Hacentest

Plataforma de exámenes de conducir teóricos para autoescuelas.



3ª Evaluación

Adán Jarillo Mérida

Tabla de contenido

[ANALISIS 4](#_Toc515325868)

[1.1 Introducción 4](#_Toc515325869)

[1.1.1 Propósito 4](#_Toc515325870)

[1.1.2 Ámbito del sistema 4](#_Toc515325871)

[1.1.3 Definiciones, siglas y abreviaciones 5](#_Toc515325872)

[1.1.4 Referencias 5](#_Toc515325873)

[1.1.5 Apreciación global 5](#_Toc515325874)

[1.2 Descripción Global 6](#_Toc515325875)

[1.2.1 Perspectiva del producto 6](#_Toc515325876)

[1.2.2 Funciones del producto 6](#_Toc515325877)

[1.2.3 Características de los usuarios 7](#_Toc515325878)

[1.2.4 Suposiciones y dependencias. 8](#_Toc515325879)

[1.2.5 Requisitos futuros. 9](#_Toc515325880)

[1.3 Requisitos específicos 9](#_Toc515325881)

[1.3.1 Requisitos del sistema 9](#_Toc515325882)

[1.3.2 Requisitos hardware 10](#_Toc515325883)

[1.3.3 Requisitos externos 10](#_Toc515325884)

[DISEÑO 11](#_Toc515325885)

[Diagrama de Gannt 11](#_Toc515325886)

[Diagrama entidad relacion 11](#_Toc515325887)

[Diagrama casos de uso 12](#_Toc515325888)

[Diagrama de clases 13](#_Toc515325889)

[Diagrama de secuencia inciar test 14](#_Toc515325890)

[Diagrama de flujo Bucle principal 15](#_Toc515325891)

[Grafo Bucle principal 16](#_Toc515325892)

[PRUEBAS 17](#_Toc515325893)

[Prueba valor limite completar test 17](#_Toc515325894)

[Prueba valores limite form. soporte 17](#_Toc515325895)

[Prueba filtrar usuario 18](#_Toc515325896)

[Prueba comprobar dni 19](#_Toc515325897)

[Prueba de rendimiento 20](#_Toc515325898)

[CODIFICACION 21](#_Toc515325899)

[Paquete acceso\_a\_datos 21](#_Toc515325900)

[Checker 21](#_Toc515325901)

[Encriptador 33](#_Toc515325902)

[GestorConsultas 33](#_Toc515325903)

[GestorSesiones 36](#_Toc515325904)

[GestorTest 39](#_Toc515325905)

[GestorUsuario 52](#_Toc515325906)

[InfoMsg 56](#_Toc515325907)

[MysqlC 57](#_Toc515325908)

[Paquete clases 61](#_Toc515325909)

[Administrador 61](#_Toc515325910)

[Alumno 62](#_Toc515325911)

[Consulta 62](#_Toc515325912)

[Desarrollador 64](#_Toc515325913)

[Enunciada 65](#_Toc515325914)

[Pregunta 65](#_Toc515325915)

[Profesor 68](#_Toc515325916)

[Respuesta 69](#_Toc515325917)

[Sesion 71](#_Toc515325918)

[Test 73](#_Toc515325919)

[Usuario 75](#_Toc515325920)

[Paquete controladores 80](#_Toc515325921)

[ChangeBtnListener 80](#_Toc515325922)

[ControlLabelListener 82](#_Toc515325923)

[EntrarComoListener 85](#_Toc515325924)

[EstadisticasBtnListener 87](#_Toc515325925)

[FormBtnListener 89](#_Toc515325926)

[FrameDrager 94](#_Toc515325927)

[JRowBtnListener 95](#_Toc515325928)

[NavMostrarListener 98](#_Toc515325929)

[NavRegListener 100](#_Toc515325930)

[OptionListener 101](#_Toc515325931)

[RadioButtonListener 104](#_Toc515325932)

[RespuestaBtnListener 105](#_Toc515325933)

[SesionBtnListener 107](#_Toc515325934)

[SoporteBtnListener 110](#_Toc515325935)

[TestBtnListener 112](#_Toc515325936)

[TextAreaListener 114](#_Toc515325937)

[TextFieldKeyListener 117](#_Toc515325938)

[TextFocusListener 121](#_Toc515325939)

[Paquete interfaz 123](#_Toc515325940)

[Ventana 123](#_Toc515325941)

[Paquete interfaz.componentes 250](#_Toc515325942)

[JChanger 250](#_Toc515325943)

[JControlLabel 251](#_Toc515325944)

[JDisplay 252](#_Toc515325945)

[JFormBtn 254](#_Toc515325946)

[JFormDisplay 257](#_Toc515325947)

[JNavLabel 260](#_Toc515325948)

[JOption 261](#_Toc515325949)

[JPopUp 264](#_Toc515325950)

[JRespuestaBtn 267](#_Toc515325951)

[JRowBtn 267](#_Toc515325952)

[JRowPanel 270](#_Toc515325953)

[Paquete interfaz.grupos 276](#_Toc515325954)

[DisplayGroup 276](#_Toc515325955)

[OptionGroup 278](#_Toc515325956)

[SesionSteup 280](#_Toc515325957)

[TextFieldGroup 280](#_Toc515325958)

[TextFieldGroupRel 286](#_Toc515325959)

[Paquete interfaz.listas 288](#_Toc515325960)

[JConsultaRowList 288](#_Toc515325961)

[JRowList 290](#_Toc515325962)

[JUserRowList 290](#_Toc515325963)

[Paquete principal 292](#_Toc515325964)

[Launcher 292](#_Toc515325965)

[IMPLEMENTACION 294](#_Toc515325966)

[Preparación del sistema 294](#_Toc515325967)

[Instalación de la aplicación 294](#_Toc515325968)

[Despliegue base de datos 294](#_Toc515325969)

[Entorno de desarrollo 294](#_Toc515325970)

[Versión ejecutable 294](#_Toc515325971)

[Futuras versiones 294](#_Toc515325972)

[MANTENIMIENTO 295](#_Toc515325973)

[Servicio interno de consultas 295](#_Toc515325974)

[Actualizaciones 295](#_Toc515325975)

# ANALISIS

## 1.1 Introducción

Este documento es una especificación de requisitos software (ERS) para el sistema de gestión de consultas. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE practica recomendada para, especificaciones de requisitos software ANSI/EEE 830, 1998.

### Propósito

Este documento ERS está dirigido tanto a desarrolladores que cuyo objetivo sea implementar mejoras o modificaciones del producto como a los posibles clientes finales, y el propósito del mismo es detallar el desarrollo de la aplicación, su implantación en el entorno del cliente y las diferentes tareas de mantenimiento que pudieran ser necesarias a lo largo del uso del producto.

### 1.1.2 Ámbito del sistema

El nombre del producto en durante sus versiones de desarrollo es “Hacentest” y su objetivo es ofrecer a diferentes autoescuelas una plataforma donde los alumnos puedan realizar test de conducir teóricos y llevar un registro de los mismos para obtener diversas estadísticas y ponerlas a disposición de los profesores.

El software ofrecerá tanto un sistema de registro integrado en la propia aplicación y ampliable a una versión web como una validación de Log-in a la propia plataforma.

La funcionalidad principal será la de generar diferentes test aleatorios o basados en las necesidades del usuario. El resultado de estos será registrado y posteriormente utilizado para obtener diferentes estadísticas.

La plataforma integrara además un sistema para actualizar los datos de los diferentes usuarios, dar de baja a usuarios de niveles inferiores y cambiar el nivel de los mismos en función de las necesidades del negocio.

Dentro de la aplicación no se gestionara ningún aspecto administrativo del negocio. Su funcionalidad es exclusivamente de nivel educativo.

### 1.1.3 Definiciones, siglas y abreviaciones

A continuación se exponen diferentes términos que se utilizaran a lo largo de la documentación:

BBDD: Base de datos.

Checker: Método que comprueba la validez de un dato y devuelve un valor lógico en consecuencia.

Componente: Referencia a los distintos objetos que conforman la interfaz gráfica de la aplicación.

Data: Conjunto de datos que comparten un objeto y la tabla que lo representa en la base de datos.

Formulario: Conjunto de componentes que recogen datos con el fin de introducirlos a la base de datos.

JRE: Versión de java instalada en el equipo del usuario.

Test: Conjunto de 30 preguntas con cuatro posibles opciones elegidas al azar.

S.O.: Sistema operativo.

### 1.1.4 Referencias

**Requisitos Java:**

Oracle JDK 8 and JRE 8 Certified System Configurations (<http://www.oracle.com/technetwork/java/javase/certconfig-2095354.html>)

**Requsitos Mysql:**

MySQL 5.7.21 (2018-01-15, General Availability) (https://dev.mysql.com/doc/relnotes/mysql/5.7/en/news-5-7-21.html)

### 1.1.5 Apreciación global

Este documento constara de:

Un apartado de análisis donde se expondrá el objetivo del proyecto detallando las partes que lo conforman y el comportamiento esperado de las mismas durante la ejecución del producto. También se describirá el cometido y contenido del documento en relación al desarrollo del producto.

Un apartado de Diseño donde mediante distintos diagramas gráficos se describirá la composición de la plataforma, su comportamiento y como se relaciona entre sus distintas partes.

Un apartado de pruebas que verificara la correcta funcionalidad de la plataforma y documentara los métodos utilizados para garantizarla.

Un apartado de codificación donde se pondrá a disposición de los interesados el código fuente de la aplicación.

Un apartado de implantación que describe los pasos necesarios para la correcta integración de la plataforma en el entorno del cliente.

Un apartado de mantenimiento donde se detallaran las posibles necesidades del cliente en el tiempo que disfrute del producto y la forma en las que se llevarían a cabo.

## 1.2 Descripción Global

### 1.2.1 Perspectiva del producto

La plataforma actúa como un sistema independiente, no dependerá de terceros más allá de un S.O. y una conexión con el servidor que almacene la B.B.D.D. En caso de una futura adición de un sistema web ambos podrían funcionar de manera independiente mientras este relacionados con la misma B.B.D.D.

### 1.2.2 Funciones del producto

**Gestión de usuarios:**

La plataforma permite el registro de nuevos usuarios en la categoría más baja (alumnos), más adelante por medios de un administrador se podrá actualizar su categoría, el sistema de registro se basa en un formulario que mediante distintos componentes recogerá los distintos datos necesarios para crear un nuevo registro en la B.B.D.D y estos mismos serán validados mediante un Checker.

Mediante otro formulario que recogerá el Nick y la contraseña del usuario ambos datos se comprobaran en la B.B.D.D y permitirá el ingreso a la plataforma en función del nivel de acceso del usuario comprobado.

**Checker:**

La plataforma dispondrá de una clase destina a validar los datos introducidos con el fin de controlar los registros introducidos en la base de datos. Esta clase tendrá como objetivo controlar la longitud y tipo de caracteres que los usuarios introducirán en los campos de texto de los formularios, además en casos especiales realizara comprobaciones añadidas en función de patrones o criterios propios de los datos.

**Generador de Test:**

Mediante este sistema se podrán generar diferentes test en funciones de las necesidades del usuario, incluyendo más preguntas de un tema u otro según una decisión previa del usuario. Cada vez que un usuario contesta una pregunta se generara un registro en la B.B.D.D que constara de la elección del usuario, si fue correcta, el id del test en el que se seleccionó la pregunta de forma que más adelante podremos utilizar estos datos para la gestión de estadísticas.

**Gestión de estadísticas:**

La plataforma podrá generar a partir de los registros generados al realizar test distintas estadísticas (Preguntas más o menos acertadas, Temas más difíciles, porcentajes de test aprobados y suspensos, coeficiente de mejora de cada alumno). Estos datos estarán a disposición de los usuarios según sus niveles de acceso y podrán utilizarse para generar informes.

**Interfaz gráfica:**

La aplicación contara con una interfaz gráfica destinada a facilitar el uso de la aplicación haciéndola más intuitiva y descriptiva, consiguiendo de esta manera una experiencia de usuario más cómoda y fluida.

**Acceso a datos:**

El acceso a datos se realizará mediante un driver guardado en el apartado de librerías (lib) de la aplicación especificado más adelante.

El software contara con una clase dedica a la conexión con la B.B.D.D. y la generación de sentencias SQL necesarias para el acceso a datos, esta clase no contara con información propia de la B.B.D.D. (más allá del nombre y la ruta de acceso).

### 1.2.3 Características de los usuarios

La plataforma esta destina a distintos tipos de usuarios en función de las tareas que vayan a realizar dentro de la misma, sin embargo contaran con datos comunes:

Nombre, apellidos, nick, contraseña, email, DNI, fecha de ingreso y fecha de última sesión. Estos datos conformaran también partes de los datos de los usuarios registrados en la B.B.D.D.

Además todos los usuarios contaran con un nivel de acceso que los dividirá según su rol en la plataforma.

**Nivel de acceso 4 y 3 (Master y Desarrollador):**

Rol destinado a usuarios con conocimientos de programación y gestión de base de datos, su tarea sea el testeo y depuración de la propia aplicación. Este nivel tendrá acceso total a la aplicación con el fin de poder probar cada una de las funcionalidades de la misma.

Pueden existir varios desarrolladores pero solo uno de ellos puede ser Master, que tendrá la función añadida de gestionar al resto de desarrolladores pudiendo dar de alta nuevos o eliminar a los ya existentes.

**Nivel de acceso 2 (Administrador):**

Rol destinado a usuarios con conocimientos de la gestión del negocio en el cual se desea implementar la aplicación, su función principal es la de gestionar al resto de usuarios (salvo desarrolladores, invisibles en las listas de usuarios) pudiendo modificar datos clave de otros usuarios, darles de baja y cambiar en nivel de acceso de usuarios inferiores (Salvo para ascenderles a desarrollador o master).

**Nivel de acceso 1 (Profesor):**

Rol destinado a los distintos trabajadores de la autoescuela, su función será la de supervisar a los alumnos mediante las diferentes estadísticas de todos los alumnos que ofrecerá la plataforma.

**Nivel de acceso 0 (Alumno):**

Rol destinado a los clientes del negocio, su nivel de acceso es el más bajo y su función será la de realizar test y ver las estadísticas generales o de sus propios test.

### 1.2.4 Suposiciones y dependencias.

**Implementación en varios negocios**

La aplicación en su versión de desarrollo está destinada a una única autoescuela, aunque está diseñada para poder ser escalada a un modelo de múltiples autoescuela teniendo en cuenta la necesidad de implementar un sistema para diferenciar los alumnos, profesores y administradores de diferentes escuelas. Las posibles opciones son bases de datos divididas, códigos de registro diferentes para los alumnos, selección de centro durante el registro. La opción principal es la de una única base de datos en un servidor remoto y un código secreto único para cada centro que recibirá el primer administrador y que deberá establecer en forma de archivo en cada equipo durante la implementación de la aplicación. Este archivo será leído durante la ejecución del programa para saber en qué centro se está ejecutando.

**Cambios de criterios/modelo del test.**

Se valora la posibilidad de que el actual modelo de examen teórico de la DGT (Dirección General de Tráfico, España) o La implementación de la aplicación en países con un modelo diferente. Los cambios serán dependientes del modelo a implementar y por lo tanto tendrán que ser considerados una vez que se conozcan los nuevos criterios.

**Traducción de la interfaz grafica**

En el caso de querer implementar aplicación en países con un idioma diferente e incompatible con el Castellano, idioma utilizado durante el desarrollo de la interfaz, será necesario reconstruir la inicialización de los componentes de la interfaz, estos deberán tener en cuenta un parámetro que haga referencia al idioma en que se desea ejecutar la aplicación y establecer el texto en consecuencia. Este sistema se deberá implementar también en cada uno de los métodos destinados a generar textos dentro de la aplicación

### 1.2.5 Requisitos futuros.

**Servidor Web:**

En el caso de una posible adaptación de la B.B.D.D. o la propia aplicación a un entorno web, será necesario que el cliente cuente con una conexión a internet con capacidad suficiente para conectar los diferentes equipos con el servidor.

## 1.3 Requisitos específicos

### 1.3.1 Requisitos del sistema

**Java JRE Versión 1.8 o posterior:**

Aquellos equipos que deseen ejecutar la aplicación deberán contar con una versión de Java compatible. Los requisitos de esta versión están descritos en el documento Oracle JDK 8 and JRE 8 Certified System Configurations en el apartado de referencias (1.1.4).

**MySQL Version 5.7.21:**

El servidor local deberá ser compatible con la versión 5.7.21 de MySQL. Las especificaciones de esta versión están descritas en el documento MySQL 5.7.21 (2018-01-15, General Availability) en el apartado de referencias (1.1.4).

**S.O. compatible con Java JRE 1.8:**

El sistema operativo sobre el que se ejecutara la aplicación debe ser compatible con la versión anteriormente mencionada:

* Windows
* Windows 10 (8u51 y superiores)
* Windows 8.x (escritorio)
* Windows 7 SP1
* Windows Vista SP2
* Windows Server 2008 R2 SP1 (64 bits)
* Windows Server 2012 y 2012 R2 (64 bits)
* RAM: 128 MB
* Espacio en disco: 124 MB para JRE; 2 MB para Java Update
* Procesador: Mínimo Pentium 2 a 266 MHz
* Exploradores: Internet Explorer 9 y superior, Firefox
* Mac OS X
* Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
* Privilegios de administrador para la instalación
* Explorador de 64 bits
* Se requiere un explorador de 64 bits (Safari, por ejemplo) para ejecutar Oracle Java en Mac.
* Linux
* Oracle Linux 5.5+1
* Oracle Linux 6.x (32 bits), 6.x (64 bits)2
* Oracle Linux 7.x (64 bits)2 (8u20 y superiores)
* Red Hat Enterprise Linux 5.5+1, 6.x (32 bits), 6.x (64 bits)2
* Red Hat Enterprise Linux 7.x (64 bits)2 (8u20 y superiores)
* Suse Linux Enterprise Server 10 SP2+, 11.x
* Suse Linux Enterprise Server 12.x (64 bits)2 (8u31 y superiores)
* Ubuntu Linux 12.04 LTS, 13.x
* Ubuntu Linux 14.x (8u25 y superiores)
* Ubuntu Linux 15.04 (8u45 y superiores)
* Ubuntu Linux 15.10 (8u65 y superiores)
* Exploradores: Firefox

### 1.3.2 Requisitos hardware

**Equipo:**

* Compatible con alguno de los S.O. anteriormente mencionados.
* RAM: 512mb
* Espacio en disco: 1GB
* Procesador: Mínimo Pentium 2 a 266 MHz

### 1.3.3 Requisitos externos

**Servidor Local:**

Compatible con la versión de MySQL anteriormente mencionada en este documento.

# DISEÑO

## Diagrama de Gannt



## Diagrama entidad relacion



## Diagrama casos de uso



## Diagrama de clases



## Diagrama de secuencia inciar test



## Diagrama de flujo Bucle principal



## Grafo Bucle principal



## Diagrama de flujo Vista invitado



## Grafo Vista invitado



## Diagrama de flujo Vista alumno



## Grafo Vista alumno



## Diagrama de flujo Vista profesor



## Grafo Vista profesor



## Diagrama de flujo Vista administrador



## Grafo Vista administrador



## Diagrama de flujo Vista desarrollador



## Grafo Vista desarrollador



## Diagrama de flujo Display Login



## Grafo de Display login



# PRUEBAS

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba valor limite completar test | | | |
| Partimos desde un punto determina durante la ejecución de un test, en el que hemos respondido a 27 preguntas. La condición comprobada para dar por finalizado el test es que el tamaño de la lista de preguntas respondidas sea menor que 30. En caso positivo la respuesta esperada es la llamada del método \*actualizar pregunta. En caso negativo la respuesta esperada es la llamada al método \*completar test. | | | |
| VALOR INTRODUCIDO | CONDICION | RESPUESTA ESPERADA | RESPUESTA OBTENIDA |
| 27 | <30 | POSITIVA | POSITIVA |
| 28 | <30 | POSITIVA | POSITIVA |
| 29 | <30 | POSITIVA | POSITIVA |
| 30 | <30 | NEGATIVA | NEGATIVA |

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba valores limite form. soporte | | | |
| Comprobaremos valores límites del campo descripción del formulario de soporte:   * 0 caracteres * 300 caracteres * 301 caracteres | | | |
| VALOR INTRODUCIDO | CONDICION | RESPUESTA ESPERADA | RESPUESTA OBTENIDA |
| 0 caracteres | >300 || equals(“”) | POSITIVA | POSITIVA |
| 300 caracteres | >300 || equals(“”) | NEGATIVA | NEGATIVA |
| 301 | >300 || equals(“”) | POSITIVA | POSITIVA |

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba filtrar usuario | | | |
| Se debe filtrar de entre 4 usuarios según su estado, Nick, DNI o acceso) comparando la cadena introducida con cada uno de estos valores, los datos de los usuarios a filtrar son:  Usuario 1: Nick: n1, Estado: online, Acceso: Alumno, DNI: 00000001R  Usuario 2: Nick: n2, Estado: online, Acceso: Profesor, DNI: 00000002W  Usuario 3: Nick: n3, Estado: offline, Acceso: Administrador, DNI: 00000003A  Usuario 4: Nick: n4, Estado: online, Acceso: Desarrollador, DNI: 00000004G | | | |
| VALOR INTRODUCIDO | CONDICION | RESPUESTA ESPERADA | RESPUESTA OBTENIDA |
| n | ? like %n% | Usuario 1, Usuario 2, Usuario 3 Usuario 4 | Usuario 1, Usuario 2, Usuario 3 Usuario 4 |
| N2 | ? like %n% | Usuario 1 | Usuario1 |
| 0000000 | ? like %n% | Usuario 1, Usuario 2, Usuario 3 Usuario 4 | Usuario 1, Usuario 2, Usuario 3 Usuario 4 |
| ONLINE | ? like %n% | Usuario 1, Usuario 2 | Usuario 1, Usuario 2 |
| Administrador | ? like %n% | Usuario 3 | Usuario 3 |
| Ajministrairt | ? like %n% | - | - |
| \_\_ | ? like %n% | \_ | Usuario 1, Usuario 2, Usuario 3 Usuario 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba comprobar dni | | | |
| El método \*checkDNI de la clase checker debe retornarnos un objeto InfoMsg compuesto de cuatro valores (texto, icono, booleano y color). Durante la comprobación de la cadena introducida se podrán tomar caminos diferentes:   * Camino 1: No hay coincidencia de formato * Camino 2: No hay coincidencia de letra * Camino 3: Hay coincidencia con registro en la base de datos * Camino 4: Cumple formato, letra y no está registrado | | | |
| VALOR INTRODUCIDO | CONDICION | RESPUESTA ESPERADA | RESPUESTA OBTENIDA |
| ¿¿?¡kjdlkasjdfhX | Formato coincide con "\\d{8}[a-zA-Z]" | CAMINO 1 | CAMINO 1 |
| 51512452X | Formato coincide con \\d{8}[a-zA-Z] && letra coincide con posición[E(DIGITOS)%23] | CAMINO 2 | CAMINO 2 |
| 51512452L | Formato coincide con \\d{8}[a-zA-Z] && letra coincide con posición[E(DIGITOS)%23] && DNI no registrado | CAMINO 3 | CAMINO 3 |
| 00000001R | Formato coincide con \\d{8}[a-zA-Z] && letra coincide con posición[E(DIGITOS)%23] && DNI no registrado | CAMINO 4 | CAMINO 4 |

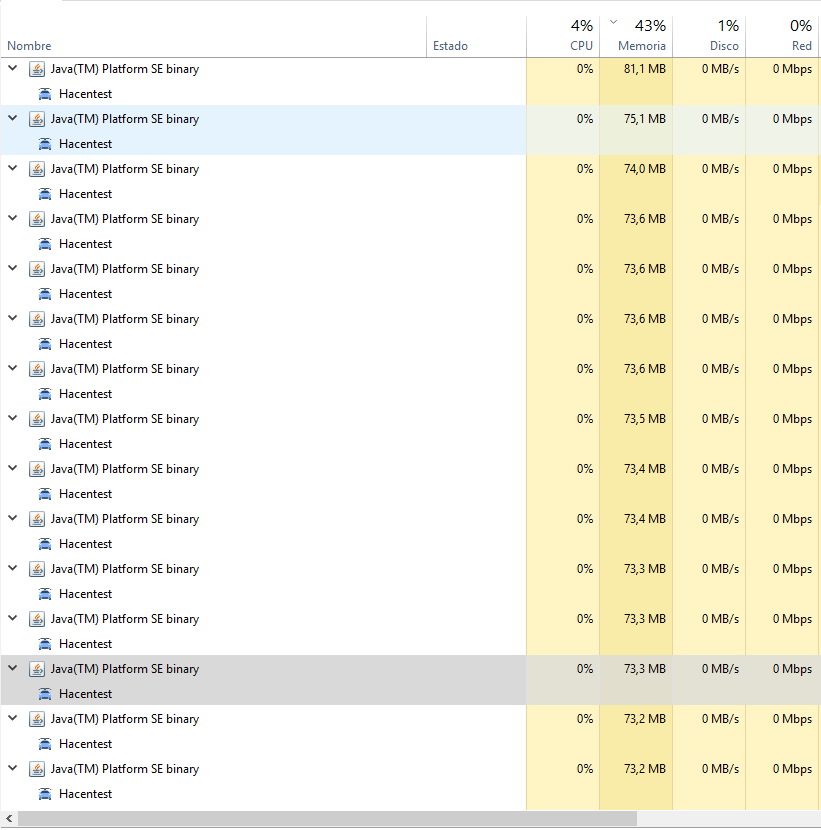
## Prueba de rendimiento

Memoria consumida por la aplicación en su versión alpha 1.0.0: 85,3MB



Memoria consumida por la aplicación ejecutándose 15 veces en su versión alpha : 964MB

El consumo de cpu, disco y red es nulo.



# CODIFICACION

## Paquete acceso\_a\_datos

### Checker

package acceso\_a\_datos;

import java.awt.Color;

import java.sql.ResultSet;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Checker {

private final Color COLOR\_ERROR = new Color(219, 186, 70);

private final Color COLOR\_CHECK = new Color(50, 205, 50);

GestorUsuarios gestorUsuarios;

GestorConsultas gestorConsultas;

public Checker(GestorUsuarios gestorUsuarios, GestorConsultas gestorConsultas) {

this.gestorUsuarios = gestorUsuarios;

this.gestorConsultas = gestorConsultas;

}

public InfoMsg checkPassword(String s) {

InfoMsg info;

Pattern pat = Pattern.compile("(?=.\*[0-9])(?=.\*[a-z])(?=.\*[A-Z])(?=\\S+$).{8,}");

Matcher mat = pat.matcher(s);

if (mat.matches()) {

info = new InfoMsg("Contraseña valida", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

Pattern pat2 = Pattern.compile("(?=.\*[0-9])(?=\\S+$).{1,}");

Matcher mat2 = pat2.matcher(s);

if (!mat2.matches()) {

info = new InfoMsg("Debe incluir un numero", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

pat2 = Pattern.compile("(?=.\*[a-z])(?=\\S+$).{1,}");

mat2 = pat2.matcher(s);

if (!mat2.matches()) {

info = new InfoMsg("Debe incluir una minuscula", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

pat = Pattern.compile("(?=.\*[A-Z])(?=\\S+$).{1,}");

mat2 = pat2.matcher(s);

if (!mat2.matches()) {

info = new InfoMsg("Debe incluir una mayuscula", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

info = new InfoMsg("Minimo 8 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

}

}

}

return info;

}

public InfoMsg checkStringStrict(String s) {

// Solo letras, sin espacios maximo 20 caracteres

InfoMsg info;

Pattern pat = Pattern.compile("^[a-zA-Z]{2,20}$");

Matcher mat = pat.matcher(s);

if (mat.matches()) {

info = new InfoMsg("Nombre valido", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

if (s.length() > 20) {

info = new InfoMsg("Maximo 20 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

if (s.length() < 2) {

info = new InfoMsg("Minimo 2 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

info = new InfoMsg("Solo puede incluir letras", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

}

}

return info;

}

public InfoMsg checkStringFlex(String s) {

// Solo letras, numeros y guiones, sin espacios maximo 20 caracteres

// Reconvertimos la cadana a primera mayuscula, siguientes minusculas.

// "^[a-zA-Z0-9\_-]{4,20}$"

InfoMsg info = null;

Pattern pat = Pattern.compile("^[a-zA-Z0-9\_-]{4,20}$");

Matcher mat = pat.matcher(s);

if (mat.matches()) {

// Cumple formato

if (!gestorUsuarios.buscarUsuarioExacto("nick", s)) {

info = new InfoMsg("Nick valido", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("El nick no esta disponible", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

} else {

if (s.length() > 20) {

info = new InfoMsg("Maximo 20 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

if (s.length() < 4) {

info = new InfoMsg("Minimo 4 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

info = new InfoMsg("Solo letras, numeros y guiones", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

}

}

return info;

}

public InfoMsg checkEmail(String s) {

// Formato Email.

InfoMsg info = null;

Pattern pat = Pattern.compile("^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$", Pattern.CASE\_INSENSITIVE);

Matcher mat = pat.matcher(s);

if(mat.matches()) {

if (!gestorUsuarios.buscarUsuarioExacto("email", s)) {

info = new InfoMsg("Email valido", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("Este email ya esta en uso", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

} else {

info = new InfoMsg("Formato no reconocido", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

return info;

}

public InfoMsg checkStringStrict(String s, int nCadenas) {

// Divide la cadena en subCadeanes en cada espacio en blanco. Comprueba que el

// numero de cadenas coincide

// con nCadenas y aplica checkStringStrict a cada una de las cadenas.

String[] apellidos = s.split(" ");

InfoMsg info, infoS1, infoS2;

if (apellidos.length == nCadenas) {

infoS1 = checkStringStrict(apellidos[0]);

infoS2 = checkStringStrict(apellidos[1]);

if (infoS1.isCheck() && infoS2.isCheck()) {

info = new InfoMsg("Apellidos validos", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

if (infoS1.isCheck()) {

if (apellidos[1].length() > 20) {

info = new InfoMsg("Maximo 20 caracteres por apellido", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

if (apellidos[1].length() < 2) {

info = new InfoMsg("Minimo 2 caracteres por apellido", "/imagenes/error\_black\_32px.png",

false, COLOR\_ERROR);

} else {

info = infoS2;

}

}

} else {

if (apellidos[0].length() > 20) {

info = new InfoMsg("Maximo 20 caracteres por apellido", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

if (apellidos[0].length() < 2) {

info = new InfoMsg("Minimo 2 caracteres por apellido", "/imagenes/error\_black\_32px.png",

false, COLOR\_ERROR);

} else {

info = infoS1;

}

}

}

}

} else {

if (apellidos.length > 2) {

info = new InfoMsg("Maximo 2 apellidos", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

info = new InfoMsg("Minimo 2 apellidos", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

}

return info;

}

public InfoMsg checkStringFlexKey(String s) {

// Solo letras, numeros y guiones, sin espacios maximo 20 caracteres

// Reconvertimos la cadana a primera mayuscula, siguientes minusculas.

