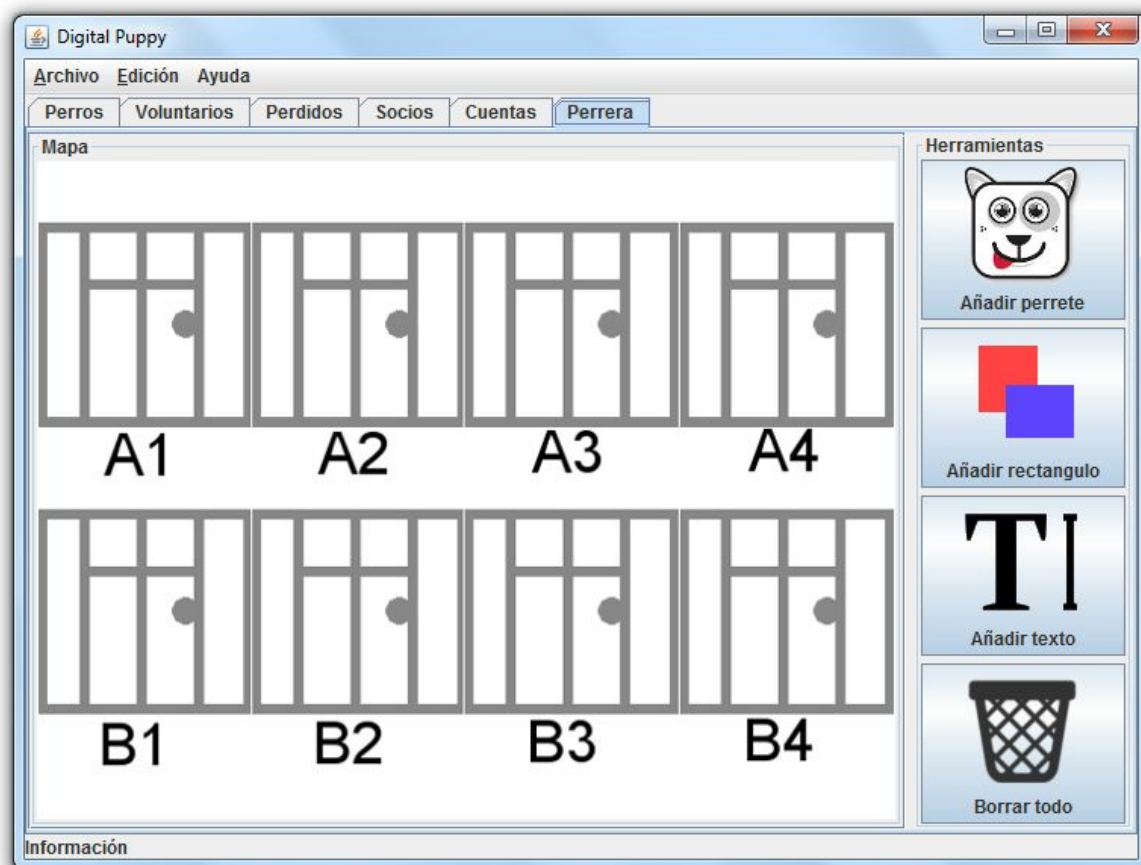


Y, como siempre, debemos recordar guardar los cambios efectuados antes de cerrar la aplicación.



4.8.- Pestaña *Perrera*

Esta pestaña está dedicada al manejo de la herramienta gráfica que permitirá a los usuarios de la aplicación manejar el mapa de la perrera.

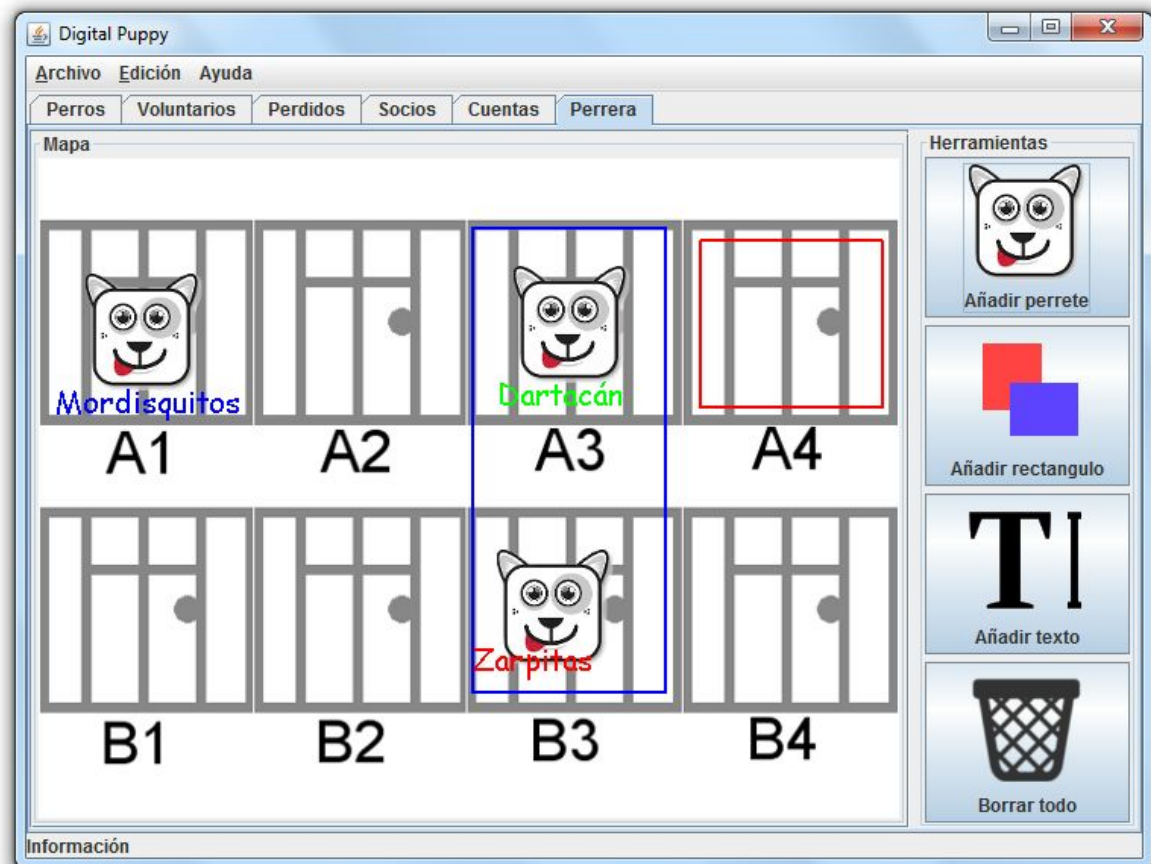


De nuevo tenemos la pantalla dividida en dos secciones, la del mapa, donde dibujaremos y la de las herramientas, que nos permitirán seleccionar diferentes instrumentos para pintar. El mapa está pensado para una perrera pequeña, con tan solo ocho cubiles.

