

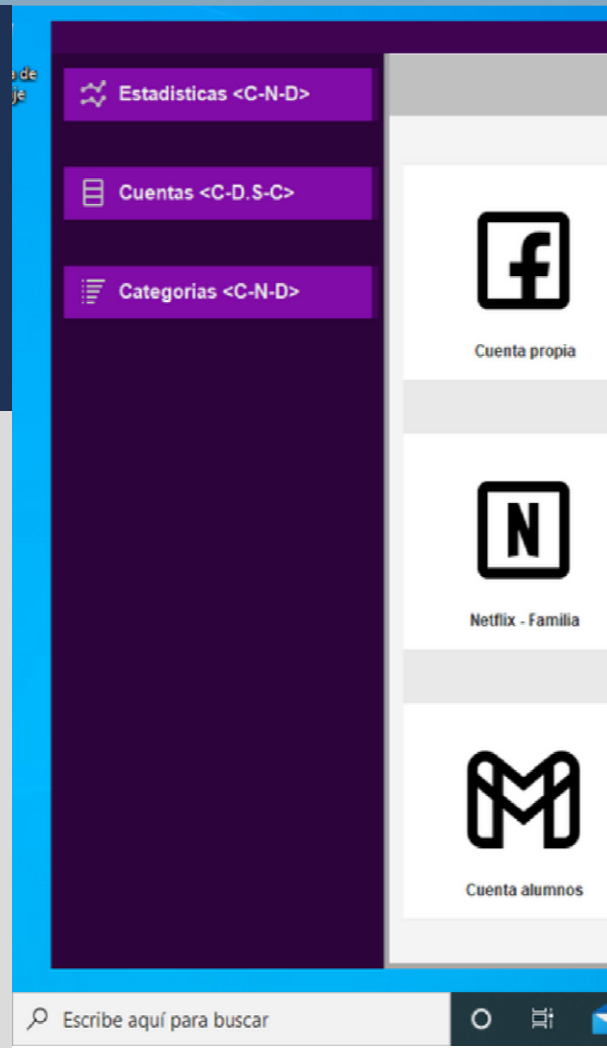


# GESTOR DE CONTRASEÑAS

manual técnico

ACTIVIDAD INTEGRADORA  
DE DESARROLLO DE SOFTWARE

DESARROLLADO POR  
Nestor Emmanuel Ascencio Sánchez  
SEMESTRE 5 DEL GRUPO A
















# MANUAL TECNICO

El siguiente manual técnico se desarrolló para el correcto mantenimiento y comprensión del software desarrollado (Gestor de Contraseñas) y su comprensión, en el siguiente manual se mencionan las especificaciones del programa, funciones del software, así como corrección de errores y algunas características del mismo.



## ANTES DE EMPEZAR...

El software esta desarrollado en Java SE-18, aun así mantiene compatibilidad con versiones de java posteriores a Java SE-8 a continuación le mostramos dos configuraciones principales guardadas dentro del proyecto para su funcionamiento correcto en su respectiva versión de java.

1- Abra la carpeta del proyecto, localice la carpeta llamada ".settings" y ábrala.

	.settings	04/12/2022 10:10 p. m.	Carpeta de archivos	
	access	04/12/2022 07:26 p. m.	Carpeta de archivos	
	bin	04/12/2022 10:19 p. m.	Carpeta de archivos	
	database	04/12/2022 10:14 p. m.	Carpeta de archivos	
	Java-1.8-settings	05/12/2022 05:39 p. m.	Carpeta de archivos	
	Java-1.18-settings	05/12/2022 05:39 p. m.	Carpeta de archivos	
	src	04/12/2022 10:07 p. m.	Carpeta de archivos	
	.classpath	04/12/2022 10:19 p. m.	Archivo CLASSPATH	1 KB
	.project	30/11/2022 07:01 p. m.	Archivo PROJECT	1 KB
	jgoodies-forms-1.8.0.jar	02/12/2022 03:57 p. m.	Executable Jar File	131 KB
	jgoodies-forms-1.8.0-sources.jar	02/12/2022 03:57 p. m.	Executable Jar File	172 KB
	miglayout15-swing.jar	02/12/2022 03:56 p. m.	Executable Jar File	86 KB
	miglayout-src.zip	02/12/2022 03:56 p. m.	Archivo WinRAR Z...	182 KB

2- Localice el archivo llamado "org.eclipse.jdt.core.prefs" y ábralo.

	org.eclipse.core.resources.prefs	30/11/2022 07:01 p. m.	Archivo PREFS	1 KB
	org.eclipse.jdt.core.prefs	04/12/2022 10:19 p. m.	Archivo PREFS	1 KB

3- localice los siguientes parámetros de configuración y cambios al número de versión que este usando.

```
org.eclipse.jdt.core.compiler.codegen.targetPlatform=18
org.eclipse.jdt.core.compiler.compliance=18
org.eclipse.jdt.core.compiler.source=18
```

4- Vuelva a abrir la carpeta principal del proyecto, localice el archivo ".classpath". y ábralo.

.settings	04/12/2022 10:10 p. m.	Carpeta de archivos	
access	04/12/2022 07:26 p. m.	Carpeta de archivos	
bin	04/12/2022 10:19 p. m.	Carpeta de archivos	
database	04/12/2022 10:14 p. m.	Carpeta de archivos	
Java-1.8-settings	05/12/2022 05:39 p. m.	Carpeta de archivos	
Java-1.18-settings	05/12/2022 05:39 p. m.	Carpeta de archivos	
src	04/12/2022 10:07 p. m.	Carpeta de archivos	
.classpath	04/12/2022 10:19 p. m.	Archivo CLASSPATH	1 KB
.project	30/11/2022 07:01 p. m.	Archivo PROJECT	1 KB
jgoodies-forms-1.8.0.jar	02/12/2022 03:57 p. m.	Executable Jar File	131 KB
jgoodies-forms-1.8.0-sources.jar	02/12/2022 03:57 p. m.	Executable Jar File	172 KB
miglayout15-swing.jar	02/12/2022 03:56 p. m.	Executable Jar File	86 KB
miglayout-src.zip	02/12/2022 03:56 p. m.	Archivo WinRAR Z...	182 KB

5- localice el siguiente classpathentry y cambie el apartado de path por la versión que esta usando.

```
<classpathentry exported="true" kind="con" |  
path="org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-18">  
  <attributes>  
    <attribute name="module" value="true"/>  
  </attributes>
```

## APLICACION DE LAS COMPETENCIAS DESARROLLADAS EN EL MODULO DE DESARROLLO DE SOFTWARE.

Las competencias que se desarrollaron en el módulo de desarrollo de software se aplicaron al utilizar una técnica de programación como lo es la programación orientada a objetos y el utilizar herramientas para desarrollar los diagramas necesarios para explicar y dar a entender el funcionamiento del software así como el abordamiento de un problema de interés social como lo es el uso excesivo e inseguro de una misma contraseña en diferentes cuentas.

### QUE ES "GESTOR DE CONTRASEÑAS"?

Para explicar el contexto del software tenemos que explicar el porqué de este. Abordamos el tema en seguridad de cuentas al tener el problema que hoy en día se presenta, la mayoría de las personas usan la misma contraseña para dos o más cuentas dejando más de una en una vulnerabilidad alta, en otras ocasiones donde usan una contraseña diferente la mayoría de las personas se les olvida las credenciales y pierden acceso a su cuenta. La funcionalidad principal de programa es poder agregar una o más credenciales de cuentas como Netflix, Twitter, Facebook, Gmail, etc. Agregando más seguridad al usuario de poder usar una contraseña diferente en cada cuenta que se crea en la internet sin necesidad de recordarla o con miedo a perderla. Utilizamos la técnica de programación orientada a objetos a razón de que es una de las más usadas y más fuertes dentro del tema de programación, al dividir los segmentos de la página el clases y objetos para el fácil manejo de ítems (Cuentas o credenciales). En el software se hizo el uso de archivos (no encriptados) para poder mantener la información a salvo de perdida de datos y seguridad de inicio de sección del software.

### ALGORITMO DEL SOFTWARE:

D.E: mail, pass, itname, itemail, itpass, ittype.

D.S: itname, itemail, itpass, ittype.

Inicio

1. Leer mail
2. Leer pass
3. Autenticar al usuario con sus credenciales
4. Mostrar el menú principal
5. Mostrar ítems creados
6. Leer itname
7. Leer itemail
8. Leer itpass
9. Leer ittype
10. Añadir ítem (Cuenta o carenciales)
11. Mostrar ítems creados