// "^[a-zA-Z0-9\_-]{4,20}$"

//La cadena debe existir dentro de la base de datos ene l campo indicado

InfoMsg info = null;

Pattern pat = Pattern.compile("^[a-zA-Z0-9\_-]{4,20}$");

Matcher mat = pat.matcher(s);

if (mat.matches()) {

// Cumple formato

if(gestorUsuarios.buscarUsuarioExacto("nick", s)){

info = new InfoMsg("Usuario encontrado", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("El usuario no existe", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

} else {

if (s.length() > 20) {

info = new InfoMsg("Maximo 20 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

if (s.length() < 4) {

info = new InfoMsg("Minimo 4 caracteres", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

} else {

info = new InfoMsg("Solo letras, numeros y guiones", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

}

}

return info;

}

public InfoMsg checkDNI(String s) {

// Comprueba longitud = 9 empieza por numero o x | y | z (NIE), comprobara que

// la letra coincide

// segun el numero y que no esta registrado en la BBDD

InfoMsg info = null;

char[] letras = { 'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H',

'L', 'C', 'K', 'E', 'T' };

if (!gestorUsuarios.buscarUsuarioExacto("dni", s)) {

if (s.length() > 0) {

char pL = s.charAt(0);

switch (pL) {

case 'x' | 'X':

s = "0" + s.substring(1);

break;

case 'y' | 'Y':

s = "1" + s.substring(1);

break;

case 'z' | 'Z':

s = "2" + s.substring(1);

break;

}

}

Pattern pat = Pattern.compile("\\d{8}[a-zA-Z]");

Matcher mat = pat.matcher(s);

if (mat.matches()) {

int nDNI = Integer.parseInt(s.substring(0, 8));

int resto = nDNI % 23;

char letra = s.charAt(8);

if (Character.toUpperCase(letra) == letras[resto]) {

info = new InfoMsg("DNI valido", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("La letra no coincide", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

} else {

info = new InfoMsg("Formato incorrecto", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

} else {

info = new InfoMsg("Este DNI ya esta en uso", "/imagenes/error\_black\_32px.png", false, COLOR\_ERROR);

}

return info;

}

public InfoMsg checkSimilitud(String sPrincipal, String sRelacionada, boolean check) {

InfoMsg info = null;

if(check) {

if (sPrincipal.equals(sRelacionada)) {

info = new InfoMsg("Las contraseñas coinciden", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("Las contraseñas no coinciden", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

} else {

info = checkPassword(sRelacionada);

if (info.isCheck()) {

info = checkSimilitud(sPrincipal, sRelacionada, true);

}

}

return info;

}

public InfoMsg checkCorrespondencia(String sPrincipal, String sRelacionada, boolean check) {

InfoMsg info = null;

Encriptador enc = new Encriptador();

if(check) {

if(enc.compararMD5(sPrincipal, gestorUsuarios.getPassword("nick", sRelacionada))) {

info = new InfoMsg("Contraseña Correcta", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

} else {

info = new InfoMsg("Contraseña incorrecta", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

} else {

info = new InfoMsg("Usuario no encontrado", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

}

return info;

}

public ResultSet buscar(String s) {

ResultSet rs = null;

rs = gestorUsuarios.getUsuarioFilter(s);

return rs;

}

public InfoMsg checkFreeString(String s, int caracteres) {

InfoMsg info = null;

if(s.length() > caracteres) {

info = new InfoMsg("Maximo "+caracteres+" caracteres", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

if(s.length() < 2) {

info = new InfoMsg("Minimo 2 caracteres", "/imagenes/error\_black\_32px.png", false,

COLOR\_ERROR);

} else {

info = new InfoMsg("Formato correcto", "/imagenes/checkbox\_32px.png", true, COLOR\_CHECK);

}

}

return info;

}

public ResultSet consultas(String s) {

ResultSet rs = null;

rs = gestorConsultas.getConsultaFilter(s);

return rs;

}

}

### Encriptador

**package** acceso\_a\_datos;

**import** org.apache.commons.codec.digest.DigestUtils;

**public** **class** Encriptador {

**public** String encriptarMD5(String cadena) {

**return** DigestUtils.md5Hex(cadena);

}

**public** **boolean** compararMD5(String cadena, String encriptada) {

**boolean** control = **false**;

**if** (**this**.encriptarMD5(cadena).equals(encriptada)) {

control = **true**;

}

**return** control;

}

}

### GestorConsultas

package acceso\_a\_datos;

import java.sql.ResultSet;

import clases.Consulta;

public class GestorConsultas {

private MysqlC mysqlc;

public GestorConsultas(MysqlC mysqlc) {

super();

this.mysqlc = mysqlc;

}

public boolean generarConsulta(Consulta consulta) {

boolean control = false;

if (mysqlc.isConectada()) {

String nombreTabla = "consulta";

String campos = "fecha, asunto, descripcion, fk\_usuario";

String value = "CURRENT\_TIMESTAMP(), '"+consulta.getAsunto()+"', '"+consulta.getDescripcion()+"',"+consulta.getUsuario().getPk\_usuario();

if(mysqlc.insertInto(nombreTabla, campos, value)){

control = true;

}

}

return control;

}

public boolean generarConsultaInvitado(Consulta consulta) {

boolean control = false;

if (mysqlc.isConectada()) {

String nombreTabla = "consulta";

String campos = "fecha, asunto, descripcion, fk\_usuario";

String value = "CURRENT\_TIMESTAMP(), '"+consulta.getAsunto()+"', '"+consulta.getDescripcion()+"', 1";

if(mysqlc.insertInto(nombreTabla, campos, value)){

control = true;

}

}

return control;

}

public ResultSet getConsultaFilter(String value) {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("consulta inner join usuario on consulta.fk\_usuario=usuario.pk\_usuario","nick LIKE '%"+value+"%' OR fecha LIKE '%"+value+"%' OR acceso LIKE '%"+value+"%'OR asunto LIKE '%"+value+"%'");

}

return rs;

}

public ResultSet getConsulta(String value) {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("consulta inner join usuario on consulta.fk\_usuario=usuario.pk\_usuario", value);

}

return rs;

}

public ResultSet getAllConsultas() {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("consulta inner join usuario on consulta.fk\_usuario=usuario.pk\_usuario");

}

return rs;

}

}

### GestorSesiones

package acceso\_a\_datos;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Random;

import clases.Sesion;

public class GestorSesiones {

private MysqlC mysqlc;

GestorUsuarios gestorUsuarios;

public GestorSesiones(MysqlC mysqlc, GestorUsuarios gestorUsuarios) {

super();

this.mysqlc = mysqlc;

this.gestorUsuarios = gestorUsuarios;

}

public ResultSet getAllSesions() {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("sesion");

}

return rs;

}

public Sesion abrirSesion(ResultSet rs) {

Sesion sesion = null;

sesion = new Sesion(rs, generarCodigo(getAllSesions()));

if (mysqlc.isConectada()) {

mysqlc.insertInto("sesion", "codigo, tiempo\_inicio, fk\_usuario",

"'" + sesion.getCodigo() + "', CURRENT\_TIMESTAMP() , " + sesion.getUsuario().getPk\_usuario());

}

return sesion;

}

public void cerrarSesion(Sesion sesion) {

if (mysqlc.isConectada()) {

mysqlc.Update("sesion", "tiempo\_inicio = tiempo\_inicio, tiempo\_final= CURRENT\_TIMESTAMP()", "codigo='"+sesion.getCodigo()+"'");

String where = "nick='"+sesion.getUsuario().getNick()+"'";

gestorUsuarios.actualizarEstadoUsuario("offline", where);

}

}

public String generarCodigo(ResultSet rs) {

boolean seguir = true;

String codigo = "";

while (seguir) {

seguir = false;

codigo = randomCodigo();

try {

rs.beforeFirst();

while (rs.next()) {

if (codigo.equals(rs.getString("codigo"))) {

seguir = true;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return codigo;

}

private String randomCodigo() {

Random rng = new Random();

String codigo = "@";

int[] numeros = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

char[] letras = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k', 'l', 'm', 'n', 'o', 'p', 'k', 'r', 's', 't',

'u', 'v', 'w', 'x', 'y', 'z' };

char c;

int n;

while (codigo.length() < 15) {

c = letras[rng.nextInt(letras.length)];

n = numeros[rng.nextInt(numeros.length)];

codigo = codigo + c + n;

}

return codigo;

}

}

### GestorTest

package acceso\_a\_datos;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Random;

import clases.Pregunta;

import clases.Respuesta;

import clases.Test;

import clases.Usuario;

public class GestorTest {

private MysqlC mysqlc;

public GestorTest(MysqlC mysqlc) {

this.mysqlc = mysqlc;

}

public boolean añadirPregunta(Pregunta pregunta) {

boolean control = false;

String codigo\_pregunta = generarCodigo(getAllPreguntas(), '#', "codigo\_pregunta");

if (mysqlc.isConectada()) {

String nombreTabla = "pregunta";

String campos = "enunciado, indice\_correcto, codigo\_pregunta";

String value = "'" + pregunta.getEnunciado() + "', '" + pregunta.getIndiceCorrecto() + "', '"

+ codigo\_pregunta + "'";

if (mysqlc.insertInto(nombreTabla, campos, value)) {

ResultSet rs = getPregunta("codigo\_pregunta", codigo\_pregunta);

Respuesta[] respuestas = pregunta.getRespuestas();

try {

if (rs.next()) {

int fk\_pregunta = rs.getInt("pk\_pregunta");

for (int y = 0; y < respuestas.length; y++) {

respuestas[y].setFk\_pregunta(fk\_pregunta);

}

int i = 0;

while (i < respuestas.length && añadirRespuesta(respuestas[i])) {

i++;

}

if (i == respuestas.length) {

control = true;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

return control;

}

private boolean añadirRespuesta(Respuesta respuesta) {

boolean control = false;

if (mysqlc.isConectada()) {

String nombreTabla = "respuesta";

String campos = "texto, indice, fk\_pregunta";

String value = "'" + respuesta.getTexto() + "', '" + respuesta.getIndice() + "', "

+ respuesta.getFk\_pregunta();

if (mysqlc.insertInto(nombreTabla, campos, value)) {

control = true;

}

}

return control;

}

public ResultSet getPregunta(String campo, String value) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("pregunta", " " + campo + "='" + value + "'");

}

return rs;

}

public ResultSet getAllPreguntas() {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("pregunta");

}

return rs;

}

public ResultSet getAllTest() {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("test");

}

return rs;

}

public String generarCodigo(ResultSet rs, char identificador, String campo) {

boolean seguir = true;

String codigo = "";

while (seguir) {

seguir = false;

codigo = randomCodigo(identificador);

try {

rs.beforeFirst();

while (rs.next()) {

if (codigo.equals(rs.getString(campo))) {

seguir = true;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return codigo;

}

private String randomCodigo(char identificador) {

Random rng = new Random();

String codigo = "" + identificador;

int[] numeros = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

char[] letras = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k', 'l', 'm', 'n', 'o', 'p', 'k', 'r', 's', 't',

'u', 'v', 'w', 'x', 'y', 'z' };

char c;

int n;

while (codigo.length() < 15) {

c = letras[rng.nextInt(letras.length)];

n = numeros[rng.nextInt(numeros.length)];

codigo = codigo + c + n;

}

return codigo;

}

public ResultSet getTest(String campo, String value) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("test", " " + campo + "='" + value + "'");

}

return rs;

}

public ResultSet getEnunciada(String campo, String value) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("enuncian", " " + campo + "='" + value + "'");

}

return rs;

}

public ResultSet getRespuesta(String where) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("respuesta", where);

}

return rs;

}

public ResultSet getTest(String where) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("test", where);

}

return rs;

}

public ResultSet getPregunta(String campo, String value, String modificador) {

ResultSet rs = null;

if (mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("pregunta", " " + campo + " " + modificador + " " + value);

}

return rs;

}

public Pregunta getPreguntaDisponible(Test test) {

Pregunta pregunta = null;

ResultSet rs = getPregunta("pk\_pregunta",

"(SELECT enuncian.fk\_pregunta FROM enuncian WHERE fk\_test=" + test.getPk\_test() + ")", "NOT IN");

try {

if (rs.next()) {

Random rng = new Random();

int nIteraciones = rng.nextInt(100);

int contador = 0;

while (contador < nIteraciones) {

if (rs.next()) {

contador++;

} else {

rs.beforeFirst();

if (rs.next()) {

contador++;

}

}

}

pregunta = new Pregunta(rs);

Respuesta ra = null;

Respuesta rb = null;

Respuesta rc = null;

ResultSet rsa = getRespuesta("fk\_pregunta=" + pregunta.getPk\_pregunta() + " AND indice ='a'");

ResultSet rsb = getRespuesta("fk\_pregunta=" + pregunta.getPk\_pregunta() + " AND indice ='b'");

ResultSet rsc = getRespuesta("fk\_pregunta=" + pregunta.getPk\_pregunta() + " AND indice ='c'");

if (rsa.next()) {

ra = new Respuesta(rsa);

}

if (rsb.next()) {

rb = new Respuesta(rsb);

}

if (rsc.next()) {

rc = new Respuesta(rsc);

}

Respuesta[] respuestas = { ra, rb, rc };

pregunta.setRespuestas(respuestas);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return pregunta;

}

public Test empezarTest(Usuario usuario) {

Test test = null;

String codigo\_test = generarCodigo(getAllTest(), '$', "codigo\_test");

if (mysqlc.isConectada()) {

String nombreTabla = "test";

String campos = "fk\_usuario, codigo\_test";

String value = usuario.getPk\_usuario() + ", '" + codigo\_test + "'";

;

mysqlc.insertInto(nombreTabla, campos, value);

}

ResultSet rs = getTest("codigo\_test='" + codigo\_test + "'");

try {

if (rs.next()) {

test = new Test(rs);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return test;

}

public void añadirEnunciada(Test test, Pregunta pregunta, char indice, boolean correcto) {

if (mysqlc.isConectada()) {

String nombreTabla = "enuncian";

String campos = "fk\_test, fk\_pregunta, respuesta, correcto";

String value = test.getPk\_test() + "," + pregunta.getPk\_pregunta() + ", '" + indice + "', " + correcto;

mysqlc.insertInto(nombreTabla, campos, value);

}

}

public int getNumeroAciertos(Test test) {

int aciertos = 0;

String nombreTabla = "enuncian";

String where = "fk\_test=" + test.getPk\_test() + " AND correcto=1";

if (mysqlc.isConectada()) {

ResultSet rs = mysqlc.selectFrom(nombreTabla, where);

try {

while (rs.next()) {

aciertos++;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return aciertos;

}

public int getNumeroTest(Usuario usuario) {

int numeroTest = 0;

String nombreTabla = "test";

String where = "fk\_usuario=" + usuario.getPk\_usuario();

if (mysqlc.isConectada()) {

ResultSet rs = mysqlc.selectFrom(nombreTabla, where);

try {

while (rs.next()) {

numeroTest++;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return numeroTest;

}

public int getNumeroAprobados(Usuario usuario) {

int numeroAprobados = 0;

String nombreTabla = "test";

String where = "fk\_usuario=" + usuario.getPk\_usuario();

if (mysqlc.isConectada()) {

ResultSet rs = mysqlc.selectFrom(nombreTabla, where);

try {

while (rs.next()) {

if(getNumeroAciertos(new Test(rs)) > 25) {

numeroAprobados++;

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return numeroAprobados;

}

public float getMediaFallos(Usuario usuario) {

int sumaFallos = 0;

float mediaFallos = 0;

ArrayList<Integer> fallos = new ArrayList<Integer>();

String nombreTabla = "test";

String where = "fk\_usuario=" + usuario.getPk\_usuario();

if (mysqlc.isConectada()) {

ResultSet rs = mysqlc.selectFrom(nombreTabla, where);

try {

while (rs.next()) {

int nAciertos = getNumeroAciertos(new Test(rs));

int nFallos = 30 - nAciertos;

fallos.add(nFallos);

}

Iterator<Integer> it = fallos.iterator();

while(it.hasNext()) {

sumaFallos = sumaFallos +it.next();

}

try {

mediaFallos = sumaFallos/fallos.size();

} catch(ArithmeticException err) {

mediaFallos = 0;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return mediaFallos;

}

}

### GestorUsuario

package acceso\_a\_datos;

import java.sql.Date;

import java.sql.ResultSet;

import java.sql.SQLException;

import clases.Usuario;

public class GestorUsuarios {

private MysqlC mysqlc;

public GestorUsuarios(MysqlC mysqlc) {

super();

this.mysqlc = mysqlc;

}

public boolean registarUsuario(Usuario usuario) {

boolean control = false;

if(mysqlc.isConectada()) {

Encriptador enc = new Encriptador();

Date fecha = new Date(new java.util.Date().getTime());

String nombreTabla = "usuario";

String campos = "nick, pass, nombre, apellidos, dni, email, fecha\_registro";

String value = "'"+usuario.getNick()+"', '"+enc.encriptarMD5(usuario.getPass())+"', '"+usuario.getNombre()+"', '"+usuario.getApellidos()+"', '"+usuario.getDni()+"', '"+usuario.getEmail()+"', '"+fecha+"'";

if(mysqlc.insertInto(nombreTabla, campos, value)){

control = true;

}

}

return control;

}

public boolean actualizarEstadoUsuario(String estado, String where) {

boolean control = false;

if(mysqlc.isConectada()) {

if(mysqlc.Update("usuario", "estado='"+estado+"'", where)){

control = true;

}

}

return control;

}

public boolean actualizarUsuario(String campo, String valor ,String where) {

boolean control = false;

if(mysqlc.isConectada()) {

if(mysqlc.Update("usuario", campo+"='"+valor+"'", where)){

control = true;

}

}

return control;

}

public boolean buscarUsuarioExacto(String campo, String value) {

boolean control = false;

if(mysqlc.isConectada()) {

// Cumple formato

ResultSet rs = mysqlc.selectFrom("usuario"," "+campo+"='" + value + "'");

try {

if (rs.next()) {

control = true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return control;

}

public ResultSet getUsuarioFilter(String value) {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("usuario","(nick LIKE '%"+value+"%' OR dni LIKE '%"+value+"%' OR estado LIKE '%"+value+"%'OR acceso LIKE '%"+value+"%') AND pk\_usuario > 1");

}

return rs;

}

public ResultSet getUsuario(String campo, String value) {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("usuario"," "+campo+"='" + value + "'");

}

return rs;

}

public ResultSet getAllUsuarios() {

ResultSet rs = null;

if(mysqlc.isConectada()) {

// Cumple formato

rs = mysqlc.selectFrom("usuario", "pk\_usuario > 1");

}

return rs;

}

public String getPassword(String campo, String value) {

String pass = "";

if(mysqlc.isConectada()) {

ResultSet rs = mysqlc.selectFrom("usuario"," "+campo+"='" + value + "'");

try {

if(rs.next()) {

pass = rs.getString("pass");

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return pass;

}

}

### InfoMsg

**package** acceso\_a\_datos;

**import** java.awt.Color;

**public** **class** InfoMsg {

**private** String text;

**private** String ico\_path;

**private** **boolean** check;

**private** Color color;

**public** InfoMsg(String text, String ico\_path, **boolean** check, Color color) {

**super**();

**this**.text = text;

**this**.ico\_path = ico\_path;

**this**.check = check;

**this**.color = color;

}

**public** String getText() {

**return** text;

}

**public** **void** setText(String text) {

**this**.text = text;

}

**public** String getIco\_path() {

**return** ico\_path;

}

**public** **void** setIco\_path(String ico\_path) {

**this**.ico\_path = ico\_path;

}

**public** **boolean** isCheck() {

**return** check;

}

**public** **void** setCheck(**boolean** check) {

**this**.check = check;

}

**public** Color getColor() {

**return** color;

}

**public** **void** setColor(Color color) {

**this**.color = color;

}

}

### MysqlC

package acceso\_a\_datos;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class MysqlC {

private static Connection con;

private static final String DRIVER = "com.mysql.jdbc.Driver";

private static final String USUARIO = "root";

private static final String PASSWORD = "";

private static final String URL = "jdbc:mysql://localhost:3306/hacentest";

private static boolean conectada = false;

public boolean isConectada() {

return conectada;

}

public boolean conectar() {

try {

Class.forName(DRIVER);

con = DriverManager.getConnection(URL, USUARIO, PASSWORD);

if(con != null) {

System.out.println("MYSQL: Conexion establecida.");

conectada = true;

}

} catch(ClassNotFoundException | SQLException e) {

System.err.println("MYSQL: "+e+".");

}

return conectada;

}

public void desconectar(){

try {

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

};

try {

if(con.isClosed()) {

System.out.println("MYSQL: Fin de la conexion.");

conectada = false;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

//SQL STATEMENTS

public boolean insertInto(String nombreTabla, String campos, String value) {

boolean control = false;

String sql="INSERT INTO "+nombreTabla+" ("+campos+") VALUES ("+value+")";

try {

PreparedStatement stm = con.prepareStatement(sql);

int n = stm.executeUpdate();

if(n > 0 ) {

control = true;

System.out.println("MYSQL: Execute Insert( -> "+sql+" <-). Rows afected: "+n);

}

} catch (SQLException e) {

System.err.println("MYSQL: "+e);

}

return control;

}

public boolean Update(String nombreTabla, String update, String where) {

boolean control = false;

String sql="UPDATE "+nombreTabla+" SET "+update+" WHERE "+where;

try {

PreparedStatement stm = con.prepareStatement(sql);

int n = stm.executeUpdate();

if(n > 0 ) {

control = true;

System.out.println("MYSQL: Execute Update( -> "+sql+" <-). Rows afected: "+n);

}

} catch (SQLException e) {

System.err.println("MYSQL: "+e);

}

return control;

}

public ResultSet selectFrom(String nombreTabla, String where) {

ResultSet rs = null;

String sql="SELECT \* FROM "+nombreTabla+" WHERE "+where;

try {

PreparedStatement stm = con.prepareStatement(sql);

rs = stm.executeQuery(sql);

System.out.println("MYSQL: Execute Query( -> "+sql+" <-)");

} catch (SQLException e) {

System.err.println("MYSQL: "+e);

}

return rs;

}

public ResultSet selectFrom(String nombreTabla) {

ResultSet rs = null;

String sql="SELECT \* FROM "+nombreTabla;

try {

PreparedStatement stm = con.prepareStatement(sql);

rs = stm.executeQuery(sql);

System.out.println("MYSQL: Execute Query( -> "+sql+" <-)");

} catch (SQLException e) {

System.err.println("MYSQL: "+e);

}

return rs;

}

public void cerrarConexion() {

try {

if(con != null){

con.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

## Paquete clases

### Administrador

**package** clases;

**public** **class** Administrador **extends** Usuario {

}

### Alumno

**package** clases;

**public** **class** Alumno **extends** Usuario {

}

### Consulta

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

public class Consulta {

private int pk\_consulta;

private Timestamp fecha;

private Usuario usuario;

private String asunto, descripcion;

public Consulta(Usuario usuario, String asunto, String descripcion) {

super();

this.usuario = usuario;

this.asunto = asunto;

this.descripcion = descripcion;

}

public Consulta(String asunto, String descripcion) {

super();

this.asunto = asunto;

this.descripcion = descripcion;

}

public Consulta(ResultSet rs) {

try {

usuario = new Usuario(rs);

pk\_consulta = rs.getInt("pk\_consulta");

fecha = rs.getTimestamp("fecha");

asunto = rs.getString("asunto");

descripcion = rs.getString("descripcion");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public int getPk\_consulta() {

return pk\_consulta;

}

public void setPk\_consulta(int pk\_consulta) {

this.pk\_consulta = pk\_consulta;

}

public Timestamp getFecha() {

return fecha;

}

public void setFecha(Timestamp fecha) {

this.fecha = fecha;

}

public Usuario getUsuario() {

return usuario;

}

public void setUsuario(Usuario usuario) {

this.usuario = usuario;

}

public String getAsunto() {

return asunto;

}

public void setAsunto(String asunto) {

this.asunto = asunto;

}

public String getDescripcion() {

return descripcion;

}

public void setDescripcion(String descripcion) {

this.descripcion = descripcion;

}

}

### Desarrollador

**package** clases;

**public** **class** Desarrollador **extends** Usuario {

}

### Enunciada

**package** clases;

**public** **class** Enunciada {

Pregunta pregunta;

**char** respuesta;

**boolean** correcta;

**public** Enunciada(Pregunta pregunta, **char** respuesta, **boolean** correcta) {

**super**();

**this**.pregunta = pregunta;

**this**.respuesta = respuesta;

**this**.correcta = correcta;

}

**public** Pregunta getPregunta() {

**return** pregunta;

}

**public** **void** setPregunta(Pregunta pregunta) {

**this**.pregunta = pregunta;

}

**public** **char** getRespuesta() {

**return** respuesta;

}

**public** **void** setRespuesta(**char** respuesta) {

**this**.respuesta = respuesta;

}

**public** **boolean** isCorrecta() {

**return** correcta;

}

**public** **void** setCorrecta(**boolean** correcta) {

**this**.correcta = correcta;

}

}

### Pregunta

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

public class Pregunta {

int pk\_pregunta;

String enunciado, codigo\_pregunta;

char indiceCorrecto;

Respuesta[] respuestas;

public Pregunta(String enunciado, char indiceCorrecto, Respuesta[] respuestas) {

super();

this.enunciado = enunciado;

this.indiceCorrecto = indiceCorrecto;

this.respuestas = respuestas;

}

public Pregunta(ResultSet rs) {

// TODO Auto-generated constructor stub

try {

pk\_pregunta = rs.getInt("pk\_pregunta");

enunciado = rs.getString("enunciado");

codigo\_pregunta = rs.getString("codigo\_pregunta");

indiceCorrecto = rs.getString("indice\_correcto").charAt(0);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public int getPk\_pregunta() {

return pk\_pregunta;

}

public void setPk\_pregunta(int pk\_pregunta) {

this.pk\_pregunta = pk\_pregunta;

}

public String getEnunciado() {

return enunciado;

}

public void setEnunciado(String enunciado) {

this.enunciado = enunciado;

}

public char getIndiceCorrecto() {

return indiceCorrecto;

}

public void setIndiceCorrecto(char indiceCorrecto) {

this.indiceCorrecto = indiceCorrecto;

}

public Respuesta[] getRespuestas() {

return respuestas;

}

public void setRespuestas(Respuesta[] respuestas) {

this.respuestas = respuestas;

}

public String getCodigo\_pregunta() {

return codigo\_pregunta;

}

public void setCodigo\_pregunta(String codigo\_pregunta) {

this.codigo\_pregunta = codigo\_pregunta;

}

}

### Profesor

**package** clases;

**public** **class** Profesor **extends** Usuario {

}

### Respuesta

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

public class Respuesta {

int pk\_respuesta, fk\_pregunta;

String texto;

char indice;

public Respuesta(String texto, char indice) {

super();

this.texto = texto;

this.indice = indice;

}

public Respuesta(ResultSet rs) {

try {

pk\_respuesta = rs.getInt("pk\_respuesta");

fk\_pregunta = rs.getInt("fk\_pregunta");

texto= rs.getString("texto");

indice= rs.getString("indice").charAt(0);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public int getPk\_respuesta() {

return pk\_respuesta;

}

public void setPk\_respuesta(int pk\_respuesta) {

this.pk\_respuesta = pk\_respuesta;

}

public int getFk\_pregunta() {

return fk\_pregunta;

}

public void setFk\_pregunta(int fk\_pregunta) {

this.fk\_pregunta = fk\_pregunta;

}

public String getTexto() {

return texto;

}

public void setTexto(String texto) {

this.texto = texto;

}

public char getIndice() {

return indice;

}

public void setIndice(char indice) {

this.indice = indice;

}

}

### Sesion

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.Random;

public class Sesion {

private Usuario usuario;

private Timestamp tiempo\_inicio;

private String codigo;

private boolean sesionPrincipal;

public Sesion(ResultSet rs , String codigo) {

super();

usuario = new Usuario(rs);

this.codigo = codigo;

sesionPrincipal = true;

}

public void setUsuario(Usuario usuario) {

this.usuario = usuario;

}

public Usuario getUsuario() {

return usuario;

}

public Timestamp getTiempo\_inicio() {

return tiempo\_inicio;

}

public String getCodigo() {

return codigo;

}

public boolean isSesionPrincipal() {

return sesionPrincipal;

}

public void setSesionPrincipal(boolean sesionPrincipal) {

this.sesionPrincipal = sesionPrincipal;

}

}

### Test

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.ArrayList;

public class Test {

int pk\_test;

Usuario usuario;

Timestamp tiempo\_inicio, tiempo\_final;

String codigo\_test;

ArrayList<Enunciada> enunciadas;

public Test(Usuario usuario, Timestamp tiempo\_inicio, Timestamp tiempo\_final) {

super();

this.usuario = usuario;

this.tiempo\_inicio = tiempo\_inicio;

this.tiempo\_final = tiempo\_final;

}

public Test(ResultSet rs) {

enunciadas = new ArrayList<Enunciada>();

try {

pk\_test= rs.getInt("pk\_test");

tiempo\_inicio = rs.getTimestamp("tiempo\_inicio");

tiempo\_final = rs.getTimestamp("tiempo\_final");

codigo\_test = rs.getString("codigo\_test");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public int getPk\_test() {

return pk\_test;

}

public void setPk\_test(int pk\_test) {

this.pk\_test = pk\_test;

}

public Usuario getUsuario() {

return usuario;

}

public void setUsuario(Usuario usuario) {

this.usuario = usuario;

}

public Timestamp getTiempo\_inicio() {

return tiempo\_inicio;

}

public void setTiempo\_inicio(Timestamp tiempo\_inicio) {

this.tiempo\_inicio = tiempo\_inicio;

}

public Timestamp getTiempo\_final() {

return tiempo\_final;

}

public void setTiempo\_final(Timestamp tiempo\_final) {

this.tiempo\_final = tiempo\_final;

}

public ArrayList<Enunciada> getEnunciadas() {

return enunciadas;

}

public void setEnunciadas(ArrayList<Enunciada> enunciadas) {

this.enunciadas = enunciadas;

}

}

### Usuario

package clases;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Date;

public class Usuario {

private String nick, pass, nombre, apellidos, dni, email, acceso, estado;

private Date fecha\_registro;

private int pk\_usuario;

public Usuario() {

}

public Usuario(String nick, String pass, String nombre, String apellidos, String dni, String email,

Date fecha\_registro, Date fecha\_ultima\_sesion) {

super();

this.nick = nick;

this.pass = pass;

this.nombre = nombre;

this.apellidos = apellidos;

this.dni = dni;

this.email = email;

this.fecha\_registro = fecha\_registro;

}

public Usuario(ResultSet rs) {

try {

this.nick =(rs.getString("nick"));

this.nombre = (rs.getString("nombre"));

this.apellidos =(rs.getString("apellidos"));

this.dni = (rs.getString("dni"));

this.email =(rs.getString("email"));

this.acceso =(rs.getString("acceso"));

this.fecha\_registro = (rs.getDate("fecha\_Registro"));

this.pk\_usuario = (rs.getInt("pk\_usuario"));

this.estado =(rs.getString("estado"));

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public String getNick() {

return nick;

}

public void setNick(String nick) {

this.nick = nick;

}

public String getPass() {

return pass;

}

public void setPass(String pass) {

this.pass = pass;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getApellidos() {

return apellidos;

}

public void setApellidos(String apellidos) {

this.apellidos = apellidos;

}

public String getDni() {

return dni;

}

public void setDni(String dni) {

this.dni = dni;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public Date getFecha\_registro() {

return fecha\_registro;

}

public void setFecha\_registro(Date fecha\_registro) {

this.fecha\_registro = fecha\_registro;

}

public String getAcceso() {

return acceso;

}

public void setAcceso(String acceso) {

this.acceso = acceso;

}

public int getPk\_usuario() {

return pk\_usuario;

}

public String getEstado() {

return estado;

}

public void setEstado(String estado) {

this.estado = estado;

}

}

## Paquete controladores

### ChangeBtnListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.ImageIcon;

import interfaz.Ventana;

import interfaz.componentes.JChanger;

public class ChangerBtnListener implements MouseListener {

JChanger changer;

Ventana ventana;

public ChangerBtnListener(JChanger changer, Ventana ventana) {

super();

this.changer = changer;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

ventana.getGestorUsuarios().actualizarUsuario(changer.getTipo(), changer.getEstado(), "pk\_usuario="+ventana.getUsuarioInspeccionado().getPk\_usuario());

switch(changer.getTipo()) {

case "acceso":

ventana.getUsuarioInspeccionado().setAcceso(changer.getEstado());

break;

case "estado":

ventana.getUsuarioInspeccionado().setEstado(changer.getEstado());

break;

}

ventana.actualizarMostraUsuario(ventana.getUsuarioInspeccionado());

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

changer.setIcon(new ImageIcon(Ventana.class.getResource(changer.getIco\_hover())));

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

changer.setIcon(new ImageIcon(Ventana.class.getResource(changer.getIco\_path())));

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### ControlLabelListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import interfaz.Ventana;

import interfaz.componentes.JControlLabel;

public class ControlLabelListener implements MouseListener {

private JControlLabel lbl;

private Ventana ventana;

public ControlLabelListener(JControlLabel lbl, Ventana ventana) {

this.lbl = lbl;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

switch(lbl.getFunction()) {

case 0:

if(ventana.getSesionActual() != null) {

ventana.getGestorSesiones().cerrarSesion(ventana.getSesionActual());

}

try {

ventana.cerrarConexion();

}catch(NullPointerException err) {

}

System.exit(0);

break;

case 1:

ventana.getFocusFixer().requestFocus();

ventana.getFrame().setState(JFrame.ICONIFIED);

break;

case 2:

ventana.hidePopUp();

break;

}

}

@Override

public void mouseEntered(MouseEvent e) {

lbl.setIcon(new ImageIcon(Ventana.class.getResource(lbl.getPathIcoHover())));

}

@Override

public void mouseExited(MouseEvent e) {

lbl.setIcon(new ImageIcon(Ventana.class.getResource(lbl.getPathIco())));

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### EntrarComoListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.JLabel;

import interfaz.Ventana;

public class EntrarComoListener implements MouseListener {

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

private JLabel lbl;

public EntrarComoListener(Ventana ventana, JLabel lbl) {

super();

this.ventana = ventana;

this.lbl = lbl;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

ventana.cambiarVistaSesion(lbl.getText());

ventana.getSesionActual().setSesionPrincipal(false);

ventana.showPopUp("entrarcomo"+lbl.getText());

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_HOVER);

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_BACKGROUND);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### EstadisticasBtnListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import clases.Usuario;

import interfaz.Ventana;

public class EstadisticasBtnListener implements MouseListener {

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

public EstadisticasBtnListener(Ventana ventana) {

super();

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

Usuario usuario = ventana.getUsuarioInspeccionado();

int numeroTest = ventana.getGestorTest().getNumeroTest(usuario);

int numeroAprobados = ventana.getGestorTest().getNumeroAprobados(usuario);

float mediaFallos = ventana.getGestorTest().getMediaFallos(usuario);

ventana.actualizarMostrarEstadisticas(numeroTest, numeroAprobados, mediaFallos);

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_HOVER);

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_BACKGROUND);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### FormBtnListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.sql.ResultSet;

import java.sql.SQLException;

import clases.Pregunta;

import interfaz.Ventana;

import interfaz.componentes.JFormBtn;

public class FormBtnListener implements MouseListener {

JFormBtn formBtn;

private Ventana ventana;

public FormBtnListener(JFormBtn formBtn, Ventana ventana) {

super();

this.formBtn = formBtn;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

if (formBtn.isGroupAllTrue()) {

switch (formBtn.getFunction()) {

case "reg":

if (ventana.getGestorUsuarios().registarUsuario(ventana.getRegData())) {

ventana.resetRegForm();

ventana.setRegDisplay(2, "back");

ventana.showPopUp("regcomplete");

ventana.goTo("display\_login");

} else {

ventana.showPopUp("mysqlerror");

}

break;

case "login":

ResultSet rs = ventana.getLoginResultSet();

try {

if (rs.next()) {

String estado = rs.getString("estado");

if (estado.equals("offline")) {

int pk\_usuario = rs.getInt("pk\_usuario");

String where = "pk\_usuario=" + pk\_usuario;

ventana.getGestorUsuarios().actualizarEstadoUsuario("online", where);

ventana.iniciarSesion(ventana.getGestorSesiones().abrirSesion(rs));

switch (ventana.getSesionActual().getUsuario().getAcceso()) {

case "alumno":

//sesion alumno

ventana.cambiarVistaSesion("alumno");

break;

case "profesor":