La herramienta está pensada para que los voluntarios de la protectora puedan dibujar en ella la disposición de los diferentes perros, agruparlos mediante rectángulos y añadir anotaciones para así ayudarles sirviendo de apoyo gráfico en las reuniones que tengan.

Mediante el botón “Añadir perrete” podremos dibujar perretes en el mapa, añadir anotaciones con la herramienta “Añadir texto” y rectángulos haciendo click “Añadir rectángulo”. Podemos cambiar el color de rectángulos y textos seleccionando el que queramos en el menú Edición>Color, así como el tamaño y tipo de letra.

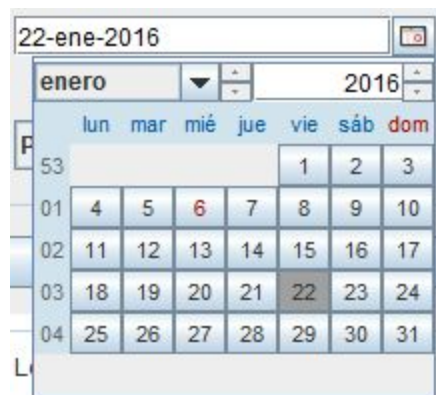
Una vez hayamos acabado con el mapa, o por el contrario si nos hemos equivocado en algo, podremos borrar lo dibujado hasta entonces con el botón “Borrar todo”.



5.- Tecnología y recursos utilizados

En esta sección se comentarán en profundidad los apartados y funcionalidades del programa más interesantes y que hemos encontrado más complejos.

- **JCalendar:** Es un plugin de Windows Builder que facilita la usabilidad de las fechas, permitiendo al usuario seleccionar en un calendario la fecha que quiera de forma muy fácil e intuitiva.



El uso de JCalendar es muy similar al de un JSpinner de tipo Date

```
{
    datechooser = new JDateChooser();
    datechooser.setDateFormatString("dow mon dd hh:mm:ss zzz yyyy");
    GridBagConstraints gbc_datechooser = new GridBagConstraints();
    gbc_datechooser.anchor = GridBagConstraints.ABOVE_BASELINE;
    gbc_datechooser.insets = new Insets(0, 0, 5, 0);
    gbc_datechooser.fill = GridBagConstraints.HORIZONTAL;
    gbc_datechooser.gridx = 1;
    gbc_datechooser.gridy = 4;
    panel_info.add(datechooser, gbc_datechooser);
}
```

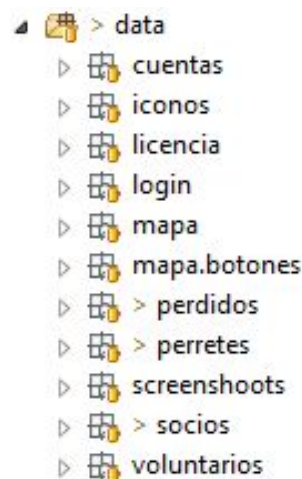
- **JFormattedTextField:** Permite al usuario rellenar de una forma más intuitiva a modo de plantilla un cuadro de texto evitando así posibles fallos con respecto al formato esperado.

Telefono	***_***_***	NIF	XXXXXXXX-X
----------	-------------	-----	------------

Para poder moldear las plantillas hay que usar un objeto de la clase MaskFormatter. En el fragmento siguiente se ve el código utilizado para dar formato a los teléfonos.

```
MaskFormatter formatoTlfno;  
try {  
    formatoTlfno = new MaskFormatter("###-###-###"); //$NON-NLS-1$  
    formatoTlfno.setPlaceholderCharacter('*');  
    tf_telefono = new JFormattedTextField(formatoTlfno);  
} catch (ParseException e) {  
    e.printStackTrace();  
}  
panel_info.add(tf_telefono, gbc_tf_telefono);
```

- **Sistema de almacenamiento (persistencia de datos):** Para poder guardar y cargar todos los datos necesarios para el sistema de información se ha diseñado un sistema de gestión de ficheros.
 - *Jerarquía de datos:* todos los datos se almacenan en la carpeta “data”. Esta contiene varios subdirectorios, los cuales serán accedidos por las diferentes pestañas para obtener los recursos necesarios.



- *Guardar ficheros:* Para almacenar los datos el programa creará un fichero de tipo .perdido (por ejemplo, en el caso de perros perdidos) y lo guardará en el directorio correspondiente a esta pestaña. Los diferentes datos se separan en el fichero mediante saltos de línea, almacenando así datos como: fechas, rutas a las imágenes, comentarios...


```
String stringToSave = nombre + "\r\n" +
    sp_edad.getValue() + "\r\n" +
    raza + "\r\n" +
    telefono + "\r\n" +
    parseFecha(datechooser.getDate()) + "\r\n" +
    cb_tamaño.getSelectedIndex() + "\r\n" +
    lblFoto.getIcon() + "\r\n" +
    tb_comentarios.getText();

if (reply == JOptionPane.YES_OPTION) {
    File file = new File("data/perdidos/" + nombre + ".perdido");

    try {
        FileWriter fw = new FileWriter (file, false);
        fw.write(stringToSave);
        fw.close();
        vp.lblInfo.setText(Messages.getString("PanelPerdidos.61")+ file.getAbsolutePath());
    }
}
```