Fin

## DIAGRAMA DE CLASES:

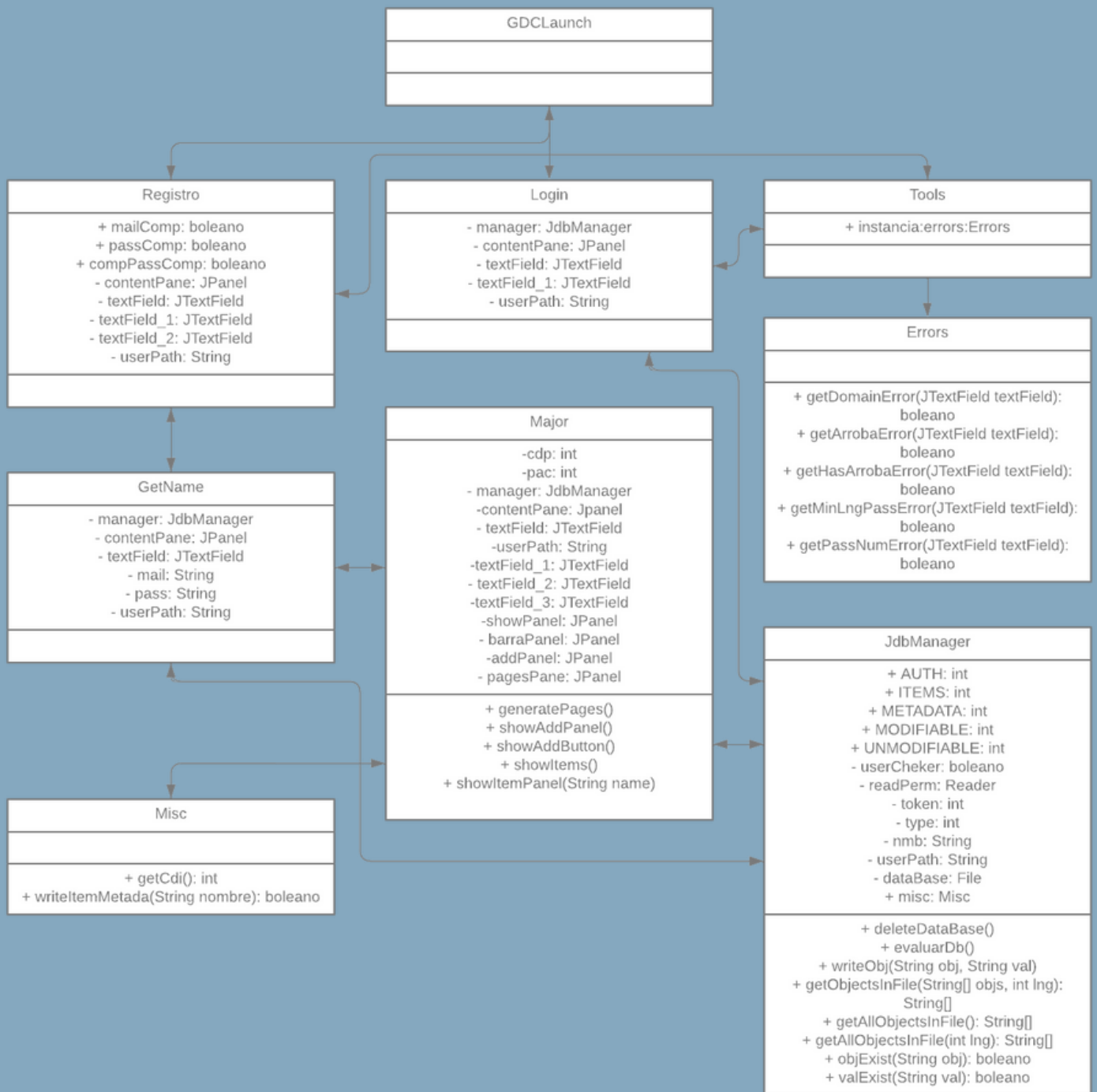
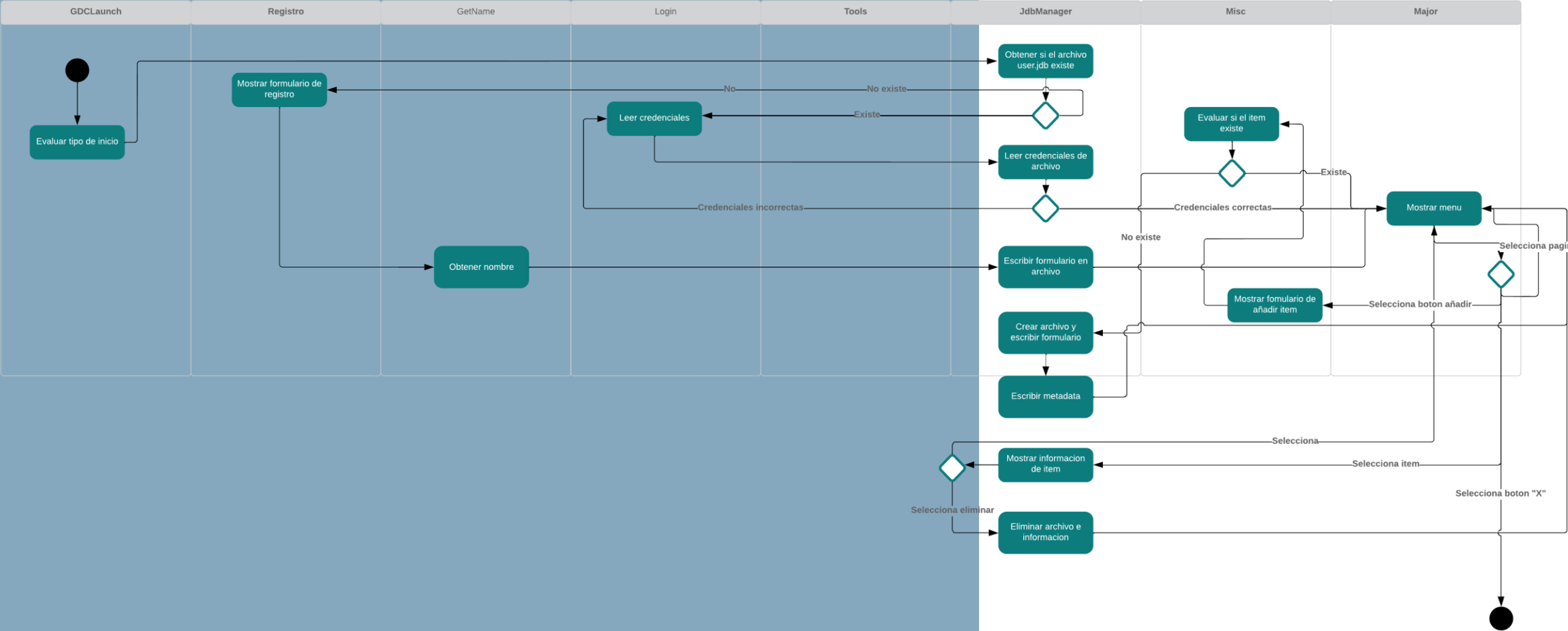
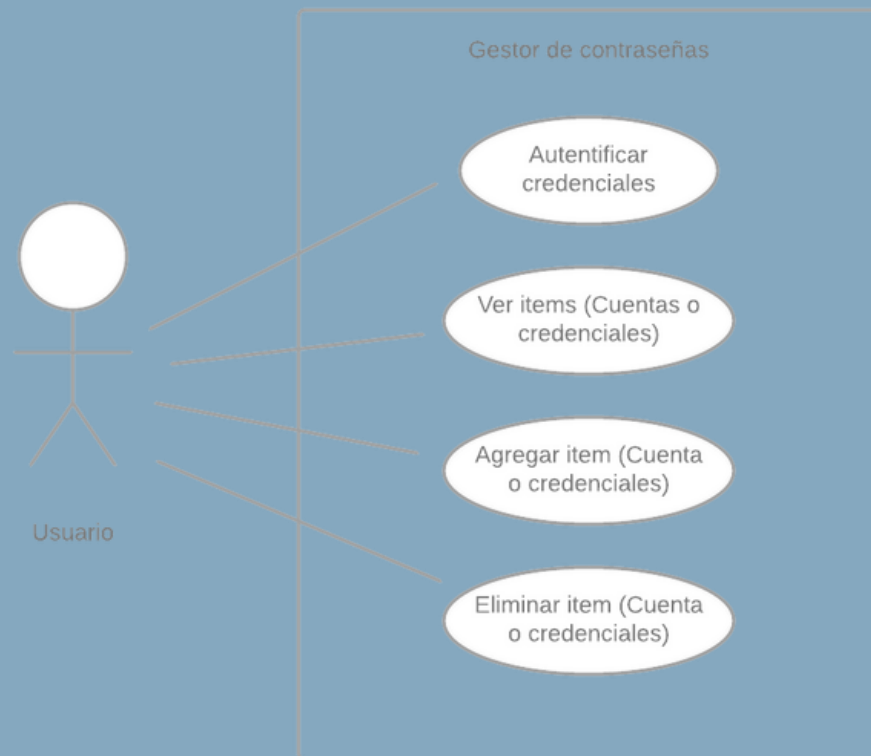


DIAGRAMA DE ACTIVIDADES:



## DIAGRAMA DE CASOS DE USO:



## CODIGO EN JAVA:

Antes de ejecutar el Código algunos errores que existen son al cambiar de versión de java no funcionara de manera correcta el autenticador para saber si el programa es usado por primera vez o no. El programa se inicia con la clase GDCLaunch.java en lunarsoftware.gdc.

Se a agregado una etiqueta <E-D> que es la etiqueta de una función aun en desarrollo.

De preferencia borrar la carpeta "database" ubicada en la carpeta principal del proyecto para evitar errores de autenticación.

```
package lunarsoftware.gdc;
```

```
import java.awt.Desktop;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import javax.swing.JOptionPane;
```

```
import lunarsoftware.gdc.gui.auth.Login;
```

```
import lunarsoftware.gdc.gui.auth.Registro;
```

```
import lunarsoftware.lib.jdbmanager.JdbManager;
```

```
/**
```

```
 *   Nombre del software: Gestor de contraseñas
```

```
 *   @author: Nestor Emmanuel
```

```
 *
```

```
 *   Fecha de inicio: 1/12/2022
```

```
 *   Ultima modificacion: 4/12/2022
```

```
 */
```

```
public class GDCLaunch{
```

```
    static boolean DEV = true;
```

```
    public static void main(String[] args) {
```

```
        JdbManager write = new JdbManager(JdbManager.AUTH, JdbManager.UNMODIFIABLE, null);
```

```
        /**
```

```
         *   Aqui checo que el software no se encuentre en desarrollo y
```

```
         *   sea seguro ingresar datos personales, de otra forma le informo
```

```
         *   al usuario que el software esta en modo desarrollo y no es seguro.
```

```
        */
```

```
        JOptionPane.showMessageDialog(null, "Alerta: El software esta en modo desarrollo, los datos aqui guardados\n"
```

```
        + ""
```

```
        + ""
```

```
        + "no son de ninguna forma encriptados, porfavor no ingrese ningun tipo\n"
```

```
        + "de dato personal, el programa solo muestra la funcionalidad principal."
```

```
        + "", null, 2, null);
```

```
        //Posdata: No alcance a terminar todas las funcionalidades que queria agregar :C
```

```
        /*try {
```

```
            Desktop.getDesktop().open(new
```

```
File("D:\\Compu\\Projects\\JavaWorkspaces\\GestorDeContraseñas\\src\\lunarsoftware\\gdc\\oas.docx"));
```

```
        } catch (IOException e) {
```

```
            // TODO Auto-generated catch block
```

```
            e.printStackTrace();
```



```

    }*/

/**
 * Aqui lanza el frame(Ventana) correnpodiente a si el usuario esta o no registrado
 */

if(write.getUserCheker()) {
    Login logging = new Login();
    Login.main(null);
}else {
    Registro registro = new Registro();
    registro.main(null);
}
}
}

package lunarsoftware.gdc;

import javax.swing.JTextField;

/**
 * Clase usada como caja de herrmanietas
 */
public class Tools{
    /**
     * Variable estatica par acceder a la clase Errors
     * de obtencion de errores
     */
    public static Errors error = new Errors();

    /**
     * Clase creada para la obtencion de errores
     */
    public static class Errors{

        public boolean getHasArrobaError(JTextField textField) {
            /**
             * Funcion que devuelve un booleano si el
             * correo contiene un "@"
             */
            boolean rtn = false;
            if(!textField.getText().contains("@")) {
                rtn = true;
            }
            return rtn;
        }

        public boolean getArrobaError(JTextField textField) {
            /**
             * Fuencion que devuelve un booleano si el correo
             * contiene mas de dos "@"
             */
            boolean rtn = false;
            for(int i=0;i<textField.getText().length();i++) {
                if(textField.getText().charAt(i) == '@') {
                    for(int o=i+1;o<textField.getText().length();o++) {
                        if(textField.getText().charAt(o) == '@') {
                            rtn = true;
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

```

        break;
    }else {rtn = false;}
}
return rtn;
}

public boolean getDomainError(JTextField textField) {
/**
 * Funcion que devuelve un booleano si el dominio
 * contiene una direccion ip o dns valida(like .com .net etc)
 */
boolean rtn = false;
for(int i=0;i<textField.getText().length();i++) {
    if(textField.getText().charAt(i) == '@') {
        for(int o=i+1;o<textField.getText().length();o++) {
            if(textField.getText().charAt(o) == '.' &&
textField.getText().length() != o + 1) {
                rtn = false;
                break;
            }else {
                rtn = true;
            }
        }
        break;
    }else {rtn = true;}
}
return rtn;
}

public boolean getMinLngPassError(JTextField textField) {
/**
 * Funcion que retorna un booleano si la contraseña no es mayor
 * de 8 caracteres
 */
boolean rtn = false;
if(textField.getText().length() < 8) {
    rtn = true;
}
return rtn;
}

public boolean getPassNumError(JTextField textField) {
/**
 * Funcion que retorna un booleano si la contraseña
 * contiene uno o mas numeros
 */
boolean rtn = false;
for(int ot=0;ot<=9;ot++) {
    if(textField.getText().contains(""+ot)) {
        rtn = false;
        break;
    }else rtn = true;
}
return rtn;
}
}
}
}

```

```
package lunarsoftware.lib.jdbmanager;
```

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;

```

```

import java.io.Reader;

import lunarsoftware.lib.jdbmanager.misc.Misc;

public class JdbManager{
    /**
     * Variables estaticas para identificar al tipo
     * de base de datos a la que se va a acceder
     */
    public static int AUTH = 0;
    public static int ITEMS = 1;
    public static int METADATA = 2;

    /**
     * Variables estaticas para identificar el
     * permiso de escritura
     */
    public static int MODIFIABLE = 0;
    public static int UNMODIFIABLE = 1;

    private boolean userCheker;
    private Reader readPerm;
    protected int token;
    protected int type;
    protected String nmb;
    protected String userPath;
    protected File dataBase;
    Misc misc;

    // Constructor para escribir
    public JdbManager(int Tipo, int Permiso, String Nombre) {
        userPath = System.getProperty("user.dir");
        userCheker = false;
        if(Tipo == JdbManager.AUTH) {
            dataBase = (File) new File(userPath + "\\database\\auth",
                File.separator + "user.jdb");
        }else if(Tipo == JdbManager.ITEMS) {
            dataBase = (File) new File(userPath + "\\database\\items",
                File.separator + Nombre + ".item");
        }else if(Tipo == JdbManager.METADATA) {
            dataBase = (File) new File(userPath + "\\database\\metdata",
                File.separator + Nombre + ".data");
        }
        this.type = Tipo;
        this.nmb = Nombre;
        token = Permiso;

        evaluarDb();
    }
}