// sesion profesor

ventana.cambiarVistaSesion("profesor");

break;

case "administrador":

// sesion administrador

ventana.cambiarVistaSesion("administrador");

break;

case "desarrollador":

// sesion desarrollador

ventana.cambiarVistaSesion("desarrollador");

break;

}

int numeroTest = ventana.getGestorTest().getNumeroTest(ventana.getSesionActual().getUsuario());

int numeroAprobados = ventana.getGestorTest().getNumeroAprobados(ventana.getSesionActual().getUsuario());

float mediaFallos = ventana.getGestorTest().getMediaFallos(ventana.getSesionActual().getUsuario());

ventana.actualizarMostrarEstadisticas(numeroTest, numeroAprobados, mediaFallos);

ventana.actualizarListaUsuarios(ventana.getGestorUsuarios().getAllUsuarios());

ventana.actualizarListaConsultas(ventana.getGestorConsultas().getAllConsultas());

ventana.getSesionActual().getUsuario().setEstado("online");

ventana.actualizarMostraUsuario(ventana.getSesionActual().getUsuario());

ventana.showPopUp("logincomplete");

} else {

switch (estado) {

case "online":

ventana.showPopUp("useronline");

break;

case "suspendido":

ventana.showPopUp("usersuspend");

break;

}

}

}

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

break;

case "pregunta":

Pregunta pregunta = ventana.getPreguntaData();

if(ventana.getGestorTest().añadirPregunta(pregunta)) {

ventana.showPopUp("addpregunta");

ventana.reniciarFormularioPregunta();

}

break;

}

}

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

if (formBtn.isGroupAllTrue()) {

formBtn.setBackground(formBtn.getCOLOR\_HOVER());

formBtn.setToolTipText("Todo correcto! pulsa aqui para registrarte.");

} else {

formBtn.setBackground(formBtn.getCOLOR\_DISABLED());

formBtn.setToolTipText("Algunos de los campos requeridos no estan correctamente complentados.");

}

}

@Override

public void mouseExited(MouseEvent e) {

formBtn.setBackground(formBtn.getCOLOR\_CHECK());

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### FrameDrager

package controladores;

import java.awt.Point;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.JFrame;

public class FrameDrager extends MouseAdapter {

private final JFrame frame;

private Point mouseDownCompCoords = null;

public FrameDrager(JFrame frame) {

this.frame = frame;

}

public void mouseReleased(MouseEvent e) {

//mouseDownCompCoords = null;

}

public void mousePressed(MouseEvent e) {

mouseDownCompCoords = e.getPoint();

}

public void mouseDragged(MouseEvent e) {

Point currCoords = e.getLocationOnScreen();

frame.setLocation(currCoords.x - mouseDownCompCoords.x, currCoords.y - mouseDownCompCoords.y);

}

}

### JRowBtnListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.sql.ResultSet;

import java.sql.SQLException;

import clases.Consulta;

import clases.Usuario;

import interfaz.Ventana;

import interfaz.componentes.JRowBtn;

public class JRowBtnListener implements MouseListener {

private JRowBtn btn;

private Ventana ventana;

public JRowBtnListener(JRowBtn btn, Ventana ventana) {

super();

this.btn = btn;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent arg0) {

// TODO Auto-generated method stub

ventana.getFocusFixer().requestFocus();

switch(btn.getTipo()) {

case "usuario":

ResultSet rsu = ventana.getGestorUsuarios().getUsuario("pk\_usuario", ""+btn.getPk());

try {

if(rsu.next()) {

Usuario usuario= new Usuario(rsu);

ventana.setUsuarioInspeccionado(usuario);

ventana.actualizarMostraUsuario(usuario);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

break;

case "consulta":

ResultSet rsc = ventana.getGestorConsultas().getConsulta("pk\_consulta ="+btn.getPk());

try {

if(rsc.next()) {

Consulta consulta= new Consulta(rsc);

ventana.actualizarMostrarConsulta(consulta);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

break;

}

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

btn.setBackground(btn.getCOLOR\_HOVER());

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

btn.setBackground(btn.getCOLOR\_CHECK());

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

}

}

### NavMostrarListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import interfaz.Ventana;

import interfaz.componentes.JNavLabel;

public class NavMostrarListener implements MouseListener {

private JNavLabel navLabel;

private Ventana ventana;

@Override

public void mouseClicked(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseEntered(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseExited(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mousePressed(MouseEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent arg0) {

// TODO Auto-generated method stub

}

}

### NavRegListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.ImageIcon;

import interfaz.Ventana;

import interfaz.componentes.JNavLabel;

public class NavRegListener implements MouseListener {

private JNavLabel navLabel;

private Ventana ventana;

public NavRegListener(JNavLabel navLabel, Ventana ventana) {

super();

this.navLabel = navLabel;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

ventana.getFocusFixer().requestFocus();

ventana.setRegDisplay(navLabel.getActualIndex(), navLabel.getFunction());

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

navLabel.setIcon(new ImageIcon(Ventana.class.getResource(navLabel.getIco\_hover\_path())));

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

navLabel.setIcon(new ImageIcon(Ventana.class.getResource(navLabel.getIco\_path())));

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### OptionListener

package controladores;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.util.Iterator;

import interfaz.componentes.JOption;

public class OptionListener implements MouseListener {

private JOption opt;

public OptionListener(JOption opt ){

super();

this.opt = opt;

}

@Override

public void mouseClicked(MouseEvent e) {

Iterator<JOption> it = opt.getGrupo().getGrupo().iterator();

while(it.hasNext()) {

JOption o = it.next();

if(!o.equals(opt)) {

o.setSelected(false);

}

}

opt.setSelected(true);

if(opt.getDisplay() == null) {

System.out.println("falla");

}

opt.getDisplay().getGrupo().getVentana().getFocusFixer().requestFocus();

opt.getDisplay().getGrupo().mostarDisplay(opt.getDisplay());

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

opt.setBackground(opt.getCOLOR\_HOVER());

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

if(opt.isSelected()) {

opt.setBackground(opt.getCOLOR\_SELECTED());

}else {

opt.setBackground(opt.getCOLOR\_BACKGROUND());

}

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### RadioButtonListener

package controladores;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JRadioButton;

public class RadioButtonListener implements ActionListener {

JRadioButton rbp, rb1, rb2;

public RadioButtonListener(JRadioButton rbp, JRadioButton rb1, JRadioButton rb2) {

super();

this.rbp = rbp;

this.rb1 = rb1;

this.rb2 = rb2;

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if(rbp.isSelected()) {

rb1.setSelected(false);

rb2.setSelected(false);

} else {

rb1.setSelected(true);

}

}

}

### RespuestaBtnListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import clases.Enunciada;

import interfaz.Ventana;

import interfaz.componentes.JRespuestaBtn;

public class RespuestaBtnListener implements MouseListener {

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

private JRespuestaBtn btn;

public RespuestaBtnListener(Ventana ventana, JRespuestaBtn btn) {

super();

this.ventana = ventana;

this.btn = btn;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

boolean correcto = false;

if(ventana.getPreguntaActual().getIndiceCorrecto() == btn.getIndice()) {

correcto = true;

}

ventana.getGestorTest().añadirEnunciada(ventana.getTestActual(), ventana.getPreguntaActual(), btn.getIndice(), correcto);

ventana.getTestActual().getEnunciadas().add(new Enunciada(ventana.getPreguntaActual(), btn.getIndice(), correcto));

if(ventana.getTestActual().getEnunciadas().size() < 30) {

ventana.actualizarPregunta(ventana.getGestorTest().getPreguntaDisponible(ventana.getTestActual()));

} else {

ventana.completarTest(ventana.getGestorTest().getNumeroAciertos(ventana.getTestActual()));

}

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

btn.setBackground(COLOR\_HOVER);

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

btn.setBackground(COLOR\_BACKGROUND);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### SesionBtnListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import interfaz.Ventana;

public class SesionBtnListener implements MouseListener {

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

public SesionBtnListener(Ventana ventana) {

super();

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

if(ventana.getSesionActual().isSesionPrincipal()) {

ventana.getGestorSesiones().cerrarSesion(ventana.getSesionActual());

ventana.cambiarVistaSesion("invitado");

} else {

ventana.cambiarVistaSesion(ventana.getSesionActual().getUsuario().getAcceso());

ventana.getSesionActual().setSesionPrincipal(true);

}

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_HOVER);

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_BACKGROUND);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### SoporteBtnListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import clases.Consulta;

import interfaz.Ventana;

import interfaz.grupos.TextFieldGroup;

public class SoporteBtnListener implements MouseListener {

JTextArea textArea;

TextFieldGroup textFieldGroup;

Ventana ventana;

private final Color COLOR\_CHECK = new Color(50, 205, 50);

private final Color COLOR\_DISABLED = new Color(128, 128, 128);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

public SoporteBtnListener(JTextArea textArea, TextFieldGroup textFieldGroup, Ventana ventana) {

super();

this.textArea = textArea;

this.textFieldGroup = textFieldGroup;

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

boolean control = false;

if(ventana.getSesionActual() == null) {

control = ventana.getGestorConsultas().generarConsultaInvitado(new Consulta(textFieldGroup.getTextF().getText(), textArea.getText()));

} else {

control = ventana.getGestorConsultas().generarConsulta(new Consulta(ventana.getSesionActual().getUsuario(), textFieldGroup.getTextF().getText(), textArea.getText()));

}

if(control) {

ventana.reniciarFormularioSoporte();

ventana.showPopUp("nuevaconsulta");

ventana.getFocusFixer().requestFocus();

}

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

if(!textFieldGroup.isCheck() || textArea.getText().equals("") || textArea.getText().length() > 300) {

e.getComponent().setEnabled(false);

e.getComponent().setBackground(COLOR\_DISABLED);

} else {

e.getComponent().setEnabled(true);

e.getComponent().setBackground(COLOR\_HOVER);

}

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_CHECK);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### TestBtnListener

package controladores;

import java.awt.Color;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import interfaz.Ventana;

public class TestBtnListener implements MouseListener {

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

public TestBtnListener(Ventana ventana) {

super();

this.ventana = ventana;

}

@Override

public void mouseClicked(MouseEvent e) {

// TODO Auto-generated method stub

if(ventana.getTestActual() == null) {

ventana.setTestActual(ventana.getGestorTest().empezarTest(ventana.getSesionActual().getUsuario()));

ventana.actualizarPregunta(ventana.getGestorTest().getPreguntaDisponible(ventana.getTestActual()));

} else {

ventana.ReiniciarTest();

}

}

@Override

public void mouseEntered(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_HOVER);

}

@Override

public void mouseExited(MouseEvent e) {

// TODO Auto-generated method stub

e.getComponent().setBackground(COLOR\_BACKGROUND);

}

@Override

public void mousePressed(MouseEvent e) {

// TODO Auto-generated method stub

}

@Override

public void mouseReleased(MouseEvent e) {

// TODO Auto-generated method stub

}

}

### TextAreaListener

package controladores;

import java.awt.Color;

import java.awt.event.FocusEvent;

import java.awt.event.FocusListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import javax.swing.JLabel;

import javax.swing.JTextArea;

public class TextAreaListener implements FocusListener, KeyListener {

JTextArea textArea;

JLabel textLbl;

private final Color COLOR\_DANGER = new Color(222, 69, 69);

public TextAreaListener(JTextArea textArea, JLabel textLbl) {

super();

this.textArea = textArea;

this.textLbl = textLbl;

}

@Override

public void keyPressed(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyReleased(KeyEvent e) {

// TODO Auto-generated method stub

if(textArea.getText().equals("")) {

textLbl.setText("Maximo 300 caracteres");

} else {

textLbl.setText(textArea.getText().length()+"/300 caracteres.");

if(textArea.getText().length() > 300) {

textLbl.setForeground(COLOR\_DANGER);

}else {

textLbl.setForeground(Color.WHITE);

}

}

}

@Override

public void keyTyped(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void focusGained(FocusEvent e) {

// TODO Auto-generated method stub

if(textArea.getText().equals("Describre detalladamente la incidencia.")) {

textArea.setText("");

}

}

@Override

public void focusLost(FocusEvent e) {

// TODO Auto-generated method stub

if(textArea.getText().equals("")) {

textArea.setText("Describre detalladamente la incidencia.");

}

}

}

### TextFieldKeyListener

package controladores;

import java.awt.Color;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import acceso\_a\_datos.Checker;

import acceso\_a\_datos.InfoMsg;

import interfaz.Ventana;

import interfaz.grupos.TextFieldGroup;

import interfaz.grupos.TextFieldGroupRel;

public class TextFieldKeyListener implements KeyListener {

TextFieldGroup tFGroup;

Checker checker;

Ventana ventana;

private final Color COLOR\_DANGER = new Color(222, 69, 69);

public TextFieldKeyListener(TextFieldGroup tFGroup, Checker checker) {

super();

this.tFGroup = tFGroup;

this.checker = checker;

}

public TextFieldKeyListener(TextFieldGroup tFGroup, Checker checker, Ventana ventana) {

super();

this.tFGroup = tFGroup;

this.checker = checker;

this.ventana = ventana;

}

@Override

public void keyPressed(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyReleased(KeyEvent e){

InfoMsg info;

if(tFGroup.isConnected()) {

switch(tFGroup.getTipo()) {

case "strict":

info = checker.checkStringStrict(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "pass":

info = checker.checkPassword(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "flex":

info = checker.checkStringFlex(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "strict2":

info = checker.checkStringStrict(tFGroup.getTextF().getText(), 2);

tFGroup.showInfo(info);

break;

case "dni":

info = checker.checkDNI(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "email":

info = checker.checkEmail(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "flex\_key":

info = checker.checkStringFlexKey(tFGroup.getTextF().getText());

tFGroup.showInfo(info);

break;

case "rel":

if (tFGroup instanceof TextFieldGroupRel) {

TextFieldGroupRel tFGroupRel = (TextFieldGroupRel)tFGroup;

TextFieldGroup rel = tFGroupRel.getRel();

switch(tFGroupRel.getRelType()) {

case "similitud":

info = checker.checkSimilitud(tFGroupRel.getTextF().getText(), rel.getTextF().getText(),rel.isCheck());

tFGroupRel.showInfo(info);

break;

case "correspondencia":

info = checker.checkCorrespondencia(tFGroupRel.getTextF().getText(), rel.getTextF().getText(),rel.isCheck());

tFGroupRel.showInfo(info);

break;

}

}

break;

case "buscar":

ventana.actualizarListaUsuarios(checker.buscar(tFGroup.getTextF().getText()));

break;

case "consulta":

ventana.actualizarListaConsultas(checker.consultas(tFGroup.getTextF().getText()));

break;

case "soporte":

info = checker.checkFreeString(tFGroup.getTextF().getText(), 60);

tFGroup.showInfo(info);

break;

case "enunciado":

info = checker.checkFreeString(tFGroup.getTextF().getText(), 300);

tFGroup.showInfo(info);

break;

case "respuesta":

info = checker.checkFreeString(tFGroup.getTextF().getText(), 120);

tFGroup.showInfo(info);

break;

}

} else {

info = new InfoMsg("Imposible conectar a la base de datos", "/imagenes/error\_black\_32px.png", false, COLOR\_DANGER);

tFGroup.showInfo(info);

}

}

@Override

public void keyTyped(KeyEvent e) {

}

}

### TextFocusListener

package controladores;

import java.awt.Color;

import java.awt.event.FocusEvent;

import java.awt.event.FocusListener;

import acceso\_a\_datos.Checker;

import acceso\_a\_datos.InfoMsg;

import acceso\_a\_datos.MysqlC;

import interfaz.grupos.TextFieldGroup;

public class TextFocusListener implements FocusListener {

TextFieldGroup tFGroup;

Checker checker;

MysqlC mysqlc = new MysqlC();

private final Color COLOR\_DANGER = new Color(222, 69, 69);

public TextFocusListener(TextFieldGroup tFGroup, Checker checker) {

super();

this.tFGroup = tFGroup;

this.checker = checker;

}

@Override

public void focusGained(FocusEvent arg0) {

if (tFGroup.getTextF().getText().equals(tFGroup.getDefaultText())) {

tFGroup.getTextF().setText("");

}

if(mysqlc.isConectada()) {

tFGroup.setConnected(true);

} else {

tFGroup.setConnected(false);

tFGroup.showInfo(new InfoMsg("Imposible conectar a la base de datos", "/imagenes/error\_black\_32px.png", false, COLOR\_DANGER));

}

}

@Override

public void focusLost(FocusEvent arg0) {

if (tFGroup.getTextF().getText().equals("")) {

tFGroup.reset();

}

}

}

## Paquete interfaz

### Ventana

package interfaz;

import java.awt.Color;

import java.awt.Component;

import java.awt.Container;

import javax.swing.\*;

import acceso\_a\_datos.Checker;

import acceso\_a\_datos.GestorConsultas;

import acceso\_a\_datos.GestorSesiones;

import acceso\_a\_datos.GestorTest;

import acceso\_a\_datos.GestorUsuarios;

import acceso\_a\_datos.MysqlC;

import clases.Consulta;

import clases.Pregunta;

import clases.Respuesta;

import clases.Sesion;

import clases.Test;

import clases.Usuario;

import controladores.FormBtnListener;

import controladores.FrameDrager;

import controladores.NavRegListener;

import controladores.OptionListener;

import controladores.RadioButtonListener;

import controladores.RespuestaBtnListener;

import controladores.SesionBtnListener;

import controladores.SoporteBtnListener;

import controladores.TestBtnListener;

import controladores.TextAreaListener;

import controladores.TextFieldKeyListener;

import controladores.TextFocusListener;

import interfaz.componentes.JDisplay;

import interfaz.componentes.JFormBtn;

import interfaz.componentes.JFormDisplay;

import interfaz.componentes.JChanger;

import interfaz.componentes.JControlLabel;

import interfaz.componentes.JOption;

import interfaz.componentes.JPopUp;

import interfaz.componentes.JRespuestaBtn;

import interfaz.componentes.JNavLabel;

import interfaz.grupos.DisplayGroup;

import interfaz.grupos.OptionGroup;

import interfaz.grupos.SesionSteup;

import interfaz.grupos.TextFieldGroup;

import interfaz.grupos.TextFieldGroupRel;

import interfaz.listas.JConsultaRowList;

import interfaz.listas.JUserRowList;

import controladores.ChangerBtnListener;

import controladores.ControlLabelListener;

import controladores.EntrarComoListener;

import controladores.EstadisticasBtnListener;

import java.awt.Font;

import java.awt.GridLayout;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.text.DecimalFormat;

import java.util.ArrayList;

import java.util.Iterator;

import java.awt.Toolkit;

public class Ventana {

// ACCESO A DATOS INICIO

private MysqlC mysqlc;

private GestorUsuarios gestorUsuarios;

private GestorSesiones gestorSesiones;

private GestorConsultas gestorConsultas;

private GestorTest gestorTest;

private Checker checker;

// DATOS DE SESION

private Sesion sesionActual;

private Test testActual;

private Pregunta preguntaActual;

private Usuario usuarioInspeccionado;

// DATA

private Usuario RegData = new Usuario();

// GRUPOS

private ArrayList<JOption> listaOpciones\_invitado = new ArrayList<JOption>();

private ArrayList<JOption> listaOpciones\_alumno = new ArrayList<JOption>();

private ArrayList<JOption> listaOpciones\_profesor = new ArrayList<JOption>();

private ArrayList<JOption> listaOpciones\_administrador = new ArrayList<JOption>();

private ArrayList<JOption> listaOpciones\_desarrollador = new ArrayList<JOption>();

private ArrayList<JDisplay> listaDisplays\_invitado = new ArrayList<JDisplay>();

private ArrayList<JDisplay> listaDisplays\_alumno = new ArrayList<JDisplay>();

private ArrayList<JDisplay> listaDisplays\_profesor = new ArrayList<JDisplay>();

private ArrayList<JDisplay> listaDisplays\_administrador = new ArrayList<JDisplay>();

private ArrayList<JDisplay> listaDisplays\_desarrollador = new ArrayList<JDisplay>();

private ArrayList<JFormDisplay> grupo\_reg = new ArrayList<JFormDisplay>();

// COLORES

private final Color COLOR\_ERROR = new Color(219, 186, 70);

private final Color COLOR\_SELECTED = new Color(85, 65, 118);

private final Color COLOR\_BACKGROUND = new Color(54, 33, 89);

private final Color COLOR\_CHECK = new Color(50, 205, 50);

private final Color COLOR\_DANGER = new Color(222, 69, 69);

// SESIONS\_STEUP

private SesionSteup invitado, alumno, profesor, administrador, desarrollador;

private OptionGroup opciones\_invitado, opciones\_alumno, opciones\_profesor, opciones\_administrador,

opciones\_desarrollador;

private DisplayGroup displays;

// COMPONENTES INICIO

// Estructura

private JFrame frmHacentest;

private JPanel background\_panel, side\_panel, center\_panel;

private JPopUp popUp\_panel;

private JButton focusFixer;

// 1-Side panel

private JPanel logo\_panel, nav\_panel;

// 1.1-Logo panel

private JLabel logo\_imagen, logo\_texto;

// 1.2-Nav Panel

private JPanel navPanel\_invitado, navPanel\_alumno, navPanel\_profesor, navPanel\_administrador,

navPanel\_desarrollador;

// 1.2.1-Index nav

// Invitado

private JOption opt\_login, opt\_reg, opt\_info, opt\_soporte;

private JLabel opt\_login\_ico, opt\_login\_text, opt\_reg\_ico, opt\_reg\_text, opt\_info\_ico, opt\_info\_text,

opt\_soporte\_ico, opt\_soporte\_text;

// Alumno

private JOption opt\_iniciarTest, opt\_estadisticas\_alumno, opt\_cerrarSesion\_alumno, opt\_soporte\_alumno;

private JLabel opt\_iniciarTest\_ico, opt\_iniciarTest\_text, opt\_estadisticas\_alumno\_ico, opt\_estadisticas\_alumno\_text,

opt\_soporte\_alumno\_ico, opt\_soporte\_alumno\_text, opt\_cerrarSesion\_alumno\_ico, opt\_cerrarSesion\_alumno\_text;

// Profesor

private JOption opt\_estadisticas\_profesor, opt\_entrarComo\_profesor, opt\_cerrarSesion\_profesor, opt\_soporte\_profesor;

private JLabel opt\_estadisticas\_profesor\_ico, opt\_entrarComo\_profesor\_ico, opt\_entrarComo\_profesor\_text,

opt\_soporte\_profesor\_ico, opt\_soporte\_profesor\_text, opt\_cerrarSesion\_profesor\_ico,

opt\_cerrarSesion\_profesor\_text;

// Admin

private JOption opt\_verUsuarios, opt\_soporte\_administrador, opt\_entrarComo\_administrador,

opt\_cerrarSesion\_administrador;

private JLabel opt\_verUsuarios\_ico, opt\_verUsuarios\_text, opt\_soporte\_administrador\_ico,

opt\_soporte\_administrador\_text, opt\_entrarComo\_administrador\_ico, opt\_entrarComo\_administrador\_text,

opt\_cerrarSesion\_administrador\_ico, opt\_cerrarSesion\_administrador\_text;

// Desarrollador

private JOption opt\_verUsuarios\_desarrollador, opt\_entrarComo\_desarrollador, opt\_anadirPregunta,

opt\_cerrarSesion\_desarrollador;

private JLabel opt\_verUsuarios\_desarrollador\_ico, opt\_verUsuarios\_desarrollador\_text,

opt\_entrarComo\_desarrollador\_ico, opt\_entrarComo\_desarrollador\_text, opt\_anadirPregunta\_ico,

opt\_anadirPregunta\_text, opt\_cerrarSesion\_desarrollador\_ico, opt\_cerrarSesion\_desarrollador\_text;

// 2-Center panel

private JPanel titulo\_panel, display\_panel;

// 2.1-Titulo panel

private JControlLabel close\_lbl, minimize\_lbl;

private JLabel titulo\_ico, titulo\_text;

private JSeparator separador\_titulo;

// 2.2-Displays

private JDisplay display\_index, display\_login, display\_reg, display\_info, display\_soporte, display\_verUsuarios,

display\_mostrarUsuario;

// 2.2.1-Index display

private JLabel mensaje\_index;

// 2.2.2-Login display

private JPanel login\_password\_panel;

private JLabel imagen\_central\_login;

private JFormBtn login\_iniciar\_FormBtn;

private JLabel login\_iniciar\_formBtn\_text;

// 2.2.2.1-Login\_nick\_panel

private JPanel login\_nick\_info\_panel;

private JLabel login\_nick\_info\_ico, login\_nick\_info\_text;

// 2.2.2.2-Login\_pass\_panel

private JPanel login\_password\_info\_panel;

private JLabel login\_password\_icon, login\_password\_info\_ico, login\_password\_info\_text;

private JPasswordField login\_password\_textF;

private JSeparator login\_password\_separator;

// 2.2.3-ReG display

private JPanel reg\_stepbar\_panel;

private JFormDisplay reg\_form\_step\_1\_panel, reg\_form\_step\_2\_panel, reg\_form\_step\_3\_panel;

private JNavLabel reg\_next\_navLabel, reg\_back\_navLabel;

private JLabel reg\_form\_ico, reg\_form\_titulo;

private JSeparator reg\_form\_separator;

// 2.2.3.1-Stepbar

private JSeparator reg\_stepbar\_1\_to\_2\_separator, reg\_stepbar\_2\_to\_3\_separator;

private JLabel reg\_stepbar\_1\_ico, reg\_stepbar\_2\_ico, reg\_stepbar\_3\_ico, reg\_stepbar\_1\_arrow\_ico,

reg\_stepbar\_2\_arrow\_ico, reg\_stepbar\_3\_arrow\_ico;

// 2.2.3.2-Form 1

private JPanel reg\_nick\_panel, reg\_password\_panel, reg\_password\_check\_panel;

// 2.2.3.2.1-Nick panel

private JSeparator reg\_nick\_separator;

private JLabel reg\_nick\_ico, reg\_nick\_info\_ico, reg\_nick\_info\_text;

private JTextField reg\_nick\_textF;

private JPanel reg\_nick\_info\_panel;

// 2.2.3.2.2-Password panel

private JSeparator reg\_password\_separator;

private JLabel reg\_password\_ico, reg\_password\_info\_ico, reg\_password\_info\_text;

private JPasswordField reg\_password\_textF;

private JPanel reg\_password\_info\_panel;

// 2.2.3.2.3-Password check panel

private JPanel reg\_password\_check\_info\_panel;

private JSeparator reg\_password\_check\_separator;

private JLabel reg\_password\_check\_ico, reg\_password\_check\_info\_ico, reg\_password\_check\_info\_text;

private JPasswordField reg\_password\_check\_textF;

// 2.2.3.3-Form 2

private JPanel reg\_name\_panel, reg\_apellido\_panel, reg\_dni\_panel;

// 2.2.3.3.1-Name panel

private JPanel reg\_name\_info\_panel;

private JSeparator reg\_name\_separator;

private JLabel reg\_name\_ico, reg\_name\_info\_ico, reg\_name\_info\_text;

private JTextField reg\_name\_textF;

// 2.2.3.3.2-Apellido panel

private JPanel reg\_apellido\_info\_panel;

private JSeparator reg\_apellido\_separator;

private JLabel reg\_apellido\_ico, reg\_apellido\_info\_ico, reg\_apellido\_info\_text;

private JTextField reg\_apellido\_textF;

// 2.2.3.3.3-DNI panel

private JPanel reg\_dni\_info\_panel;

private JSeparator reg\_dni\_separator;

private JLabel reg\_dni\_ico, reg\_dni\_info\_ico, reg\_dni\_info\_text;

private JTextField reg\_dni\_textF;

// 2.2.3.3-Form 3

private JPanel reg\_email\_panel;

// 2.2.3.3.1-Email panel

private JPanel reg\_email\_info\_panel;

private JSeparator reg\_email\_separator;

private JLabel reg\_email\_ico, reg\_email\_info\_ico, reg\_email\_info\_text;

private JTextField reg\_email\_textF;

// 2.2.3.3.2-Confirmar Button

private JFormBtn reg\_confirmar\_FormBtn;

private JLabel reg\_confirmar\_formBtn\_text;

// 2.2.4-Info display

// 2.2.5-Soporte display

// 2.2.6- Mostrar Usuario

// 2.2.7-Ver usuarios

private JPanel verUsuarios\_contenedor, verUsuarios\_nick\_info\_panel, verUsuarios\_nick\_panel;

private JScrollPane verUsuarios\_scroll;

private JSeparator verUsuarios\_nick\_separator;

private JTextField verUsuarios\_nick\_textF;

private JLabel verUsuarios\_nick\_info\_ico, verUsuarios\_nick\_info\_text, verUsuarios\_nick\_icon;

private JUserRowList listUpdater\_verUsuarios;

// 3-PopUp Panel

private JLabel popUp\_ico, popUp\_text;

private JControlLabel popUp\_salir\_labelControl;

// [ ]COMPONENTES LOGICOS

// Formulario Login

private ArrayList<TextFieldGroup> login\_grupo\_logico = new ArrayList<TextFieldGroup>();

private TextFieldGroup login\_nick\_tfg;

private TextFieldGroup login\_password\_tfg;

// Formulario registro

private ArrayList<TextFieldGroup> reg\_grupo\_logico = new ArrayList<TextFieldGroup>();

private TextFieldGroup reg\_nick\_tfg, reg\_password\_tfg, reg\_name\_tfg, reg\_apellido\_tfg, reg\_dni\_tfg, reg\_email\_tfg;

private TextFieldGroup reg\_password\_check\_tfg;

// Formulario soporte

private TextFieldGroup soporte\_asunto\_tfg;

// Buscador verUsuarios

private TextFieldGroup verUsuarios\_nick\_tfg;

private JLabel acceso\_text\_mostrarUsuario;

private JPanel login\_nick\_panel;

private JSeparator login\_nick\_separator;

private JLabel login\_nick\_icon;

private JTextField login\_nick\_textF;

private JPanel nick\_panel\_mostrarUsuario;

private JSeparator nick\_separator\_mostrarUsuario;

private JLabel nick\_ico\_mostrarUsuario;

private JLabel nick\_text\_mostrarUsuario;

private JPanel nombre\_panel\_mostrarUsuario;

private JSeparator nombre\_separator\_mostrarUsuario;

private JLabel nombre\_ico\_mostrarUsuario;

private JLabel nombre\_text\_mostrarUsuario;

private JPanel apellidos\_panel\_mostrarUsuario;

private JSeparator apellidos\_separator\_mostrarUsuario;

private JLabel apellidos\_ico\_mostrarUsuario;

private JLabel apellidos\_text\_mostrarUsuario;

private JPanel dni\_panel\_mostrarUsuario;

private JSeparator dni\_separator\_mostrarUsuario;

private JLabel dni\_ico\_mostrarUsuario;

private JLabel dni\_text\_mostrarUsuario;

private JPanel email\_panel\_mostrarUsuario;

private JSeparator email\_separator\_mostrarUsuario;

private JLabel email\_ico\_mostrarUsuario;

private JLabel email\_text\_mostrarUsuario;

private JPanel fechar\_panel\_mostrarUsuario;

private JSeparator fechar\_separator\_mostrarUsuario;

private JLabel fechar\_ico\_mostrarUsuario;

private JLabel fechar\_text\_mostrarUsuario;

private JLabel tiempoDeUso\_ico\_mostrarUsuario;

private JLabel tiempoDeUso\_text\_mostrarUsuario;

private JLabel acceso\_ico\_mostrarUsuario;

private JLabel estado\_text\_mostrarUsuario;

private JLabel estado\_ico\_mostrarUsuario;

private JPanel accesoCambio\_panel\_mostrarUsuario;

private JPanel estadoCambio\_panel\_mostrarUsuario;

// ---------------------------------------------------------------

private JDisplay display\_cerrarSesion;

private JPanel cerrarSesion\_btn\_panel;

private JLabel cerrarSesion\_btn\_text;

// ---------------------------------------------------------------

private JDisplay display\_entrarComo;

private JPanel entrarComo\_alumno\_btn\_panel;

private JLabel entrarComo\_alumno\_btn\_text;

private JLabel entrarComo\_alumno\_btn\_ico;

private JPanel entrarComo\_profesor\_btn\_panel;

private JLabel entrarComo\_profesor\_btn\_text;

private JLabel entrarComo\_profesor\_btn\_ico;

private JPanel entrarComo\_administrador\_btn\_panel;

private JLabel entrarComo\_administrador\_btn\_text;

private JLabel entrarComo\_administrador\_btn\_ico;

// --------------------------------------------------------------

private JLabel mensaje\_info;

private JLabel java\_ico;

private JLabel eclipse\_ico;

private JLabel mysql\_ico;

private JPanel soporte\_asunto\_panel;

// --------------------------------------------------------------

private JSeparator soporte\_asunto\_separator;

private JLabel soporte\_asunto\_icon;

private JTextField soporte\_asunto\_textF;

private JPanel soporte\_asunto\_info\_panel;

private JLabel soporte\_asunto\_info\_ico;

private JLabel soporte\_asunto\_info\_text;

private JLabel soporte\_caracteres\_text;

private JFormBtn soporte\_enviar\_FormBtn;

private JLabel soporte\_enviar\_formBtn\_text;

private JTextArea soporte\_textArea;

private JDisplay display\_verConsultas;

// ---------------------------------------------------------------

private JPanel verConsultas\_filtro\_panel;

private JSeparator verConsultas\_filtro\_separator;

private JLabel verConsultas\_filtro\_icon;

private JTextField verConsultas\_filtro\_textF;

private JPanel verConsultas\_filtro\_info\_panel;

private JLabel verConsultas\_filtro\_info\_ico;

private JLabel verConsultas\_filtro\_info\_text;

private JScrollPane verConsultas\_scroll;

private JPanel verConsultas\_\_contenedor;

private JConsultaRowList listUpdater\_verConsultas;

private TextFieldGroup verConsultas\_filto\_tfg;