- *Carga de ficheros:* como ya ha sido explicado en el apartado anterior debido a la distribución y dependiendo del tipo de archivo (.perrete, .socio, .voluntario...) el programa accederá a todos los ficheros contenidos en el directorio correspondiente a la pestaña, los leerá, los convertirá al tipo de formato correspondiente (Date, Int, String..) y cargará los datos en las entradas correspondientes para permitir al usuario consultarlos y/o modificarlos.

```
File file = new File("data/perdidos/" + nombreArchivo); //$NON-NLS-1$



try{
    FileReader fr = new FileReader (file.getAbsolutePath());
    BufferedReader br = new BufferedReader(fr);
    tf_nombre.setText(br.readLine());
    sp_edad.setValue(new Integer(br.readLine()));
    tf_raza.setText(br.readLine());
    tf_telefono.setText(br.readLine());
    datechooser.setDate(parseFecha(br.readLine()));
    cb_tamaño.setSelectedIndex(new Integer(br.readLine()));
    String image = br.readLine();
    if (image.equals("null")) image = "data/perdidos/default.jpg";
    lblFoto.setIcon (new ImageIcon (image));
    String txt = null;
    while ((txt = br.readLine()) != null) tb_comentarios.append(txt+'\n');

    br.close();
    vp.lblInfo.setText(Messages.getString("PanelPerdidos.33")+ file.getAbsolutePath());
}
```

- *Modificar fichero*: para modificar un fichero, el programa lo sobrescribe entero con los nuevos datos proporcionados por el usuario.
- **Sistema de Internacionalización**: El sistema contiene un JButton al inicio de la aplicación, en la ventana login, que permite al usuario cambiar el idioma usado (Inglés-Español). Mediante la clase Messages.java el programa controla el idioma seleccionado.

```
public static String getString(String key) {
    try {
        return RESOURCE_BUNDLE.getString(key);
    } catch (MissingResourceException e) {
        return '!' + key + '!';
    }
}
```

La función *getString(String key)* permite obtener el String correcto, y *RESOURCE_BUNDLE* accede a los ficheros *.properties* que almacenan los valores ya traducidos.

 messages_en.properties
 messages.properties

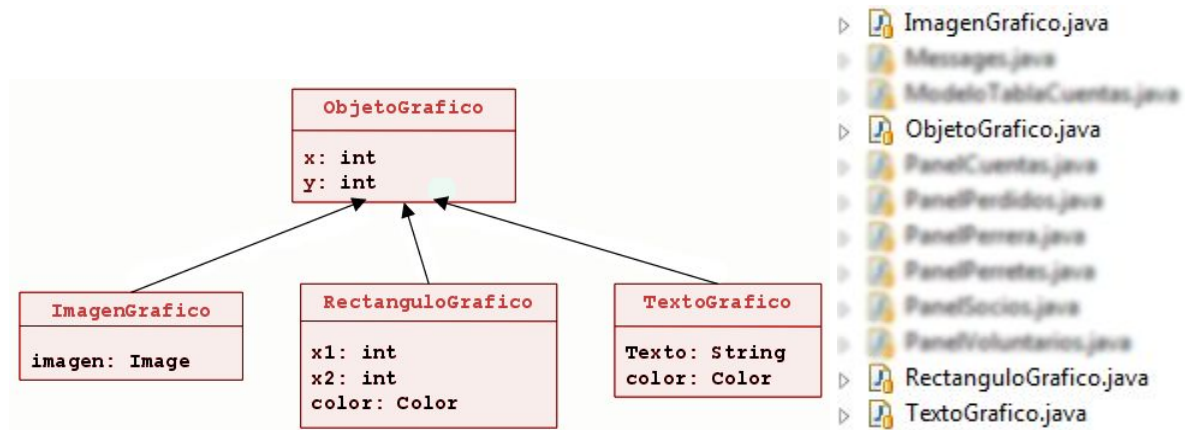
Ejemplo de fichero de traducción messages_en y messages.

```
PanelCuentas.40=Saved
PanelCuentas.41=Error saving file. Retry again
PanelCuentas.5=Manage money
```

```
PanelCuentas.40=Guardado
PanelCuentas.41=Error guardando el archivo. Reinicie el programa y vuelva a intentarlo
PanelCuentas.5=Gestionar dinero
```

- **Sistema de dibujo**: la aplicación, como ya ha sido comentando antes, consta de una pestaña en la que el usuario puede dibujar sobre la imagen de un mapa que actuará como lienzo. Todos los elementos proporcionados por Swing tienen un elemento Graphics asociado y trabajaremos con estos para desarrollar esta funcionalidad.

Cuatro clases han sido definidas para esta función. La primera es ObjetoGrafico, de la que extienden las otras tres, cada una centrada en un tipo de gráfico distinto; incrustar imágenes, dibujar rectángulos o incluir texto.