```

```
//Constructor para leer
public JdbManager(int Tipo, String Nombre) {
    userPath = System.getProperty("user.dir");
    userCheker = false;
    if(Tipo == JdbManager.AUTH) {
        dataBase = (File) new File(userPath + "\\database\\auth",
                                    File.separator + "user.jdb");
    }else if(Tipo == JdbManager.ITEMS) {
        dataBase = (File) new File(userPath + "\\database\\items",
                                    File.separator + Nombre + ".item");
    }else if(Tipo == JdbManager.METADATA) {
        dataBase = (File) new File(userPath + "\\database\\.metdata",
                                    File.separator + Nombre + ".data");
    }
    this.type = Tipo;
    this.nmb = Nombre;

    evaluarDb();
}
}
```

```
public void deleteDataBase() {
    if(!dataBase.delete()) {
        System.out.println("Consola: Error al eliminar el archivo");
    }
}
}
```

```
public void evaluarDb() {
    /**
     * Evaluacion de la existencia del achivo(base de datos)
     * y creacion de tal.
     */
    if (!dataBase.exists()) {
        try {
            dataBase.getParentFile().mkdirs();
            dataBase.createNewFile();
        } catch (Exception e) {
            e.printStackTrace();
        }
        setUserCheker(false);
    }else{
        setUserCheker(true);
    }
}
}
```

```
public void writeObj(String obj, String val) {
    /**
     * Funcion de escritura de objeto-valor
     */
    String aux[], objs[];
    FileWriter writePerm = null;
```

```
PrintWriter writer = null;
```

```
if(!objExist(obj)) {  
    /**  
     * Si el objeto no existe lo crea  
     */  
  
    try {  
        writePerm = new FileWriter(dataBase,true);  
        writer = new PrintWriter(writePerm);  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
    writer.println(obj + ": " + val);  
    try {  
        writePerm.close();  
        writer.close();  
    }catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
else if(token == 0) {  
    /**  
     * Si el objeto no existe y si el permiso para escribir  
     * es MODIFICABLE entonces cambia el valor del objeto  
     */  
    objs = getAllObjectsInFile();  
    dataBase.delete();  
    evaluarDb();  
    try {  
        writePerm = new FileWriter(dataBase,true);  
        writer = new PrintWriter(writePerm);  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
    for(int i = 0; i< objs.length; i+=2) {  
        if(objs[i] == null) {  
            break;  
        }  
        if(objs[i].equals(obj)) {  
            writer.println(obj + ": " + val);  
        }else {  
            writer.println(objs[i] + ": " + objs[i+1]);  
        }  
    }  
    try {  
        writePerm.close();  
        writer.close();  
    }catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
}
```

```
public String[] getObjectsInFile(String[] objs, int lng) {  
    /**  
     * Obtiene ciertos objetos especificados en "objs" de  
     * la base de datos  
     */  
    String[] rtn = new String[lng * 2];  
    int cont = 0; int i = 0;  
    String aux = "";  
    Reader readPerm = null;  
    BufferedReader read = null;  
  
    do {  
        try {  
            readPerm = new FileReader(dataBase);  
            read = new BufferedReader(readPerm);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        do {  
            try {  
                aux = read.readLine();  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
  
            if(aux != null && aux.contains(objs[i] + ":")) {  
                for(int it = 0; it<= aux.length(); it++) {  
                    if(aux.charAt(it) == objs[i].charAt(0)) {  
                        aux = aux.substring(it + objs[i].length() + 2);  
                        break;  
                    }  
                }  
  
                rtn[cont] = objs[i];  
                rtn[cont + 1] = aux;  
                aux = null;  
            } else {  
                rtn[cont] = objs[i];  
                rtn[cont + 1] = null;  
            }  
        } while(aux != null);  
  
        try {  
            readPerm.close();  
            read.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        i ++;  
    }  
}
```

```

        cont += 2;
    }while(i != lng);
    return rtn;
}

```

```

public String[] getAllObjectsInFile() {
    /**
     * Obtencion de todos los objetos dentro de la base de datos
     *
     * Nota: por el momento se usa misc.getCdi() ya que la funcion
     * solo se usa para obtener todos los items dentro de la base
     * de datos de la metadata
     */
    Misc misc = new Misc();
    String[] rtn = new String[misc.getCdi() * 2 + 2];
    int cont = 0; int ot;
    String aux = "";String aux2 = "";
    String rdr = "";
    readPerm = null;
    BufferedReader read = null;

    try {
        readPerm = new FileReader(dataBase);
        read = new BufferedReader(readPerm);
    }catch(Exception e) {
        e.printStackTrace();
    }

    do {
        try {
            aux = read.readLine();
        }catch (Exception e) {
            e.printStackTrace();
        }
        if(aux != null && aux.contains(":")) {
            for(int it = 0; it<= aux.length(); it++) {
                if(aux.charAt(it) != ' ') {
                    for(ot = 0; ot <= aux.length(); ot++) {
                        if(aux.charAt(ot) == ':') {
                            break;
                        }
                    }
                }
                rdr = aux.substring(it, ot);
                aux2 = aux.substring(ot + 2,aux.length());
                break;
            }
        }
        rtn[cont] = rdr;
        rtn[cont + 1] = aux2;
        cont += 2;
    }
}

```

```

        }
    }while(aux != null);

    try {
        readPerm.close();
        read.close();
    }catch (Exception e) {
        e.printStackTrace();
    }

    return rtn;
}

```

```

public String[] getAllObjectsInFile(int lng) {
    /**
     * Obtencion de todos los objetos dentro de la base de datos
     * especificando la cantidad de objetos a buscar
     */
    String[] rtn = new String[lng * 2];
    int cont = 0; int ot;
    String aux = "";String aux2 = "";
    String rdr = "";
    readPerm = null;
    BufferedReader read = null;

    try {
        readPerm = new FileReader(dataBase);
        read = new BufferedReader(readPerm);
    }catch(Exception e) {
        e.printStackTrace();
    }

    do {
        try {
            aux = read.readLine();
        }catch (Exception e) {
            e.printStackTrace();
        }
        if(aux != null && aux.contains(":")) {
            for(int it = 0; it<= aux.length(); it++) {
                if(aux.charAt(it) != ' ') {
                    for(ot = 0; ot <= aux.length(); ot++) {
                        if(aux.charAt(ot) == ':') {
                            break;
                        }
                    }
                }

                rdr = aux.substring(it, ot);
                aux2 = aux.substring(ot + 2,aux.length());
                break;
            }
        }
    } while (aux != null);

    return new String[] {rdr, aux2};
}

```



```

        }
        rtn[cont] = rdr;
        rtn[cont + 1] = aux2;
        cont += 2;
    }
}while(aux != null);

try {
    readPerm.close();
    read.close();
}catch (Exception e) {
    e.printStackTrace();
}

return rtn;
}

```

```

public boolean objExist(String obj) {
    /**
     * Retorno de un booleano si el objeto especificado
     * en "obj" existe o no en la base de datos
     */
    boolean band = false;
    String aux = "";
    Reader readPerm = null;
    String str = "";
    BufferedReader read = null;

    try {
        readPerm = new FileReader(dataBase);
        read = new BufferedReader(readPerm);
    }catch(Exception e) {
        e.printStackTrace();
    }

    do {
        try {
            aux = read.readLine();
        }catch (Exception e) {
            e.printStackTrace();
        }
        if(aux != null && aux.length() != 0) {
            for(int it = 0; it < aux.length(); it++) {
                if(aux.charAt(it) != ' ') {
                    for(int ot = 0; ot <= aux.length(); ot++) {
                        if(aux.charAt(ot) == ':') {
                            str = aux.substring(it, ot);
                            break;
                        }else {
                            str = null;
                        }
                    }
                }
            }
        }
    } while(aux != null);

    return band;
}

```

```

        }
        break;
    }else {
        str = null;
    }
    }
}
if(str != null && str.equals(obj)) {
    band = true;
    break;
}
}while(aux != null);

try {
    readPerm.close();
    read.close();
}catch(Exception e) {
    e.printStackTrace();
}
return band;
}

```

```

public boolean valExist(String val) {
    /**
     * Retorno de un booleano si el valor especificado
     * en "val" existe o no en la base de datos
     */
    boolean band = false;
    String aux = "";
    Reader readPerm = null;
    String str = "";
    BufferedReader read = null;

    try {
        readPerm = new FileReader(dataBase);
        read = new BufferedReader(readPerm);
    }catch(Exception e) {
        e.printStackTrace();
    }

    do {
        try {
            aux = read.readLine();
        }catch (Exception e) {
            e.printStackTrace();
        }
        if(aux != null && aux.length() != 0) {
            for(int it = 0; it < aux.length(); it++) {
                if(aux.charAt(it) != ' ') {
                    for(int ot = 0; ot <= aux.length(); ot++) {
                        if(aux.charAt(ot) == ':') {

```

```

                str = aux.substring(ot + 2, aux.length());
                break;
            }else {
                str = null;
            }
        }
        break;
    }else {
        str = null;
    }
}

if(str != null && str.equals(val)) {
    band = true;
    break;
}
}while(aux != null);

try {
    readPerm.close();
    read.close();
}catch(Exception e) {
    e.printStackTrace();
}
return band;
}

public boolean getUserCheker() {
    return userCheker;
}

public void setUserCheker(boolean userCheker) {
    this.userCheker = userCheker;
}
}

package lunarsoftware.lib.jdbmanager.misc;

import lunarsoftware.lib.jdbmanager.JdbManager;

public class Misc{
    public Misc() {

    }

    public int getCdi() {
        /**
         * Funcion que retorna la cantidad de items (Cuentas contraseñas)
         * agregados por el usuario
         */
        int rtn = 0;
        JdbManager write = new JdbManager(JdbManager.METADATA, JdbManager.MODIFIABLE,
"items");
        String[] cdoToSearch = {"Cdo"};
        String[] cdoReturn = write.getObjectsInFile(cdoToSearch, 1);
    }
}

```

```

        if(cdoReturn[1] != null) {
            rtn = Integer.parseInt(cdoReturn[1]);
        }else {
            rtn = 0;
        }

        return rtn;
    }

    public boolean writeItemMetada(String nombre) {
        /**
         * Funcion que escribe la metadata del item especificado
         * en "nombre"
         */
        boolean band = false;

        JdbManager write = new JdbManager(JdbManager.METADATA, JdbManager.MODIFIABLE,
"items");

        String[] cdoToSearch = {"Cdo"};
        String[] cdoReturn = write.getObjectsInFile(cdoToSearch, 1);

        System.out.println(write.valExist(nombre));

        if(!write.valExist(nombre)) {
            /**
             * Si el item no existe lo agrega
             */
            if(cdoReturn[1] != null) {
                write.writeObj("Cdo", "" + (Integer.parseInt(cdoReturn[1]) + 1));
                write.writeObj("item" + (Integer.parseInt(cdoReturn[1]) + 1), nombre);
            }else {
                write.writeObj("Cdo", "1");
                write.writeObj("item" + 1, nombre);
            }
            band = true;
        }else {
            /**
             * Si el item existe le retorna un booleano para especificar
             * que el item ya esta creado y que no puede ser agregado dos
             * veces o mas
             */
            band = false;
        }

        return band;
    }
}

package lunarsoftware.gdc.gui;

import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

import javax.swing.JFrame;

public class MoverFrame extends MouseAdapter {

    private final JFrame frame;
    private Point mouseComp = null;

    public MoverFrame(JFrame frame) {

```

```

        this.frame = frame;
    }

    public void mouseReleased(MouseEvent e) {
        mouseComp = null;
    }

    public void mousePressed(MouseEvent e) {
        //AL mouse seleccionar el contenedor le retorna un Point
        mouseComp = e.getPoint();
    }

    public void mouseDragged(MouseEvent e) {
        /**Al mause ser arrastrado dentro del contenedor obtiene las cordenadas
        * y relocaliza la ventana al lugar en donde se encuentra el mouse
        */
        Point cords = e.getLocationOnScreen();
        frame.setLocation(cords.x - mouseComp.x, cords.y - mouseComp.y);
    }
}