// ----------------------------------------------------------------

private JDisplay display\_mostrarConsulta;

private JPanel asunto\_panel\_mostrarConsulta;

private JSeparator asunto\_separator\_mostrarConsulta;

private JLabel asunto\_ico\_mostrarConsulta;

private JLabel asunto\_text\_mostrarConsulta;

private JPanel nick\_panel\_mostrarConsulta;

private JSeparator nick\_separator\_mostrarConsulta;

private JLabel nick\_ico\_mostrarConsulta;

private JLabel nick\_text\_mostrarConsulta;

private JPanel fecha\_panel\_mostrarConsulta;

private JSeparator fecha\_separator\_mostrarConsulta;

private JLabel fecha\_ico\_mostrarConsulta;

private JLabel fecha\_text\_mostrarConsulta;

private JLabel descripcion\_mostrarConsulta;

// ---------------------------------------------------------------

private JDisplay display\_anadirPregunta;

private JPanel anadirPregunta\_enunciado\_panel;

private JSeparator anadirPregunta\_enunciado\_separator;

private JLabel anadirPregunta\_enunciado\_icon;

private JTextField anadirPregunta\_enunciado\_textF;

private JPanel anadirPregunta\_enunciado\_info\_panel;

private JLabel anadirPregunta\_enunciado\_info\_ico;

private JLabel anadirPregunta\_enunciado\_info\_text;

private JPanel anadirPregunta\_respuestaB\_panel;

private JSeparator anadirPregunta\_respuestaB\_separator;

private JTextField anadirPregunta\_respuestaB\_textF;

private JPanel anadirPregunta\_respuestaB\_info\_panel;

private JLabel anadirPregunta\_respuestaB\_info\_ico;

private JLabel anadirPregunta\_respuestaB\_info\_text;

private JLabel anadirPregunta\_respuestaB\_icon;

private JPanel anadirPregunta\_respuestaC\_panel;

private JSeparator anadirPregunta\_respuestaC\_separator;

private JLabel anadirPregunta\_respuestaC\_icon;

private JTextField anadirPregunta\_respuestaC\_textF;

private JPanel anadirPregunta\_respuestaC\_info\_panel;

private JLabel anadirPregunta\_respuestaC\_info\_ico;

private JLabel anadirPregunta\_respuestaC\_info\_text;

private JPanel anadirPregunta\_respuestaA\_panel;

private JSeparator anadirPregunta\_respuestaA\_separator;

private JLabel anadirPregunta\_respuestaA\_icon;

private JTextField anadirPregunta\_respuestaA\_textF;

private JPanel anadirPregunta\_respuestaA\_info\_panel;

private JLabel anadirPregunta\_respuestaA\_info\_ico;

private JLabel anadirPregunta\_respuestaA\_info\_text;

private JFormBtn anadirPregunta\_FormBtn;

private JLabel anadirPregunta\_FormBtn\_text;

private JRadioButton anadirPregunta\_respuestaA\_radioBtn;

private JRadioButton anadirPregunta\_respuestaB\_radioBtn;

private JRadioButton anadirPregunta\_respuestaC\_radioBtn;

private TextFieldGroup anadirPregunta\_enunciado\_tfg;

private TextFieldGroup anadirPregunta\_respuestaA\_tfg;

private TextFieldGroup anadirPregunta\_respuestaB\_tfg;

private TextFieldGroup anadirPregunta\_respuestaC\_tfg;

private ArrayList<TextFieldGroup> anadirPregunta\_grupo\_logico = new ArrayList<TextFieldGroup>();

// ----------------------------------------------

private JDisplay display\_iniciarTest;

private JLabel iniciarTest\_respuestaA\_ico;

private JLabel iniciarTest\_respuestaA\_text;

private JRespuestaBtn iniciarTest\_respuestaB\_panel;

private JLabel iniciarTest\_respuestaB\_ico;

private JLabel iniciarTest\_respuestaB\_text;

private JRespuestaBtn iniciarTest\_respuestaC\_panel;

private JLabel iniciarTest\_respuestaC\_ico;

private JLabel iniciarTest\_respuestaC\_text;

private JPanel iniciarTest\_pregunta\_panel;

private JLabel iniciarTest\_pregunta\_enunciado;

private JSeparator iniciarTest\_pregunta\_separator;

private JLabel iniciarTest\_pregunta\_contadorPregunta;

private JRespuestaBtn iniciarTest\_respuestaA\_panel;

private JPanel iniciarTest\_index\_panel;

private JPanel iniciarTest\_index\_btn\_panel;

private JLabel iniciarTest\_index\_btn\_text;

private JPanel iniciarTest\_end\_panel;

private JPanel iniciarTest\_end\_btn\_panel;

private JLabel iniciarTest\_end\_btn\_text;

private JLabel iniciarTest\_fallos\_numero;

private JLabel iniciarTest\_fallos\_text;

private JLabel iniciarTest\_resultado\_text;

private JLabel iniciarTest\_aciertos\_numero;

private JLabel iniciarTest\_aciertos\_text;

// ------------------

private JPanel mostrarUsuario\_estadisticas\_btn;

private JLabel mostrarUsuario\_estadisticas\_btn\_text;

// ---------------------------------------------------------

private JDisplay display\_mostrarEstadisticas;

private JLabel mostrarEstadisticas\_aprobados\_numero;

private JLabel mostrarEstadisticas\_aprobados\_text;

private JLabel mostrarEstadisticas\_suspendos\_numero;

private JLabel mostrarEstadisticas\_suspendos\_text;

private JLabel mostrarEstadisticas\_nick\_text;

private JSeparator mostrarEstadisticas\_separator2;

private JSeparator mostrarEstadisticas\_separator1;

private JLabel mostrarEstadisticas\_test\_text;

private JLabel mostrarEstadisticas\_test\_numero;

private JLabel mostrarEstadisticas\_media\_text;

private JLabel mostrarEstadisticas\_media\_numero;

private JChanger estadoCambio\_online\_mostrarUsuario;

private JChanger estadoCambio\_offline\_mostrarUsuario;

private JChanger estadoCambio\_suspendido\_mostrarUsuario;

private JChanger accesoCambio\_alumno\_mostrarUsuario;

private JChanger accesoCambio\_profesor\_mostrarUsuario;

private JChanger accesoCambio\_admin\_mostrarUsuario;

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* INICIALIZACION COMPONENTES

\*/

public Ventana() {

mysqlc = new MysqlC();

gestorUsuarios = new GestorUsuarios(mysqlc);

gestorSesiones = new GestorSesiones(mysqlc, gestorUsuarios);

gestorConsultas = new GestorConsultas(mysqlc);

gestorTest = new GestorTest(mysqlc);

checker = new Checker(gestorUsuarios, gestorConsultas);

mysqlc.conectar();

initialize();

}

/\*\*

\* @wbp.parser.entryPoint

\*/

// Inicializacion de componentes

private void initialize() {

// Frame

frmHacentest = new JFrame();

frmHacentest.setIconImage(

Toolkit.getDefaultToolkit().getImage(Ventana.class.getResource("/imagenes/hacentestico.png")));

frmHacentest.setTitle("Hacentest");

frmHacentest.setBounds(100, 100, 1100, 700);

frmHacentest.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frmHacentest.setUndecorated(true);

frmHacentest.setVisible(true);

frmHacentest.getContentPane().setLayout(null);

frmHacentest.setResizable(false);

FrameDrager drager = new FrameDrager(frmHacentest);

frmHacentest.addMouseListener(drager);

frmHacentest.addMouseMotionListener(drager);

cargarPopUpPanel();

cargarEstructura();

cargarSesionSteups();

cargarDisplays();

cargarNavPanelInvitado();

cargarNavPanelAlumno();

cargarNavPanelProfesor();

cargarNavPanelAdministrador();

cargarNavPanelDesarrollador();

// Comprobar driver

if (mysqlc.isConectada()) {

} else {

this.showPopUp("mysqlerror");

}

}

// Estructura

public void setTestActual(Test testActual) {

this.testActual = testActual;

}

public Test getTestActual() {

return testActual;

}

public Usuario getUsuarioInspeccionado() {

return usuarioInspeccionado;

}

public void setUsuarioInspeccionado(Usuario usuarioInspeccionado) {

this.usuarioInspeccionado = usuarioInspeccionado;

}

private void cargarEstructura() {

// Background Panel

background\_panel = new JPanel();

background\_panel.setBounds(0, 0, 1100, 700);

frmHacentest.getContentPane().add(background\_panel);

background\_panel.setLayout(null);

// Estructura panels

side\_panel = new JPanel();

side\_panel.setBounds(0, 0, 300, 700);

background\_panel.add(side\_panel);

side\_panel.setLayout(null);

side\_panel.setBackground(COLOR\_BACKGROUND);

center\_panel = new JPanel();

center\_panel.setBounds(300, 0, 800, 700);

background\_panel.add(center\_panel);

center\_panel.setBackground(COLOR\_SELECTED);

center\_panel.setLayout(null);

focusFixer = new JButton("");

focusFixer.setBounds(42, 0, 1, 1);

frmHacentest.getContentPane().add(focusFixer);

cargarCenterPanel();

cargarSidePanel();

}

private void cargarSidePanel() {

// Logo panel

logo\_panel = new JPanel();

logo\_panel.setBounds(0, 0, 300, 137);

side\_panel.add(logo\_panel);

logo\_panel.setOpaque(false);

logo\_panel.setLayout(null);

logo\_imagen = new JLabel("");

logo\_imagen.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/logo\_96px.png")));

logo\_imagen.setBounds(92, 11, 96, 84);

logo\_panel.add(logo\_imagen);

logo\_texto = new JLabel("HACENTEST");

logo\_texto.setFont(new Font("Segoe UI Light", Font.BOLD, 40));

logo\_texto.setForeground(Color.WHITE);

logo\_texto.setBounds(35, 84, 223, 53);

logo\_panel.add(logo\_texto);

// Nav panel

nav\_panel = new JPanel();

nav\_panel.setBounds(0, 136, 300, 464);

side\_panel.add(nav\_panel);

nav\_panel.setLayout(null);

nav\_panel.setOpaque(false);

}

private void cargarCenterPanel() {

// Titulo panel

titulo\_panel = new JPanel();

titulo\_panel.setBounds(0, 0, 800, 148);

center\_panel.add(titulo\_panel);

titulo\_panel.setLayout(null);

titulo\_panel.setOpaque(false);

titulo\_ico = new JLabel("");

titulo\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/index\_96px.png")));

titulo\_ico.setBounds(25, 23, 101, 102);

titulo\_panel.add(titulo\_ico);

titulo\_text = new JLabel("Bienvenid@");

titulo\_text.setForeground(Color.WHITE);

titulo\_text.setFont(new Font("Tahoma", Font.PLAIN, 60));

titulo\_text.setBounds(136, 23, 529, 102);

titulo\_panel.add(titulo\_text);

separador\_titulo = new JSeparator();

separador\_titulo.setBounds(10, 136, 780, 12);

titulo\_panel.add(separador\_titulo);

// Control labels

close\_lbl = new JControlLabel(0, "/imagenes/salir\_32px.png", "/imagenes/salir\_hover\_32px.png");

close\_lbl.setIcon(new ImageIcon(Ventana.class.getResource(close\_lbl.getPathIco())));

close\_lbl.setBounds(762, 0, 32, 40);

titulo\_panel.add(close\_lbl);

close\_lbl.addMouseListener(new ControlLabelListener(close\_lbl, this));

minimize\_lbl = new JControlLabel(1, "/imagenes/minimizar\_32px.png", "/imagenes/minimizar\_hover\_32px.png");

minimize\_lbl.setIcon(new ImageIcon(Ventana.class.getResource(minimize\_lbl.getPathIco())));

minimize\_lbl.setBounds(730, 0, 32, 40);

titulo\_panel.add(minimize\_lbl);

minimize\_lbl.addMouseListener(new ControlLabelListener(minimize\_lbl, this));

// Display panel

display\_panel = new JPanel();

display\_panel.setBounds(0, 148, 800, 552);

center\_panel.add(display\_panel);

display\_panel.setLayout(null);

display\_panel.setOpaque(false);

}

private void cargarPopUpPanel() {

// PopUp Panel

popUp\_panel = new JPopUp();

popUp\_panel.setBounds(5, 540, 290, 150);

frmHacentest.getContentPane().add(popUp\_panel);

popUp\_panel.setBackground(Color.RED);

popUp\_panel.setLayout(null);

popUp\_ico = new JLabel("");

popUp\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/info\_96px.png")));

popUp\_ico.setBounds(8, 31, 96, 96);

popUp\_panel.add(popUp\_ico);

popUp\_text = new JLabel("");

popUp\_text.setVerticalAlignment(SwingConstants.TOP);

popUp\_text.setHorizontalAlignment(SwingConstants.LEFT);

popUp\_text.setFont(new Font("Tahoma", Font.PLAIN, 16));

popUp\_text.setForeground(Color.WHITE);

popUp\_text.setBounds(105, 32, 173, 96);

popUp\_panel.add(popUp\_text);

popUp\_panel.setVisible(false);

popUp\_salir\_labelControl = new JControlLabel(2, "/imagenes/salir\_black\_16px.png", "/imagenes/salir\_16px.png");

popUp\_salir\_labelControl.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/salir\_black\_16px.png")));

popUp\_salir\_labelControl.setBounds(256, 13, 16, 16);

popUp\_panel.add(popUp\_salir\_labelControl);

popUp\_salir\_labelControl.addMouseListener(new ControlLabelListener(popUp\_salir\_labelControl, this));

popUp\_panel.setIco(popUp\_ico);

popUp\_panel.setText(popUp\_text);

popUp\_panel.setControl(popUp\_salir\_labelControl);

}

// Inicializacion de Grupos

private void cargarSesionSteups() {

// Carga los grupos de componentes utilizados durante la aplicacion

opciones\_invitado = new OptionGroup();

opciones\_alumno = new OptionGroup();

opciones\_profesor = new OptionGroup();

opciones\_administrador = new OptionGroup();

opciones\_desarrollador = new OptionGroup();

displays = new DisplayGroup(titulo\_ico, titulo\_text, this);

invitado = new SesionSteup(opciones\_invitado, displays);

alumno = new SesionSteup(opciones\_alumno, displays);

profesor = new SesionSteup(opciones\_profesor, displays);

administrador = new SesionSteup(opciones\_administrador, displays);

desarrollador = new SesionSteup(opciones\_desarrollador, displays);

}

// DISPLAYS

private void cargarDisplays() {

cargarDisplayIndex();

cargarDisplayLogin();

cargarDisplayReg();

cargarDisplaySoporte();

cargarDisplayInfo();

cargarDisplayVerUsuarios();

cargarDisplayMostrarUsuario();

cargarDisplayCerrarSesion();

cargarDisplayEntrarComo();

cargarDisplayVerConsultas();

cargarDisplayMostrarConsulta();

cargarDisplayAñadirPregunta();

cargarDisplayIniciarTest();

cargarDisplayMostrarEstadisticas();

}

private void cargarDisplayIndex() {

// --> Display index

display\_index = new JDisplay("Bienvenid@", "/imagenes/index\_96px.png", displays);

display\_index.setBounds(0, 0, 800, 552);

display\_panel.add(display\_index);

display\_index.setLayout(null);

display\_index.setOpaque(false);

mensaje\_index = new JLabel("<html><p>Gracias por instalar Hacentest<br><br>"

+ "Para comenzar a utilizar la aplicación debes registrarte<br><br>"

+ "Si ya dispones de una cuenta puedes iniciar sesión directamente<br><br>"

+ "Para conocer más acerca de la aplicación visita la sección de información<br><br>"

+ "Si necesitas asistencia consulta la sección de soporte<p><html>");

mensaje\_index.setHorizontalAlignment(SwingConstants.CENTER);

mensaje\_index.setForeground(Color.WHITE);

mensaje\_index.setFont(new Font("Tahoma", Font.PLAIN, 22));

mensaje\_index.setBounds(26, 46, 753, 276);

display\_index.add(mensaje\_index);

}

private void cargarDisplayLogin() {

// --> Display Login

display\_login = new JDisplay("Iniciar sesion", "/imagenes/login\_96px.png", displays);

display\_login.setBounds(0, 0, 800, 552);

display\_panel.add(display\_login);

display\_login.setLayout(null);

display\_login.setOpaque(false);

display\_login.setVisible(false);

imagen\_central\_login = new JLabel("");

imagen\_central\_login.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_96px.png")));

imagen\_central\_login.setBounds(352, 27, 96, 96);

display\_login.add(imagen\_central\_login);

login\_nick\_panel = new JPanel();

login\_nick\_panel.setBounds(204, 144, 390, 96);

display\_login.add(login\_nick\_panel);

login\_nick\_panel.setOpaque(false);

login\_nick\_panel.setLayout(null);

login\_nick\_separator = new JSeparator();

login\_nick\_separator.setBounds(52, 41, 328, 2);

login\_nick\_panel.add(login\_nick\_separator);

login\_nick\_icon = new JLabel("");

login\_nick\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_text\_32px.png")));

login\_nick\_icon.setBounds(10, 11, 32, 32);

login\_nick\_panel.add(login\_nick\_icon);

login\_nick\_textF = new JTextField("Usuario");

login\_nick\_textF.setHorizontalAlignment(SwingConstants.LEFT);

login\_nick\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

login\_nick\_textF.setForeground(Color.WHITE);

login\_nick\_textF.setBounds(52, 11, 328, 25);

login\_nick\_panel.add(login\_nick\_textF);

login\_nick\_textF.setColumns(10);

login\_nick\_textF.setBackground(COLOR\_SELECTED);

login\_nick\_textF.setBorder(null);

login\_nick\_info\_panel = new JPanel();

login\_nick\_info\_panel.setBounds(10, 50, 370, 32);

login\_nick\_panel.add(login\_nick\_info\_panel);

login\_nick\_info\_panel.setBackground(COLOR\_ERROR);

login\_nick\_info\_panel.setLayout(null);

login\_nick\_info\_panel.setVisible(false);

login\_nick\_info\_ico = new JLabel("");

login\_nick\_info\_ico.setBounds(0, 0, 32, 32);

login\_nick\_info\_panel.add(login\_nick\_info\_ico);

login\_nick\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

login\_nick\_info\_text = new JLabel("Error");

login\_nick\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

login\_nick\_info\_text.setBounds(42, 0, 328, 32);

login\_nick\_info\_panel.add(login\_nick\_info\_text);

login\_password\_panel = new JPanel();

login\_password\_panel.setLayout(null);

login\_password\_panel.setOpaque(false);

login\_password\_panel.setBounds(204, 251, 390, 96);

display\_login.add(login\_password\_panel);

login\_password\_separator = new JSeparator();

login\_password\_separator.setBounds(52, 41, 328, 2);

login\_password\_panel.add(login\_password\_separator);

login\_password\_icon = new JLabel("");

login\_password\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/pass\_32px.png")));

login\_password\_icon.setBounds(10, 11, 32, 32);

login\_password\_panel.add(login\_password\_icon);

login\_password\_textF = new JPasswordField("Contrase\u00F1a");

login\_password\_textF.setHorizontalAlignment(SwingConstants.LEFT);

login\_password\_textF.setForeground(Color.WHITE);

login\_password\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

login\_password\_textF.setColumns(10);

login\_password\_textF.setBorder(null);

login\_password\_textF.setBackground(new Color(85, 65, 118));

login\_password\_textF.setBounds(52, 11, 328, 25);

login\_password\_panel.add(login\_password\_textF);

login\_password\_info\_panel = new JPanel();

login\_password\_info\_panel.setLayout(null);

login\_password\_info\_panel.setBackground(new Color(219, 186, 70));

login\_password\_info\_panel.setBounds(10, 50, 370, 32);

login\_password\_panel.add(login\_password\_info\_panel);

login\_password\_info\_panel.setVisible(false);

login\_password\_info\_ico = new JLabel("");

login\_password\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

login\_password\_info\_ico.setBounds(0, 0, 32, 32);

login\_password\_info\_panel.add(login\_password\_info\_ico);

login\_password\_info\_text = new JLabel("Error");

login\_password\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

login\_password\_info\_text.setBounds(42, 0, 328, 32);

login\_password\_info\_panel.add(login\_password\_info\_text);

login\_iniciar\_FormBtn = new JFormBtn("login", login\_grupo\_logico);

login\_iniciar\_FormBtn.setBounds(275, 450, 258, 62);

display\_login.add(login\_iniciar\_FormBtn);

login\_iniciar\_FormBtn.setLayout(null);

login\_iniciar\_FormBtn.setBackground(new Color(50, 205, 50));

login\_iniciar\_FormBtn.addMouseListener(new FormBtnListener(login\_iniciar\_FormBtn, this));

login\_iniciar\_formBtn\_text = new JLabel("Iniciar sesi\u00F3n");

login\_iniciar\_formBtn\_text.setHorizontalAlignment(SwingConstants.CENTER);

login\_iniciar\_formBtn\_text.setForeground(Color.WHITE);

login\_iniciar\_formBtn\_text.setFont(new Font("Tahoma", Font.BOLD, 25));

login\_iniciar\_formBtn\_text.setBounds(10, 0, 238, 62);

login\_iniciar\_FormBtn.add(login\_iniciar\_formBtn\_text);

// Componentes logicos formulario Login

login\_nick\_tfg = new TextFieldGroup(login\_nick\_panel, login\_nick\_info\_panel, login\_nick\_icon,

login\_nick\_info\_ico, login\_nick\_info\_text, login\_nick\_textF, login\_nick\_separator, "flex\_key",

login\_grupo\_logico);

login\_password\_tfg = new TextFieldGroupRel(login\_password\_panel, login\_password\_info\_panel, login\_password\_icon,

login\_password\_info\_ico, login\_password\_info\_text, login\_password\_textF, login\_password\_separator,

login\_grupo\_logico, login\_nick\_tfg, "correspondencia");

// Listener login

login\_nick\_textF.addKeyListener(new TextFieldKeyListener(login\_nick\_tfg, checker));

login\_password\_textF.addKeyListener(new TextFieldKeyListener(login\_password\_tfg, checker));

login\_nick\_textF.addKeyListener(new TextFieldKeyListener(login\_password\_tfg, checker));

login\_nick\_textF.addFocusListener(new TextFocusListener(login\_nick\_tfg, checker));

login\_password\_textF.addFocusListener(new TextFocusListener(login\_password\_tfg, checker));

}

private void cargarDisplayReg() {

// --> Display Reg

display\_reg = new JDisplay("Crear una cuenta", "/imagenes/reg\_96px.png", displays);

display\_reg.setBounds(0, 0, 800, 552);

display\_panel.add(display\_reg);

display\_reg.setLayout(null);

display\_reg.setOpaque(false);

display\_reg.setVisible(false);

// Reg Stepbar

reg\_stepbar\_panel = new JPanel();

reg\_stepbar\_panel.setBounds(10, 11, 780, 64);

display\_reg.add(reg\_stepbar\_panel);

reg\_stepbar\_panel.setOpaque(false);

reg\_stepbar\_panel.setLayout(null);

reg\_stepbar\_1\_ico = new JLabel("");

reg\_stepbar\_1\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_1\_32px.png")));

reg\_stepbar\_1\_ico.setBounds(10, 24, 32, 32);

reg\_stepbar\_panel.add(reg\_stepbar\_1\_ico);

reg\_stepbar\_2\_ico = new JLabel("");

reg\_stepbar\_2\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_2\_32px.png")));

reg\_stepbar\_2\_ico.setBounds(375, 24, 32, 32);

reg\_stepbar\_panel.add(reg\_stepbar\_2\_ico);

reg\_stepbar\_3\_ico = new JLabel("");

reg\_stepbar\_3\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_3\_32px.png")));

reg\_stepbar\_3\_ico.setBounds(738, 24, 32, 32);

reg\_stepbar\_panel.add(reg\_stepbar\_3\_ico);

reg\_stepbar\_1\_to\_2\_separator = new JSeparator();

reg\_stepbar\_1\_to\_2\_separator.setBounds(37, 39, 342, 2);

reg\_stepbar\_panel.add(reg\_stepbar\_1\_to\_2\_separator);

reg\_stepbar\_2\_to\_3\_separator = new JSeparator();

reg\_stepbar\_2\_to\_3\_separator.setBounds(401, 39, 342, 2);

reg\_stepbar\_panel.add(reg\_stepbar\_2\_to\_3\_separator);

reg\_stepbar\_1\_arrow\_ico = new JLabel("");

reg\_stepbar\_1\_arrow\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/arrow\_down\_32px.png")));

reg\_stepbar\_1\_arrow\_ico.setBounds(10, 0, 32, 28);

reg\_stepbar\_panel.add(reg\_stepbar\_1\_arrow\_ico);

reg\_stepbar\_2\_arrow\_ico = new JLabel("");

reg\_stepbar\_2\_arrow\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/arrow\_down\_32px.png")));

reg\_stepbar\_2\_arrow\_ico.setBounds(375, 0, 32, 28);

reg\_stepbar\_panel.add(reg\_stepbar\_2\_arrow\_ico);

reg\_stepbar\_2\_arrow\_ico.setVisible(false);

reg\_stepbar\_3\_arrow\_ico = new JLabel("");

reg\_stepbar\_3\_arrow\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/arrow\_down\_32px.png")));

reg\_stepbar\_3\_arrow\_ico.setBounds(738, 0, 32, 28);

reg\_stepbar\_panel.add(reg\_stepbar\_3\_arrow\_ico);

reg\_stepbar\_3\_arrow\_ico.setVisible(false);

// Reg form 1

reg\_form\_step\_1\_panel = new JFormDisplay("Datos de la cuenta", "/imagenes/user\_info\_64px.png", grupo\_reg, 1,

false, true);

reg\_form\_step\_1\_panel.setOpaque(false);

reg\_form\_step\_1\_panel.setBounds(173, 161, 449, 363);

display\_reg.add(reg\_form\_step\_1\_panel);

reg\_form\_step\_1\_panel.setLayout(null);

reg\_nick\_panel = new JPanel();

reg\_nick\_panel.setLayout(null);

reg\_nick\_panel.setOpaque(false);

reg\_nick\_panel.setBounds(31, 11, 390, 96);

reg\_form\_step\_1\_panel.add(reg\_nick\_panel);

reg\_nick\_separator = new JSeparator();

reg\_nick\_separator.setBounds(52, 41, 328, 2);

reg\_nick\_panel.add(reg\_nick\_separator);

reg\_nick\_ico = new JLabel("");

reg\_nick\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_text\_32px.png")));

reg\_nick\_ico.setBounds(10, 11, 32, 32);

reg\_nick\_panel.add(reg\_nick\_ico);

reg\_nick\_textF = new JTextField("Nick");

reg\_nick\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_nick\_textF.setForeground(Color.WHITE);

reg\_nick\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_nick\_textF.setColumns(10);

reg\_nick\_textF.setBorder(null);

reg\_nick\_textF.setBackground(new Color(85, 65, 118));

reg\_nick\_textF.setBounds(52, 11, 328, 25);

reg\_nick\_panel.add(reg\_nick\_textF);

reg\_nick\_info\_panel = new JPanel();

reg\_nick\_info\_panel.setLayout(null);

reg\_nick\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_nick\_info\_panel.setBounds(10, 50, 370, 32);

reg\_nick\_panel.add(reg\_nick\_info\_panel);

reg\_nick\_info\_panel.setVisible(false);

reg\_nick\_info\_ico = new JLabel("");

reg\_nick\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_nick\_info\_ico.setBounds(0, 0, 32, 32);

reg\_nick\_info\_panel.add(reg\_nick\_info\_ico);

reg\_nick\_info\_text = new JLabel("Error");

reg\_nick\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_nick\_info\_text.setBounds(42, 0, 328, 32);

reg\_nick\_info\_panel.add(reg\_nick\_info\_text);

reg\_password\_panel = new JPanel();

reg\_password\_panel.setLayout(null);

reg\_password\_panel.setOpaque(false);

reg\_password\_panel.setBounds(31, 118, 390, 96);

reg\_form\_step\_1\_panel.add(reg\_password\_panel);

reg\_password\_separator = new JSeparator();

reg\_password\_separator.setBounds(52, 41, 328, 2);

reg\_password\_panel.add(reg\_password\_separator);

reg\_password\_ico = new JLabel("");

reg\_password\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/pass\_32px.png")));

reg\_password\_ico.setBounds(10, 11, 32, 32);

reg\_password\_panel.add(reg\_password\_ico);

reg\_password\_textF = new JPasswordField("Contrase\u00F1a");

reg\_password\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_password\_textF.setForeground(Color.WHITE);

reg\_password\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_password\_textF.setColumns(10);

reg\_password\_textF.setBorder(null);

reg\_password\_textF.setBackground(new Color(85, 65, 118));

reg\_password\_textF.setBounds(52, 11, 328, 25);

reg\_password\_panel.add(reg\_password\_textF);

reg\_password\_info\_panel = new JPanel();

reg\_password\_info\_panel.setLayout(null);

reg\_password\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_password\_info\_panel.setBounds(10, 50, 370, 32);

reg\_password\_panel.add(reg\_password\_info\_panel);

reg\_password\_info\_panel.setVisible(false);

reg\_password\_info\_ico = new JLabel("");

reg\_password\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_password\_info\_ico.setBounds(0, 0, 32, 32);

reg\_password\_info\_panel.add(reg\_password\_info\_ico);

reg\_password\_info\_text = new JLabel("Error");

reg\_password\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_password\_info\_text.setBounds(42, 0, 328, 32);

reg\_password\_info\_panel.add(reg\_password\_info\_text);

reg\_password\_check\_panel = new JPanel();

reg\_password\_check\_panel.setLayout(null);

reg\_password\_check\_panel.setOpaque(false);

reg\_password\_check\_panel.setBounds(31, 225, 390, 96);

reg\_form\_step\_1\_panel.add(reg\_password\_check\_panel);

reg\_password\_check\_separator = new JSeparator();

reg\_password\_check\_separator.setBounds(52, 41, 328, 2);

reg\_password\_check\_panel.add(reg\_password\_check\_separator);

reg\_password\_check\_ico = new JLabel("");

reg\_password\_check\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/pass\_check\_32px.png")));

reg\_password\_check\_ico.setBounds(10, 11, 32, 32);

reg\_password\_check\_panel.add(reg\_password\_check\_ico);

reg\_password\_check\_textF = new JPasswordField("Repite tu contrase\u00F1a");

reg\_password\_check\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_password\_check\_textF.setForeground(Color.WHITE);

reg\_password\_check\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_password\_check\_textF.setColumns(10);

reg\_password\_check\_textF.setBorder(null);

reg\_password\_check\_textF.setBackground(new Color(85, 65, 118));

reg\_password\_check\_textF.setBounds(52, 11, 328, 25);

reg\_password\_check\_panel.add(reg\_password\_check\_textF);

reg\_password\_check\_info\_panel = new JPanel();

reg\_password\_check\_info\_panel.setLayout(null);

reg\_password\_check\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_password\_check\_info\_panel.setBounds(10, 50, 370, 32);

reg\_password\_check\_panel.add(reg\_password\_check\_info\_panel);

reg\_password\_check\_info\_panel.setVisible(false);

reg\_password\_check\_info\_ico = new JLabel("");

reg\_password\_check\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_password\_check\_info\_ico.setBounds(0, 0, 32, 32);

reg\_password\_check\_info\_panel.add(reg\_password\_check\_info\_ico);

reg\_password\_check\_info\_text = new JLabel("Error");

reg\_password\_check\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_password\_check\_info\_text.setBounds(42, 0, 328, 32);

reg\_password\_check\_info\_panel.add(reg\_password\_check\_info\_text);