Cuando el usuario clickea en un botón de la pestaña de dibujo una variable interna (*modo*) cambia en consecuencia, para así recordar su selección y se espera a que el usuario interactúe con el mapa. A continuación se muestra el código ejecutado por el programa cuando el usuario hace click en el lienzo, muy parecido al visto en clase.

```

private class AreaDibujoMouseListener extends MouseAdapter {
    @Override
    public void mousePressed(MouseEvent e) {
        System.out.println(modo);
        x = e.getX();
        y = e.getY();

        switch (modo) {
            case PERRETE:
                areaDibujo.addObjetoGrafico(new ImagenGrafico(x, y, imgPerrete));
                areaDibujo.repaint();
                break;

            case RECTANGULO:
                areaDibujo.addObjetoGrafico(new RectanguloGrafico(x,y,x,y, colorDibujado));
                break;

            case TEXTO:
                txtTexto.setBounds(x, y, 200,30);
                txtTexto.setVisible(true);
                txtTexto.requestFocus();
                txtTexto.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent arg) {

                        if(!txtTexto.getText().equals("")) {
                            areaDibujo.addObjetoGrafico(new TextoGrafico(x, y+15,
                                txtTexto.getText(), colorDibujado));
                        }
                        txtTexto.setText("");
                        txtTexto.setVisible(false);
                        areaDibujo.repaint();
                    }
                });
                areaDibujo.add(txtTexto);
                break;
        }
    }
}

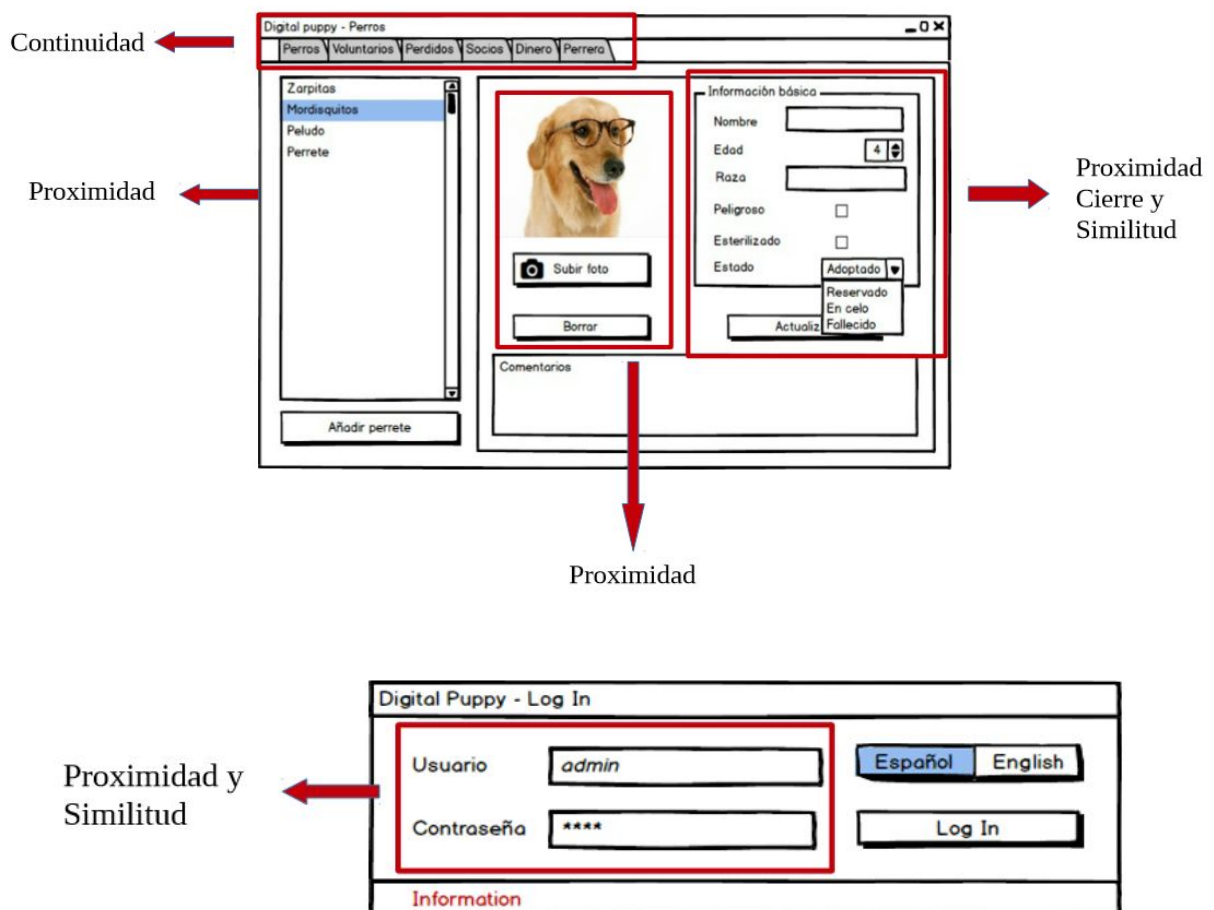
```

Dependiendo del modo en el que estemos el sistema añadirá el dibujo de un perrete al mapa en las coordenadas que el usuario haya elegido, definirá un nuevo rectángulo desde donde el usuario pinchó hasta donde dejó de arrastrar o creará una textbox temporal en la que el usuario podrá escribir el texto que quiera añadir al lienzo.

6.- Justificación del diseño de la GUI en base a lo estudiado en teoría

Para la elaboración de este prototipo de software se ha realizado previamente un boceto mediante la herramienta Balsamiq Mockups, que nos sirvió para tener una idea conceptual del actual prototipo. Gracias al boceto pudimos hacer una pre evaluación antes de empezar a desarrollar el prototipo.

El boceto fue analizado acorde con las leyes de Gestalt como se muestra a continuación:



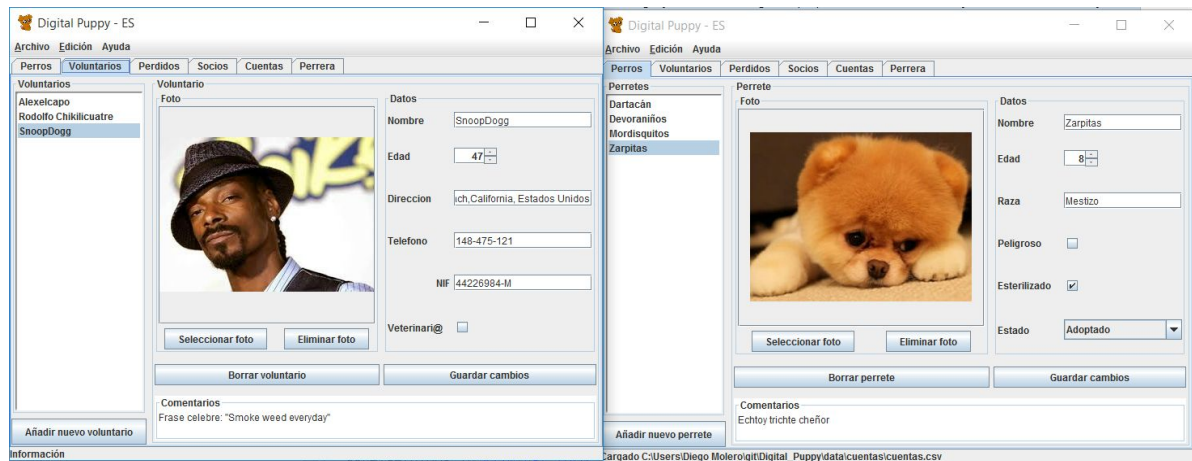
Salvo pequeños cambios, el prototipo ha seguido a grandes rasgos el boceto, como se mostrará en el apartado 7.