```

```
package lunarsoftware.gdc.gui;
```

```

import java.awt.EventQueue;
import java.awt.Label;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowEvent;

```

```

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

```

```

import lunarsoftware.gdc.Tools;
import lunarsoftware.gdc.gui.auth.Registro;
import lunarsoftware.lib.jdbmanager.JdbManager;
import lunarsoftware.lib.jdbmanager.misc.Misc;

```

```

import java.awt.Color;
import java.awt.Cursor;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Component;
import java.awt.GridLayout;
import java.awt.Font;

```

```
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JScrollBar;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import java.awt.ComponentOrientation;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
```

```
import java.awt.Insets;
import javax.swing.JComboBox;
import java.awt.BorderLayout;
import com.jgoodies.forms.layout.FormLayout;
import com.jgoodies.forms.layout.ColumnSpec;
import com.jgoodies.forms.layout.RowSpec;
import net.miginfocom.swing.MigLayout;
import javax.swing.SpringLayout;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
```

```
public class Major extends JFrame {
    private int cdp;
    private int pac;
```

```
    private JdbManager manager;
    private JPanel contentPane;
    private JTextField textField;
    private String userPath;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JPanel showPanel;
    private JPanel barraPanel;
    private JPanel addPanel;
    private JPanel pagesPanel;
```

```
/**
 * Inizidor del frame(Ventana)
 */
public static void main(String[] args) {
    JOptionPane.showMessageDialog(null, "Alerta: El software esta en modo desarrollo, los datos aqui guardados\n"
        + ""
        + ""
        + "no son de ninguna forma encriptados, porfavor no ingrese ningun tipo\n"
        + "de dato personal, el programa solo muestra la funcionalidad principal."
        + "", null, 2, null);
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
```

```

        Major frame = new Major();
        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

});

}

/**
 * Creacion del frame(Ventana)
 */
public Major() {
    userPath = System.getProperty("user.dir");
    Misc misc = new Misc();
    pac = 1;
    int cdl = misc.getCdi();
    cdp = (int) Math.ceil(((float)cdl) / 15);

    setUndecorated(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 1300, 700);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(244, 244, 244));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(null);
    setContentPane(contentPane);

    Label label = new Label("");
    label.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    //Agrega la funcionalidad de mover el frame en un label
    MoverFrame mov = new MoverFrame(this);
    label.addMouseListener(mov);
    label.addMouseMotionListener(mov);
    label.setBackground(new Color(63, 3, 82));
    label.setBounds(0, 0, 1226, 24);
    contentPane.add(label);

    JLabel ButtonExit = new JLabel("");
    ButtonExit.setOpaque(true);
    ButtonExit.addMouseListener(new MouseAdapter() {
        /**
         * Efecto hover sobre el label
         */
        @Override
        public void mouseEntered(MouseEvent e) {
            ButtonExit.setBackground(new Color(255, 70, 70));
        }
        @Override
        public void mouseExited(MouseEvent e) {

```

```

        ButtonExit.setBackground(new Color(45, 3, 60));
    }
    /**
     * Cierre de ventana y finalizacion del programa
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
        System.exit(0);
    }
});
ButtonExit.setBackground(new Color(45, 3, 60));
ButtonExit.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
ButtonExit.setToolTipText("");
ButtonExit.setHorizontalAlignment(SwingConstants.CENTER);
ButtonExit.setHorizontalTextPosition(SwingConstants.CENTER);
ButtonExit.setSize(new Dimension(100, 1000));
ButtonExit.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\close.png"));
ButtonExit.setBounds(1263, 0, 37, 24);
contentPane.add(ButtonExit);

JLabel ButtonMinimize = new JLabel("");
ButtonMinimize.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
ButtonMinimize.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        ButtonMinimize.setBackground(new Color(250, 75, 170));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        ButtonMinimize.setBackground(new Color(45, 3, 60));
    }
    /**
     * Minimizacion de la ventana
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        setState(JFrame.ICONIFIED);
    }
});
ButtonMinimize.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\minimize.png"));
ButtonMinimize.setToolTipText("");
ButtonMinimize.setSize(new Dimension(100, 1000));
ButtonMinimize.setOpaque(true);
ButtonMinimize.setHorizontalTextPosition(SwingConstants.CENTER);
ButtonMinimize.setHorizontalAlignment(SwingConstants.CENTER);
ButtonMinimize.setBackground(new Color(45, 3, 60));
ButtonMinimize.setBounds(1226, 0, 37, 24);

```



```
contentPane.add(ButtonMinimize);
```

```
JPanel panel = new JPanel();  
panel.setBackground(new Color(45, 3, 59));  
panel.setBounds(0, 24, 261, 676);  
contentPane.add(panel);  
panel.setLayout(null);
```

```
JPanel panel_1 = new JPanel();  
panel_1.addMouseListener(new MouseAdapter() {  
    /**  
     * Efecto hover sobre el panel  
     */  
    @Override  
    public void mouseEntered(MouseEvent e) {  
        panel_1.setBackground(new Color(168, 16, 218));  
    }  
    @Override  
    public void mouseExited(MouseEvent e) {  
        panel_1.setBackground(new Color(129, 12, 168));  
    }  
});
```

```
panel_1.setBackground(new Color(129, 12, 168));  
panel_1.setBounds(10, 11, 241, 39);  
panel.add(panel_1);  
panel_1.setLayout(null);
```

```
JLabel lblNewLabel_1_1 = new JLabel("");  
lblNewLabel_1_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\metrics.png"));  
lblNewLabel_1_1.setHorizontalTextPosition(SwingConstants.CENTER);  
lblNewLabel_1_1.setHorizontalAlignment(SwingConstants.CENTER);  
lblNewLabel_1_1.setBounds(0, 0, 47, 39);  
panel_1.add(lblNewLabel_1_1);
```

```
JLabel lblEstadisticas = new JLabel("Estadisticas <C-N-D>");  
lblEstadisticas.setForeground(new Color(233, 233, 233));  
lblEstadisticas.setFont(new Font("Arial", Font.BOLD, 15));  
lblEstadisticas.setBounds(43, 0, 160, 39);  
panel_1.add(lblEstadisticas);
```

```
JPanel panel_1_1 = new JPanel();  
panel_1_1.addMouseListener(new MouseAdapter() {  
    /**  
     * Efecto hover sobre el panel  
     */  
    @Override  
    public void mouseEntered(MouseEvent e) {  
        panel_1_1.setBackground(new Color(168, 16, 218));  
    }  
    @Override  
    public void mouseExited(MouseEvent e) {
```

```

        panel_1_1.setBackground(new Color(129, 12, 168));
    }
});
panel_1_1.setBackground(new Color(129, 12, 168));
panel_1_1.setBounds(10, 84, 241, 39);
panel.add(panel_1_1);
panel_1_1.setLayout(null);

JLabel lblNewLabel_1 = new JLabel("");
lblNewLabel_1.setBounds(0, 0, 47, 39);
lblNewLabel_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\database.png"));
panel_1_1.add(lblNewLabel_1);

JLabel lblNewLabel = new JLabel("Cuentas <C-D.S-C>");
lblNewLabel.setBounds(43, 0, 168, 39);
lblNewLabel.setForeground(new Color(233, 233, 233));
lblNewLabel.setFont(new Font("Arial", Font.BOLD, 15));
panel_1_1.add(lblNewLabel);

JPanel panel_1_2 = new JPanel();
panel_1_2.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el panel
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        panel_1_2.setBackground(new Color(168, 16, 218));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        panel_1_2.setBackground(new Color(129, 12, 168));
    }
});
panel_1_2.setBackground(new Color(129, 12, 168));
panel_1_2.setBounds(10, 157, 241, 39);
panel.add(panel_1_2);
panel_1_2.setLayout(null);

JLabel lblNewLabel_1_1_1 = new JLabel("");
lblNewLabel_1_1_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\category.png"));
lblNewLabel_1_1_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_1_1.setBounds(0, 0, 47, 39);
panel_1_2.add(lblNewLabel_1_1_1);

JLabel lblCategorias = new JLabel("Categorias <C-N-D>");
lblCategorias.setForeground(new Color(233, 233, 233));
lblCategorias.setFont(new Font("Arial", Font.BOLD, 15));
lblCategorias.setBounds(43, 0, 169, 39);

```

```
panel_1_2.add(lblCategorias);
```

```
JPanel panel_1_4 = new JPanel();
panel_1_4.setBounds(250, 11, 6, 39);
panel.add(panel_1_4);
panel_1_4.setBackground(new Color(70, 3, 92));
```

```
JPanel panel_1_4_1 = new JPanel();
panel_1_4_1.setBackground(new Color(70, 3, 92));
panel_1_4_1.setBounds(250, 84, 6, 39);
panel.add(panel_1_4_1);
```

```
JPanel panel_1_4_1_1 = new JPanel();
panel_1_4_1_1.setBackground(new Color(70, 3, 92));
panel_1_4_1_1.setBounds(250, 157, 6, 39);
panel.add(panel_1_4_1_1);
```

```

JPanel panel_2 = new JPanel();
panel_2.setLayout(null);
panel_2.setBackground(new Color(166, 166, 166));
panel_2.setBounds(259, 24, 6, 676);
contentPane.add(panel_2);

```

```
showPanel = new JPanel();
showItems();
contentPane.add(showPanel);
```

```
barraPanel = new JPanel();
barraPanel.setBackground(new Color(192, 192, 192));
barraPanel.setBounds(265, 25, 1032, 45);
contentPane.add(barraPanel);
barraPanel.setLayout(null);
```

```

/*
 * Funcion de buscador <E-D>
 *
 */

JLabel lblNewLabel_2 = new JLabel("");
lblNewLabel_2.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_2.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\search.png"));
lblNewLabel_2.setBounds(300, 10, 29, 24);
barraPanel.add(lblNewLabel_2);

textField = new JTextField();
textField.getDocument().addDocumentListener(new DocumentListener() {
    @Override public void changedUpdate(DocumentEvent e) {}
    @Override
    public void insertUpdate(DocumentEvent e) {}
})
}

```

```

        @Override
        public void removeUpdate(DocumentEvent e) {

        }

});
textField.setMargin(new Insets(2, 18, 2, 2));
textField.setBackground(new Color(233, 233, 233));
textField.setText("Buscar");
textField.setForeground(new Color(73, 73, 73));
textField.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 14));
textField.setColumns(10);
textField.setBorder(null);
textField.setBounds(21, 10, 279, 24);
barraPanel.add(textField);

JPanel panel_5 = new JPanel();
panel_5.setBackground(new Color(233, 233, 233));
panel_5.setBounds(8, 9, 320, 26);
barraPanel.add(panel_5);
*/
addPanel = new JPanel();
addPanel.addMouseListener(new MouseAdapter() {
    /**
     * Aqui el software llama a las funciones necesarias para mostrar el
     * panel para añadir un nuevo item(Cuenta o contraseñas)
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        showPanel.removeAll();
        showPanel.revalidate();
        showPanel.repaint();
        barraPanel.remove(addPanel);
        barraPanel.revalidate();
        barraPanel.repaint();
        showAddPanel();
    }
    /**
     * Efecto hover sobre el panel
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        addPanel.setBackground(new Color(17, 196, 62));
    }
    public void mouseExited(MouseEvent e) {
        addPanel.setBackground(new Color(16, 182, 57));
    }
});
addPanel.setBackground(new Color(16, 182, 57));
addPanel.setBounds(930, 10, 92, 28);
barraPanel.add(addPanel);