// Reg form 2

reg\_form\_step\_2\_panel = new JFormDisplay("Datos del usuario", "/imagenes/user\_info\_64px.png", grupo\_reg, 2,

true, true);

reg\_form\_step\_2\_panel.setOpaque(false);

reg\_form\_step\_2\_panel.setBounds(173, 161, 449, 363);

display\_reg.add(reg\_form\_step\_2\_panel);

reg\_form\_step\_2\_panel.setLayout(null);

reg\_form\_step\_2\_panel.setVisible(false);

reg\_name\_panel = new JPanel();

reg\_name\_panel.setLayout(null);

reg\_name\_panel.setOpaque(false);

reg\_name\_panel.setBounds(31, 11, 390, 96);

reg\_form\_step\_2\_panel.add(reg\_name\_panel);

reg\_name\_separator = new JSeparator();

reg\_name\_separator.setBounds(52, 41, 328, 2);

reg\_name\_panel.add(reg\_name\_separator);

reg\_name\_ico = new JLabel("");

reg\_name\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/name\_32px.png")));

reg\_name\_ico.setBounds(10, 11, 32, 32);

reg\_name\_panel.add(reg\_name\_ico);

reg\_name\_textF = new JTextField("Nombre");

reg\_name\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_name\_textF.setForeground(Color.WHITE);

reg\_name\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_name\_textF.setColumns(10);

reg\_name\_textF.setBorder(null);

reg\_name\_textF.setBackground(new Color(85, 65, 118));

reg\_name\_textF.setBounds(52, 11, 328, 25);

reg\_name\_panel.add(reg\_name\_textF);

reg\_name\_info\_panel = new JPanel();

reg\_name\_info\_panel.setLayout(null);

reg\_name\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_name\_info\_panel.setBounds(10, 50, 370, 32);

reg\_name\_panel.add(reg\_name\_info\_panel);

reg\_name\_info\_panel.setVisible(false);

reg\_name\_info\_ico = new JLabel("");

reg\_name\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_name\_info\_ico.setBounds(0, 0, 32, 32);

reg\_name\_info\_panel.add(reg\_name\_info\_ico);

reg\_name\_info\_text = new JLabel("Error");

reg\_name\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_name\_info\_text.setBounds(42, 0, 328, 32);

reg\_name\_info\_panel.add(reg\_name\_info\_text);

reg\_apellido\_panel = new JPanel();

reg\_apellido\_panel.setLayout(null);

reg\_apellido\_panel.setOpaque(false);

reg\_apellido\_panel.setBounds(31, 118, 390, 96);

reg\_form\_step\_2\_panel.add(reg\_apellido\_panel);

reg\_apellido\_separator = new JSeparator();

reg\_apellido\_separator.setBounds(52, 41, 328, 2);

reg\_apellido\_panel.add(reg\_apellido\_separator);

reg\_apellido\_ico = new JLabel("");

reg\_apellido\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/apellidos\_32px.png")));

reg\_apellido\_ico.setBounds(10, 11, 32, 32);

reg\_apellido\_panel.add(reg\_apellido\_ico);

reg\_apellido\_textF = new JTextField("Apellidos");

reg\_apellido\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_apellido\_textF.setForeground(Color.WHITE);

reg\_apellido\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_apellido\_textF.setColumns(10);

reg\_apellido\_textF.setBorder(null);

reg\_apellido\_textF.setBackground(new Color(85, 65, 118));

reg\_apellido\_textF.setBounds(52, 11, 328, 25);

reg\_apellido\_panel.add(reg\_apellido\_textF);

reg\_apellido\_info\_panel = new JPanel();

reg\_apellido\_info\_panel.setLayout(null);

reg\_apellido\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_apellido\_info\_panel.setBounds(10, 50, 370, 32);

reg\_apellido\_panel.add(reg\_apellido\_info\_panel);

reg\_apellido\_info\_panel.setVisible(false);

reg\_apellido\_info\_ico = new JLabel("");

reg\_apellido\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_apellido\_info\_ico.setBounds(0, 0, 32, 32);

reg\_apellido\_info\_panel.add(reg\_apellido\_info\_ico);

reg\_apellido\_info\_text = new JLabel("Error");

reg\_apellido\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_apellido\_info\_text.setBounds(42, 0, 328, 32);

reg\_apellido\_info\_panel.add(reg\_apellido\_info\_text);

reg\_dni\_panel = new JPanel();

reg\_dni\_panel.setLayout(null);

reg\_dni\_panel.setOpaque(false);

reg\_dni\_panel.setBounds(31, 225, 390, 96);

reg\_form\_step\_2\_panel.add(reg\_dni\_panel);

reg\_dni\_separator = new JSeparator();

reg\_dni\_separator.setBounds(52, 41, 328, 2);

reg\_dni\_panel.add(reg\_dni\_separator);

reg\_dni\_ico = new JLabel("");

reg\_dni\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/dni\_32px.png")));

reg\_dni\_ico.setBounds(10, 11, 32, 32);

reg\_dni\_panel.add(reg\_dni\_ico);

reg\_dni\_textF = new JTextField("DNI");

reg\_dni\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_dni\_textF.setForeground(Color.WHITE);

reg\_dni\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_dni\_textF.setColumns(10);

reg\_dni\_textF.setBorder(null);

reg\_dni\_textF.setBackground(new Color(85, 65, 118));

reg\_dni\_textF.setBounds(52, 11, 328, 25);

reg\_dni\_panel.add(reg\_dni\_textF);

reg\_dni\_info\_panel = new JPanel();

reg\_dni\_info\_panel.setLayout(null);

reg\_dni\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_dni\_info\_panel.setBounds(10, 50, 370, 32);

reg\_dni\_panel.add(reg\_dni\_info\_panel);

reg\_dni\_info\_panel.setVisible(false);

reg\_dni\_info\_ico = new JLabel("");

reg\_dni\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_dni\_info\_ico.setBounds(0, 0, 32, 32);

reg\_dni\_info\_panel.add(reg\_dni\_info\_ico);

reg\_dni\_info\_text = new JLabel("Error");

reg\_dni\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_dni\_info\_text.setBounds(42, 0, 328, 32);

reg\_dni\_info\_panel.add(reg\_dni\_info\_text);

// Reg form 3

reg\_form\_step\_3\_panel = new JFormDisplay("Confirmar datos", "/imagenes/user\_info\_64px.png", grupo\_reg, 3, true,

false);

reg\_form\_step\_3\_panel.setOpaque(false);

reg\_form\_step\_3\_panel.setBounds(173, 161, 449, 363);

display\_reg.add(reg\_form\_step\_3\_panel);

reg\_form\_step\_3\_panel.setLayout(null);

reg\_form\_step\_3\_panel.setVisible(false);

reg\_email\_panel = new JPanel();

reg\_email\_panel.setLayout(null);

reg\_email\_panel.setOpaque(false);

reg\_email\_panel.setBounds(31, 11, 390, 96);

reg\_form\_step\_3\_panel.add(reg\_email\_panel);

reg\_email\_separator = new JSeparator();

reg\_email\_separator.setBounds(52, 41, 328, 2);

reg\_email\_panel.add(reg\_email\_separator);

reg\_email\_ico = new JLabel("");

reg\_email\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/email\_32px.png")));

reg\_email\_ico.setBounds(10, 11, 32, 32);

reg\_email\_panel.add(reg\_email\_ico);

reg\_email\_textF = new JTextField("Correo electronico");

reg\_email\_textF.setHorizontalAlignment(SwingConstants.LEFT);

reg\_email\_textF.setForeground(Color.WHITE);

reg\_email\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_email\_textF.setColumns(10);

reg\_email\_textF.setBorder(null);

reg\_email\_textF.setBackground(new Color(85, 65, 118));

reg\_email\_textF.setBounds(52, 11, 328, 25);

reg\_email\_panel.add(reg\_email\_textF);

reg\_email\_info\_panel = new JPanel();

reg\_email\_info\_panel.setLayout(null);

reg\_email\_info\_panel.setBackground(new Color(219, 186, 70));

reg\_email\_info\_panel.setBounds(10, 50, 370, 32);

reg\_email\_panel.add(reg\_email\_info\_panel);

reg\_email\_info\_panel.setVisible(false);

reg\_email\_info\_ico = new JLabel("");

reg\_email\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

reg\_email\_info\_ico.setBounds(0, 0, 32, 32);

reg\_email\_info\_panel.add(reg\_email\_info\_ico);

reg\_email\_info\_text = new JLabel("Error");

reg\_email\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

reg\_email\_info\_text.setBounds(42, 0, 328, 32);

reg\_email\_info\_panel.add(reg\_email\_info\_text);

reg\_confirmar\_FormBtn = new JFormBtn("reg", reg\_grupo\_logico);

reg\_confirmar\_FormBtn.setBackground(COLOR\_CHECK);

reg\_confirmar\_FormBtn.setBounds(100, 290, 258, 62);

reg\_form\_step\_3\_panel.add(reg\_confirmar\_FormBtn);

reg\_confirmar\_FormBtn.setLayout(null);

reg\_confirmar\_FormBtn.addMouseListener(new FormBtnListener(reg\_confirmar\_FormBtn, this));

reg\_confirmar\_formBtn\_text = new JLabel("Crear cuenta");

reg\_confirmar\_formBtn\_text.setBounds(10, 0, 238, 62);

reg\_confirmar\_FormBtn.add(reg\_confirmar\_formBtn\_text);

reg\_confirmar\_formBtn\_text.setForeground(Color.WHITE);

reg\_confirmar\_formBtn\_text.setFont(new Font("Tahoma", Font.BOLD, 25));

reg\_confirmar\_formBtn\_text.setHorizontalAlignment(SwingConstants.CENTER);

// Reg Nav labels

reg\_next\_navLabel = new JNavLabel("/imagenes/siguiente\_96px.png", "/imagenes/siguiente\_hover\_96px.png", "reg",

"next");

reg\_next\_navLabel.setIcon(new ImageIcon(Ventana.class.getResource(reg\_next\_navLabel.getIco\_path())));

reg\_next\_navLabel.setBounds(694, 279, 96, 96);

display\_reg.add(reg\_next\_navLabel);

reg\_next\_navLabel.addMouseListener(new NavRegListener(reg\_next\_navLabel, this));

reg\_back\_navLabel = new JNavLabel("/imagenes/atras\_96px.png", "/imagenes/atras\_hover\_96px.png", "reg", "back");

reg\_back\_navLabel.setIcon(new ImageIcon(Ventana.class.getResource(reg\_back\_navLabel.getIco\_path())));

reg\_back\_navLabel.setBounds(10, 279, 96, 96);

display\_reg.add(reg\_back\_navLabel);

reg\_back\_navLabel.addMouseListener(new NavRegListener(reg\_back\_navLabel, this));

reg\_back\_navLabel.setVisible(false);

// Reg tittle

reg\_form\_separator = new JSeparator();

reg\_form\_separator.setBounds(183, 150, 429, 2);

display\_reg.add(reg\_form\_separator);

reg\_form\_titulo = new JLabel("Datos de la cuenta");

reg\_form\_titulo.setBounds(262, 86, 338, 64);

display\_reg.add(reg\_form\_titulo);

reg\_form\_titulo.setHorizontalAlignment(SwingConstants.CENTER);

reg\_form\_titulo.setFont(new Font("Tahoma", Font.BOLD, 32));

reg\_form\_titulo.setForeground(Color.WHITE);

reg\_form\_ico = new JLabel("");

reg\_form\_ico.setBounds(205, 86, 63, 62);

display\_reg.add(reg\_form\_ico);

reg\_form\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_info\_64px.png")));

// Componentes logicos formulario Registro

reg\_nick\_tfg = new TextFieldGroup(reg\_nick\_panel, reg\_nick\_info\_panel, reg\_nick\_ico, reg\_nick\_info\_ico,

reg\_nick\_info\_text, reg\_nick\_textF, reg\_nick\_separator, "flex", reg\_grupo\_logico);

reg\_password\_tfg = new TextFieldGroup(reg\_password\_panel, reg\_password\_info\_panel, reg\_password\_ico,

reg\_password\_info\_ico, reg\_password\_info\_text, reg\_password\_textF, reg\_password\_separator, "pass",

reg\_grupo\_logico);

reg\_name\_tfg = new TextFieldGroup(reg\_name\_panel, reg\_name\_info\_panel, reg\_name\_ico, reg\_name\_info\_ico,

reg\_name\_info\_text, reg\_name\_textF, reg\_name\_separator, "strict", reg\_grupo\_logico);

reg\_apellido\_tfg = new TextFieldGroup(reg\_apellido\_panel, reg\_apellido\_info\_panel, reg\_apellido\_ico,

reg\_apellido\_info\_ico, reg\_apellido\_info\_text, reg\_apellido\_textF, reg\_apellido\_separator, "strict2",

reg\_grupo\_logico);

reg\_dni\_tfg = new TextFieldGroup(reg\_dni\_panel, reg\_dni\_info\_panel, reg\_dni\_ico, reg\_dni\_info\_ico,

reg\_dni\_info\_text, reg\_dni\_textF, reg\_dni\_separator, "dni", reg\_grupo\_logico);

reg\_email\_tfg = new TextFieldGroup(reg\_email\_panel, reg\_email\_info\_panel, reg\_email\_ico, reg\_email\_info\_ico,

reg\_email\_info\_text, reg\_email\_textF, reg\_email\_separator, "email", reg\_grupo\_logico);

reg\_password\_check\_tfg = new TextFieldGroupRel(reg\_password\_check\_panel, reg\_password\_check\_info\_panel,

reg\_password\_check\_ico, reg\_password\_check\_info\_ico, reg\_password\_check\_info\_text,

reg\_password\_check\_textF, reg\_password\_check\_separator, reg\_grupo\_logico, reg\_password\_tfg,

"similitud");

// Listeners reg

reg\_nick\_textF.addKeyListener(new TextFieldKeyListener(reg\_nick\_tfg, checker));

reg\_password\_textF.addKeyListener(new TextFieldKeyListener(reg\_password\_tfg, checker));

reg\_name\_textF.addKeyListener(new TextFieldKeyListener(reg\_name\_tfg, checker));

reg\_apellido\_textF.addKeyListener(new TextFieldKeyListener(reg\_apellido\_tfg, checker));

reg\_dni\_textF.addKeyListener(new TextFieldKeyListener(reg\_dni\_tfg, checker));

reg\_email\_textF.addKeyListener(new TextFieldKeyListener(reg\_email\_tfg, checker));

reg\_password\_check\_textF.addKeyListener(new TextFieldKeyListener(reg\_password\_check\_tfg, checker));

reg\_password\_textF.addKeyListener(new TextFieldKeyListener(reg\_password\_check\_tfg, checker));

reg\_nick\_textF.addFocusListener(new TextFocusListener(reg\_nick\_tfg, checker));

reg\_password\_textF.addFocusListener(new TextFocusListener(reg\_password\_tfg, checker));

reg\_name\_textF.addFocusListener(new TextFocusListener(reg\_name\_tfg, checker));

reg\_apellido\_textF.addFocusListener(new TextFocusListener(reg\_apellido\_tfg, checker));

reg\_dni\_textF.addFocusListener(new TextFocusListener(reg\_dni\_tfg, checker));

reg\_email\_textF.addFocusListener(new TextFocusListener(reg\_email\_tfg, checker));

reg\_password\_check\_textF.addFocusListener(new TextFocusListener(reg\_password\_check\_tfg, checker));

}

private void cargarDisplaySoporte() {

// --> Display Soporte

display\_soporte = new JDisplay("Soporte", "/imagenes/soporte\_96px.png", displays);

display\_soporte.setBounds(0, 0, 800, 552);

display\_panel.add(display\_soporte);

display\_soporte.setLayout(null);

display\_soporte.setOpaque(false);

display\_soporte.setVisible(false);

soporte\_asunto\_panel = new JPanel();

soporte\_asunto\_panel.setBounds(3, 29, 636, 96);

display\_soporte.add(soporte\_asunto\_panel);

soporte\_asunto\_panel.setOpaque(false);

soporte\_asunto\_panel.setLayout(null);

soporte\_asunto\_separator = new JSeparator();

soporte\_asunto\_separator.setBounds(52, 41, 574, 2);

soporte\_asunto\_panel.add(soporte\_asunto\_separator);

soporte\_asunto\_icon = new JLabel("");

soporte\_asunto\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/asunto\_32px.png")));

soporte\_asunto\_icon.setBounds(10, 11, 32, 32);

soporte\_asunto\_panel.add(soporte\_asunto\_icon);

soporte\_asunto\_textF = new JTextField("Asunto");

soporte\_asunto\_textF.setHorizontalAlignment(SwingConstants.LEFT);

soporte\_asunto\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

soporte\_asunto\_textF.setForeground(Color.WHITE);

soporte\_asunto\_textF.setBounds(52, 11, 328, 25);

soporte\_asunto\_panel.add(soporte\_asunto\_textF);

soporte\_asunto\_textF.setColumns(10);

soporte\_asunto\_textF.setBackground(COLOR\_SELECTED);

soporte\_asunto\_textF.setBorder(null);

soporte\_asunto\_info\_panel = new JPanel();

soporte\_asunto\_info\_panel.setBounds(10, 50, 616, 32);

soporte\_asunto\_panel.add(soporte\_asunto\_info\_panel);

soporte\_asunto\_info\_panel.setBackground(COLOR\_ERROR);

soporte\_asunto\_info\_panel.setLayout(null);

soporte\_asunto\_info\_panel.setVisible(false);

soporte\_asunto\_info\_ico = new JLabel("");

soporte\_asunto\_info\_ico.setBounds(0, 0, 32, 32);

soporte\_asunto\_info\_panel.add(soporte\_asunto\_info\_ico);

soporte\_asunto\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

soporte\_asunto\_info\_text = new JLabel("Error");

soporte\_asunto\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

soporte\_asunto\_info\_text.setBounds(42, 0, 572, 32);

soporte\_asunto\_info\_panel.add(soporte\_asunto\_info\_text);

soporte\_textArea = new JTextArea();

soporte\_textArea.setLineWrap(true);

soporte\_textArea.setFont(new Font("Monospaced", Font.PLAIN, 22));

soporte\_textArea.setText("Describre detalladamente la incidencia.");

soporte\_textArea.setBounds(12, 127, 776, 239);

display\_soporte.add(soporte\_textArea);

soporte\_caracteres\_text = new JLabel("Maximo 300 caracteres");

soporte\_caracteres\_text.setForeground(Color.WHITE);

soporte\_caracteres\_text.setBounds(12, 366, 329, 47);

display\_soporte.add(soporte\_caracteres\_text);

soporte\_caracteres\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

TextAreaListener textAreaListener = new TextAreaListener(soporte\_textArea, soporte\_caracteres\_text);

soporte\_textArea.addKeyListener(textAreaListener);

soporte\_textArea.addFocusListener(textAreaListener);

soporte\_asunto\_tfg = new TextFieldGroup(soporte\_asunto\_panel, soporte\_asunto\_info\_panel, soporte\_asunto\_icon,

soporte\_asunto\_info\_ico, soporte\_asunto\_info\_text, soporte\_asunto\_textF, soporte\_asunto\_separator,

"soporte");

soporte\_asunto\_textF.addKeyListener(new TextFieldKeyListener(soporte\_asunto\_tfg, checker));

soporte\_asunto\_textF.addFocusListener(new TextFocusListener(soporte\_asunto\_tfg, checker));

soporte\_enviar\_FormBtn = new JFormBtn("soporte");

soporte\_enviar\_FormBtn.setBackground(COLOR\_CHECK);

soporte\_enviar\_FormBtn.setBounds(297, 436, 258, 62);

display\_soporte.add(soporte\_enviar\_FormBtn);

soporte\_enviar\_FormBtn.setLayout(null);

soporte\_enviar\_FormBtn.addMouseListener(new SoporteBtnListener(soporte\_textArea, soporte\_asunto\_tfg, this));

soporte\_enviar\_formBtn\_text = new JLabel("Enviar");

soporte\_enviar\_formBtn\_text.setBounds(10, 0, 238, 62);

soporte\_enviar\_FormBtn.add(soporte\_enviar\_formBtn\_text);

soporte\_enviar\_formBtn\_text.setForeground(Color.WHITE);

soporte\_enviar\_formBtn\_text.setFont(new Font("Tahoma", Font.BOLD, 25));

soporte\_enviar\_formBtn\_text.setHorizontalAlignment(SwingConstants.CENTER);

}

private void cargarDisplayInfo() {

// --> Display Info

display\_info = new JDisplay("Información", "/imagenes/info\_96px.png", displays);

display\_info.setBounds(0, 0, 800, 552);

display\_panel.add(display\_info);

display\_info.setLayout(null);

display\_info.setOpaque(false);

display\_info.setVisible(false);

mensaje\_info = new JLabel(

"<html><p>HACENTEST\u00A9<br><br>Aplicacion de simalacion de test para el carnet de conducir en Espa\u00F1a.<br><br>"

+ "Version: D.0.7.1<br><br>Autor: Adan Jarillo Merida.<br><br>"

+ "Desarrollado como proyecto de programacion para el ciclo de grado superiror de D.A.M. en CES Fuencarral curso 2017/2018.<p><html>");

mensaje\_info.setHorizontalAlignment(SwingConstants.CENTER);

mensaje\_info.setForeground(Color.WHITE);

mensaje\_info.setFont(new Font("Tahoma", Font.PLAIN, 22));

mensaje\_info.setBounds(50, 22, 700, 351);

display\_info.add(mensaje\_info);

java\_ico = new JLabel("");

java\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/javaico\_96px.png")));

java\_ico.setBounds(225, 384, 96, 96);

display\_info.add(java\_ico);

eclipse\_ico = new JLabel("");

eclipse\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/eclipseico\_96px.png")));

eclipse\_ico.setBounds(456, 384, 96, 96);

display\_info.add(eclipse\_ico);

mysql\_ico = new JLabel("");

mysql\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/mysqlico\_96px.png")));

mysql\_ico.setBounds(341, 384, 96, 96);

display\_info.add(mysql\_ico);

}

private void cargarDisplayVerUsuarios() {

display\_verUsuarios = new JDisplay("Buscar usuario", "/imagenes/buscar\_96px.png", displays);

display\_verUsuarios.setBounds(0, 0, 800, 552);

display\_panel.add(display\_verUsuarios);

display\_verUsuarios.setLayout(null);

display\_verUsuarios.setOpaque(false);

display\_verUsuarios.setVisible(false);

verUsuarios\_nick\_panel = new JPanel();

verUsuarios\_nick\_panel.setBounds(204, 0, 390, 96);

display\_verUsuarios.add(verUsuarios\_nick\_panel);

verUsuarios\_nick\_panel.setOpaque(false);

verUsuarios\_nick\_panel.setLayout(null);

verUsuarios\_nick\_separator = new JSeparator();

verUsuarios\_nick\_separator.setBounds(52, 41, 328, 2);

verUsuarios\_nick\_panel.add(verUsuarios\_nick\_separator);

verUsuarios\_nick\_icon = new JLabel("");

verUsuarios\_nick\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/buscar\_32px.png")));

verUsuarios\_nick\_icon.setBounds(10, 11, 32, 32);

verUsuarios\_nick\_panel.add(verUsuarios\_nick\_icon);

verUsuarios\_nick\_textF = new JTextField("Buscar");

verUsuarios\_nick\_textF.setHorizontalAlignment(SwingConstants.LEFT);

verUsuarios\_nick\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

verUsuarios\_nick\_textF.setForeground(Color.WHITE);

verUsuarios\_nick\_textF.setBounds(52, 11, 328, 25);

verUsuarios\_nick\_panel.add(verUsuarios\_nick\_textF);

verUsuarios\_nick\_textF.setColumns(10);

verUsuarios\_nick\_textF.setBackground(COLOR\_SELECTED);

verUsuarios\_nick\_textF.setBorder(null);

verUsuarios\_nick\_info\_panel = new JPanel();

verUsuarios\_nick\_info\_panel.setBounds(10, 50, 370, 32);

verUsuarios\_nick\_panel.add(verUsuarios\_nick\_info\_panel);

verUsuarios\_nick\_info\_panel.setBackground(COLOR\_ERROR);

verUsuarios\_nick\_info\_panel.setLayout(null);

verUsuarios\_nick\_info\_panel.setVisible(false);

verUsuarios\_nick\_info\_ico = new JLabel("");

verUsuarios\_nick\_info\_ico.setBounds(0, 0, 32, 32);

verUsuarios\_nick\_info\_panel.add(verUsuarios\_nick\_info\_ico);

verUsuarios\_nick\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

verUsuarios\_nick\_info\_text = new JLabel("Error");

verUsuarios\_nick\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

verUsuarios\_nick\_info\_text.setBounds(42, 0, 328, 32);

verUsuarios\_nick\_info\_panel.add(verUsuarios\_nick\_info\_text);

verUsuarios\_scroll = new JScrollPane();

verUsuarios\_scroll.setBorder(null);

verUsuarios\_scroll.setOpaque(false);

verUsuarios\_scroll.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

verUsuarios\_scroll.setBounds(10, 90, 780, 450);

display\_verUsuarios.add(verUsuarios\_scroll);

verUsuarios\_contenedor = new JPanel();

verUsuarios\_contenedor.setBackground(COLOR\_SELECTED);

verUsuarios\_scroll.setViewportView(verUsuarios\_contenedor);

verUsuarios\_contenedor.setLayout(new GridLayout(0, 1, 0, 20));

ArrayList<Usuario> listaUsers = new ArrayList<Usuario>();

listUpdater\_verUsuarios = new JUserRowList(listaUsers, verUsuarios\_contenedor, this);

listUpdater\_verUsuarios.cargarLista();

verUsuarios\_nick\_tfg = new TextFieldGroup(verUsuarios\_nick\_panel, verUsuarios\_nick\_info\_panel,

verUsuarios\_nick\_icon, verUsuarios\_nick\_info\_ico, verUsuarios\_nick\_info\_text, verUsuarios\_nick\_textF,

verUsuarios\_nick\_separator, "buscar");

verUsuarios\_nick\_textF.addKeyListener(new TextFieldKeyListener(verUsuarios\_nick\_tfg, checker, this));

verUsuarios\_nick\_textF.addFocusListener(new TextFocusListener(verUsuarios\_nick\_tfg, checker));

}

private void cargarDisplayMostrarUsuario() {

display\_mostrarUsuario = new JDisplay("Mostrar usuario", "/imagenes/info\_96px.png", displays);

display\_mostrarUsuario.setBounds(0, 0, 800, 552);

display\_panel.add(display\_mostrarUsuario);

display\_mostrarUsuario.setLayout(null);

display\_mostrarUsuario.setOpaque(false);

display\_mostrarUsuario.setVisible(false);

nick\_panel\_mostrarUsuario = new JPanel();

nick\_panel\_mostrarUsuario.setLayout(null);

nick\_panel\_mostrarUsuario.setOpaque(false);

nick\_panel\_mostrarUsuario.setBounds(10, 11, 390, 52);

display\_mostrarUsuario.add(nick\_panel\_mostrarUsuario);

nick\_separator\_mostrarUsuario = new JSeparator();

nick\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

nick\_panel\_mostrarUsuario.add(nick\_separator\_mostrarUsuario);

nick\_ico\_mostrarUsuario = new JLabel("");

nick\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_text\_32px.png")));

nick\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

nick\_panel\_mostrarUsuario.add(nick\_ico\_mostrarUsuario);

nick\_text\_mostrarUsuario = new JLabel("Nick");

nick\_text\_mostrarUsuario.setForeground(Color.WHITE);

nick\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

nick\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

nick\_panel\_mostrarUsuario.add(nick\_text\_mostrarUsuario);

nombre\_panel\_mostrarUsuario = new JPanel();

nombre\_panel\_mostrarUsuario.setLayout(null);

nombre\_panel\_mostrarUsuario.setOpaque(false);

nombre\_panel\_mostrarUsuario.setBounds(10, 74, 390, 52);

display\_mostrarUsuario.add(nombre\_panel\_mostrarUsuario);

nombre\_separator\_mostrarUsuario = new JSeparator();

nombre\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

nombre\_panel\_mostrarUsuario.add(nombre\_separator\_mostrarUsuario);

nombre\_ico\_mostrarUsuario = new JLabel("");

nombre\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/name\_32px.png")));

nombre\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

nombre\_panel\_mostrarUsuario.add(nombre\_ico\_mostrarUsuario);

nombre\_text\_mostrarUsuario = new JLabel("Nombre");

nombre\_text\_mostrarUsuario.setForeground(Color.WHITE);

nombre\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

nombre\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

nombre\_panel\_mostrarUsuario.add(nombre\_text\_mostrarUsuario);

apellidos\_panel\_mostrarUsuario = new JPanel();

apellidos\_panel\_mostrarUsuario.setLayout(null);

apellidos\_panel\_mostrarUsuario.setOpaque(false);

apellidos\_panel\_mostrarUsuario.setBounds(10, 137, 390, 52);

display\_mostrarUsuario.add(apellidos\_panel\_mostrarUsuario);

apellidos\_separator\_mostrarUsuario = new JSeparator();

apellidos\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

apellidos\_panel\_mostrarUsuario.add(apellidos\_separator\_mostrarUsuario);

apellidos\_ico\_mostrarUsuario = new JLabel("");

apellidos\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/apellidos\_32px.png")));

apellidos\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

apellidos\_panel\_mostrarUsuario.add(apellidos\_ico\_mostrarUsuario);

apellidos\_text\_mostrarUsuario = new JLabel("Apellidos");

apellidos\_text\_mostrarUsuario.setForeground(Color.WHITE);

apellidos\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

apellidos\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

apellidos\_panel\_mostrarUsuario.add(apellidos\_text\_mostrarUsuario);

dni\_panel\_mostrarUsuario = new JPanel();

dni\_panel\_mostrarUsuario.setLayout(null);

dni\_panel\_mostrarUsuario.setOpaque(false);

dni\_panel\_mostrarUsuario.setBounds(10, 200, 390, 52);

display\_mostrarUsuario.add(dni\_panel\_mostrarUsuario);

dni\_separator\_mostrarUsuario = new JSeparator();

dni\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

dni\_panel\_mostrarUsuario.add(dni\_separator\_mostrarUsuario);

dni\_ico\_mostrarUsuario = new JLabel("");

dni\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/dni\_32px.png")));

dni\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

dni\_panel\_mostrarUsuario.add(dni\_ico\_mostrarUsuario);

dni\_text\_mostrarUsuario = new JLabel("DNI");

dni\_text\_mostrarUsuario.setForeground(Color.WHITE);

dni\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

dni\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

dni\_panel\_mostrarUsuario.add(dni\_text\_mostrarUsuario);

email\_panel\_mostrarUsuario = new JPanel();

email\_panel\_mostrarUsuario.setLayout(null);

email\_panel\_mostrarUsuario.setOpaque(false);

email\_panel\_mostrarUsuario.setBounds(10, 263, 390, 52);

display\_mostrarUsuario.add(email\_panel\_mostrarUsuario);

email\_separator\_mostrarUsuario = new JSeparator();

email\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

email\_panel\_mostrarUsuario.add(email\_separator\_mostrarUsuario);

email\_ico\_mostrarUsuario = new JLabel("");

email\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/email\_32px.png")));

email\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

email\_panel\_mostrarUsuario.add(email\_ico\_mostrarUsuario);

email\_text\_mostrarUsuario = new JLabel("Correo electronico");

email\_text\_mostrarUsuario.setForeground(Color.WHITE);

email\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

email\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

email\_panel\_mostrarUsuario.add(email\_text\_mostrarUsuario);

fechar\_panel\_mostrarUsuario = new JPanel();

fechar\_panel\_mostrarUsuario.setLayout(null);

fechar\_panel\_mostrarUsuario.setOpaque(false);

fechar\_panel\_mostrarUsuario.setBounds(10, 326, 390, 52);

display\_mostrarUsuario.add(fechar\_panel\_mostrarUsuario);

fechar\_separator\_mostrarUsuario = new JSeparator();

fechar\_separator\_mostrarUsuario.setBounds(52, 41, 328, 2);

fechar\_panel\_mostrarUsuario.add(fechar\_separator\_mostrarUsuario);

fechar\_ico\_mostrarUsuario = new JLabel("");

fechar\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/fecha\_32px.png")));

fechar\_ico\_mostrarUsuario.setBounds(10, 11, 32, 32);

fechar\_panel\_mostrarUsuario.add(fechar\_ico\_mostrarUsuario);

fechar\_text\_mostrarUsuario = new JLabel("Fecha registro");

fechar\_text\_mostrarUsuario.setForeground(Color.WHITE);

fechar\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 20));

fechar\_text\_mostrarUsuario.setBounds(52, 11, 328, 30);

fechar\_panel\_mostrarUsuario.add(fechar\_text\_mostrarUsuario);

tiempoDeUso\_ico\_mostrarUsuario = new JLabel("");

tiempoDeUso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/time\_96px.png")));

tiempoDeUso\_ico\_mostrarUsuario.setBounds(410, 30, 96, 96);

display\_mostrarUsuario.add(tiempoDeUso\_ico\_mostrarUsuario);

tiempoDeUso\_text\_mostrarUsuario = new JLabel("DD:HH:MM");

tiempoDeUso\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 26));

tiempoDeUso\_text\_mostrarUsuario.setForeground(Color.WHITE);

tiempoDeUso\_text\_mostrarUsuario.setBounds(516, 50, 221, 47);

display\_mostrarUsuario.add(tiempoDeUso\_text\_mostrarUsuario);

acceso\_ico\_mostrarUsuario = new JLabel("");

acceso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/dev\_96px.png")));

acceso\_ico\_mostrarUsuario.setBounds(410, 282, 96, 96);

display\_mostrarUsuario.add(acceso\_ico\_mostrarUsuario);

estado\_text\_mostrarUsuario = new JLabel("Acceso");

estado\_text\_mostrarUsuario.setForeground(Color.WHITE);

estado\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 26));

estado\_text\_mostrarUsuario.setBounds(516, 175, 221, 47);

display\_mostrarUsuario.add(estado\_text\_mostrarUsuario);

estado\_ico\_mostrarUsuario = new JLabel("");

estado\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/online\_96px.png")));

estado\_ico\_mostrarUsuario.setBounds(410, 156, 96, 96);

display\_mostrarUsuario.add(estado\_ico\_mostrarUsuario);

acceso\_text\_mostrarUsuario = new JLabel("Acceso");

acceso\_text\_mostrarUsuario.setForeground(Color.WHITE);

acceso\_text\_mostrarUsuario.setFont(new Font("Tahoma", Font.BOLD, 26));

acceso\_text\_mostrarUsuario.setBounds(516, 304, 221, 47);

display\_mostrarUsuario.add(acceso\_text\_mostrarUsuario);

estadoCambio\_panel\_mostrarUsuario = new JPanel();

estadoCambio\_panel\_mostrarUsuario.setOpaque(false);

estadoCambio\_panel\_mostrarUsuario.setBounds(526, 233, 123, 32);

display\_mostrarUsuario.add(estadoCambio\_panel\_mostrarUsuario);

estadoCambio\_panel\_mostrarUsuario.setLayout(null);

estadoCambio\_online\_mostrarUsuario = new JChanger("/imagenes/online\_32px.png",

"/imagenes/online\_hover\_32px.png", "estado", "online");

estadoCambio\_online\_mostrarUsuario.setBounds(0, 0, 32, 32);

estadoCambio\_online\_mostrarUsuario

.setIcon(new ImageIcon(Ventana.class.getResource(estadoCambio\_online\_mostrarUsuario.getIco\_path())));

estadoCambio\_panel\_mostrarUsuario.add(estadoCambio\_online\_mostrarUsuario);

estadoCambio\_offline\_mostrarUsuario = new JChanger("/imagenes/offline\_32px.png",

"/imagenes/offline\_hover\_32px.png", "estado", "offline");

estadoCambio\_offline\_mostrarUsuario.setBounds(42, 0, 32, 32);

estadoCambio\_offline\_mostrarUsuario

.setIcon(new ImageIcon(Ventana.class.getResource(estadoCambio\_offline\_mostrarUsuario.getIco\_path())));

estadoCambio\_panel\_mostrarUsuario.add(estadoCambio\_offline\_mostrarUsuario);

estadoCambio\_suspendido\_mostrarUsuario = new JChanger("/imagenes/banned\_32px.png",