El uso de la aplicación y sus funciones es muy orientativo (*affordances*), usa un sistema de pestañas con el que el usuario se siente familiarizado y cómodo.

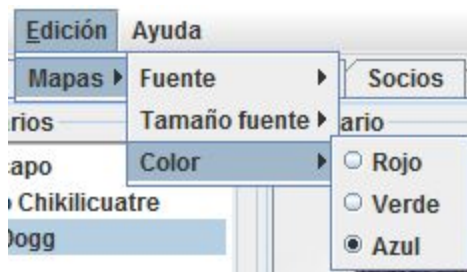
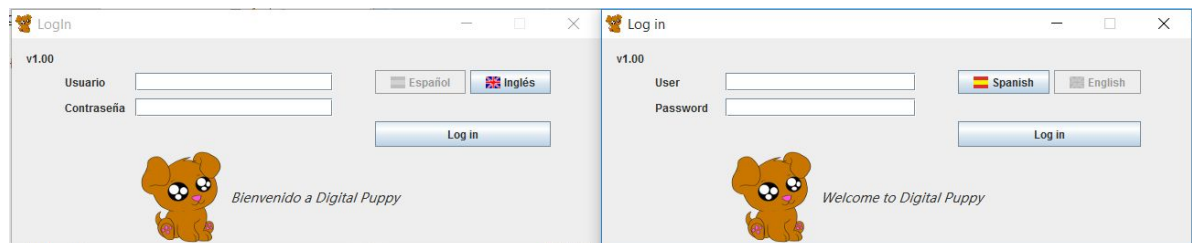
Se ha realizado una evaluación de las heurísticas de usabilidad de la interfaz de usuario:

Comunicación: la aplicación ofrece un dialogo con el usuario empleando un lenguaje común y entendible.

Consistencia: hay uniformidad tanto en el funcionamiento como en la representación de mensajes:



Flexibilidad: el programa facilita tanto el uso de la diversidad de usuarios (ingles-español) como la posibilidad de cambiar opciones estéticas:



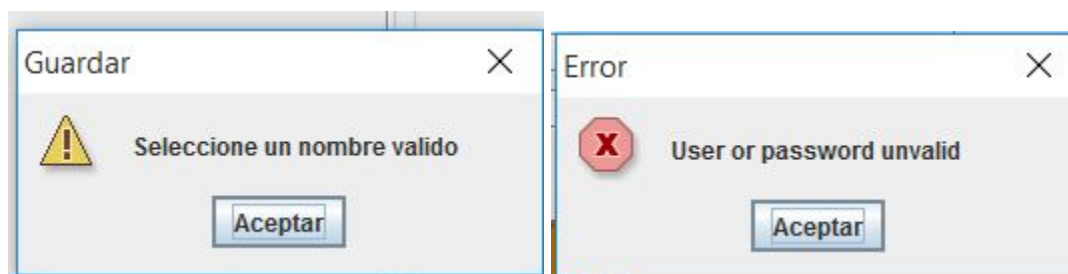
Realimentación: mantiene al usuario en todo momento informado de la actividad que realiza:



Ayuda: El usuario tiene información adicional debajo de la ventana y en las herramientas de la parte superior.

Robustez: El sistema funciona correctamente, se han realizado muchas pruebas de testing.

Minimizar errores: el programa avisa de error mediante mensajes de error:



Atractivo: la aplicación es atractiva para el usuario:

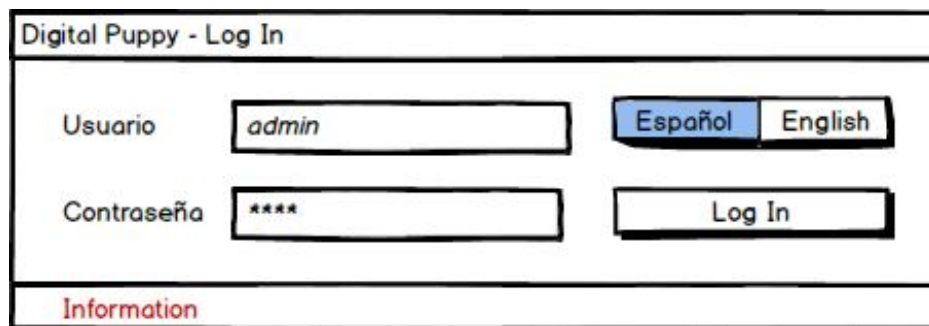


Estandarización: el aplicación usa el sistema de ventanas y pestañas, asi como una barra de herramientas y botones, que está muy familiarizado y estandarizado

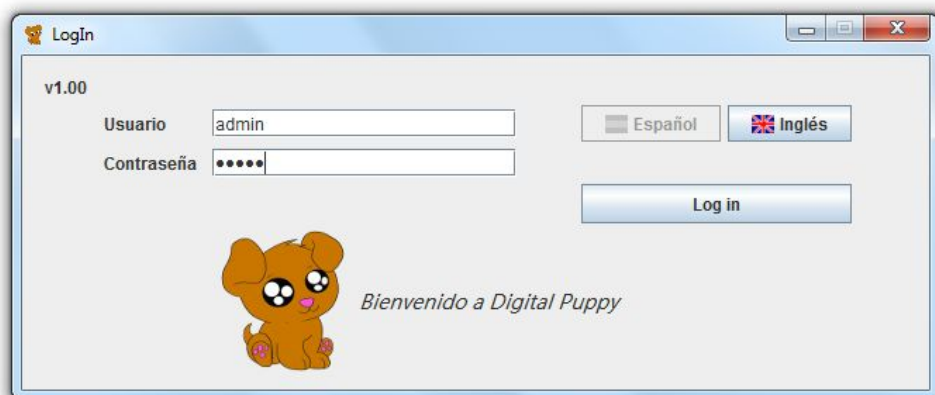
7.- Mockups vs Implementaciones

En esta sección se compararán y comentarán las similitudes y diferencias entre el boceto realizado para el primer hito y el resultado final del prototipo.