```

```

JLabel lblNewLabel_5 = new JLabel("");
addPanel.add(lblNewLabel_5);
lblNewLabel_5.setBackground(new Color(0, 198, 50));
lblNewLabel_5.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_5.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 12));
lblNewLabel_5.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\add.png"));

JLabel lblNewLabel_5_1 = new JLabel("Añadir");
lblNewLabel_5_1.setForeground(new Color(233, 233, 233));
addPanel.add(lblNewLabel_5_1);
lblNewLabel_5_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_5_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5_1.setFont(new Font("Arial", Font.BOLD, 13));
lblNewLabel_5_1.setBackground(new Color(0, 198, 50));

JPanel panel_2_1 = new JPanel();
panel_2_1.setLayout(null);
panel_2_1.setBackground(new Color(166, 166, 166));
panel_2_1.setBounds(261, 696, 1039, 5);
contentPane.add(panel_2_1);

JPanel panel_2_2 = new JPanel();
panel_2_2.setLayout(null);
panel_2_2.setBackground(new Color(166, 166, 166));
panel_2_2.setBounds(1297, 24, 6, 676);
contentPane.add(panel_2_2);

//Llamada al metodo para generar las paginas necesarias
pagesPanel = new JPanel();
generatePages();

contentPane.add(pagesPanel);

setLocationRelativeTo(null);
}

public void generatePages() {
    /**
     * Aqui el software itera sobre los items y los divide en multiplos de 20
     * para generar las paginas(< 1 2 3 4 5 6 7 8 X>)
     */
    pagesPanel.removeAll();
    pagesPanel.revalidate();
    pagesPanel.repaint();
    FlowLayout flowLayout = (FlowLayout) pagesPanel.getLayout();
    flowLayout.setVgap(10);
    pagesPanel.setBackground(new Color(244, 244, 244));
    pagesPanel.setBounds(275, 70, 1012, 35);
    contentPane.add(pagesPanel);
}

```

```

JLabel substracLabel = new JLabel("<");
substracLabel.addMouseListener(new MouseAdapter() {
    /**
     * Restador de pagina actual
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        if(pac != 1) {
            pac--;
            generatePages();
            showItems();
            contentPane.add(pagesPanel);
        }
        showItems();
    }
});

```

```

int inc;int fnl = pac;for(fnl = pac;!(fnl%20 == 0);fnl +=1);inc = fnl - 19;

```

```

/**
 * Generador de paginas
 */
pagesPanel.add(substracLabel);
for(int i=inc;i<=fnl + 1;i++) {
    if(i > cdp) {
        break;
    }else {
        JLabel numberOfPage;
        if(i <= fnl) {
            numberOfPage = new JLabel(""+i);
            final int param = i;
            numberOfPage.addMouseListener(new MouseAdapter() {
                /**
                 * Aqui se le añade la pagina correcta a la que tiene que apuntar
                 * al ser precionado
                 */
                @Override
                public void mouseClicked(MouseEvent e) {
                    pac = param;
                    generatePages();
                    showItems();
                    contentPane.add(pagesPanel);
                }
            });
            pagesPanel.add(numberOfPage);
        }else {
            /**
             * Cuando el numero de paginas exede las 20 paginas se le
             * agrega un label " a - b" donde a indica donde incia la siguiente
             * seccion y b donde termina

```

```

        */
        if((i + 19) > cdp) {
            numberOfPage = new JLabel(""+ i + "-" + cdp);
            pagesPanel.add(numberOfPage);
        }else {
            numberOfPage = new JLabel(""+ i + " - " + (i + 19));
            pagesPanel.add(numberOfPage);
        }
    }

    /**
     * Si el label a agregar es igual a la pagina en la que se encuentra
     * se le agregara un efecto de seleccion
     */
    if(i == pac) {
        numberOfPage.setBorder(new LineBorder(new Color(212, 212, 212), 1, true));
        numberOfPage.setOpaque(true);
        numberOfPage.setBackground(new Color(231, 231, 231));
        pagesPanel.add(numberOfPage);
    }
}

/**
 * Sumador de pagina actual
 */
JLabel plussLabel = new JLabel(">");
plussLabel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        if(pac < cdp) {
            pac ++;
            generatePages();
            contentPane.add(pagesPanel);
        }
        showItems();
    }
});
pagesPanel.add(plussLabel);

pagesPanel.revalidate();
pagesPanel.repaint();
}

```

```

public void showAddPanel() {
    /**
     * Genera un formulario para la obtencion de los datos del
     * item(Cuenta o contraseña)
     */
    showPanel.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
    showPanel.setBackground(new Color(228, 228, 228));
}

```

```
showPanel.setBounds(276, 105, 703, 299);
contentPane.add(showPanel);
showPanel.setLayout(null);
```

```
JPanel imgPanel = new JPanel();
imgPanel.setBounds(40, 55, 98, 98);
showPanel.add(imgPanel);
imgPanel.setLayout(null);
```

```
JLabel icoLabel = new JLabel("");
icoLabel.setHorizontalTextPosition(SwingConstants.CENTER);
icoLabel.setHorizontalAlignment(SwingConstants.CENTER);
icoLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\acounticons\\default.png"));
icoLabel.setBounds(0, 0, 98, 98);
imgPanel.add(icoLabel);
```

```
JLabel nombreLabel = new JLabel("Nombre:");
nombreLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
nombreLabel.setBounds(181, 40, 90, 25);
showPanel.add(nombreLabel);
```

```
JTextField nombreField = new JTextField();
nombreField.setBounds(254, 40, 205, 25);
showPanel.add(nombreField);
nombreField.setColumns(10);
```

```
JLabel correoLabel = new JLabel("Correo o nombre de usuario:");
correoLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
correoLabel.setBounds(181, 76, 205, 25);
showPanel.add(correoLabel);
```

```
JTextField correoField = new JTextField();
correoField.setColumns(10);
correoField.setBounds(387, 77, 205, 25);
showPanel.add(correoField);
```

```
JLabel passLabel = new JLabel("Contraseña:");
passLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
passLabel.setBounds(181, 113, 90, 25);
showPanel.add(passLabel);
```

```
JLabel typeLabel = new JLabel("Tipo:");
typeLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
typeLabel.setBounds(181, 151, 90, 25);
showPanel.add(typeLabel);
```

```
JComboBox<String> comboBox = new JComboBox<String>();
comboBox.addItemListener(new ItemListener() {
    /**
     * Generacion de imagen segun seleccion
     */
});
```



```

        public void itemStateChanged(ItemEvent e) {
            icoLabel.removeAll();
            icoLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\acounticons\\" + (String)
comboBox.getSelectedItemAt() + ".png"));
            icoLabel.revalidate();
            icoLabel.repaint();
        }
    });
    comboBox.addItem("Otro");
    comboBox.addItem("Netflix");
    comboBox.addItem("Facebook");
    comboBox.addItem("Twitter");
    comboBox.addItem("Instagram");
    comboBox.addItem("Git Hub");
    comboBox.addItem("Office");
    comboBox.addItem("Gmail");
    comboBox.setBounds(229, 150, 205, 25);
    showPanel.add(comboBox);

    JTextField passField = new JTextField();
    passField.setColumns(10);
    passField.setBounds(277, 113, 205, 25);
    showPanel.add(passField);

    JLabel errorLabel = new JLabel("El nombre de la cuenta ya existe, intenta con otro nombre...");
    errorLabel.setVisible(false);
    errorLabel.setForeground(new Color(234, 0, 0));
    errorLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 14));
    errorLabel.setBounds(180, 206, 402, 14);
    showPanel.add(errorLabel);

    JPanel panelAdd = new JPanel();
    panelAdd.addMouseListener(new MouseAdapter() {
        /**
         * Aqui el software añade el item(Cuenta o contraseña)
         * a la base de datos el programa
         */
        @Override
        public void mouseClicked(MouseEvent e) {
            Misc misc = new Misc();

            if(misc.writeItemMetada(nombreField.getText())) {
                JdbManager itemManager = new JdbManager(JdbManager.ITEMS, JdbManager.MODIFIABLE,
nombreField.getText());

                itemManager.writeObj("Name", nombreField.getText());
                itemManager.writeObj("Mail", correoField.getText());
                itemManager.writeObj("Pass", passField.getText());
                itemManager.writeObj("Type", (String) comboBox.getSelectedItemAt());
                showPanel.removeAll();
                showPanel.revalidate();
            }
        }
    });

```

```

        showPanel.repaint();
        barraPanel.remove(addPanel);
        barraPanel.revalidate();
        barraPanel.repaint();
        showAddButton();
        showItems();
        int cdl = misc.getCdi();
        cdp = (int) Math.ceil(((float)cdl) / 15);
        generatePages();
        errorLabel.setVisible(false);
    }else {
        errorLabel.setVisible(true);
    }
}

/**
 * Efecto hover sobre el panel
 */
@Override
public void mouseEntered(MouseEvent e) {
    panelAdd.setBackground(new Color(17, 196, 62));
}
public void mouseExited(MouseEvent e) {
    panelAdd.setBackground(new Color(16, 182, 57));
}
});
panelAdd.setBackground(new Color(16, 182, 57));
panelAdd.setBounds(581, 248, 92, 28);
showPanel.add(panelAdd);

JLabel ButtonAdd = new JLabel("Añadir");
ButtonAdd.setBackground(new Color(16, 182, 57));
ButtonAdd.setHorizontalTextPosition(SwingConstants.CENTER);
ButtonAdd.setHorizontalAlignment(SwingConstants.CENTER);
ButtonAdd.setForeground(new Color(233, 233, 233));
ButtonAdd.setFont(new Font("Arial", Font.BOLD, 13));
ButtonAdd.setBackground(new Color(0, 198, 50));
panelAdd.add(ButtonAdd);

JPanel panelCancel = new JPanel();
panelCancel.setOpaque(true);
panelCancel.addMouseListener(new MouseAdapter() {
    /**
     * Vuelve a generar los items y cancela la operacion
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        showItems();
        showAddButton();
    }
}
/**

```

```

        * Efecto hover sobre el panel
        */
        @Override
        public void mouseEntered(MouseEvent e) {
            panelCancel.setBackground(new Color(237, 50, 50));
        }
        public void mouseExited(MouseEvent e) {
            panelCancel.setBackground(new Color(232, 21, 21));
        }
    });
    panelCancel.setBackground(new Color(232, 21, 21));
    panelCancel.setBounds(460, 248, 92, 28);
    showPanel.add(panelCancel);

    JLabel ButtonCancel = new JLabel("Cancelar");
    ButtonCancel.setHorizontalTextPosition(SwingConstants.CENTER);
    ButtonCancel.setHorizontalAlignment(SwingConstants.CENTER);
    ButtonCancel.setForeground(new Color(233, 233, 233));
    ButtonCancel.setFont(new Font("Arial", Font.BOLD, 13));
    ButtonCancel.setBackground(new Color(232, 21, 21));
    panelCancel.add(ButtonCancel);
}

```

```

public void showAddButton() {
    /**
     * Vuelve a generar el boton de Añadir item(Cuenta o contraseña)
     */
    addPanel = new JPanel();
    addPanel.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {

            showPanel.removeAll();
            showPanel.revalidate();
            showPanel.repaint();
            barraPanel.remove(addPanel);
            barraPanel.revalidate();
            barraPanel.repaint();
            showAddPanel();

        }
        @Override
        public void mouseEntered(MouseEvent e) {
            addPanel.setBackground(new Color(17, 196, 62));
        }
        public void mouseExited(MouseEvent e) {
            addPanel.setBackground(new Color(16, 182, 57));
        }
    });
    addPanel.setBackground(new Color(16, 182, 57));
    addPanel.setBounds(930, 10, 92, 28);
    barraPanel.add(addPanel);
}

```

```

JLabel lblNewLabel_5 = new JLabel("");
addPanel.add(lblNewLabel_5);
lblNewLabel_5.setBackground(new Color(0, 198, 50));
lblNewLabel_5.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_5.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 12));
lblNewLabel_5.setIcon(new ImageIcon(userPath + "\\access\\icons\\panel\\add.png"));

JLabel lblNewLabel_5_1 = new JLabel("Añadir");
lblNewLabel_5_1.setForeground(new Color(233, 233, 233));
addPanel.add(lblNewLabel_5_1);
lblNewLabel_5_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_5_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_5_1.setFont(new Font("Arial", Font.BOLD, 13));
lblNewLabel_5_1.setBackground(new Color(0, 198, 50));
}