"/imagenes/banned\_hover\_32px.png", "estado", "suspendido");

estadoCambio\_suspendido\_mostrarUsuario.setBounds(84, 0, 32, 32);

estadoCambio\_suspendido\_mostrarUsuario.setIcon(

new ImageIcon(Ventana.class.getResource(estadoCambio\_suspendido\_mostrarUsuario.getIco\_path())));

estadoCambio\_panel\_mostrarUsuario.add(estadoCambio\_suspendido\_mostrarUsuario);

accesoCambio\_panel\_mostrarUsuario = new JPanel();

accesoCambio\_panel\_mostrarUsuario.setLayout(null);

accesoCambio\_panel\_mostrarUsuario.setOpaque(false);

accesoCambio\_panel\_mostrarUsuario.setBounds(516, 361, 123, 32);

display\_mostrarUsuario.add(accesoCambio\_panel\_mostrarUsuario);

accesoCambio\_alumno\_mostrarUsuario = new JChanger("/imagenes/alumno\_32px.png",

"/imagenes/alumno\_hover\_32px.png", "acceso", "alumno");

accesoCambio\_alumno\_mostrarUsuario.setBounds(0, 0, 32, 32);

accesoCambio\_panel\_mostrarUsuario.add(accesoCambio\_alumno\_mostrarUsuario);

accesoCambio\_alumno\_mostrarUsuario

.setIcon(new ImageIcon(Ventana.class.getResource(accesoCambio\_alumno\_mostrarUsuario.getIco\_path())));

accesoCambio\_profesor\_mostrarUsuario = new JChanger("/imagenes/profesor\_32px.png",

"/imagenes/profesor\_hover\_32px.png", "acceso", "profesor");

accesoCambio\_profesor\_mostrarUsuario.setBounds(42, 0, 32, 32);

accesoCambio\_panel\_mostrarUsuario.add(accesoCambio\_profesor\_mostrarUsuario);

accesoCambio\_profesor\_mostrarUsuario

.setIcon(new ImageIcon(Ventana.class.getResource(accesoCambio\_profesor\_mostrarUsuario.getIco\_path())));

accesoCambio\_admin\_mostrarUsuario = new JChanger("/imagenes/admin\_32px.png", "/imagenes/admin\_hover\_32px.png",

"acceso", "administrador");

accesoCambio\_admin\_mostrarUsuario.setBounds(81, 0, 32, 32);

accesoCambio\_panel\_mostrarUsuario.add(accesoCambio\_admin\_mostrarUsuario);

accesoCambio\_admin\_mostrarUsuario

.setIcon(new ImageIcon(Ventana.class.getResource(accesoCambio\_admin\_mostrarUsuario.getIco\_path())));

estadoCambio\_online\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(estadoCambio\_online\_mostrarUsuario, this));

estadoCambio\_offline\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(estadoCambio\_offline\_mostrarUsuario, this));

estadoCambio\_suspendido\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(estadoCambio\_suspendido\_mostrarUsuario, this));

accesoCambio\_alumno\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(accesoCambio\_alumno\_mostrarUsuario, this));

accesoCambio\_profesor\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(accesoCambio\_profesor\_mostrarUsuario, this));

accesoCambio\_admin\_mostrarUsuario

.addMouseListener(new ChangerBtnListener(accesoCambio\_admin\_mostrarUsuario, this));

mostrarUsuario\_estadisticas\_btn = new JPanel();

mostrarUsuario\_estadisticas\_btn.setBackground(COLOR\_BACKGROUND);

mostrarUsuario\_estadisticas\_btn.setBounds(210, 427, 356, 87);

display\_mostrarUsuario.add(mostrarUsuario\_estadisticas\_btn);

mostrarUsuario\_estadisticas\_btn.setLayout(null);

mostrarUsuario\_estadisticas\_btn.addMouseListener(new EstadisticasBtnListener(this));

mostrarUsuario\_estadisticas\_btn\_text = new JLabel("Estadisticas");

mostrarUsuario\_estadisticas\_btn\_text.setBounds(0, 0, 356, 87);

mostrarUsuario\_estadisticas\_btn.add(mostrarUsuario\_estadisticas\_btn\_text);

mostrarUsuario\_estadisticas\_btn\_text.setHorizontalAlignment(SwingConstants.CENTER);

mostrarUsuario\_estadisticas\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 34));

mostrarUsuario\_estadisticas\_btn\_text.setForeground(Color.WHITE);

mostrarUsuario\_estadisticas\_btn.addMouseListener(new TestBtnListener(this));

}

private void cargarDisplayCerrarSesion() {

// --> Display Info

display\_cerrarSesion = new JDisplay("Cerrar sesion", "/imagenes/cerrar\_96px.png", displays);

display\_cerrarSesion.setBounds(0, 0, 800, 552);

display\_panel.add(display\_cerrarSesion);

display\_cerrarSesion.setLayout(null);

display\_cerrarSesion.setOpaque(false);

display\_cerrarSesion.setVisible(false);

cerrarSesion\_btn\_panel = new JPanel();

cerrarSesion\_btn\_panel.setBackground(COLOR\_BACKGROUND);

cerrarSesion\_btn\_panel.setBounds(228, 200, 356, 150);

display\_cerrarSesion.add(cerrarSesion\_btn\_panel);

cerrarSesion\_btn\_panel.setLayout(null);

cerrarSesion\_btn\_panel.addMouseListener(new SesionBtnListener(this));

cerrarSesion\_btn\_text = new JLabel("Cerrar sesion");

cerrarSesion\_btn\_text.setHorizontalAlignment(SwingConstants.CENTER);

cerrarSesion\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 34));

cerrarSesion\_btn\_text.setForeground(Color.WHITE);

cerrarSesion\_btn\_text.setBounds(10, 44, 336, 60);

cerrarSesion\_btn\_panel.add(cerrarSesion\_btn\_text);

}

private void cargarDisplayEntrarComo() {

// --> Display Info

display\_entrarComo = new JDisplay("Cerrar sesion", "/imagenes/cerrar\_96px.png", displays);

display\_entrarComo.setBounds(0, 0, 800, 552);

display\_panel.add(display\_entrarComo);

display\_entrarComo.setLayout(null);

display\_entrarComo.setOpaque(false);

display\_entrarComo.setVisible(false);

entrarComo\_alumno\_btn\_panel = new JPanel();

entrarComo\_alumno\_btn\_panel.setBackground(COLOR\_BACKGROUND);

entrarComo\_alumno\_btn\_panel.setBounds(144, 38, 493, 150);

display\_entrarComo.add(entrarComo\_alumno\_btn\_panel);

entrarComo\_alumno\_btn\_panel.setLayout(null);

entrarComo\_alumno\_btn\_text = new JLabel("Alumno");

entrarComo\_alumno\_btn\_text.setHorizontalAlignment(SwingConstants.LEFT);

entrarComo\_alumno\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 36));

entrarComo\_alumno\_btn\_text.setForeground(Color.WHITE);

entrarComo\_alumno\_btn\_text.setBounds(163, 46, 360, 60);

entrarComo\_alumno\_btn\_panel.add(entrarComo\_alumno\_btn\_text);

entrarComo\_alumno\_btn\_ico = new JLabel("");

entrarComo\_alumno\_btn\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/alumno\_96px.png")));

entrarComo\_alumno\_btn\_ico.setBounds(41, 29, 96, 96);

entrarComo\_alumno\_btn\_panel.add(entrarComo\_alumno\_btn\_ico);

entrarComo\_profesor\_btn\_panel = new JPanel();

entrarComo\_profesor\_btn\_panel.setLayout(null);

entrarComo\_profesor\_btn\_panel.setBackground(new Color(54, 33, 89));

entrarComo\_profesor\_btn\_panel.setBounds(144, 199, 493, 150);

display\_entrarComo.add(entrarComo\_profesor\_btn\_panel);

entrarComo\_profesor\_btn\_text = new JLabel("Profesor");

entrarComo\_profesor\_btn\_text.setHorizontalAlignment(SwingConstants.LEFT);

entrarComo\_profesor\_btn\_text.setForeground(Color.WHITE);

entrarComo\_profesor\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 36));

entrarComo\_profesor\_btn\_text.setBounds(163, 46, 360, 60);

entrarComo\_profesor\_btn\_panel.add(entrarComo\_profesor\_btn\_text);

entrarComo\_profesor\_btn\_ico = new JLabel("");

entrarComo\_profesor\_btn\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/profesor\_96px.png")));

entrarComo\_profesor\_btn\_ico.setBounds(41, 29, 96, 96);

entrarComo\_profesor\_btn\_panel.add(entrarComo\_profesor\_btn\_ico);

entrarComo\_administrador\_btn\_panel = new JPanel();

entrarComo\_administrador\_btn\_panel.setLayout(null);

entrarComo\_administrador\_btn\_panel.setBackground(new Color(54, 33, 89));

entrarComo\_administrador\_btn\_panel.setBounds(144, 360, 493, 150);

display\_entrarComo.add(entrarComo\_administrador\_btn\_panel);

entrarComo\_administrador\_btn\_text = new JLabel("Administrador");

entrarComo\_administrador\_btn\_text.setHorizontalAlignment(SwingConstants.LEFT);

entrarComo\_administrador\_btn\_text.setForeground(Color.WHITE);

entrarComo\_administrador\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 36));

entrarComo\_administrador\_btn\_text.setBounds(163, 46, 360, 60);

entrarComo\_administrador\_btn\_panel.add(entrarComo\_administrador\_btn\_text);

entrarComo\_administrador\_btn\_ico = new JLabel("");

entrarComo\_administrador\_btn\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/admin\_96px.png")));

entrarComo\_administrador\_btn\_ico.setBounds(41, 29, 96, 96);

entrarComo\_administrador\_btn\_panel.add(entrarComo\_administrador\_btn\_ico);

entrarComo\_alumno\_btn\_panel.addMouseListener(new EntrarComoListener(this, entrarComo\_alumno\_btn\_text));

entrarComo\_profesor\_btn\_panel.addMouseListener(new EntrarComoListener(this, entrarComo\_profesor\_btn\_text));

entrarComo\_administrador\_btn\_panel

.addMouseListener(new EntrarComoListener(this, entrarComo\_administrador\_btn\_text));

}

private void cargarDisplayVerConsultas() {

display\_verConsultas = new JDisplay("Buscar consulta", "/imagenes/buscar\_96px.png", displays);

display\_verConsultas.setBounds(0, 0, 800, 552);

display\_panel.add(display\_verConsultas);

display\_verConsultas.setLayout(null);

display\_verConsultas.setOpaque(false);

display\_verConsultas.setVisible(false);

verConsultas\_filtro\_panel = new JPanel();

verConsultas\_filtro\_panel.setBounds(204, 0, 390, 96);

display\_verConsultas.add(verConsultas\_filtro\_panel);

verConsultas\_filtro\_panel.setOpaque(false);

verConsultas\_filtro\_panel.setLayout(null);

verConsultas\_filtro\_separator = new JSeparator();

verConsultas\_filtro\_separator.setBounds(52, 41, 328, 2);

verConsultas\_filtro\_panel.add(verConsultas\_filtro\_separator);

verConsultas\_filtro\_icon = new JLabel("");

verConsultas\_filtro\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/buscar\_32px.png")));

verConsultas\_filtro\_icon.setBounds(10, 11, 32, 32);

verConsultas\_filtro\_panel.add(verConsultas\_filtro\_icon);

verConsultas\_filtro\_textF = new JTextField("Buscar");

verConsultas\_filtro\_textF.setHorizontalAlignment(SwingConstants.LEFT);

verConsultas\_filtro\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

verConsultas\_filtro\_textF.setForeground(Color.WHITE);

verConsultas\_filtro\_textF.setBounds(52, 11, 328, 25);

verConsultas\_filtro\_panel.add(verConsultas\_filtro\_textF);

verConsultas\_filtro\_textF.setColumns(10);

verConsultas\_filtro\_textF.setBackground(COLOR\_SELECTED);

verConsultas\_filtro\_textF.setBorder(null);

verConsultas\_filtro\_info\_panel = new JPanel();

verConsultas\_filtro\_info\_panel.setBounds(10, 50, 370, 32);

verConsultas\_filtro\_panel.add(verConsultas\_filtro\_info\_panel);

verConsultas\_filtro\_info\_panel.setBackground(COLOR\_ERROR);

verConsultas\_filtro\_info\_panel.setLayout(null);

verConsultas\_filtro\_info\_panel.setVisible(false);

verConsultas\_filtro\_info\_ico = new JLabel("");

verConsultas\_filtro\_info\_ico.setBounds(0, 0, 32, 32);

verConsultas\_filtro\_info\_panel.add(verConsultas\_filtro\_info\_ico);

verConsultas\_filtro\_info\_ico

.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

verConsultas\_filtro\_info\_text = new JLabel("Error");

verConsultas\_filtro\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

verConsultas\_filtro\_info\_text.setBounds(42, 0, 328, 32);

verConsultas\_filtro\_info\_panel.add(verConsultas\_filtro\_info\_text);

verConsultas\_scroll = new JScrollPane();

verConsultas\_scroll.setBorder(null);

verConsultas\_scroll.setOpaque(false);

verConsultas\_scroll.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER);

verConsultas\_scroll.setBounds(10, 90, 780, 450);

display\_verConsultas.add(verConsultas\_scroll);

verConsultas\_\_contenedor = new JPanel();

verConsultas\_\_contenedor.setBackground(COLOR\_SELECTED);

verConsultas\_scroll.setViewportView(verConsultas\_\_contenedor);

verConsultas\_\_contenedor.setLayout(new GridLayout(0, 1, 0, 20));

ArrayList<Consulta> listaConsultas = new ArrayList<Consulta>();

listUpdater\_verConsultas = new JConsultaRowList(listaConsultas, verConsultas\_\_contenedor, this);

listUpdater\_verConsultas.cargarLista();

verConsultas\_filto\_tfg = new TextFieldGroup(verConsultas\_filtro\_panel, verConsultas\_filtro\_info\_panel,

verConsultas\_filtro\_icon, verConsultas\_filtro\_info\_ico, verConsultas\_filtro\_info\_text,

verConsultas\_filtro\_textF, verConsultas\_filtro\_separator, "consulta");

verConsultas\_filtro\_textF.addKeyListener(new TextFieldKeyListener(verConsultas\_filto\_tfg, checker, this));

verConsultas\_filtro\_textF.addFocusListener(new TextFocusListener(verConsultas\_filto\_tfg, checker));

}

private void cargarDisplayAñadirPregunta() {

display\_anadirPregunta = new JDisplay("Añadir pregunta", "/imagenes/pregunta\_96px.png", displays);

display\_anadirPregunta.setBounds(0, 0, 790, 552);

display\_panel.add(display\_anadirPregunta);

display\_anadirPregunta.setLayout(null);

display\_anadirPregunta.setOpaque(false);

display\_anadirPregunta.setVisible(false);

anadirPregunta\_enunciado\_panel = new JPanel();

anadirPregunta\_enunciado\_panel.setBounds(10, 29, 780, 96);

display\_anadirPregunta.add(anadirPregunta\_enunciado\_panel);

anadirPregunta\_enunciado\_panel.setOpaque(false);

anadirPregunta\_enunciado\_panel.setLayout(null);

anadirPregunta\_enunciado\_separator = new JSeparator();

anadirPregunta\_enunciado\_separator.setBounds(52, 41, 718, 2);

anadirPregunta\_enunciado\_panel.add(anadirPregunta\_enunciado\_separator);

anadirPregunta\_enunciado\_icon = new JLabel("");

anadirPregunta\_enunciado\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/enunciado\_32px.png")));

anadirPregunta\_enunciado\_icon.setBounds(10, 11, 32, 32);

anadirPregunta\_enunciado\_panel.add(anadirPregunta\_enunciado\_icon);

anadirPregunta\_enunciado\_textF = new JTextField("Enunciado");

anadirPregunta\_enunciado\_textF.setHorizontalAlignment(SwingConstants.LEFT);

anadirPregunta\_enunciado\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_enunciado\_textF.setForeground(Color.WHITE);

anadirPregunta\_enunciado\_textF.setBounds(52, 11, 718, 25);

anadirPregunta\_enunciado\_panel.add(anadirPregunta\_enunciado\_textF);

anadirPregunta\_enunciado\_textF.setColumns(10);

anadirPregunta\_enunciado\_textF.setBackground(COLOR\_SELECTED);

anadirPregunta\_enunciado\_textF.setBorder(null);

anadirPregunta\_enunciado\_info\_panel = new JPanel();

anadirPregunta\_enunciado\_info\_panel.setBounds(10, 50, 760, 32);

anadirPregunta\_enunciado\_panel.add(anadirPregunta\_enunciado\_info\_panel);

anadirPregunta\_enunciado\_info\_panel.setBackground(COLOR\_ERROR);

anadirPregunta\_enunciado\_info\_panel.setLayout(null);

anadirPregunta\_enunciado\_info\_panel.setVisible(false);

anadirPregunta\_enunciado\_info\_ico = new JLabel("");

anadirPregunta\_enunciado\_info\_ico.setBounds(0, 0, 32, 32);

anadirPregunta\_enunciado\_info\_panel.add(anadirPregunta\_enunciado\_info\_ico);

anadirPregunta\_enunciado\_info\_ico

.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

anadirPregunta\_enunciado\_info\_text = new JLabel("Error");

anadirPregunta\_enunciado\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_enunciado\_info\_text.setBounds(42, 0, 572, 32);

anadirPregunta\_enunciado\_info\_panel.add(anadirPregunta\_enunciado\_info\_text);

// --

anadirPregunta\_respuestaA\_panel = new JPanel();

anadirPregunta\_respuestaA\_panel.setBounds(10, 136, 517, 96);

display\_anadirPregunta.add(anadirPregunta\_respuestaA\_panel);

anadirPregunta\_respuestaA\_panel.setOpaque(false);

anadirPregunta\_respuestaA\_panel.setLayout(null);

anadirPregunta\_respuestaA\_separator = new JSeparator();

anadirPregunta\_respuestaA\_separator.setBounds(52, 37, 450, 6);

anadirPregunta\_respuestaA\_panel.add(anadirPregunta\_respuestaA\_separator);

anadirPregunta\_respuestaA\_icon = new JLabel("");

anadirPregunta\_respuestaA\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/a\_32px.png")));

anadirPregunta\_respuestaA\_icon.setBounds(10, 11, 32, 32);

anadirPregunta\_respuestaA\_panel.add(anadirPregunta\_respuestaA\_icon);

anadirPregunta\_respuestaA\_textF = new JTextField("Respuesta A");

anadirPregunta\_respuestaA\_textF.setHorizontalAlignment(SwingConstants.LEFT);

anadirPregunta\_respuestaA\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaA\_textF.setForeground(Color.WHITE);

anadirPregunta\_respuestaA\_textF.setBounds(52, 11, 450, 25);

anadirPregunta\_respuestaA\_panel.add(anadirPregunta\_respuestaA\_textF);

anadirPregunta\_respuestaA\_textF.setColumns(10);

anadirPregunta\_respuestaA\_textF.setBackground(COLOR\_SELECTED);

anadirPregunta\_respuestaA\_textF.setBorder(null);

anadirPregunta\_respuestaA\_info\_panel = new JPanel();

anadirPregunta\_respuestaA\_info\_panel.setBounds(10, 50, 492, 32);

anadirPregunta\_respuestaA\_panel.add(anadirPregunta\_respuestaA\_info\_panel);

anadirPregunta\_respuestaA\_info\_panel.setBackground(COLOR\_ERROR);

anadirPregunta\_respuestaA\_info\_panel.setLayout(null);

anadirPregunta\_respuestaA\_info\_panel.setVisible(false);

anadirPregunta\_respuestaA\_info\_ico = new JLabel("");

anadirPregunta\_respuestaA\_info\_ico.setBounds(0, 0, 32, 32);

anadirPregunta\_respuestaA\_info\_panel.add(anadirPregunta\_respuestaA\_info\_ico);

anadirPregunta\_respuestaA\_info\_ico

.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

anadirPregunta\_respuestaA\_info\_text = new JLabel("Error");

anadirPregunta\_respuestaA\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaA\_info\_text.setBounds(42, 0, 440, 32);

anadirPregunta\_respuestaA\_info\_panel.add(anadirPregunta\_respuestaA\_info\_text);

// --

anadirPregunta\_respuestaB\_panel = new JPanel();

anadirPregunta\_respuestaB\_panel.setBounds(10, 243, 517, 96);

display\_anadirPregunta.add(anadirPregunta\_respuestaB\_panel);

anadirPregunta\_respuestaB\_panel.setOpaque(false);

anadirPregunta\_respuestaB\_panel.setLayout(null);

anadirPregunta\_respuestaB\_separator = new JSeparator();

anadirPregunta\_respuestaB\_separator.setBounds(52, 37, 450, 6);

anadirPregunta\_respuestaB\_panel.add(anadirPregunta\_respuestaB\_separator);

anadirPregunta\_respuestaB\_icon = new JLabel("");

anadirPregunta\_respuestaB\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/b\_32px.png")));

anadirPregunta\_respuestaB\_icon.setBounds(10, 11, 32, 32);

anadirPregunta\_respuestaB\_panel.add(anadirPregunta\_respuestaB\_icon);

anadirPregunta\_respuestaB\_textF = new JTextField("Respuesta B");

anadirPregunta\_respuestaB\_textF.setHorizontalAlignment(SwingConstants.LEFT);

anadirPregunta\_respuestaB\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaB\_textF.setForeground(Color.WHITE);

anadirPregunta\_respuestaB\_textF.setBounds(52, 11, 450, 25);

anadirPregunta\_respuestaB\_panel.add(anadirPregunta\_respuestaB\_textF);

anadirPregunta\_respuestaB\_textF.setColumns(10);

anadirPregunta\_respuestaB\_textF.setBackground(COLOR\_SELECTED);

anadirPregunta\_respuestaB\_textF.setBorder(null);

anadirPregunta\_respuestaB\_info\_panel = new JPanel();

anadirPregunta\_respuestaB\_info\_panel.setBounds(10, 50, 492, 32);

anadirPregunta\_respuestaB\_panel.add(anadirPregunta\_respuestaB\_info\_panel);

anadirPregunta\_respuestaB\_info\_panel.setBackground(COLOR\_ERROR);

anadirPregunta\_respuestaB\_info\_panel.setLayout(null);

anadirPregunta\_respuestaB\_info\_panel.setVisible(false);

anadirPregunta\_respuestaB\_info\_ico = new JLabel("");

anadirPregunta\_respuestaB\_info\_ico.setBounds(0, 0, 32, 32);

anadirPregunta\_respuestaB\_info\_panel.add(anadirPregunta\_respuestaB\_info\_ico);

anadirPregunta\_respuestaB\_info\_ico

.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

anadirPregunta\_respuestaB\_info\_text = new JLabel("Error");

anadirPregunta\_respuestaB\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaB\_info\_text.setBounds(42, 0, 440, 32);

anadirPregunta\_respuestaB\_info\_panel.add(anadirPregunta\_respuestaB\_info\_text);

// -

anadirPregunta\_respuestaC\_panel = new JPanel();

anadirPregunta\_respuestaC\_panel.setBounds(10, 350, 517, 96);

display\_anadirPregunta.add(anadirPregunta\_respuestaC\_panel);

anadirPregunta\_respuestaC\_panel.setOpaque(false);

anadirPregunta\_respuestaC\_panel.setLayout(null);

anadirPregunta\_respuestaC\_separator = new JSeparator();

anadirPregunta\_respuestaC\_separator.setBounds(52, 37, 450, 6);

anadirPregunta\_respuestaC\_panel.add(anadirPregunta\_respuestaC\_separator);

anadirPregunta\_respuestaC\_icon = new JLabel("");

anadirPregunta\_respuestaC\_icon.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/c\_32px.png")));

anadirPregunta\_respuestaC\_icon.setBounds(10, 11, 32, 32);

anadirPregunta\_respuestaC\_panel.add(anadirPregunta\_respuestaC\_icon);

anadirPregunta\_respuestaC\_textF = new JTextField("Respuesta C");

anadirPregunta\_respuestaC\_textF.setHorizontalAlignment(SwingConstants.LEFT);

anadirPregunta\_respuestaC\_textF.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaC\_textF.setForeground(Color.WHITE);

anadirPregunta\_respuestaC\_textF.setBounds(52, 11, 450, 25);

anadirPregunta\_respuestaC\_panel.add(anadirPregunta\_respuestaC\_textF);

anadirPregunta\_respuestaC\_textF.setColumns(10);

anadirPregunta\_respuestaC\_textF.setBackground(COLOR\_SELECTED);

anadirPregunta\_respuestaC\_textF.setBorder(null);

anadirPregunta\_respuestaC\_info\_panel = new JPanel();

anadirPregunta\_respuestaC\_info\_panel.setBounds(10, 50, 492, 32);

anadirPregunta\_respuestaC\_panel.add(anadirPregunta\_respuestaC\_info\_panel);

anadirPregunta\_respuestaC\_info\_panel.setBackground(COLOR\_ERROR);

anadirPregunta\_respuestaC\_info\_panel.setLayout(null);

anadirPregunta\_respuestaC\_info\_panel.setVisible(false);

anadirPregunta\_respuestaC\_info\_ico = new JLabel("");

anadirPregunta\_respuestaC\_info\_ico.setBounds(0, 0, 32, 32);

anadirPregunta\_respuestaC\_info\_panel.add(anadirPregunta\_respuestaC\_info\_ico);

anadirPregunta\_respuestaC\_info\_ico

.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/error\_black\_32px.png")));

anadirPregunta\_respuestaC\_info\_text = new JLabel("Error");

anadirPregunta\_respuestaC\_info\_text.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaC\_info\_text.setBounds(42, 0, 440, 32);

anadirPregunta\_respuestaC\_info\_panel.add(anadirPregunta\_respuestaC\_info\_text);

anadirPregunta\_respuestaA\_radioBtn = new JRadioButton("Opcion correcta");

anadirPregunta\_respuestaA\_radioBtn.setSelected(true);

anadirPregunta\_respuestaA\_radioBtn.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaA\_radioBtn.setForeground(Color.WHITE);

anadirPregunta\_respuestaA\_radioBtn.setOpaque(false);

anadirPregunta\_respuestaA\_radioBtn.setBounds(533, 159, 254, 50);

display\_anadirPregunta.add(anadirPregunta\_respuestaA\_radioBtn);

anadirPregunta\_respuestaB\_radioBtn = new JRadioButton("Opcion correcta");

anadirPregunta\_respuestaB\_radioBtn.setOpaque(false);

anadirPregunta\_respuestaB\_radioBtn.setForeground(Color.WHITE);

anadirPregunta\_respuestaB\_radioBtn.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaB\_radioBtn.setBounds(536, 259, 254, 50);

display\_anadirPregunta.add(anadirPregunta\_respuestaB\_radioBtn);

anadirPregunta\_respuestaC\_radioBtn = new JRadioButton("Opcion correcta");

anadirPregunta\_respuestaC\_radioBtn.setOpaque(false);

anadirPregunta\_respuestaC\_radioBtn.setForeground(Color.WHITE);

anadirPregunta\_respuestaC\_radioBtn.setFont(new Font("Tahoma", Font.BOLD, 20));

anadirPregunta\_respuestaC\_radioBtn.setBounds(536, 370, 254, 50);

display\_anadirPregunta.add(anadirPregunta\_respuestaC\_radioBtn);

anadirPregunta\_FormBtn = new JFormBtn("pregunta", anadirPregunta\_grupo\_logico);

anadirPregunta\_FormBtn.setBackground(COLOR\_CHECK);

anadirPregunta\_FormBtn.setBounds(280, 457, 258, 62);

display\_anadirPregunta.add(anadirPregunta\_FormBtn);

anadirPregunta\_FormBtn.setLayout(null);

anadirPregunta\_FormBtn.addMouseListener(new FormBtnListener(anadirPregunta\_FormBtn, this));

anadirPregunta\_FormBtn\_text = new JLabel("Añadir");

anadirPregunta\_FormBtn\_text.setBounds(10, 0, 238, 62);

anadirPregunta\_FormBtn.add(anadirPregunta\_FormBtn\_text);

anadirPregunta\_FormBtn\_text.setForeground(Color.WHITE);

anadirPregunta\_FormBtn\_text.setFont(new Font("Tahoma", Font.BOLD, 25));

anadirPregunta\_FormBtn\_text.setHorizontalAlignment(SwingConstants.CENTER);

// --

anadirPregunta\_enunciado\_tfg = new TextFieldGroup(anadirPregunta\_enunciado\_panel,

anadirPregunta\_enunciado\_info\_panel, anadirPregunta\_enunciado\_icon, anadirPregunta\_enunciado\_info\_ico,

anadirPregunta\_enunciado\_info\_text, anadirPregunta\_enunciado\_textF, anadirPregunta\_enunciado\_separator,

"enunciado", anadirPregunta\_grupo\_logico);

anadirPregunta\_respuestaA\_tfg = new TextFieldGroup(anadirPregunta\_respuestaA\_panel,

anadirPregunta\_respuestaA\_info\_panel, anadirPregunta\_respuestaA\_icon,

anadirPregunta\_respuestaA\_info\_ico, anadirPregunta\_respuestaA\_info\_text,

anadirPregunta\_respuestaA\_textF, anadirPregunta\_respuestaA\_separator, "respuesta",

anadirPregunta\_grupo\_logico);

anadirPregunta\_respuestaB\_tfg = new TextFieldGroup(anadirPregunta\_respuestaB\_panel,

anadirPregunta\_respuestaB\_info\_panel, anadirPregunta\_respuestaB\_icon,

anadirPregunta\_respuestaB\_info\_ico, anadirPregunta\_respuestaB\_info\_text,

anadirPregunta\_respuestaB\_textF, anadirPregunta\_respuestaB\_separator, "respuesta",

anadirPregunta\_grupo\_logico);

anadirPregunta\_respuestaC\_tfg = new TextFieldGroup(anadirPregunta\_respuestaC\_panel,

anadirPregunta\_respuestaC\_info\_panel, anadirPregunta\_respuestaC\_icon,

anadirPregunta\_respuestaC\_info\_ico, anadirPregunta\_respuestaC\_info\_text,

anadirPregunta\_respuestaC\_textF, anadirPregunta\_respuestaC\_separator, "respuesta",

anadirPregunta\_grupo\_logico);

anadirPregunta\_enunciado\_textF.addKeyListener(new TextFieldKeyListener(anadirPregunta\_enunciado\_tfg, checker));

anadirPregunta\_enunciado\_textF.addFocusListener(new TextFocusListener(anadirPregunta\_enunciado\_tfg, checker));

anadirPregunta\_respuestaA\_textF

.addKeyListener(new TextFieldKeyListener(anadirPregunta\_respuestaA\_tfg, checker));

anadirPregunta\_respuestaA\_textF.addFocusListener(new TextFocusListener(anadirPregunta\_respuestaA\_tfg, checker));

anadirPregunta\_respuestaB\_textF

.addKeyListener(new TextFieldKeyListener(anadirPregunta\_respuestaB\_tfg, checker));

anadirPregunta\_respuestaB\_textF.addFocusListener(new TextFocusListener(anadirPregunta\_respuestaB\_tfg, checker));

anadirPregunta\_respuestaC\_textF

.addKeyListener(new TextFieldKeyListener(anadirPregunta\_respuestaC\_tfg, checker));

anadirPregunta\_respuestaC\_textF.addFocusListener(new TextFocusListener(anadirPregunta\_respuestaC\_tfg, checker));

anadirPregunta\_respuestaA\_radioBtn.addActionListener(new RadioButtonListener(anadirPregunta\_respuestaA\_radioBtn,

anadirPregunta\_respuestaB\_radioBtn, anadirPregunta\_respuestaC\_radioBtn));

anadirPregunta\_respuestaB\_radioBtn.addActionListener(new RadioButtonListener(anadirPregunta\_respuestaB\_radioBtn,

anadirPregunta\_respuestaC\_radioBtn, anadirPregunta\_respuestaA\_radioBtn));

anadirPregunta\_respuestaC\_radioBtn.addActionListener(new RadioButtonListener(anadirPregunta\_respuestaC\_radioBtn,

anadirPregunta\_respuestaA\_radioBtn, anadirPregunta\_respuestaB\_radioBtn));