7.1.- Ventana de login

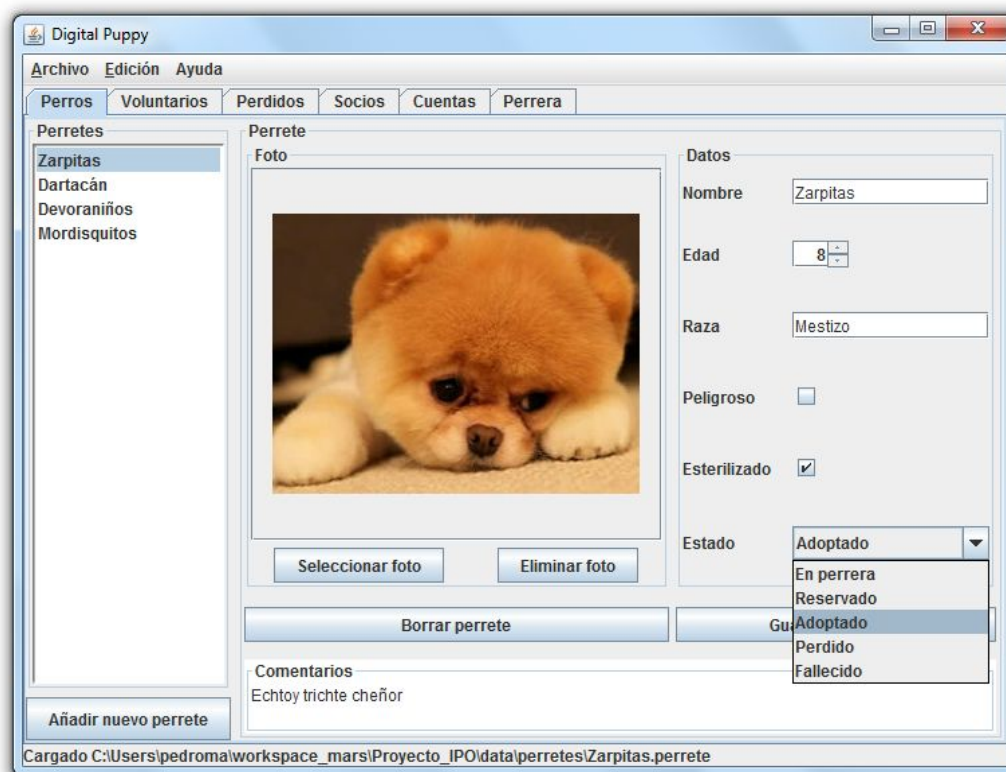
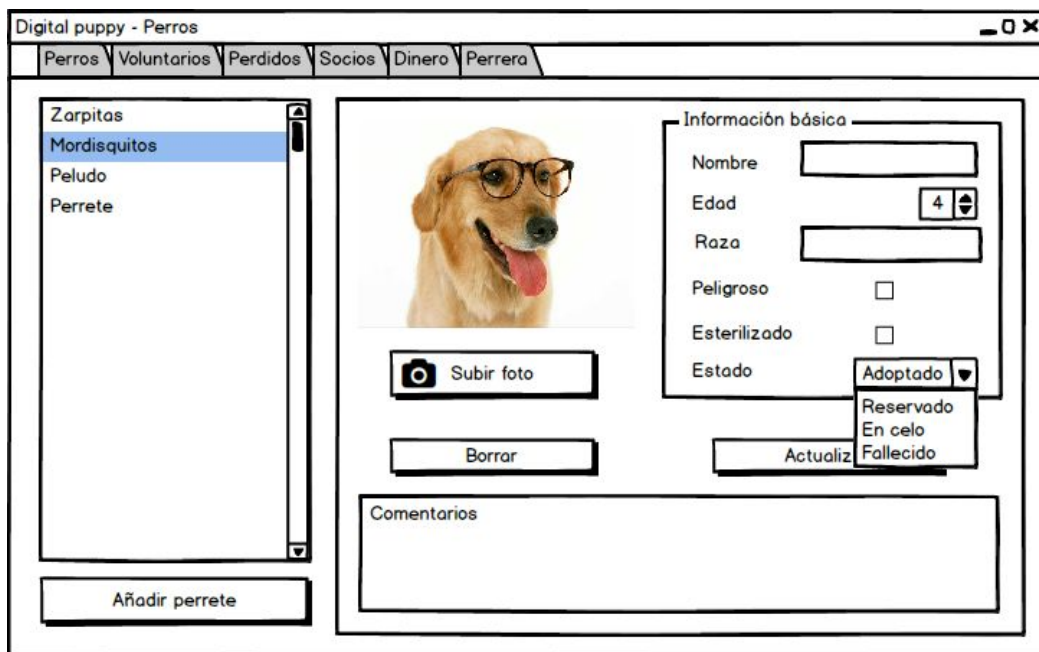


A hand-drawn mockup of a login window titled "Digital Puppy - Log In". It features two input fields: "Usuario" with the text "admin" and "Contraseña" with four asterisks. To the right of the password field is a "Log In" button. Above the password field are two language selection buttons: "Español" (highlighted in blue) and "English". At the bottom of the window is a red "Information" label.



La ventana de inicio ha acabado siendo muy parecida, aunque finalmente se ha eliminado la etiqueta de información en pos de los diálogos de información y añadido el logo de la protectora.

7.2.- Pestaña Perros



En este caso la implementación final se ha mantenido fiel en casi todos los aspectos, sin contar nuevas áreas bordeadas y el nuevo botón "Eliminar foto" que aporta una funcionalidad en la que no pensamos mientras desarrollamos el mockup.