```

```

public void showItems() {
    /**
     * Genera el panel principal y genera los items(Cuentas o contraseñas)
     * que se hallan en la base de datos
     */
    showPanel.removeAll();
    showPanel.revalidate();
    showPanel.repaint();
    String userPath = System.getProperty("user.dir");
    JdbManager manager = new JdbManager(JdbManager.METADATA, "items");

    showPanel.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
    showPanel.setBackground(new Color(233, 233, 233));
    showPanel.setBounds(276, 105, 1011, 559);
    showPanel.setLayout(new GridLayout(0, 5, 15, 40));
    int inc = (pac * 15) - 14;
    int fnl = pac * 15;

    for(int i=inc;i<=fnl;i++) {
        String[] objToSearch = {"item" + i};
        String[] objReturn;
        objReturn = manager.getObjectsInFile(objToSearch, 1);
        JdbManager item = new JdbManager(JdbManager.ITEMS, objReturn[1]);

        if(objReturn[1] != null) {
            /**
             * Si el objeto existe se agrega el panel de mandera normal
             * y toma las propiedades del item(Cuenta o contraseña)
             * para generarlo
             */
            String[] panelObjs = {"Name", "Type"};
            String[] panelObjsRtn;
            panelObjsRtn = item.getObjectsInFile(panelObjs, 2);

```

```

JPanel subPanel = new JPanel();
subPanel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        subPanel.setBackground(new Color(245, 245, 245));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        subPanel.setBackground(new Color(255, 255, 255));
    }
    @Override
    public void mouseClicked(MouseEvent e) {
        showItemPanel(((JLabel) subPanel.getComponent(1)).getText());
    }
});
subPanel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
subPanel.setBackground(new Color(255, 255, 255));
SpringLayout sl_subPanel = new SpringLayout();
subPanel.setLayout(sl_subPanel);

JLabel icon = new JLabel("");
sl_subPanel.putConstraint(SpringLayout.NORTH, icon, 22, SpringLayout.NORTH, subPanel);
sl_subPanel.putConstraint(SpringLayout.WEST, icon, 45, SpringLayout.WEST, subPanel);
sl_subPanel.putConstraint(SpringLayout.EAST, icon, -47, SpringLayout.EAST, subPanel);
subPanel.add(icon);
icon.setHorizontalAlignment(SwingConstants.CENTER);
icon.setIcon(new ImageIcon(userPath + "\\access\\icons\\acounticons\\" + panelObjsRtn[3] + ".png"));

JLabel name = new JLabel(panelObjsRtn[1]);

sl_subPanel.putConstraint(SpringLayout.SOUTH, icon, -6, SpringLayout.NORTH, name);
sl_subPanel.putConstraint(SpringLayout.NORTH, name, 126, SpringLayout.NORTH, subPanel);
sl_subPanel.putConstraint(SpringLayout.WEST, name, 7, SpringLayout.WEST, subPanel);
sl_subPanel.putConstraint(SpringLayout.SOUTH, name, -10, SpringLayout.SOUTH, subPanel);
sl_subPanel.putConstraint(SpringLayout.EAST, name, -7, SpringLayout.EAST, subPanel);
name.setVerticalTextPosition(SwingConstants.TOP);
name.setHorizontalAlignment(SwingConstants.CENTER);
subPanel.add(name);

showPanel.add(subPanel);
}else{
    /**
     * Si el item no existe agrega un panel de relleno(Panel en blanco)
     */
    JPanel nullPanel = new JPanel();
    nullPanel.setLayout(null);
    nullPanel.setBackground(new Color(233, 233, 233));
    showPanel.add(nullPanel);
}
}

```

```

        contentPane.add(showPanel);
        showPanel.revalidate();
        showPanel.repaint();
    }

    public void showItemPanel(String name) {
        /**
         * Creacion del panel para visualizar las propiedades
         * del item(Cuenta o contraseña)
         */

        showPanel.removeAll();
        showPanel.revalidate();
        showPanel.repaint();

        JdbManager localManager = new JdbManager(JdbManager.ITEMS, JdbManager.MODIFIABLE, name);
        String[] rtn;
        String[] ico = {"Type", "Name", "Mail", "Pass"}; rtn = localManager.getObjectsInFile(ico, 4);
        showPanel.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
        showPanel.setBackground(new Color(228, 228, 228));
        showPanel.setBounds(276, 105, 703, 299);
        contentPane.add(showPanel);
        showPanel.setLayout(null);

        JPanel imgPanel = new JPanel();
        imgPanel.setBounds(40, 55, 98, 98);
        showPanel.add(imgPanel);
        imgPanel.setLayout(null);

        JLabel icoLabel = new JLabel("");
        icoLabel.setHorizontalTextPosition(SwingConstants.CENTER);
        icoLabel.setHorizontalAlignment(SwingConstants.CENTER);
        icoLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\acounticons\\" + rtn[1] + ".png"));
        icoLabel.setBounds(0, 0, 98, 98);
        imgPanel.add(icoLabel);

        JLabel nombreLabel = new JLabel("Nombre:");
        nombreLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
        nombreLabel.setBounds(181, 40, 90, 25);
        showPanel.add(nombreLabel);

        JLabel nombreLabelres = new JLabel(rtn[3]);
        nombreLabelres.setForeground(new Color(60, 60, 60));
        nombreLabelres.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
        nombreLabelres.setBounds(254, 40, 205, 25);
        showPanel.add(nombreLabelres);

        JLabel correoLabel = new JLabel("Correo:");
        correoLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
        correoLabel.setBounds(181, 76, 205, 25);
        showPanel.add(correoLabel);
    }

```

```

JLabel correoField = new JLabel(rtn[5]);
correoField.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
correoField.setBounds(250, 77, 300, 25);
showPanel.add(correoField);

```

```

JLabel passLabel = new JLabel("Contraseña:");
passLabel.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
passLabel.setBounds(181, 113, 90, 25);
showPanel.add(passLabel);

```

```

JLabel passField = new JLabel(rtn[7]);
passField.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 15));
passField.setBounds(277, 113, 205, 25);
showPanel.add(passField);

```

```

JPanel panelBorrar = new JPanel();
panelBorrar.setOpaque(true);
panelBorrar.addMouseListener(new MouseAdapter() {
    /**
     * Eliminacion del item(Cuenta o contraseña)
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        JdbManager met = new JdbManager(JdbManager.METADATA, JdbManager.MODIFIABLE, "items");
        int ot = 0;
        String[] AllObjs = met.getAllObjectsInFile();
        met.deleteDataBase();
        met.evaluarDb();
        /**
         * Regeneracion del archivo sin el item eliminado
         */
        for(int i=1;i<=(AllObjs.length - 1);i+=2) {
            ot++;
            if(AllObjs != null) {
                if(AllObjs[i].equals(name)) {
                    for(int it = i + 2; it<=(AllObjs.length - 1);it+=2) {
                        met.writeObj("item" + (it - 2 - ot), AllObjs[it]);
                        ot++;
                    }
                    break;
                }else {
                    met.writeObj(AllObjs[i-1], AllObjs[i]);
                }
            }
            met.writeObj("Cdo", ""+(Integer.parseInt(AllObjs[1]) - 1));
            JdbManager file = new JdbManager(JdbManager.ITEMS, JdbManager.MODIFIABLE, name);
            file.deleteDataBase();
            showItems();
        }
    }
    /**

```

```

        * Efecto hover sobre el panel
        */
        @Override
        public void mouseEntered(MouseEvent e) {
            panelBorrar.setBackground(new Color(237, 50, 50));
        }
        public void mouseExited(MouseEvent e) {
            panelBorrar.setBackground(new Color(232, 21, 21));
        }
    });
    panelBorrar.setBackground(new Color(232, 21, 21));
    panelBorrar.setBounds(460, 248, 92, 28);
    showPanel.add(panelBorrar);

    JLabel ButtonDelete = new JLabel("Borrar");
    ButtonDelete.setHorizontalTextPosition(SwingConstants.CENTER);
    ButtonDelete.setHorizontalAlignment(SwingConstants.CENTER);
    ButtonDelete.setForeground(new Color(233, 233, 233));
    ButtonDelete.setFont(new Font("Arial", Font.BOLD, 13));
    ButtonDelete.setBackground(new Color(232, 21, 21));
    panelBorrar.add(ButtonDelete);

    JPanel panelRegresar = new JPanel();
    panelRegresar.setOpaque(true);
    panelRegresar.addMouseListener(new MouseAdapter() {
        /**
         * Regresa el panel a su estado principal llamando
         * a la funcion para generar los items
         */
        @Override
        public void mouseClicked(MouseEvent e) {
            showItems();
        }
        /**
         * Efecto hover sobre el panel
         */
        @Override
        public void mouseEntered(MouseEvent e) {
            panelRegresar.setBackground(new Color(220, 220, 220));
        }
        public void mouseExited(MouseEvent e) {
            panelRegresar.setBackground(new Color(233, 233, 233));
        }
    });
    panelRegresar.setBackground(new Color(233, 233, 233));
    panelRegresar.setBounds(581, 248, 92, 28);
    showPanel.add(panelRegresar);

    JLabel ButtonRegresar = new JLabel("Regresar");
    ButtonRegresar.setHorizontalTextPosition(SwingConstants.CENTER);
    ButtonRegresar.setHorizontalAlignment(SwingConstants.CENTER);

```



```
        ButtonRegresar.setForeground(new Color(40, 40, 40));
        ButtonRegresar.setFont(new Font("Arial", Font.BOLD, 13));
        ButtonRegresar.setBackground(new Color(232, 21, 21));
        panelRegresar.add(ButtonRegresar);

        showPanel.revalidate();
        showPanel.repaint();
    }
}
```

```
package lunarsoftware.gdc.gui.auth;
```

```
import java.io.PrintStream;
```

```
import java.awt.EventQueue;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.border.EmptyBorder;
```

```
import javax.swing.SpringLayout;
```

```
import java.awt.Color;
```

```
import javax.swing.BoxLayout;
```

```
import java.awt.CardLayout;
```

```
import java.awt.BorderLayout;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.ImageIcon;
```

```
import java.awt.Dimension;
```

```
import javax.swing.SwingConstants;
```

```
import java.awt.Cursor;
```

```
import java.awt.event.MouseAdapter;
```

```
import java.awt.event.MouseEvent;
```

```
import java.awt.event.WindowEvent;
```

```
import java.awt.Rectangle;
```

```
import javax.swing.JTextField;
```

```
import javax.swing.border.TitledBorder;
```

```
import javax.swing.event.DocumentEvent;
```

```
import javax.swing.event.DocumentListener;
```

```
import lunarsoftware.gdc.Tools;
```

```
import lunarsoftware.gdc.gui.MoverFrame;
```

```
import javax.swing.border.EtchedBorder;
```

```
import java.awt.Font;
```

```
import java.awt.ComponentOrientation;
```

```
import java.awt.Insets;
```

```
import java.awt.Label;
```

```
import javax.swing.JButton;
```

```
import java.awt.Component;
```

```
import java.awt.event.InputMethodListener;
```

```
import java.awt.event.InputMethodEvent;
```

```
import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeEvent;
import java.awt.event.ContainerAdapter;
import java.awt.event.ContainerEvent;

public class Registro extends JFrame {
    boolean mailComp;

    boolean passComp;
    boolean compPassComp;
    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private String userPath;

    /**
     * Inziador del frame(Ventana)
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Registro frame = new Registro();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Creacion del frame(Ventana)
     */
    public Registro() {
        userPath = System.getProperty("user.dir");
        mailComp = false;

        setUndecorated(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 377, 484);
        contentPane = new JPanel();
        contentPane.setSize(new Dimension(0, 500000));
        contentPane.setBackground(new Color(45, 3, 59));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("");
        lblNewLabel.setOpaque(true);
```

```

lblNewLabel.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */

    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel.setBackground(new Color(255, 70, 70));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel.setBackground(new Color(45, 3, 60));
    }

    /**
     * Cierre de ventana y finalizacion del programa
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
        System.exit(0);
    }
});
lblNewLabel.setBackground(new Color(45, 3, 60));
lblNewLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel.setToolTipText("");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel.setSize(new Dimension(100, 1000));
lblNewLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\close.png"));
lblNewLabel.setBounds(340, 0, 37, 24);
contentPane.add(lblNewLabel);

```

```

JLabel lblNewLabel_1 = new JLabel("");
lblNewLabel_1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel_1.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */

    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_1.setBackground(new Color(250, 75, 170));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_1.setBackground(new Color(45, 3, 60));
    }

    /**
     * Minimizacion de la ventana
     */