}

private void cargarDisplayMostrarConsulta() {

display\_mostrarConsulta = new JDisplay("Mostrar Consulta", "/imagenes/login\_96px.png", displays);

display\_mostrarConsulta.setBounds(0, 0, 800, 552);

display\_panel.add(display\_mostrarConsulta);

display\_mostrarConsulta.setLayout(null);

display\_mostrarConsulta.setOpaque(false);

display\_mostrarConsulta.setVisible(false);

asunto\_panel\_mostrarConsulta = new JPanel();

asunto\_panel\_mostrarConsulta.setLayout(null);

asunto\_panel\_mostrarConsulta.setOpaque(false);

asunto\_panel\_mostrarConsulta.setBounds(10, 11, 780, 52);

display\_mostrarConsulta.add(asunto\_panel\_mostrarConsulta);

asunto\_separator\_mostrarConsulta = new JSeparator();

asunto\_separator\_mostrarConsulta.setBounds(52, 41, 721, 2);

asunto\_panel\_mostrarConsulta.add(asunto\_separator\_mostrarConsulta);

asunto\_ico\_mostrarConsulta = new JLabel("");

asunto\_ico\_mostrarConsulta.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_text\_32px.png")));

asunto\_ico\_mostrarConsulta.setBounds(10, 11, 32, 32);

asunto\_panel\_mostrarConsulta.add(asunto\_ico\_mostrarConsulta);

asunto\_text\_mostrarConsulta = new JLabel("Asunto");

asunto\_text\_mostrarConsulta.setForeground(Color.WHITE);

asunto\_text\_mostrarConsulta.setFont(new Font("Tahoma", Font.BOLD, 20));

asunto\_text\_mostrarConsulta.setBounds(52, 11, 721, 30);

asunto\_panel\_mostrarConsulta.add(asunto\_text\_mostrarConsulta);

descripcion\_mostrarConsulta = new JLabel(

"<html><p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec.</p></html>");

descripcion\_mostrarConsulta.setForeground(Color.WHITE);

descripcion\_mostrarConsulta.setFont(new Font("Tahoma", Font.BOLD, 20));

descripcion\_mostrarConsulta.setBounds(20, 74, 759, 307);

display\_mostrarConsulta.add(descripcion\_mostrarConsulta);

nick\_panel\_mostrarConsulta = new JPanel();

nick\_panel\_mostrarConsulta.setLayout(null);

nick\_panel\_mostrarConsulta.setOpaque(false);

nick\_panel\_mostrarConsulta.setBounds(10, 392, 360, 52);

display\_mostrarConsulta.add(nick\_panel\_mostrarConsulta);

nick\_separator\_mostrarConsulta = new JSeparator();

nick\_separator\_mostrarConsulta.setBounds(52, 41, 300, 2);

nick\_panel\_mostrarConsulta.add(nick\_separator\_mostrarConsulta);

nick\_ico\_mostrarConsulta = new JLabel("");

nick\_ico\_mostrarConsulta.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/user\_text\_32px.png")));

nick\_ico\_mostrarConsulta.setBounds(10, 11, 32, 32);

nick\_panel\_mostrarConsulta.add(nick\_ico\_mostrarConsulta);

nick\_text\_mostrarConsulta = new JLabel("Nick");

nick\_text\_mostrarConsulta.setForeground(Color.WHITE);

nick\_text\_mostrarConsulta.setFont(new Font("Tahoma", Font.BOLD, 20));

nick\_text\_mostrarConsulta.setBounds(52, 11, 300, 30);

nick\_panel\_mostrarConsulta.add(nick\_text\_mostrarConsulta);

fecha\_panel\_mostrarConsulta = new JPanel();

fecha\_panel\_mostrarConsulta.setLayout(null);

fecha\_panel\_mostrarConsulta.setOpaque(false);

fecha\_panel\_mostrarConsulta.setBounds(430, 392, 360, 52);

display\_mostrarConsulta.add(fecha\_panel\_mostrarConsulta);

fecha\_separator\_mostrarConsulta = new JSeparator();

fecha\_separator\_mostrarConsulta.setBounds(52, 41, 300, 2);

fecha\_panel\_mostrarConsulta.add(fecha\_separator\_mostrarConsulta);

fecha\_ico\_mostrarConsulta = new JLabel("");

fecha\_ico\_mostrarConsulta.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/fecha\_32px.png")));

fecha\_ico\_mostrarConsulta.setBounds(10, 11, 32, 32);

fecha\_panel\_mostrarConsulta.add(fecha\_ico\_mostrarConsulta);

fecha\_text\_mostrarConsulta = new JLabel("Fecha");

fecha\_text\_mostrarConsulta.setForeground(Color.WHITE);

fecha\_text\_mostrarConsulta.setFont(new Font("Tahoma", Font.BOLD, 20));

fecha\_text\_mostrarConsulta.setBounds(52, 11, 300, 30);

fecha\_panel\_mostrarConsulta.add(fecha\_text\_mostrarConsulta);

}

private void cargarDisplayIniciarTest() {

display\_iniciarTest = new JDisplay("Iniciar test", "/imagenes/pregunta\_96px.png", displays);

display\_iniciarTest.setBounds(0, 0, 790, 552);

display\_panel.add(display\_iniciarTest);

display\_iniciarTest.setLayout(null);

display\_iniciarTest.setOpaque(false);

display\_iniciarTest.setVisible(false);

iniciarTest\_index\_panel = new JPanel();

iniciarTest\_index\_panel.setOpaque(false);

iniciarTest\_index\_panel.setBounds(10, 11, 780, 530);

display\_iniciarTest.add(iniciarTest\_index\_panel);

iniciarTest\_index\_panel.setLayout(null);

iniciarTest\_index\_panel.setVisible(true);

iniciarTest\_index\_btn\_panel = new JPanel();

iniciarTest\_index\_btn\_panel.setBackground(COLOR\_BACKGROUND);

iniciarTest\_index\_btn\_panel.setBounds(205, 200, 356, 150);

iniciarTest\_index\_panel.add(iniciarTest\_index\_btn\_panel);

iniciarTest\_index\_btn\_panel.setLayout(null);

iniciarTest\_index\_btn\_panel.addMouseListener(new TestBtnListener(this));

iniciarTest\_index\_btn\_text = new JLabel("Iniciar test");

iniciarTest\_index\_btn\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_index\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 34));

iniciarTest\_index\_btn\_text.setForeground(Color.WHITE);

iniciarTest\_index\_btn\_text.setBounds(10, 44, 336, 60);

iniciarTest\_index\_btn\_panel.add(iniciarTest\_index\_btn\_text);

// --

iniciarTest\_end\_panel = new JPanel();

iniciarTest\_end\_panel.setOpaque(false);

iniciarTest\_end\_panel.setBackground(Color.WHITE);

iniciarTest\_end\_panel.setBounds(10, 11, 780, 530);

display\_iniciarTest.add(iniciarTest\_end\_panel);

iniciarTest\_end\_panel.setLayout(null);

iniciarTest\_end\_panel.setVisible(false);

iniciarTest\_end\_btn\_panel = new JPanel();

iniciarTest\_end\_btn\_panel.setBackground(COLOR\_BACKGROUND);

iniciarTest\_end\_btn\_panel.setBounds(205, 350, 356, 150);

iniciarTest\_end\_panel.add(iniciarTest\_end\_btn\_panel);

iniciarTest\_end\_btn\_panel.setLayout(null);

iniciarTest\_end\_btn\_panel.addMouseListener(new TestBtnListener(this));

iniciarTest\_end\_btn\_text = new JLabel("Finalizar test");

iniciarTest\_end\_btn\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_end\_btn\_text.setFont(new Font("Tahoma", Font.BOLD, 34));

iniciarTest\_end\_btn\_text.setForeground(Color.WHITE);

iniciarTest\_end\_btn\_text.setBounds(10, 44, 336, 60);

iniciarTest\_end\_btn\_panel.add(iniciarTest\_end\_btn\_text);

iniciarTest\_aciertos\_numero = new JLabel("30");

iniciarTest\_aciertos\_numero.setForeground(COLOR\_CHECK);

iniciarTest\_aciertos\_numero.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_aciertos\_numero.setFont(new Font("Arial", Font.BOLD, 96));

iniciarTest\_aciertos\_numero.setBounds(193, 140, 114, 104);

iniciarTest\_end\_panel.add(iniciarTest\_aciertos\_numero);

iniciarTest\_aciertos\_text = new JLabel("Aciertos");

iniciarTest\_aciertos\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_aciertos\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

iniciarTest\_aciertos\_text.setForeground(COLOR\_CHECK);

iniciarTest\_aciertos\_text.setBounds(157, 238, 192, 52);

iniciarTest\_end\_panel.add(iniciarTest\_aciertos\_text);

iniciarTest\_fallos\_numero = new JLabel("30");

iniciarTest\_fallos\_numero.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_fallos\_numero.setForeground(COLOR\_DANGER);

iniciarTest\_fallos\_numero.setFont(new Font("Arial", Font.BOLD, 96));

iniciarTest\_fallos\_numero.setBounds(469, 140, 114, 104);

iniciarTest\_end\_panel.add(iniciarTest\_fallos\_numero);

iniciarTest\_fallos\_text = new JLabel("Fallos");

iniciarTest\_fallos\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_fallos\_text.setForeground(COLOR\_DANGER);

iniciarTest\_fallos\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

iniciarTest\_fallos\_text.setBounds(433, 238, 192, 52);

iniciarTest\_end\_panel.add(iniciarTest\_fallos\_text);

iniciarTest\_resultado\_text = new JLabel("Aprobado");

iniciarTest\_resultado\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_resultado\_text.setForeground(Color.GREEN);

iniciarTest\_resultado\_text.setFont(new Font("Tahoma", Font.BOLD, 62));

iniciarTest\_resultado\_text.setBounds(57, 11, 659, 91);

iniciarTest\_end\_panel.add(iniciarTest\_resultado\_text);

// --

iniciarTest\_pregunta\_panel = new JPanel();

iniciarTest\_pregunta\_panel.setOpaque(false);

iniciarTest\_pregunta\_panel.setBounds(10, 11, 780, 530);

display\_iniciarTest.add(iniciarTest\_pregunta\_panel);

iniciarTest\_pregunta\_panel.setLayout(null);

iniciarTest\_pregunta\_panel.setVisible(false);

iniciarTest\_pregunta\_enunciado = new JLabel("Enunciado");

iniciarTest\_pregunta\_enunciado.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_pregunta\_enunciado.setFont(new Font("Tahoma", Font.BOLD, 18));

iniciarTest\_pregunta\_enunciado.setForeground(Color.WHITE);

iniciarTest\_pregunta\_enunciado.setBounds(10, 43, 760, 100);

iniciarTest\_pregunta\_panel.add(iniciarTest\_pregunta\_enunciado);

iniciarTest\_pregunta\_separator = new JSeparator();

iniciarTest\_pregunta\_separator.setBounds(10, 157, 760, 2);

iniciarTest\_pregunta\_panel.add(iniciarTest\_pregunta\_separator);

iniciarTest\_pregunta\_contadorPregunta = new JLabel("Pregunta n\u00BA: 0/30");

iniciarTest\_pregunta\_contadorPregunta.setFont(new Font("Tahoma", Font.BOLD, 22));

iniciarTest\_pregunta\_contadorPregunta.setForeground(Color.WHITE);

iniciarTest\_pregunta\_contadorPregunta.setBounds(10, 0, 274, 32);

iniciarTest\_pregunta\_panel.add(iniciarTest\_pregunta\_contadorPregunta);

iniciarTest\_respuestaA\_panel = new JRespuestaBtn('a');

iniciarTest\_respuestaA\_panel.setBackground(COLOR\_BACKGROUND);

iniciarTest\_respuestaA\_panel.setBounds(10, 170, 760, 109);

iniciarTest\_pregunta\_panel.add(iniciarTest\_respuestaA\_panel);

iniciarTest\_respuestaA\_panel.setLayout(null);

iniciarTest\_respuestaA\_ico = new JLabel("");

iniciarTest\_respuestaA\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/a\_64px.png")));

iniciarTest\_respuestaA\_ico.setBounds(0, 22, 64, 64);

iniciarTest\_respuestaA\_panel.add(iniciarTest\_respuestaA\_ico);

iniciarTest\_respuestaA\_text = new JLabel("Respuesta A");

iniciarTest\_respuestaA\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_respuestaA\_text.setForeground(Color.WHITE);

iniciarTest\_respuestaA\_text.setFont(new Font("Tahoma", Font.BOLD, 18));

iniciarTest\_respuestaA\_text.setBounds(74, 11, 676, 87);

iniciarTest\_respuestaA\_panel.add(iniciarTest\_respuestaA\_text);

iniciarTest\_respuestaB\_panel = new JRespuestaBtn('b');

iniciarTest\_respuestaB\_panel.setLayout(null);

iniciarTest\_respuestaB\_panel.setBackground(new Color(54, 33, 89));

iniciarTest\_respuestaB\_panel.setBounds(10, 290, 760, 109);

iniciarTest\_pregunta\_panel.add(iniciarTest\_respuestaB\_panel);

iniciarTest\_respuestaB\_ico = new JLabel("");

iniciarTest\_respuestaB\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/b\_64px.png")));

iniciarTest\_respuestaB\_ico.setBounds(0, 22, 64, 64);

iniciarTest\_respuestaB\_panel.add(iniciarTest\_respuestaB\_ico);

iniciarTest\_respuestaB\_text = new JLabel("Respuesta B");

iniciarTest\_respuestaB\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_respuestaB\_text.setForeground(Color.WHITE);

iniciarTest\_respuestaB\_text.setFont(new Font("Tahoma", Font.BOLD, 18));

iniciarTest\_respuestaB\_text.setBounds(74, 11, 676, 87);

iniciarTest\_respuestaB\_panel.add(iniciarTest\_respuestaB\_text);

iniciarTest\_respuestaC\_panel = new JRespuestaBtn('c');

iniciarTest\_respuestaC\_panel.setLayout(null);

iniciarTest\_respuestaC\_panel.setBackground(new Color(54, 33, 89));

iniciarTest\_respuestaC\_panel.setBounds(10, 410, 760, 109);

iniciarTest\_pregunta\_panel.add(iniciarTest\_respuestaC\_panel);

iniciarTest\_respuestaC\_ico = new JLabel("");

iniciarTest\_respuestaC\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/c\_64px.png")));

iniciarTest\_respuestaC\_ico.setBounds(0, 22, 64, 64);

iniciarTest\_respuestaC\_panel.add(iniciarTest\_respuestaC\_ico);

iniciarTest\_respuestaC\_text = new JLabel("Respuesta C");

iniciarTest\_respuestaC\_text.setHorizontalAlignment(SwingConstants.CENTER);

iniciarTest\_respuestaC\_text.setForeground(Color.WHITE);

iniciarTest\_respuestaC\_text.setFont(new Font("Tahoma", Font.BOLD, 18));

iniciarTest\_respuestaC\_text.setBounds(74, 11, 676, 87);

iniciarTest\_respuestaC\_panel.add(iniciarTest\_respuestaC\_text);

iniciarTest\_respuestaA\_panel.addMouseListener(new RespuestaBtnListener(this, iniciarTest\_respuestaA\_panel));

iniciarTest\_respuestaB\_panel.addMouseListener(new RespuestaBtnListener(this, iniciarTest\_respuestaB\_panel));

iniciarTest\_respuestaC\_panel.addMouseListener(new RespuestaBtnListener(this, iniciarTest\_respuestaC\_panel));

}

private void cargarDisplayMostrarEstadisticas() {

display\_mostrarEstadisticas = new JDisplay("Estadisticas", "/imagenes/info\_96px.png", displays);

display\_mostrarEstadisticas.setBounds(0, 0, 800, 552);

display\_panel.add(display\_mostrarEstadisticas);

display\_mostrarEstadisticas.setLayout(null);

display\_mostrarEstadisticas.setOpaque(false);

display\_mostrarEstadisticas.setVisible(false);

mostrarEstadisticas\_aprobados\_numero = new JLabel("30");

mostrarEstadisticas\_aprobados\_numero.setForeground(COLOR\_CHECK);

mostrarEstadisticas\_aprobados\_numero.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_aprobados\_numero.setFont(new Font("Arial", Font.BOLD, 96));

mostrarEstadisticas\_aprobados\_numero.setBounds(54, 312, 114, 104);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_aprobados\_numero);

mostrarEstadisticas\_aprobados\_text = new JLabel("Aprobados");

mostrarEstadisticas\_aprobados\_text.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_aprobados\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

mostrarEstadisticas\_aprobados\_text.setForeground(COLOR\_CHECK);

mostrarEstadisticas\_aprobados\_text.setBounds(10, 410, 210, 52);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_aprobados\_text);

mostrarEstadisticas\_suspendos\_numero = new JLabel("30");

mostrarEstadisticas\_suspendos\_numero.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_suspendos\_numero.setForeground(COLOR\_DANGER);

mostrarEstadisticas\_suspendos\_numero.setFont(new Font("Arial", Font.BOLD, 96));

mostrarEstadisticas\_suspendos\_numero.setBounds(340, 312, 114, 104);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_suspendos\_numero);

mostrarEstadisticas\_suspendos\_text = new JLabel("Suspensos");

mostrarEstadisticas\_suspendos\_text.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_suspendos\_text.setForeground(COLOR\_DANGER);

mostrarEstadisticas\_suspendos\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

mostrarEstadisticas\_suspendos\_text.setBounds(290, 406, 216, 52);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_suspendos\_text);

mostrarEstadisticas\_media\_numero = new JLabel("30");

mostrarEstadisticas\_media\_numero.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_media\_numero.setForeground(COLOR\_ERROR);

mostrarEstadisticas\_media\_numero.setFont(new Font("Arial", Font.BOLD, 96));

mostrarEstadisticas\_media\_numero.setBounds(561, 312, 229, 104);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_media\_numero);

mostrarEstadisticas\_media\_text = new JLabel("Media fallos");

mostrarEstadisticas\_media\_text.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_media\_text.setForeground(COLOR\_ERROR);

mostrarEstadisticas\_media\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

mostrarEstadisticas\_media\_text.setBounds(561, 410, 229, 52);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_media\_text);

mostrarEstadisticas\_test\_numero = new JLabel("30");

mostrarEstadisticas\_test\_numero.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_test\_numero.setForeground(Color.WHITE);

mostrarEstadisticas\_test\_numero.setFont(new Font("Arial", Font.BOLD, 96));

mostrarEstadisticas\_test\_numero.setBounds(41, 159, 114, 104);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_test\_numero);

mostrarEstadisticas\_test\_text = new JLabel("Test realizados");

mostrarEstadisticas\_test\_text.setHorizontalAlignment(SwingConstants.CENTER);

mostrarEstadisticas\_test\_text.setForeground(Color.WHITE);

mostrarEstadisticas\_test\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

mostrarEstadisticas\_test\_text.setBounds(145, 192, 333, 52);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_test\_text);

mostrarEstadisticas\_separator1 = new JSeparator();

mostrarEstadisticas\_separator1.setBounds(10, 284, 780, 2);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_separator1);

mostrarEstadisticas\_separator2 = new JSeparator();

mostrarEstadisticas\_separator2.setBounds(10, 141, 780, 2);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_separator2);

mostrarEstadisticas\_nick\_text = new JLabel("Nick");

mostrarEstadisticas\_nick\_text.setForeground(Color.WHITE);

mostrarEstadisticas\_nick\_text.setFont(new Font("Tahoma", Font.BOLD, 38));

mostrarEstadisticas\_nick\_text.setBounds(10, 48, 780, 52);

display\_mostrarEstadisticas.add(mostrarEstadisticas\_nick\_text);

}

// NavPanel inivitado

private void cargarNavPanelInvitado() {

// NavPanel invitado

navPanel\_invitado = new JPanel();

navPanel\_invitado.setBounds(0, 0, 300, 464);

nav\_panel.add(navPanel\_invitado);

navPanel\_invitado.setOpaque(false);

navPanel\_invitado.setLayout(null);

opciones\_invitado.setNav\_panel(navPanel\_invitado);

// Nav options

// option login

opt\_login = new JOption(display\_login, "Iniciar sesi\u00F3n", "/imagenes/login\_32px.png", opciones\_invitado);

opt\_login.setBounds(0, 11, 300, 48);

navPanel\_invitado.add(opt\_login);

opt\_login.setBackground(COLOR\_BACKGROUND);

opt\_login.setLayout(null);

opt\_login.addMouseListener(new OptionListener(opt\_login));

opt\_login\_ico = new JLabel("");

opt\_login\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_login.getIco\_path())));

opt\_login\_ico.setBounds(10, 0, 48, 48);

opt\_login.add(opt\_login\_ico);

opt\_login\_text = new JLabel(opt\_login.getText());

opt\_login\_text.setForeground(Color.WHITE);

opt\_login\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_login\_text.setBounds(68, 11, 222, 26);

opt\_login.add(opt\_login\_text);

// option registro

opt\_reg = new JOption(display\_reg, "Crear una cuenta", "/imagenes/reg\_32px.png", opciones\_invitado);

opt\_reg.setLayout(null);

opt\_reg.setBackground(new Color(54, 33, 89));

opt\_reg.setBounds(0, 59, 300, 48);

navPanel\_invitado.add(opt\_reg);

opt\_reg.addMouseListener(new OptionListener(opt\_reg));

opt\_reg\_ico = new JLabel("");

opt\_reg\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_reg.getIco\_path())));

opt\_reg\_ico.setBounds(10, 0, 48, 48);

opt\_reg.add(opt\_reg\_ico);

opt\_reg\_text = new JLabel("Crear una cuenta");

opt\_reg\_text.setForeground(Color.WHITE);

opt\_reg\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_reg\_text.setBounds(68, 11, 222, 26);

opt\_reg.add(opt\_reg\_text);

// option info

opt\_info = new JOption(display\_info, "Informaci\\u00F3n", "/imagenes/info\_32px.png", opciones\_invitado);

opt\_info.setLayout(null);

opt\_info.setBackground(new Color(54, 33, 89));

opt\_info.setBounds(0, 107, 300, 48);

navPanel\_invitado.add(opt\_info);

opt\_info.addMouseListener(new OptionListener(opt\_info));

opt\_info\_ico = new JLabel("");

opt\_info\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_info.getIco\_path())));

opt\_info\_ico.setBounds(10, 0, 48, 48);

opt\_info.add(opt\_info\_ico);

opt\_info\_text = new JLabel("Informaci\u00F3n");

opt\_info\_text.setForeground(Color.WHITE);

opt\_info\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_info\_text.setBounds(68, 11, 222, 26);

opt\_info.add(opt\_info\_text);

// option soporte

opt\_soporte = new JOption(display\_soporte, "Soporte", "/imagenes/soporte\_32px.png", opciones\_invitado);

opt\_soporte.setLayout(null);

opt\_soporte.setBackground(new Color(54, 33, 89));

opt\_soporte.setBounds(0, 155, 300, 48);

navPanel\_invitado.add(opt\_soporte);

opt\_soporte.addMouseListener(new OptionListener(opt\_soporte));

opt\_soporte\_ico = new JLabel("");

opt\_soporte\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_soporte.getIco\_path())));

opt\_soporte\_ico.setBounds(10, 0, 48, 48);

opt\_soporte.add(opt\_soporte\_ico);

opt\_soporte\_text = new JLabel("Soporte");

opt\_soporte\_text.setForeground(Color.WHITE);

opt\_soporte\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_soporte\_text.setBounds(68, 11, 222, 26);

opt\_soporte.add(opt\_soporte\_text);

}

// NavPanel Alumno

private void cargarNavPanelAlumno() {

// NavPanel alumno

navPanel\_alumno = new JPanel();

navPanel\_alumno.setBounds(0, 0, 300, 464);

nav\_panel.add(navPanel\_alumno);

navPanel\_alumno.setOpaque(false);

navPanel\_alumno.setLayout(null);

navPanel\_alumno.setVisible(false);

opciones\_alumno.setNav\_panel(navPanel\_alumno);

// Nav options

// option iniciar test

opt\_iniciarTest = new JOption(display\_iniciarTest, "Iniciar test", "/imagenes/test\_32px.png", opciones\_alumno);

opt\_iniciarTest.setBounds(0, 11, 300, 48);

navPanel\_alumno.add(opt\_iniciarTest);

opt\_iniciarTest.setBackground(COLOR\_BACKGROUND);

opt\_iniciarTest.setLayout(null);

opt\_iniciarTest.addMouseListener(new OptionListener(opt\_iniciarTest));

opt\_iniciarTest\_ico = new JLabel("");

opt\_iniciarTest\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_iniciarTest.getIco\_path())));

opt\_iniciarTest\_ico.setBounds(10, 0, 48, 48);

opt\_iniciarTest.add(opt\_iniciarTest\_ico);

opt\_iniciarTest\_text = new JLabel(opt\_iniciarTest.getText());

opt\_iniciarTest\_text.setForeground(Color.WHITE);

opt\_iniciarTest\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_iniciarTest\_text.setBounds(68, 11, 222, 26);

opt\_iniciarTest.add(opt\_iniciarTest\_text);

// option estadisticas alumno

opt\_estadisticas\_alumno = new JOption(display\_mostrarEstadisticas, "Estadisticas",

"/imagenes/estadisticas\_32px.png", opciones\_alumno);

opt\_estadisticas\_alumno.setLayout(null);

opt\_estadisticas\_alumno.setBackground(new Color(54, 33, 89));

opt\_estadisticas\_alumno.setBounds(0, 59, 300, 48);

navPanel\_alumno.add(opt\_estadisticas\_alumno);

opt\_estadisticas\_alumno.addMouseListener(new OptionListener(opt\_estadisticas\_alumno));

opt\_estadisticas\_alumno\_ico = new JLabel("");

opt\_estadisticas\_alumno\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_estadisticas\_alumno.getIco\_path())));

opt\_estadisticas\_alumno\_ico.setBounds(10, 0, 48, 48);

opt\_estadisticas\_alumno.add(opt\_estadisticas\_alumno\_ico);

opt\_estadisticas\_alumno\_text = new JLabel("Estadisticas");

opt\_estadisticas\_alumno\_text.setForeground(Color.WHITE);

opt\_estadisticas\_alumno\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_estadisticas\_alumno\_text.setBounds(68, 11, 222, 26);

opt\_estadisticas\_alumno.add(opt\_estadisticas\_alumno\_text);

// option soporte alumno

opt\_soporte\_alumno = new JOption(display\_soporte, "Soporte", "/imagenes/soporte\_32px.png", opciones\_alumno);

opt\_soporte\_alumno.setLayout(null);

opt\_soporte\_alumno.setBackground(new Color(54, 33, 89));

opt\_soporte\_alumno.setBounds(0, 107, 300, 48);

navPanel\_alumno.add(opt\_soporte\_alumno);

opt\_soporte\_alumno.addMouseListener(new OptionListener(opt\_soporte\_alumno));

opt\_soporte\_alumno\_ico = new JLabel("");

opt\_soporte\_alumno\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_soporte\_alumno.getIco\_path())));

opt\_soporte\_alumno\_ico.setBounds(10, 0, 48, 48);

opt\_soporte\_alumno.add(opt\_soporte\_alumno\_ico);

opt\_soporte\_alumno\_text = new JLabel("Soporte");

opt\_soporte\_alumno\_text.setForeground(Color.WHITE);

opt\_soporte\_alumno\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_soporte\_alumno\_text.setBounds(68, 11, 222, 26);

opt\_soporte\_alumno.add(opt\_soporte\_alumno\_text);

// option cerrar sesion alumno

opt\_cerrarSesion\_alumno = new JOption(display\_cerrarSesion, "Cerrar sesion", "/imagenes/cerrar\_32px.png",

opciones\_alumno);

opt\_cerrarSesion\_alumno.setLayout(null);

opt\_cerrarSesion\_alumno.setBackground(new Color(54, 33, 89));

opt\_cerrarSesion\_alumno.setBounds(0, 155, 300, 48);

navPanel\_alumno.add(opt\_cerrarSesion\_alumno);

opt\_cerrarSesion\_alumno.addMouseListener(new OptionListener(opt\_cerrarSesion\_alumno));

opt\_cerrarSesion\_alumno\_ico = new JLabel("");

opt\_cerrarSesion\_alumno\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_cerrarSesion\_alumno.getIco\_path())));

opt\_cerrarSesion\_alumno\_ico.setBounds(10, 0, 48, 48);

opt\_cerrarSesion\_alumno.add(opt\_cerrarSesion\_alumno\_ico);

opt\_cerrarSesion\_alumno\_text = new JLabel("Cerrar sesion");

opt\_cerrarSesion\_alumno\_text.setForeground(Color.WHITE);

opt\_cerrarSesion\_alumno\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_cerrarSesion\_alumno\_text.setBounds(68, 11, 222, 26);

opt\_cerrarSesion\_alumno.add(opt\_cerrarSesion\_alumno\_text);

}

// NavPanel Profesor

private void cargarNavPanelProfesor() {

// NavPanel alumno

navPanel\_profesor = new JPanel();

navPanel\_profesor.setBounds(0, 0, 300, 464);

nav\_panel.add(navPanel\_profesor);

navPanel\_profesor.setOpaque(false);

navPanel\_profesor.setLayout(null);

navPanel\_profesor.setVisible(false);

opciones\_profesor.setNav\_panel(navPanel\_profesor);

// Nav options

// option estadisticas profesor

opt\_estadisticas\_profesor = new JOption(display\_verUsuarios, "Estadisticas", "/imagenes/estadisticas\_32px.png",

opciones\_profesor);

opt\_estadisticas\_profesor.setBounds(0, 11, 300, 48);

navPanel\_profesor.add(opt\_estadisticas\_profesor);

opt\_estadisticas\_profesor.setBackground(COLOR\_BACKGROUND);

opt\_estadisticas\_profesor.setLayout(null);

opt\_estadisticas\_profesor.addMouseListener(new OptionListener(opt\_estadisticas\_profesor));

opt\_estadisticas\_profesor\_ico = new JLabel("");

opt\_estadisticas\_profesor\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_estadisticas\_profesor.getIco\_path())));

opt\_estadisticas\_profesor\_ico.setBounds(10, 0, 48, 48);

opt\_estadisticas\_profesor.add(opt\_estadisticas\_profesor\_ico);

opt\_iniciarTest\_text = new JLabel(opt\_estadisticas\_profesor.getText());

opt\_iniciarTest\_text.setForeground(Color.WHITE);

opt\_iniciarTest\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_iniciarTest\_text.setBounds(68, 11, 222, 26);

opt\_estadisticas\_profesor.add(opt\_iniciarTest\_text);

// option entrar como - profesor

opt\_entrarComo\_profesor = new JOption(display\_entrarComo, "Entrar como", "/imagenes/cambiar\_32px.png",

opciones\_profesor);

opt\_entrarComo\_profesor.setLayout(null);

opt\_entrarComo\_profesor.setBackground(new Color(54, 33, 89));

opt\_entrarComo\_profesor.setBounds(0, 59, 300, 48);

navPanel\_profesor.add(opt\_entrarComo\_profesor);

opt\_entrarComo\_profesor.addMouseListener(new OptionListener(opt\_entrarComo\_profesor));

opt\_entrarComo\_profesor\_ico = new JLabel("");

opt\_entrarComo\_profesor\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_entrarComo\_profesor.getIco\_path())));

opt\_entrarComo\_profesor\_ico.setBounds(10, 0, 48, 48);

opt\_entrarComo\_profesor.add(opt\_entrarComo\_profesor\_ico);

opt\_entrarComo\_profesor\_text = new JLabel("Entrar como");

opt\_entrarComo\_profesor\_text.setForeground(Color.WHITE);

opt\_entrarComo\_profesor\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_entrarComo\_profesor\_text.setBounds(68, 11, 222, 26);

opt\_entrarComo\_profesor.add(opt\_entrarComo\_profesor\_text);

// option soporte profesor

opt\_soporte\_profesor = new JOption(display\_soporte, "Soporte", "/imagenes/soporte\_32px.png", opciones\_profesor);

opt\_soporte\_profesor.setLayout(null);

opt\_soporte\_profesor.setBackground(new Color(54, 33, 89));

opt\_soporte\_profesor.setBounds(0, 107, 300, 48);

navPanel\_profesor.add(opt\_soporte\_profesor);

opt\_soporte\_profesor.addMouseListener(new OptionListener(opt\_soporte\_profesor));

opt\_soporte\_profesor\_ico = new JLabel("");

opt\_soporte\_profesor\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_soporte\_profesor.getIco\_path())));

opt\_soporte\_profesor\_ico.setBounds(10, 0, 48, 48);

opt\_soporte\_profesor.add(opt\_soporte\_profesor\_ico);

opt\_soporte\_profesor\_text = new JLabel("Soporte");

opt\_soporte\_profesor\_text.setForeground(Color.WHITE);

opt\_soporte\_profesor\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_soporte\_profesor\_text.setBounds(68, 11, 222, 26);

opt\_soporte\_profesor.add(opt\_soporte\_profesor\_text);

// option cerrar sesion profesor

opt\_cerrarSesion\_profesor = new JOption(display\_cerrarSesion, "Cerrar sesion", "/imagenes/cerrar\_32px.png",

opciones\_profesor);

opt\_cerrarSesion\_profesor.setLayout(null);

opt\_cerrarSesion\_profesor.setBackground(new Color(54, 33, 89));

opt\_cerrarSesion\_profesor.setBounds(0, 155, 300, 48);

navPanel\_profesor.add(opt\_cerrarSesion\_profesor);

opt\_cerrarSesion\_profesor.addMouseListener(new OptionListener(opt\_cerrarSesion\_profesor));

opt\_cerrarSesion\_profesor\_ico = new JLabel("");

opt\_cerrarSesion\_profesor\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_cerrarSesion\_profesor.getIco\_path())));

opt\_cerrarSesion\_profesor\_ico.setBounds(10, 0, 48, 48);

opt\_cerrarSesion\_profesor.add(opt\_cerrarSesion\_profesor\_ico);

opt\_cerrarSesion\_profesor\_text = new JLabel("Cerrar sesion");

opt\_cerrarSesion\_profesor\_text.setForeground(Color.WHITE);

opt\_cerrarSesion\_profesor\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_cerrarSesion\_profesor\_text.setBounds(68, 11, 222, 26);

opt\_cerrarSesion\_profesor.add(opt\_cerrarSesion\_profesor\_text);

}

// NavPanel Administrador

private void cargarNavPanelAdministrador() {

// NavPanel administrador

navPanel\_administrador = new JPanel();

navPanel\_administrador.setBounds(0, 0, 300, 464);

nav\_panel.add(navPanel\_administrador);

navPanel\_administrador.setOpaque(false);

navPanel\_administrador.setLayout(null);

navPanel\_administrador.setVisible(false);

opciones\_administrador.setNav\_panel(navPanel\_administrador);