7.3.- Pestaña Voluntarios

Digital puppy - Voluntarios

Perros Voluntarios Perdidos Socios Dinero Perrera

Randy Orton

The Undertaker

Kane

The Rock


Mysterio


John Cena

Hulk Hogan

Stone Cold

HHH



 Subir foto

Borrar

Actualizar

Comentarios

Información básica

Nombre

Edad

Dirección

Teléfono

Conocimientos veterinarios

24

☐

Añadir voluntario

Digital Puppy - ES

Archivo Edición Ayuda

Perros Voluntarios Perdidos Socios Cuentas Perrera

Voluntarios


Alexelcapo

Rodolfo Chikilicuatre

SnoopDogg

Voluntario

Foto



Seleccionar foto

Eliminar foto

Borrar voluntario

Guardar cambios

Comentarios

Sube gameplays de "the binding of isaac"

Datos

Nombre

Edad

Dirección

Teléfono

NIF

Veterinari@

Alexelcapo

34

Las Palmas

433-551-234

59182374-F

☒

Añadir nuevo voluntario

Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\voluntarios\Alexelcapo.voluntario

Como en el caso anterior, el resultado de la implementación es casi idéntico al del boceto original.

7.4.- Pestaña Perdidos

Wireframe of the 'Perdidos' (Lost) tab in the 'Digital puppy' application. The window title is 'Digital puppy - Perdidos'. The navigation bar includes 'Perros', 'Voluntarios', 'Perdidos', 'Socios', 'Dinero', and 'Perrera'. On the left, a list shows 'Perdido 1', 'Perdido 2', '...', and 'Perdido n', with a scroll bar and an 'Añadir perrete perdido' button at the bottom. The main area features a large placeholder for a photo (a black scribble), a 'Subir foto' button, and 'Borrar' and 'Actualizar' buttons. To the right, the 'Información básica' section contains fields for 'Nombre', 'Edad' (with a spinner set to 4), 'Raza', 'Fecha perdida' (20/11/15 with a calendar icon), and 'Tamaño' (a dropdown menu with 'Grande', 'Mediano', and 'Pequeño' options). At the bottom is a 'Descripción adicional' text area.

Final design of the 'Perdidos' (Lost) tab in the 'Digital Puppy - ES' application. The window title is 'Digital Puppy - ES'. The menu bar includes 'Archivo', 'Edición', and 'Ayuda'. The navigation bar includes 'Perros', 'Voluntarios', 'Perdidos', 'Socios', 'Cuentas', and 'Perrera'. On the left, a list shows 'Perdidos', 'Darth Dog', and 'Yodita', with an 'Añadir nuevo perdido' button at the bottom. The main area features a 'Perdido' section with a 'Foto' placeholder (a large question mark), 'Seleccionar foto' and 'Eliminar foto' buttons, and 'Borrar perdido' and 'Guardar cambios' buttons. To the right, the 'Datos' section contains fields for 'Nombre' (Darth Dog), 'Edad' (2 with a spinner), 'Raza' (Dark Lord), 'Telefono' (956-235-745), 'Fecha Perdida' (16-ene-2015 with a calendar icon), and 'Tamaño' (Grande with a dropdown arrow). At the bottom is a 'Comentarios' section with the text 'Se dio a la fuga tras aniquilar a media galaxia'. The status bar at the bottom shows the file path: 'Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\perdidos\Darth Dog.perdido'.

De nuevo, el diseño final se ha basado casi por completo en el boceto que realizamos al principio del proyecto.

7.5.- Pestaña Socios

Digital puppy - Socios

Perros | Voluntarios | Perdidos | Socios | Dinero | Perrera

Diego

Jaime


Jose Antonio


Juan

Pedroma

Paco

Jaime



 Subir foto

Borrar

Actualizar

Comentarios

Información básica

Nombre

Edad

Datos bancarios

Telefono

Cantidad aportada

Añadir socio

Digital Puppy - ES

Archivo | Edición | Ayuda

Perros | Voluntarios | Perdidos | Socios | Cuentas | Perrera

Socios


FeverGuy

OldFlow

SwagBoy

Socio

Foto



Seleccionar foto

Eliminar foto

Borrar socio

Guardar cambios

Comentarios

El miedo lleva a la ira, la ira lleva al odio, el odio lleva al sufrimiento

Datos

Nombre

Edad

Email

Telefono

Datos bancarios

Cuota mensual


Añadir nuevo socio

Cargado C:\Users\pedroma\workspace_mars\Proyecto_IPO\data\socios\SwagBoy.socio

Tanto el mockup inicial como el resultado del desarrollo son casi idénticos entre sí.

7.6.- Pestaña *Cuentas*

[illegible]


Digital Puppy

Archivo Edición Ayuda

Perros Voluntarios Perdidos Socios Cuentas Perrera

Histórico

Razón	Cantidad (€)	Fecha (d/M/A)
ingreso	100	25-dic-2015
robo	-100	26-dic-2015
donativos	50	31-dic-2015
comida	-10.5	01-dic-2016
lotería	100	06-ene-2016
Vacunas	-40	03-ene-2016
Veterinario	-15	03-ene-2016