```

```

        @Override
        public void mouseClicked(MouseEvent e) {
            setState(JFrame.ICONIFIED);
        }
    });
    lblNewLabel_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\minimize.png"));
    lblNewLabel_1.setToolTipText("");
    lblNewLabel_1.setSize(new Dimension(100, 1000));
    lblNewLabel_1.setOpaque(true);
    lblNewLabel_1.setHorizontalTextPosition(SwingConstants.CENTER);
    lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel_1.setBackground(new Color(45, 3, 60));
    lblNewLabel_1.setBounds(303, 0, 37, 24);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel4 = new JLabel("El correo es incorrecto");
    lblNewLabel4.setVisible(false);
    lblNewLabel4.setHorizontalTextPosition(SwingConstants.CENTER);
    lblNewLabel4.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel4.setForeground(new Color(255, 128, 128));
    lblNewLabel4.setFont(new Font("Tahoma", Font.PLAIN, 10));
    lblNewLabel4.setBounds(36, 293, 309, 14);
    contentPane.add(lblNewLabel4);

    textField = new JTextField();
    textField.getDocument().addDocumentListener(new DocumentListener() {
        /**
         * Aqui se comprueba que el correo contenga un dominio valido
         */
        @Override public void changedUpdate(DocumentEvent e) {}
        @Override
        public void insertUpdate(DocumentEvent e) {
            if(!Tools.error.getHasArrobaError(textField)&&
                !Tools.error.getArrobaError(textField) &&
                !Tools.error.getDomainError(textField)) {
                lblNewLabel4.setVisible(false);
                setMailComp(true);
            }else {setMailComp(false);lblNewLabel4.setVisible(true);}
        }
        @Override
        public void removeUpdate(DocumentEvent e) {
            if(!Tools.error.getArrobaError(textField) &&
                !Tools.error.getDomainError(textField)&&
                !Tools.error.getHasArrobaError(textField)) {
                lblNewLabel4.setVisible(false);
                setMailComp(true);
            }else {setMailComp(false);lblNewLabel4.setVisible(true);}
        }
    });
    textField.setAlignmentY(Component.TOP_ALIGNMENT);
    textField.setAlignmentX(Component.LEFT_ALIGNMENT);

```

```

textField.setMargin(new Insets(1, 1, 1, 1));
textField.setHorizontalAlignment(SwingConstants.LEFT);
textField.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField.setForeground(new Color(239, 239, 239));
textField.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new Color(160,
160, 160)), "Correo", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
textField.setBackground(new Color(45, 3, 59));
textField.setBounds(40, 99, 295, 42);
contentPane.add(textField);
textField.setColumns(10);

JLabel lblNewLabel_2 = new JLabel("Registro");
lblNewLabel_2.setForeground(new Color(239, 239, 239));
lblNewLabel_2.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 40));
lblNewLabel_2.setBounds(114, 40, 149, 46);
contentPane.add(lblNewLabel_2);
JLabel lblNewLabel3 = new JLabel("La contraseña debe ser mayor de 8 y contener numeros");
lblNewLabel3.setVisible(false);
lblNewLabel3.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel3.setForeground(new Color(255, 128, 128));
lblNewLabel3.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel3.setBounds(37, 322, 309, 14);
contentPane.add(lblNewLabel3);

textField_1 = new JTextField();
textField_1.getDocument().addDocumentListener(new DocumentListener() {
    @Override public void changedUpdate(DocumentEvent e) {}
    /**
     * Aqui el software comprueba que la contraseña se haya escrito
     * de manera correcta (minimo 8 caracteres y que contenga numeros)
     */
    @Override
    public void insertUpdate(DocumentEvent e) {
        if(!Tools.error.getMinLngPassError(textField_1)&&
            !Tools.error.getPassNumError(textField_1)) {
            lblNewLabel3.setVisible(false);
            setPassComp(true);
        }else {lblNewLabel3.setVisible(true);setPassComp(false);}
    }
    @Override
    public void removeUpdate(DocumentEvent e) {
        if(!Tools.error.getMinLngPassError(textField_1)&&
            !Tools.error.getPassNumError(textField_1)) {
            lblNewLabel3.setVisible(false);
            setPassComp(true);
        }else {lblNewLabel3.setVisible(true);setPassComp(false);}
    }
});
textField_1.setMargin(new Insets(1, 1, 1, 1));

```

```

textField_1.setHorizontalAlignment(SwingConstants.LEFT);
textField_1.setForeground(new Color(239, 239, 239));
textField_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField_1.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField_1.setColumns(10);
textField_1.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new
Color(160, 160, 160)), "Contrase\u00F1a", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
textField_1.setBackground(new Color(45, 3, 59));
textField_1.setBounds(40, 165, 295, 42);
contentPane.add(textField_1);

JLabel lblNewLabel6 = new JLabel("Las contrase\u00F1as no son iguales");
lblNewLabel6.setVisible(false);
lblNewLabel6.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel6.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel6.setForeground(new Color(255, 128, 128));
lblNewLabel6.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel6.setBounds(37, 348, 309, 14);
contentPane.add(lblNewLabel6);

textField_2 = new JPasswordField();
textField_2.getDocument().addDocumentListener(new DocumentListener() {
    @Override public void changedUpdate(DocumentEvent e) {}
    /**
     * Aqui el software comprueba que las contrase\u00F1as coincidan
     */
    @Override
    public void insertUpdate(DocumentEvent e) {
        if(textField_2.getText().equals(textField_1.getText())) {
            lblNewLabel6.setVisible(false);
            setCompPassComp(true);
        }else {lblNewLabel6.setVisible(true);setCompPassComp(false);}
    }
    @Override
    public void removeUpdate(DocumentEvent e) {
        if(textField_2.getText().equals(textField_1.getText())) {
            lblNewLabel6.setVisible(false);
            setCompPassComp(true);
        }else {lblNewLabel6.setVisible(true);setCompPassComp(false);}
    }
});
textField_2.setMargin(new Insets(1, 1, 1, 1));
textField_2.setHorizontalAlignment(SwingConstants.LEFT);
textField_2.setForeground(new Color(239, 239, 239));
textField_2.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField_2.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField_2.setColumns(10);
textField_2.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new
Color(160, 160, 160)), "Confirmar contrase\u00F1a", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
textField_2.setBackground(new Color(45, 3, 59));
textField_2.setBounds(40, 232, 295, 42);

```

```
contentPane.add(textField_2);
```

```
JLabel lblNewLabel5 = new JLabel("Completa el registro de manera correcta");
lblNewLabel5.setVisible(false);
lblNewLabel5.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel5.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel5.setForeground(new Color(255, 128, 128));
lblNewLabel5.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel5.setBounds(37, 431, 309, 14);
contentPane.add(lblNewLabel5);
```

```
JLabel lblNewLabel_3 = new JLabel("Siguiente");
lblNewLabel_3.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(85, 5, 112));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(68, 4, 89));
    }

    /**
     * Aqui el software comprueba que el campo de correo y contraseña
     * hayan sido validados de manera correcta.
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        if(getMailComp() && getPassComp() && getCompPassComp()) {
            lblNewLabel5.setVisible(false);
            dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
            GetName frame = new GetName(textField.getText(), textField_1.getText());
            frame.main(null, textField.getText(), textField_1.getText());
        }else {
            lblNewLabel5.setVisible(true);
        }
    }
});
lblNewLabel_3.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel_3.setOpaque(true);
lblNewLabel_3.setBackground(new Color(68, 4, 89));
lblNewLabel_3.setForeground(new Color(212, 212, 212));
lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_3.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_3.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 18));
lblNewLabel_3.setBounds(127, 377, 126, 42);
contentPane.add(lblNewLabel_3);
```

```

JLabel lblNewLabel_4 = new JLabel("Manual de usuario");
lblNewLabel_4.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4.setForeground(new Color(185, 185, 185));
lblNewLabel_4.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4.setBounds(280, 459, 87, 14);
contentPane.add(lblNewLabel_4);

JLabel lblNewLabel_4_1 = new JLabel("Ayuda");
lblNewLabel_4_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_4_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4_1.setForeground(new Color(185, 185, 185));
lblNewLabel_4_1.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4_1.setBounds(233, 459, 37, 14);
contentPane.add(lblNewLabel_4_1);

Label label = new Label("");
label.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
//Agrega la funcionalidad de mover el frame en un label
MoverFrame mov = new MoverFrame(this);
label.addMouseListener(mov);
label.addMouseMotionListener(mov);
label.setBackground(new Color(63, 3, 82));
label.setBounds(0, 0, 303, 24);
contentPane.add(label);

setLocationRelativeTo(null);
}

public void setMailComp(boolean mailComp) {
    this.mailComp = mailComp;
}

public boolean getMailComp() {
    return mailComp;
}

public void setPassComp(boolean passComp) {
    this.passComp = passComp;
}

public boolean getPassComp() {
    return passComp;
}

public void setCompPassComp(boolean compPassComp) {
    this.compPassComp = compPassComp;
}

public boolean getCompPassComp() {
    return compPassComp;
}
}

```



```
package lunarsoftware.gdc.gui.auth;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.ImageIcon;
import java.awt.Dimension;
import javax.swing.SwingConstants;
import java.awt.Cursor;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowEvent;
import javax.swing.JTextField;
import javax.swing.border.TitledBorder;

import lunarsoftware.gdc.gui.Major;
import lunarsoftware.gdc.gui.MoverFrame;
import lunarsoftware.lib.jdbmanager.JdbManager;

import javax.swing.border.EtchedBorder;
import java.awt.Font;
import java.awt.ComponentOrientation;
import java.awt.Insets;

import java.awt.Component;
import java.awt.Label;

public class Login extends JFrame {
    private JdbManager manager;
    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private String userPath;

    /**
     * Iniziator del frame(Ventana)
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Login frame = new Login();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

    });
}

/**
 * Creacion del frame(Ventana)
 */
public Login() {
    userPath = System.getProperty("user.dir");
    manager = new JdbManager(JdbManager.AUTH, null);

    setUndecorated(true);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 377, 356);
    contentPane = new JPanel();
    contentPane.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    contentPane.setSize(new Dimension(0, 500000));
    contentPane.setBackground(new Color(45, 3, 59));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("");
    lblNewLabel.setOpaque(true);
    lblNewLabel.addMouseListener(new MouseAdapter() {
        /**
         * Efecto hover sobre el label
         */
        @Override
        public void mouseEntered(MouseEvent e) {
            lblNewLabel.setBackground(new Color(255, 70, 70));
        }
        @Override
        public void mouseExited(MouseEvent e) {
            lblNewLabel.setBackground(new Color(45, 3, 60));
        }
        /**
         * Cierre de ventana y finalizacion del programa
         */
        @Override
        public void mouseClicked(MouseEvent e) {
            dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
            System.exit(0);
        }
    });
    lblNewLabel.setBackground(new Color(45, 3, 60));
    lblNewLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    lblNewLabel.setToolTipText("");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setHorizontalTextPosition(SwingConstants.CENTER);
    lblNewLabel.setSize(new Dimension(100, 1000));