Gestionar dinero

Añadir

Razón

Cantidad

Añadir entrada

Borrar

Borrar entrada

Guardar

Guardar cambios

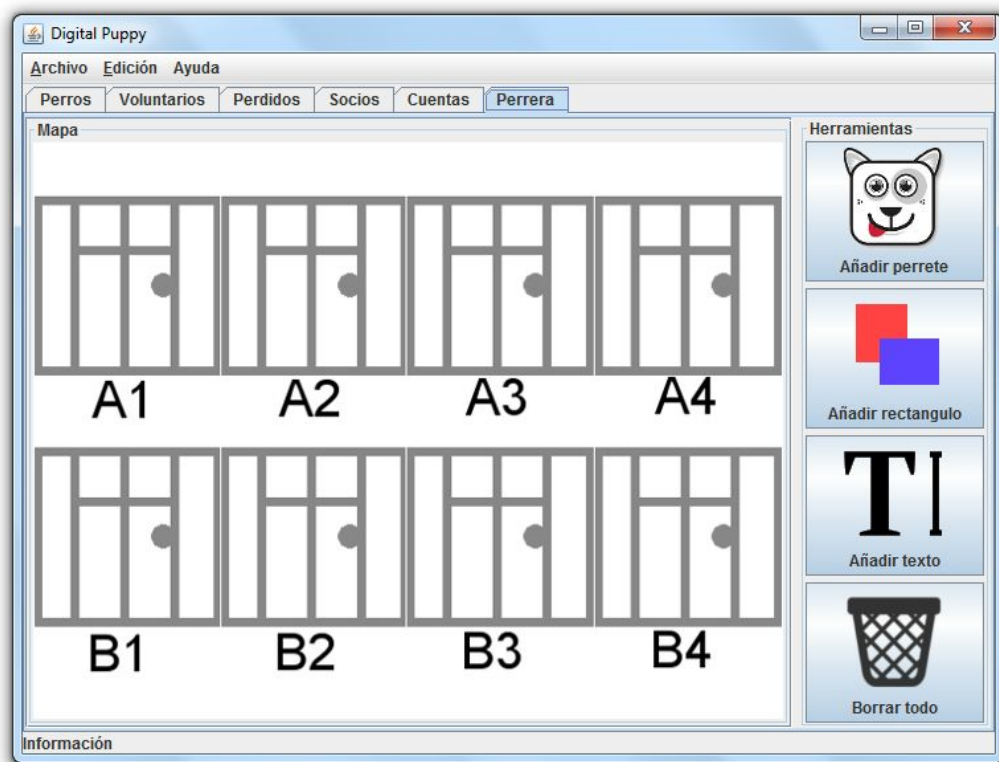
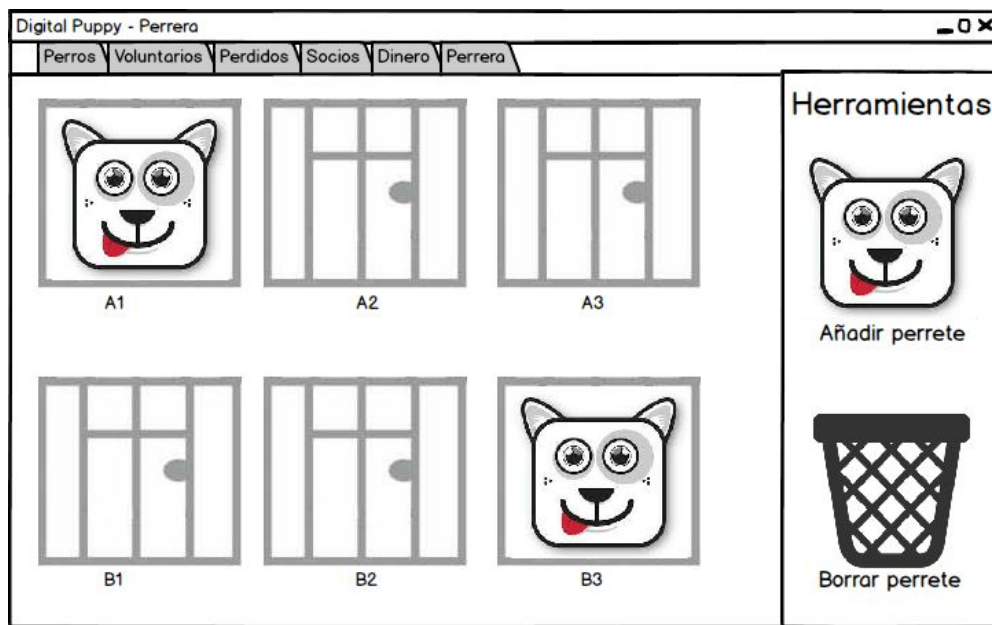
Balance actual

84.5 €

Información

Los diferentes botones “Añadir gasto” y “Añadir ganancia” que inicialmente se propusieron han sido fusionados en “Añadir entrada”, delegando de este modo la responsabilidad de decidir si es gasto o ganancia en el signo (positivo o negativo) con el que el usuario introduzca la cantidad.

7.7.- Pestaña *Perrera*



Esta pestaña ha sido la que más cambios ha sufrido en cuanto a la idea primigenia. Se han añadido botones en relieve para permitir al usuario identificar más fácilmente la funcionalidad de los mismos y se han añadido dos nuevas herramientas, “Añadir rectángulo” y “Añadir texto”, ambas autoexplicativas, así como dos jaulas más en el mapa.

8.- Webgrafía

- <http://stackoverflow.com/questions/3751791/how-to-change-default-text-file-encoding-in-eclipse>
- <http://stackoverflow.com/questions/1090098/newline-in-jlabel>
- <http://stackoverflow.com/questions/19433358/difference-between-dispose-and-exit-on-close-in-java>
- <http://stackoverflow.com/questions/527719/how-to-add-hyperlink-in-jlabel>
- <http://stackoverflow.com/questions/5175728/how-to-get-the-current-date-time-in-java>
- <http://stackoverflow.com/questions/5766318/convert-double-to-string>
- <http://stackoverflow.com/questions/5769669/convert-string-to-double-in-java>
- <http://stackoverflow.com/questions/29060364/java-how-to-save-a-file-with-current-date-and-time>
- <http://stackoverflow.com/questions/1614772/how-to-change-jframe-icon>
- <http://stackoverflow.com/questions/4548816/when-should-we-implement-serializable-interface>
- <http://stackoverflow.com/questions/7899525/how-to-split-a-string-by-space>
- <http://stackoverflow.com/questions/5889034/how-to-initialize-an-array-of-objects-in-java>
- <http://stackoverflow.com/questions/5834014/lf-will-be-replaced-by-crlf-in-git-what-is-that-and-is-it-important>
- <http://stackoverflow.com/questions/1816673/how-do-i-check-if-a-file-exists-in-java>
- <http://stackoverflow.com/questions/1016278/is-this-the-best-way-to-rewrite-the-content-of-a-file-in-java>
- <http://stackoverflow.com/questions/12461627/jlist-fires-valuechanged-twice-when-a-value-is-changed-via-mouse>
- <http://stackoverflow.com/questions/16063429/java-lang-nullpointerexception-on-fast-update-of-jlist>
- <http://stackoverflow.com/questions/9199899/java-swing-and-event-errors>
- <http://stackoverflow.com/questions/4570037/java-substring-index-range>
- <http://stackoverflow.com/questions/4917326/how-to-iterate-over-the-files-of-a-certain-directory-in-java>
- <http://stackoverflow.com/questions/21060992/how-does-java-resolve-a-relative-path-in-new-file>
- <http://stackoverflow.com/questions/2593154/get-a-resource-using-getresource>
- <http://stackoverflow.com/questions/10375115/jlist-getmodel-classcastexception>