```

```
lblNewLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\close.png"));
lblNewLabel.setBounds(340, 0, 37, 24);
contentPane.add(lblNewLabel);
```

```
JLabel lblNewLabel_1 = new JLabel("");
lblNewLabel_1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel_1.addMouseListener(new MouseAdapter() {
```

```
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_1.setBackground(new Color(250, 75, 170));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_1.setBackground(new Color(45, 3, 60));
    }
    /**
     * Minimizacion de la ventana
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        setState(JFrame.ICONIFIED);
    }
}
```

```
});
lblNewLabel_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\minimize.png"));
lblNewLabel_1.setToolTipText("");
lblNewLabel_1.setSize(new Dimension(100, 1000));
lblNewLabel_1.setOpaque(true);
lblNewLabel_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setBackground(new Color(45, 3, 60));
lblNewLabel_1.setBounds(303, 0, 37, 24);
contentPane.add(lblNewLabel_1);
```

```
textField = new JTextField();
textField.setAlignmentY(Component.TOP_ALIGNMENT);
textField.setAlignmentX(Component.LEFT_ALIGNMENT);
textField.setMargin(new Insets(1, 1, 1, 1));
textField.setHorizontalAlignment(SwingConstants.LEFT);
textField.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField.setForeground(new Color(239, 239, 239));
textField.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new Color(160, 160, 160)), "Correo", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
textField.setBackground(new Color(45, 3, 59));
textField.setBounds(40, 99, 295, 42);
contentPane.add(textField);
textField.setColumns(10);
```

```

JLabel lblNewLabel_2 = new JLabel("Iniciar sesion");
lblNewLabel_2.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_2.setForeground(new Color(239, 239, 239));
lblNewLabel_2.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 35));
lblNewLabel_2.setBounds(40, 40, 295, 46);
contentPane.add(lblNewLabel_2);

JLabel lblNewLabel3 = new JLabel("Las credenciales son invalidas");
lblNewLabel3.setVisible(false);
lblNewLabel3.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel3.setForeground(new Color(255, 128, 128));
lblNewLabel3.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel3.setBounds(37, 235, 309, 14);
contentPane.add(lblNewLabel3);

textField_1 = new JTextField();
textField_1.setMargin(new Insets(1, 1, 1, 1));
textField_1.setHorizontalAlignment(SwingConstants.LEFT);
textField_1.setForeground(new Color(239, 239, 239));
textField_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
textField_1.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField_1.setColumns(10);
textField_1.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new
Color(160, 160, 160)), "Contrase\u00F1a", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
textField_1.setBackground(new Color(45, 3, 59));
textField_1.setBounds(40, 165, 295, 42);
contentPane.add(textField_1);

JLabel lblNewLabel_3 = new JLabel("Siguiente");
lblNewLabel_3.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(85, 5, 112));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(68, 4, 89));
    }
    /**
     * Comprobacion de credenciales
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        String[] objsToSearch = {"mail", "pass"};
        String[] objsReturn;
        objsReturn = manager.getObjectsInFile(objsToSearch, 2);
    }
});

```

```

        if(objsReturn[1].equals(textField.getText()) && objsReturn[3].equals(textField_1.getText())) {
            dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
            Major major = new Major();
            major.main(null);
        }else {
            lblNewLabel3.setVisible(true);
        }
    }
});
lblNewLabel_3.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel_3.setOpaque(true);
lblNewLabel_3.setBackground(new Color(68, 4, 89));
lblNewLabel_3.setForeground(new Color(212, 212, 212));
lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_3.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_3.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 18));
lblNewLabel_3.setBounds(127, 269, 126, 42);
contentPane.add(lblNewLabel_3);

JLabel lblNewLabel_4 = new JLabel("Manual de usuario");
lblNewLabel_4.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4.setForeground(new Color(185, 185, 185));
lblNewLabel_4.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4.setBounds(280, 331, 87, 14);
contentPane.add(lblNewLabel_4);

JLabel lblNewLabel_4_1 = new JLabel("Ayuda");
lblNewLabel_4_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_4_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4_1.setForeground(new Color(185, 185, 185));
lblNewLabel_4_1.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4_1.setBounds(233, 331, 37, 14);
contentPane.add(lblNewLabel_4_1);

Label label = new Label("");
label.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
//Agrega la funcionalidad de mover el frame en un label
MoverFrame mov = new MoverFrame(this);
label.addMouseListener(mov);
label.addMouseMotionListener(mov);
label.setBackground(new Color(63, 3, 82));
label.setBounds(0, 0, 303, 24);
contentPane.add(label);

setLocationRelativeTo(null);
}
}
package lunarsoftware.gdc.gui.auth;

import java.awt.EventQueue;

```

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.ImageIcon;
import java.awt.Dimension;
import javax.swing.SwingConstants;
import java.awt.Cursor;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowEvent;
import javax.swing.JTextField;
import javax.swing.border.TitledBorder;

import lunarsoftware.gdc.gui.Major;
import lunarsoftware.gdc.gui.MoverFrame;
import lunarsoftware.lib.jdbmanager.JdbManager;
```

```
import javax.swing.border.EtchedBorder;
import java.awt.Font;
import java.awt.ComponentOrientation;
import java.awt.Insets;
import java.awt.Label;
import java.awt.Component;
```

```
public class GetName extends JFrame {
    private JdbManager manager;
    private JPanel contentPane;
    private JTextField textField;
    private String mail;
    private String pass;
    private String userPath;

    /**
     * Iniziatore del frame (Ventana)
     */
    public static void main(String[] args, String mail, String pass) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    GetName frame = new GetName(mail, pass);
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

/**
 * Creacion del frame(Ventana)
 */
public GetName(String mailparam, String passparam) {
    userPath = System.getProperty("user.dir");
    this.mail = mailparam;
    this.pass = passparam;
    manager = new JdbManager(JdbManager.AUTH, JdbManager.UNMODIFIABLE, null);

    setUndecorated(true);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 377, 270);
    contentPane = new JPanel();
    contentPane.setSize(new Dimension(0, 500000));
    contentPane.setBackground(new Color(45, 3, 59));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("");
    lblNewLabel.setOpaque(true);
    lblNewLabel.addMouseListener(new MouseAdapter() {
        /**
         * Efecto hover sobre el label
         */
        @Override
        public void mouseEntered(MouseEvent e) {
            lblNewLabel.setBackground(new Color(255, 70, 70));
        }
        @Override
        public void mouseExited(MouseEvent e) {
            lblNewLabel.setBackground(new Color(45, 3, 60));
        }
    });
    /**
     * Cierre de ventana y finalizacion del programa
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
        System.exit(0);
    }
});
lblNewLabel.setBackground(new Color(45, 3, 60));
lblNewLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel.setToolTipText("");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel.setSize(new Dimension(100, 1000));
lblNewLabel.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\close.png"));
lblNewLabel.setBounds(340, 0, 37, 24);

```

```
contentPane.add(lblNewLabel);
```

```
JLabel lblNewLabel_1 = new JLabel("");
```

```
lblNewLabel_1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
```

```
lblNewLabel_1.addMouseListener(new MouseAdapter() {
```

```
    /**
```

```
     * Efecto hover sobre el label
```

```
    */
```

```
    @Override
```

```
    public void mouseEntered(MouseEvent e) {
```

```
        lblNewLabel_1.setBackground(new Color(250, 75, 170));
```

```
    }
```

```
    @Override
```

```
    public void mouseExited(MouseEvent e) {
```

```
        lblNewLabel_1.setBackground(new Color(45, 3, 60));
```

```
    }
```

```
    /**
```

```
     * Minimizacion de la ventana
```

```
    */
```

```
    @Override
```

```
    public void mouseClicked(MouseEvent e) {
```

```
        setState(JFrame.ICONIFIED);
```

```
    }
```

```
});
```

```
lblNewLabel_1.setIcon(new ImageIcon(userPath + "\\access\\icons\\frameslash\\minimize.png"));
```

```
lblNewLabel_1.setToolTipText("");
```

```
lblNewLabel_1.setSize(new Dimension(100, 1000));
```

```
lblNewLabel_1.setOpaque(true);
```

```
lblNewLabel_1.setHorizontalTextPosition(SwingConstants.CENTER);
```

```
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblNewLabel_1.setBackground(new Color(45, 3, 60));
```

```
lblNewLabel_1.setBounds(303, 0, 37, 24);
```

```
contentPane.add(lblNewLabel_1);
```

```
textField = new JTextField();
```

```
textField.setAlignmentY(Component.TOP_ALIGNMENT);
```

```
textField.setAlignmentX(Component.LEFT_ALIGNMENT);
```

```
textField.setMargin(new Insets(1, 1, 1, 1));
```

```
textField.setHorizontalAlignment(SwingConstants.LEFT);
```

```
textField.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
```

```
textField.setForeground(new Color(239, 239, 239));
```

```
textField.setFont(new Font("Tahoma", Font.PLAIN, 12));
```

```
textField.setBorder(new TitledBorder(new EtchedBorder(EtchedBorder.LOWERED, new Color(255, 255, 255), new Color(160, 160, 160)), "Nombre", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(239, 239, 239)));
```

```
textField.setBackground(new Color(45, 3, 59));
```

```
textField.setBounds(40, 100, 295, 42);
```

```
contentPane.add(textField);
```

```
textField.setColumns(10);
```

```
JLabel lblNewLabel_2 = new JLabel("Escribe tu nombre");
```

```
lblNewLabel_2.setForeground(new Color(239, 239, 239));
```



```

lblNewLabel_2.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 25));
lblNewLabel_2.setBounds(81, 42, 211, 46);
contentPane.add(lblNewLabel_2);

JLabel lblNewLabel_3 = new JLabel("Siguiente");
lblNewLabel_3.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(85, 5, 112));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_3.setBackground(new Color(68, 4, 89));
    }
    /**
     * Aqui el software obtiene el nombre y si este es null
     * le asigna un por default.
     *
     * El software agregar la credencial de inicio de sesion
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
        if(textField.getText() == null) {
            manager.writeObj("mail", mail);
            manager.writeObj("pass", pass);
            manager.writeObj("user", "admin");
        }else {
            manager.writeObj("mail", mail);
            manager.writeObj("pass", pass);
            manager.writeObj("user", textField.getText());
        }
        Major major = new Major();
        major.main(null);
    }
});

lblNewLabel_3.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblNewLabel_3.setOpaque(true);
lblNewLabel_3.setBackground(new Color(68, 4, 89));
lblNewLabel_3.setForeground(new Color(212, 212, 212));
lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_3.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_3.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 18));
lblNewLabel_3.setBounds(40, 172, 126, 42);
contentPane.add(lblNewLabel_3);

JLabel lblNewLabel_4 = new JLabel("Manual de usuario");
lblNewLabel_4.setHorizontalTextPosition(SwingConstants.CENTER);

```

```
lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4.setForeground(new Color(185, 185, 185));
lblNewLabel_4.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4.setBounds(280, 245, 87, 14);
contentPane.add(lblNewLabel_4);
```

```
JLabel lblNewLabel_4_1 = new JLabel("Ayuda");
lblNewLabel_4_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_4_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_4_1.setForeground(new Color(185, 185, 185));
lblNewLabel_4_1.setFont(new Font("Tahoma", Font.PLAIN, 10));
lblNewLabel_4_1.setBounds(233, 245, 37, 14);
contentPane.add(lblNewLabel_4_1);
```

```
JLabel lblNewLabel_3_1 = new JLabel("Omitir");
lblNewLabel_3_1.addMouseListener(new MouseAdapter() {
    /**
     * Efecto hover sobre el label
     */
    @Override
    public void mouseEntered(MouseEvent e) {
        lblNewLabel_3_1.setBackground(new Color(54, 3, 71));
    }
    @Override
    public void mouseExited(MouseEvent e) {
        lblNewLabel_3_1.setBackground(new Color(43, 3, 59));
    }
    /**
     * Aqui el software le asigna un nombre por default.
     *
     * El software agregar la credencial de inicio de sesion
     */
    @Override
    public void mouseClicked(MouseEvent e) {
        dispatchEvent(new WindowEvent(new Registro(), WindowEvent.WINDOW_CLOSING));
        manager.writeObj("mail", mail);
        manager.writeObj("pass", pass);
        manager.writeObj("user", "admin");
        Major major = new Major();
        major.main(null);
    }
});
lblNewLabel_3_1.setOpaque(true);
lblNewLabel_3_1.setHorizontalTextPosition(SwingConstants.CENTER);
lblNewLabel_3_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_3_1.setForeground(new Color(212, 212, 212));
lblNewLabel_3_1.setFont(new Font("Microsoft Sans Serif", Font.PLAIN, 18));
lblNewLabel_3_1.setBackground(new Color(43, 3, 59));
lblNewLabel_3_1.setBounds(209, 172, 126, 42);
contentPane.add(lblNewLabel_3_1);
```

```
Label label = new Label("");
label.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
//Agrega la funcionalidad de mover el frame en un label
MoverFrame mov = new MoverFrame(this);
label.addMouseListener(mov);
label.addMouseMotionListener(mov);
label.setBackground(new Color(63, 3, 82));
label.setBounds(0, 0, 303, 24);
contentPane.add(label);

setLocationRelativeTo(null);
}
}
```