// Nav options

// option ver usuarios

opt\_verUsuarios = new JOption(display\_verUsuarios, "Buscar usuario", "/imagenes/buscar\_32px.png",

opciones\_administrador);

opt\_verUsuarios.setBounds(0, 11, 300, 48);

navPanel\_administrador.add(opt\_verUsuarios);

opt\_verUsuarios.setBackground(COLOR\_BACKGROUND);

opt\_verUsuarios.setLayout(null);

opt\_verUsuarios.addMouseListener(new OptionListener(opt\_verUsuarios));

opt\_verUsuarios\_ico = new JLabel("");

opt\_verUsuarios\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_verUsuarios.getIco\_path())));

opt\_verUsuarios\_ico.setBounds(10, 0, 48, 48);

opt\_verUsuarios.add(opt\_verUsuarios\_ico);

opt\_verUsuarios\_text = new JLabel(opt\_verUsuarios.getText());

opt\_verUsuarios\_text.setForeground(Color.WHITE);

opt\_verUsuarios\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_verUsuarios\_text.setBounds(68, 11, 222, 26);

opt\_verUsuarios.add(opt\_verUsuarios\_text);

// option soporte administrador

opt\_soporte\_administrador = new JOption(display\_soporte, "Soporte", "/imagenes/soporte\_32px.png",

opciones\_administrador);

opt\_soporte\_administrador.setLayout(null);

opt\_soporte\_administrador.setBackground(new Color(54, 33, 89));

opt\_soporte\_administrador.setBounds(0, 59, 300, 48);

navPanel\_administrador.add(opt\_soporte\_administrador);

opt\_soporte\_administrador.addMouseListener(new OptionListener(opt\_soporte\_administrador));

opt\_soporte\_administrador\_ico = new JLabel("");

opt\_soporte\_administrador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_soporte\_administrador.getIco\_path())));

opt\_soporte\_administrador\_ico.setBounds(10, 0, 48, 48);

opt\_soporte\_administrador.add(opt\_soporte\_administrador\_ico);

opt\_soporte\_administrador\_text = new JLabel("Soporte");

opt\_soporte\_administrador\_text.setForeground(Color.WHITE);

opt\_soporte\_administrador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_soporte\_administrador\_text.setBounds(68, 11, 222, 26);

opt\_soporte\_administrador.add(opt\_soporte\_administrador\_text);

// option entrar como (administrador)

opt\_entrarComo\_administrador = new JOption(display\_entrarComo, "Entrar como", "/imagenes/cambiar\_32px.png",

opciones\_administrador);

opt\_entrarComo\_administrador.setLayout(null);

opt\_entrarComo\_administrador.setBackground(new Color(54, 33, 89));

opt\_entrarComo\_administrador.setBounds(0, 107, 300, 48);

navPanel\_administrador.add(opt\_entrarComo\_administrador);

opt\_entrarComo\_administrador.addMouseListener(new OptionListener(opt\_entrarComo\_administrador));

opt\_entrarComo\_administrador\_ico = new JLabel("");

opt\_entrarComo\_administrador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_entrarComo\_administrador.getIco\_path())));

opt\_entrarComo\_administrador\_ico.setBounds(10, 0, 48, 48);

opt\_entrarComo\_administrador.add(opt\_entrarComo\_administrador\_ico);

opt\_entrarComo\_administrador\_text = new JLabel("Entrar como");

opt\_entrarComo\_administrador\_text.setForeground(Color.WHITE);

opt\_entrarComo\_administrador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_entrarComo\_administrador\_text.setBounds(68, 11, 222, 26);

opt\_entrarComo\_administrador.add(opt\_entrarComo\_administrador\_text);

// option cerrar sesion

opt\_cerrarSesion\_administrador = new JOption(display\_cerrarSesion, "Cerrar sesion", "/imagenes/cerrar\_32px.png",

opciones\_administrador);

opt\_cerrarSesion\_administrador.setLayout(null);

opt\_cerrarSesion\_administrador.setBackground(new Color(54, 33, 89));

opt\_cerrarSesion\_administrador.setBounds(0, 155, 300, 48);

navPanel\_administrador.add(opt\_cerrarSesion\_administrador);

opt\_cerrarSesion\_administrador.addMouseListener(new OptionListener(opt\_cerrarSesion\_administrador));

opt\_cerrarSesion\_administrador\_ico = new JLabel("");

opt\_cerrarSesion\_administrador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_cerrarSesion\_administrador.getIco\_path())));

opt\_cerrarSesion\_administrador\_ico.setBounds(10, 0, 48, 48);

opt\_cerrarSesion\_administrador.add(opt\_cerrarSesion\_administrador\_ico);

opt\_cerrarSesion\_administrador\_text = new JLabel("Cerrar sesion");

opt\_cerrarSesion\_administrador\_text.setForeground(Color.WHITE);

opt\_cerrarSesion\_administrador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_cerrarSesion\_administrador\_text.setBounds(68, 11, 222, 26);

opt\_cerrarSesion\_administrador.add(opt\_cerrarSesion\_administrador\_text);

}

// NavPanel Desarrollador

private void cargarNavPanelDesarrollador() {

// NavPanel desarollador

navPanel\_desarrollador = new JPanel();

navPanel\_desarrollador.setBounds(0, 0, 300, 464);

nav\_panel.add(navPanel\_desarrollador);

navPanel\_desarrollador.setOpaque(false);

navPanel\_desarrollador.setLayout(null);

navPanel\_desarrollador.setVisible(false);

opciones\_desarrollador.setNav\_panel(navPanel\_desarrollador);

// Nav options

// option ver usuarios desarollador

opt\_verUsuarios\_desarrollador = new JOption(display\_verConsultas, "Ver consultas", "/imagenes/buscar\_32px.png",

opciones\_desarrollador);

opt\_verUsuarios\_desarrollador.setBounds(0, 11, 300, 48);

navPanel\_desarrollador.add(opt\_verUsuarios\_desarrollador);

opt\_verUsuarios\_desarrollador.setBackground(COLOR\_BACKGROUND);

opt\_verUsuarios\_desarrollador.setLayout(null);

opt\_verUsuarios\_desarrollador.addMouseListener(new OptionListener(opt\_verUsuarios\_desarrollador));

opt\_verUsuarios\_desarrollador\_ico = new JLabel("");

opt\_verUsuarios\_desarrollador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_verUsuarios\_desarrollador.getIco\_path())));

opt\_verUsuarios\_desarrollador\_ico.setBounds(10, 0, 48, 48);

opt\_verUsuarios\_desarrollador.add(opt\_verUsuarios\_desarrollador\_ico);

opt\_verUsuarios\_desarrollador\_text = new JLabel(opt\_verUsuarios\_desarrollador.getText());

opt\_verUsuarios\_desarrollador\_text.setForeground(Color.WHITE);

opt\_verUsuarios\_desarrollador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_verUsuarios\_desarrollador\_text.setBounds(68, 11, 222, 26);

opt\_verUsuarios\_desarrollador.add(opt\_verUsuarios\_desarrollador\_text);

// option entrar como - desarollador

opt\_entrarComo\_desarrollador = new JOption(display\_entrarComo, "Entrar como", "/imagenes/cambiar\_32px.png",

opciones\_desarrollador);

opt\_entrarComo\_desarrollador.setLayout(null);

opt\_entrarComo\_desarrollador.setBackground(new Color(54, 33, 89));

opt\_entrarComo\_desarrollador.setBounds(0, 59, 300, 48);

navPanel\_desarrollador.add(opt\_entrarComo\_desarrollador);

opt\_entrarComo\_desarrollador.addMouseListener(new OptionListener(opt\_entrarComo\_desarrollador));

opt\_entrarComo\_desarrollador\_ico = new JLabel("");

opt\_entrarComo\_desarrollador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_entrarComo\_desarrollador.getIco\_path())));

opt\_entrarComo\_desarrollador\_ico.setBounds(10, 0, 48, 48);

opt\_entrarComo\_desarrollador.add(opt\_entrarComo\_desarrollador\_ico);

opt\_entrarComo\_desarrollador\_text = new JLabel("Entrar como");

opt\_entrarComo\_desarrollador\_text.setForeground(Color.WHITE);

opt\_entrarComo\_desarrollador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_entrarComo\_desarrollador\_text.setBounds(68, 11, 222, 26);

opt\_entrarComo\_desarrollador.add(opt\_entrarComo\_desarrollador\_text);

// option añadir pregunta

opt\_anadirPregunta = new JOption(display\_anadirPregunta, "Añadir pregunta", "/imagenes/pregunta\_32px.png",

opciones\_desarrollador);

opt\_anadirPregunta.setLayout(null);

opt\_anadirPregunta.setBackground(new Color(54, 33, 89));

opt\_anadirPregunta.setBounds(0, 107, 300, 48);

navPanel\_desarrollador.add(opt\_anadirPregunta);

opt\_anadirPregunta.addMouseListener(new OptionListener(opt\_anadirPregunta));

opt\_anadirPregunta\_ico = new JLabel("");

opt\_anadirPregunta\_ico.setIcon(new ImageIcon(Ventana.class.getResource(opt\_anadirPregunta.getIco\_path())));

opt\_anadirPregunta\_ico.setBounds(10, 0, 48, 48);

opt\_anadirPregunta.add(opt\_anadirPregunta\_ico);

opt\_anadirPregunta\_text = new JLabel("Añadir pregunta");

opt\_anadirPregunta\_text.setForeground(Color.WHITE);

opt\_anadirPregunta\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_anadirPregunta\_text.setBounds(68, 11, 222, 26);

opt\_anadirPregunta.add(opt\_anadirPregunta\_text);

// option cerrar sesion desarolloador

opt\_cerrarSesion\_desarrollador = new JOption(display\_cerrarSesion, "Cerrar sesion", "/imagenes/cerrar\_32px.png",

opciones\_desarrollador);

opt\_cerrarSesion\_desarrollador.setLayout(null);

opt\_cerrarSesion\_desarrollador.setBackground(new Color(54, 33, 89));

opt\_cerrarSesion\_desarrollador.setBounds(0, 155, 300, 48);

navPanel\_desarrollador.add(opt\_cerrarSesion\_desarrollador);

opt\_cerrarSesion\_desarrollador.addMouseListener(new OptionListener(opt\_cerrarSesion\_desarrollador));

opt\_cerrarSesion\_desarrollador\_ico = new JLabel("");

opt\_cerrarSesion\_desarrollador\_ico

.setIcon(new ImageIcon(Ventana.class.getResource(opt\_cerrarSesion\_desarrollador.getIco\_path())));

opt\_cerrarSesion\_desarrollador\_ico.setBounds(10, 0, 48, 48);

opt\_cerrarSesion\_desarrollador.add(opt\_cerrarSesion\_desarrollador\_ico);

opt\_cerrarSesion\_desarrollador\_text = new JLabel("Cerrar sesion");

opt\_cerrarSesion\_desarrollador\_text.setForeground(Color.WHITE);

opt\_cerrarSesion\_desarrollador\_text.setFont(new Font("Tahoma", Font.PLAIN, 18));

opt\_cerrarSesion\_desarrollador\_text.setBounds(68, 11, 222, 26);

opt\_cerrarSesion\_desarrollador.add(opt\_cerrarSesion\_desarrollador\_text);

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* POPUP CONTROL

\*/

public void showPopUp(String type) {

switch (type) {

case "mysqlerror":

popUp\_panel.errorStyle();

popUp\_panel.showPanel("<HTML>MYSQL: No se ha podido establecer la conexion con la base de datos</HTML>",

"/imagenes/error\_black\_96px.png");

break;

case "useronline":

popUp\_panel.errorStyle();

popUp\_panel.showPanel("<HTML>El usuario con el que intentas inciar sesion esta: <b>ONLINE</b></HTML>",

"/imagenes/error\_black\_96px.png");

break;

case "usersuspend":

popUp\_panel.errorStyle();

popUp\_panel.showPanel("<HTML>El usuario con el que intentas inciar sesion esta: <b>SUSPENDIDO</b></HTML>",

"/imagenes/error\_black\_96px.png");

break;

case "regcomplete":

popUp\_panel.infoStyle();

popUp\_panel.showPanel("<HTML>Cuenta creada con exito. Inicia sesion para disfrutar de la aplicacion</HTML>",

"/imagenes/reg\_96px.png");

break;

case "logincomplete":

popUp\_panel.infoStyle();

popUp\_panel.showPanel(

"<HTML>Sesion inciada con exito. Bienvenid@ " + sesionActual.getUsuario().getNick() + "</HTML>",

"/imagenes/user\_96px.png");

break;

case "entrarcomoAlumno":

popUp\_panel.infoStyle();

popUp\_panel.showPanel("<HTML>Sesion cambiada a modo alumno. Cierra la sesion para volver al modo "

+ sesionActual.getUsuario().getAcceso() + "</HTML>", "/imagenes/alumno\_96px.png");

break;

case "entrarcomoProfesor":

popUp\_panel.infoStyle();

popUp\_panel.showPanel("<HTML>Sesion cambiada a modo profesor. Cierra la sesion para volver al modo "

+ sesionActual.getUsuario().getAcceso() + "</HTML>", "/imagenes/profesor\_96px.png");

break;

case "entrarcomoAdministrador":

popUp\_panel.infoStyle();

popUp\_panel.showPanel("<HTML>Sesion cambiada a modo administrador. Cierra la sesion para volver al modo "

+ sesionActual.getUsuario().getAcceso() + "</HTML>", "/imagenes/admin\_96px.png");

break;

case "nuevaconsulta":

popUp\_panel.infoStyle();

popUp\_panel.showPanel(

"<HTML>Gracias por notificar tu incidencia, un desarrollador revisara la consulta en cuanto sea posible</HTML>",

"/imagenes/asunto\_96px.png");

break;

case "addpregunta":

popUp\_panel.infoStyle();

popUp\_panel.showPanel("<HTML>Pregunta añadida con exito.</HTML>", "/imagenes/asunto\_96px.png");

break;

}

// solucion al bug de solapamiento

popUp\_panel.setVisible(false);

popUp\_panel.setVisible(true);

}

public void hidePopUp() {

popUp\_panel.setVisible(false);

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* DISPLAY CONTROL

\*/

public void goTo(String displayName) {

switch (displayName) {

case "display\_login":

opt\_login.setSelected(true);

opt\_reg.setSelected(false);

opt\_info.setSelected(false);

displays.mostarDisplay(display\_login);

break;

}

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* CONTROL FORMULARIOS

\*/

public ResultSet getLoginResultSet() {

ResultSet rs = gestorUsuarios.getUsuario("nick", login\_nick\_textF.getText());

return rs;

}

public void resetRegForm() {

reg\_nick\_tfg.reset();

reg\_password\_tfg.reset();

reg\_name\_tfg.reset();

reg\_apellido\_tfg.reset();

reg\_dni\_tfg.reset();

reg\_email\_tfg.reset();

reg\_password\_check\_tfg.reset();

}

public void setRegDisplay(int index, String function) throws java.awt.IllegalComponentStateException {

focusFixer.requestFocus();

switch (function) {

case "next":

index++;

break;

case "back":

index--;

break;

}

JFormDisplay f;

JFormDisplay selected = null;

Iterator<JFormDisplay> it = grupo\_reg.iterator();

while (it.hasNext()) {

f = it.next();

f.setVisible(false);

if (f.getIndex() == index) {

selected = f;

}

}

selected.setVisible(true);

reg\_form\_ico.setIcon(new ImageIcon(Ventana.class.getResource(selected.getIcoPath())));

reg\_form\_titulo.setText(selected.getTitulo());

reg\_next\_navLabel.setVisible(selected.isNextChecker());

reg\_back\_navLabel.setVisible(selected.isBackChecker());

reg\_next\_navLabel.setActualIndex(index);

reg\_back\_navLabel.setActualIndex(index);

switch (index) {

case 1:

reg\_stepbar\_1\_arrow\_ico.setVisible(true);

reg\_stepbar\_2\_arrow\_ico.setVisible(false);

reg\_stepbar\_3\_arrow\_ico.setVisible(false);

reg\_stepbar\_1\_to\_2\_separator.setBackground(Color.WHITE);

reg\_stepbar\_2\_to\_3\_separator.setBackground(Color.WHITE);

reg\_stepbar\_1\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_1\_32px.png")));

reg\_stepbar\_2\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_2\_32px.png")));

reg\_stepbar\_3\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_3\_32px.png")));

break;

case 2:

reg\_stepbar\_1\_arrow\_ico.setVisible(false);

reg\_stepbar\_2\_arrow\_ico.setVisible(true);

reg\_stepbar\_3\_arrow\_ico.setVisible(false);

reg\_stepbar\_1\_to\_2\_separator.setBackground(COLOR\_CHECK);

reg\_stepbar\_2\_to\_3\_separator.setBackground(Color.WHITE);

reg\_stepbar\_1\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_1\_green\_32px.png")));

reg\_stepbar\_2\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_2\_32px.png")));

reg\_stepbar\_3\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_3\_32px.png")));

break;

case 3:

reg\_stepbar\_1\_arrow\_ico.setVisible(false);

reg\_stepbar\_2\_arrow\_ico.setVisible(false);

reg\_stepbar\_3\_arrow\_ico.setVisible(true);

reg\_stepbar\_1\_to\_2\_separator.setBackground(COLOR\_CHECK);

reg\_stepbar\_2\_to\_3\_separator.setBackground(COLOR\_CHECK);

reg\_stepbar\_1\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_1\_green\_32px.png")));

reg\_stepbar\_2\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_2\_green\_32px.png")));

reg\_stepbar\_3\_ico.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/circ\_3\_green\_32px.png")));

break;

}

}

private void setRegData() {

RegData.setNick(reg\_nick\_textF.getText());

RegData.setPass(String.valueOf(reg\_password\_textF.getPassword()));

RegData.setNombre(reg\_name\_textF.getText());

RegData.setApellidos(reg\_apellido\_textF.getText());

RegData.setDni(reg\_dni\_textF.getText());

RegData.setEmail(reg\_email\_textF.getText());

}

public Usuario getRegData() {

setRegData();

return RegData;

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* SESION CONTROL

\*/

public void iniciarSesion(Sesion sesion) {

sesionActual = sesion;

}

public Sesion getSesionActual() {

return sesionActual;

}

public void cambiarVistaSesion(String vista) {

reniciarFormularioSoporte();

switch (vista.toLowerCase()) {

case "invitado":

alumno.cerrarVistaDeSesion();

profesor.cerrarVistaDeSesion();

administrador.cerrarVistaDeSesion();

desarrollador.cerrarVistaDeSesion();

// --

sesionActual = null;

// --

displays.mostarDisplay(display\_index);

// --

invitado.abrirVistaDeSesion();

break;

case "alumno":

invitado.cerrarVistaDeSesion();

profesor.cerrarVistaDeSesion();

administrador.cerrarVistaDeSesion();

desarrollador.cerrarVistaDeSesion();

actualizarMostraUsuario(sesionActual.getUsuario());

// --

alumno.abrirVistaDeSesion();

// --

usuarioInspeccionado = sesionActual.getUsuario();

break;

case "profesor":

invitado.cerrarVistaDeSesion();

alumno.cerrarVistaDeSesion();

administrador.cerrarVistaDeSesion();

desarrollador.cerrarVistaDeSesion();

// --

actualizarMostraUsuario(sesionActual.getUsuario());

entrarComo\_alumno\_btn\_panel.setVisible(true);

entrarComo\_profesor\_btn\_panel.setVisible(false);

entrarComo\_administrador\_btn\_panel.setVisible(false);

// --

profesor.abrirVistaDeSesion();

break;

case "administrador":

invitado.cerrarVistaDeSesion();

alumno.cerrarVistaDeSesion();

profesor.cerrarVistaDeSesion();

desarrollador.cerrarVistaDeSesion();

// --

actualizarMostraUsuario(sesionActual.getUsuario());

entrarComo\_alumno\_btn\_panel.setVisible(true);

entrarComo\_profesor\_btn\_panel.setVisible(true);

entrarComo\_administrador\_btn\_panel.setVisible(false);

// --

administrador.abrirVistaDeSesion();

break;

case "desarrollador":

invitado.cerrarVistaDeSesion();

alumno.cerrarVistaDeSesion();

profesor.cerrarVistaDeSesion();

administrador.cerrarVistaDeSesion();

// --

actualizarMostraUsuario(sesionActual.getUsuario());

entrarComo\_alumno\_btn\_panel.setVisible(true);

entrarComo\_profesor\_btn\_panel.setVisible(true);

entrarComo\_administrador\_btn\_panel.setVisible(true);

// --

desarrollador.abrirVistaDeSesion();

break;

}

hidePopUp();

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* FRAME CONTROL

\*/

// Getters

public JFrame getFrame() {

return frmHacentest;

}

public JButton getFocusFixer() {

return focusFixer;

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* GESTORES DE DATOS

\*/

public GestorUsuarios getGestorUsuarios() {

return gestorUsuarios;

}

public GestorSesiones getGestorSesiones() {

return gestorSesiones;

}

public GestorConsultas getGestorConsultas() {

return gestorConsultas;

}

public GestorTest getGestorTest() {

return gestorTest;

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* ACTUALIZAR LISTAS

\*/

public void actualizarListaUsuarios(ResultSet rs) {

ArrayList<Usuario> listaUsers = new ArrayList<Usuario>();

try {

while (rs.next()) {

listaUsers.add(new Usuario(rs));

}

} catch (SQLException e) {

e.printStackTrace();

}

listUpdater\_verUsuarios.limpiarLista();

listUpdater\_verUsuarios.setLista(listaUsers);

listUpdater\_verUsuarios.cargarLista();

}

public void actualizarListaConsultas(ResultSet rs) {

ArrayList<Consulta> listaConsultas = new ArrayList<Consulta>();

try {

while (rs.next()) {

listaConsultas.add(new Consulta(rs));

}

} catch (SQLException e) {

e.printStackTrace();

}

listUpdater\_verConsultas.limpiarLista();

listUpdater\_verConsultas.setLista(listaConsultas);

listUpdater\_verConsultas.cargarLista();

}

/\*

\* |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

\* ACTUALIZAR USUARIOS

\*/

public void actualizarMostraUsuario(Usuario usuario) {

usuarioInspeccionado = usuario;

nick\_text\_mostrarUsuario.setText(usuario.getNick());

nombre\_text\_mostrarUsuario.setText(usuario.getNombre());

apellidos\_text\_mostrarUsuario.setText(usuario.getApellidos());

dni\_text\_mostrarUsuario.setText(usuario.getDni());

email\_text\_mostrarUsuario.setText(usuario.getEmail());

fechar\_text\_mostrarUsuario.setText(usuario.getFecha\_registro().toString());

acceso\_text\_mostrarUsuario.setText(usuario.getAcceso());

estado\_text\_mostrarUsuario.setText(usuario.getEstado());

switch (usuario.getAcceso()) {

case "alumno":

acceso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/alumno\_96px.png")));

break;

case "profesor":

acceso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/profesor\_96px.png")));

break;

case "administrador":

acceso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/admin\_96px.png")));

break;

case "desarrollador":

acceso\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/dev\_96px.png")));

break;

}

switch (usuario.getEstado()) {

case "online":

estado\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/online\_96px.png")));

break;

case "offline":

estado\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/offline\_96px.png")));

break;

case "suspendido":

estado\_ico\_mostrarUsuario.setIcon(new ImageIcon(Ventana.class.getResource("/imagenes/banned\_96px.png")));

break;

}

if (sesionActual.getUsuario().getAcceso().equals("administrador")

|| sesionActual.getUsuario().getAcceso().equals("desarollador")) {

estadoCambio\_panel\_mostrarUsuario.setVisible(true);

accesoCambio\_panel\_mostrarUsuario.setVisible(true);

} else {

estadoCambio\_panel\_mostrarUsuario.setVisible(false);

accesoCambio\_panel\_mostrarUsuario.setVisible(false);

}

displays.mostarDisplay(display\_mostrarUsuario);

}

public void actualizarMostrarConsulta(Consulta consulta) {

asunto\_text\_mostrarConsulta.setText(consulta.getAsunto());

descripcion\_mostrarConsulta.setText("<html><p>" + consulta.getDescripcion() + "</p></html>");

nick\_text\_mostrarConsulta.setText(consulta.getUsuario().getNick());

fecha\_text\_mostrarConsulta.setText(consulta.getFecha().toString());

displays.mostarDisplay(display\_mostrarConsulta);

}

public void reniciarFormularioSoporte() {

soporte\_asunto\_textF.setText(soporte\_asunto\_tfg.getDefaultText());

soporte\_asunto\_tfg.setCheck(false);

soporte\_textArea.setText("Describre detalladamente la incidencia.");

}

public Pregunta getPreguntaData() {

Pregunta pregunta = null;

Respuesta ra, rb, rc;

ra = new Respuesta(anadirPregunta\_respuestaA\_textF.getText(), 'a');

rb = new Respuesta(anadirPregunta\_respuestaB\_textF.getText(), 'b');

rc = new Respuesta(anadirPregunta\_respuestaC\_textF.getText(), 'c');

Respuesta[] respuestas = { ra, rb, rc };

char indiceCorrecto = 'a';

if (anadirPregunta\_respuestaB\_radioBtn.isSelected()) {

indiceCorrecto = 'b';

} else {

if (anadirPregunta\_respuestaC\_radioBtn.isSelected()) {

indiceCorrecto = 'c';

}

}

pregunta = new Pregunta(anadirPregunta\_enunciado\_textF.getText(), indiceCorrecto, respuestas);

return pregunta;

}

public void reniciarFormularioPregunta() {

anadirPregunta\_respuestaA\_textF.setText(anadirPregunta\_respuestaA\_tfg.getDefaultText());

anadirPregunta\_respuestaB\_textF.setText(anadirPregunta\_respuestaB\_tfg.getDefaultText());

anadirPregunta\_respuestaC\_textF.setText(anadirPregunta\_respuestaC\_tfg.getDefaultText());

anadirPregunta\_enunciado\_textF.setText(anadirPregunta\_enunciado\_tfg.getDefaultText());

anadirPregunta\_respuestaA\_tfg.setCheck(false);

anadirPregunta\_respuestaB\_tfg.setCheck(false);

anadirPregunta\_respuestaC\_tfg.setCheck(false);

anadirPregunta\_enunciado\_tfg.setCheck(false);

anadirPregunta\_respuestaA\_radioBtn.setSelected(true);

anadirPregunta\_respuestaB\_radioBtn.setSelected(false);

anadirPregunta\_respuestaC\_radioBtn.setSelected(false);

}

public void actualizarPregunta(Pregunta pregunta) {

preguntaActual = pregunta;

Respuesta[] respuestas = pregunta.getRespuestas();

for (int i = 0; i < respuestas.length; i++) {

switch (respuestas[i].getIndice()) {

case 'a':

iniciarTest\_respuestaA\_text.setText("<html><p>" + respuestas[i].getTexto() + "</p></html>");

break;

case 'b':

iniciarTest\_respuestaB\_text.setText("<html><p>" + respuestas[i].getTexto() + "</p></html>");

break;

case 'c':

iniciarTest\_respuestaC\_text.setText("<html><p>" + respuestas[i].getTexto() + "</p></html>");

break;

}

}

iniciarTest\_pregunta\_enunciado.setText("<html><p>" + pregunta.getEnunciado() + "</p></html>");

iniciarTest\_pregunta\_contadorPregunta

.setText("Pregunta n\u00BA: " + (testActual.getEnunciadas().size() + 1) + "/30");

iniciarTest\_index\_panel.setVisible(false);

iniciarTest\_pregunta\_panel.setVisible(true);

}

public Pregunta getPreguntaActual() {

return preguntaActual;

}

public void ReiniciarTest() {

iniciarTest\_index\_panel.setVisible(true);

iniciarTest\_end\_panel.setVisible(false);

iniciarTest\_pregunta\_panel.setVisible(false);

testActual = null;

}

public void completarTest(int aciertos) {

iniciarTest\_pregunta\_panel.setVisible(false);

iniciarTest\_end\_panel.setVisible(true);

iniciarTest\_aciertos\_numero.setText("" + aciertos);

iniciarTest\_fallos\_numero.setText("" + (30 - aciertos));

if ((30 - aciertos) > 3) {

iniciarTest\_resultado\_text.setText("Suspendido");

iniciarTest\_resultado\_text.setForeground(COLOR\_DANGER);

} else {

iniciarTest\_resultado\_text.setText("Aprobado");

iniciarTest\_resultado\_text.setForeground(COLOR\_CHECK);

}

}

public void actualizarMostrarEstadisticas(int numeroTest, int numeroAprobados, float mediaFallos) {

DecimalFormat df = new DecimalFormat();

df.setMaximumFractionDigits(1);

mostrarEstadisticas\_media\_numero.setText("" + df.format(mediaFallos));

mostrarEstadisticas\_aprobados\_numero.setText("" + numeroAprobados);

mostrarEstadisticas\_suspendos\_numero.setText("" + (numeroTest - numeroAprobados));

mostrarEstadisticas\_test\_numero.setText("" + numeroTest);

mostrarEstadisticas\_nick\_text.setText(usuarioInspeccionado.getNick());

displays.mostarDisplay(display\_mostrarEstadisticas);

}

public void cerrarConexion() {

mysqlc.desconectar();

}

}

## Paquete interfaz.componentes

### JChanger

**package** interfaz.componentes;

**import** javax.swing.JLabel;

**public** **class** JChanger **extends** JLabel {

/\*\*

\*

\*/

**private** **static** **final** **long** serialVersionUID = 1L;

String ico\_path, ico\_hover, tipo, estado;

**public** JChanger(String ico\_path, String ico\_hover, String tipo, String estado) {

**super**();

**this**.ico\_path = ico\_path;

**this**.ico\_hover = ico\_hover;

**this**.tipo = tipo;

**this**.estado = estado;

}

**public** String getIco\_path() {

**return** ico\_path;

}

**public** **void** setIco\_path(String ico\_path) {

**this**.ico\_path = ico\_path;

}

**public** String getIco\_hover() {

**return** ico\_hover;

}

**public** **void** setIco\_hover(String ico\_hover) {

**this**.ico\_hover = ico\_hover;

}

**public** String getTipo() {

**return** tipo;

}

**public** **void** setTipo(String tipo) {

**this**.tipo = tipo;

}

**public** String getEstado() {

**return** estado;

}

**public** **void** setEstado(String estado) {

**this**.estado = estado;

}

}

### JControlLabel

**package** interfaz.componentes;

**import** javax.swing.JLabel;

**public** **class** JControlLabel **extends** JLabel {

/\*\*

\*

\*/

**private** **static** **final** **long** serialVersionUID = 1L;

**private** String pathIco, pathIcoHover;

**private** **int** function;

**public** JControlLabel(**int** function, String pathIco, String pathIcoHover) {

**super**();

**this**.function = function;

**this**.pathIco = pathIco;

**this**.pathIcoHover = pathIcoHover;

}

**public** String getPathIco() {

**return** pathIco;

}

**public** **void** setPathIco(String pathIco) {

**this**.pathIco = pathIco;

}

**public** String getPathIcoHover() {

**return** pathIcoHover;

}

**public** **void** setPathIcoHover(String pathIcoHover) {

**this**.pathIcoHover = pathIcoHover;

}

**public** **int** getFunction() {

**return** function;

}

**public** **void** setFunction(**int** function) {

**this**.function = function;

}

}

### JDisplay

package interfaz.componentes;

import javax.swing.JPanel;

import interfaz.grupos.DisplayGroup;

public class JDisplay extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private String titulo;

private String icoPath;

private DisplayGroup grupo;

public JDisplay(String titulo, String icoPath, DisplayGroup grupo) {

super();

this.titulo = titulo;

this.icoPath = icoPath;

this.grupo = grupo;

grupo.addDisplay(this);

}

public String getTitulo() {

return titulo;

}

public void setTitulo(String titulo) {

this.titulo = titulo;

}

public String getIcoPath() {

return icoPath;

}

public void setIcoPath(String icoPath) {

this.icoPath = icoPath;

}

public DisplayGroup getGrupo() {

return grupo;

}

public void setGrupo(DisplayGroup grupo) {

this.grupo = grupo;

}

}

### JFormBtn

package interfaz.componentes;

import java.awt.Color;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.JPanel;

import interfaz.grupos.TextFieldGroup;

public class JFormBtn extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private final Color COLOR\_CHECK = new Color(50, 205, 50);

private final Color COLOR\_DISABLED = new Color(128, 128, 128);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

ArrayList<TextFieldGroup> grupo\_logico;

private String function;

public JFormBtn(String function, ArrayList<TextFieldGroup> grupo\_logico) {

super();

this.function = function;

this.grupo\_logico = grupo\_logico;

}

public JFormBtn(String string) {

// TODO Auto-generated constructor stub

}

public void setStatus() {

if(this.isEnabled()) {

this.setBackground(COLOR\_CHECK);

} else {

this.setBackground(COLOR\_DISABLED);

}

}

public Color getCOLOR\_CHECK() {

return COLOR\_CHECK;

}

public Color getCOLOR\_DISABLED() {

return COLOR\_DISABLED;

}

public Color getCOLOR\_HOVER() {

return COLOR\_HOVER;

}

public String getFunction() {

return function;

}

public void setFunction(String function) {

this.function = function;

}

public boolean isGroupAllTrue() {

Iterator<TextFieldGroup> it = grupo\_logico.iterator();

boolean control = true;

TextFieldGroup tFGroup;

while(it.hasNext()) {

tFGroup = it.next();

if(!tFGroup.isCheck()) {

control = false;

}

}

return control;

}

}

### JFormDisplay

package interfaz.componentes;

import java.util.ArrayList;

import javax.swing.JPanel;

public class JFormDisplay extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private String titulo, icoPath;

private ArrayList<JFormDisplay> grupo;

private int index;

private boolean backChecker, nextChecker;

public JFormDisplay(String titulo, String icoPath, ArrayList<JFormDisplay> grupo, int index, boolean backChecker,

boolean nextChecker) {

super();

this.titulo = titulo;

this.icoPath = icoPath;

this.grupo = grupo;

this.index = index;

this.backChecker = backChecker;

this.nextChecker = nextChecker;

grupo.add(this);

}

public boolean isBackChecker() {

return backChecker;

}

public void setBackChecker(boolean backChecker) {

this.backChecker = backChecker;

}

public boolean isNextChecker() {

return nextChecker;

}

public void setNextChecker(boolean nextChecker) {

this.nextChecker = nextChecker;

}

public int getIndex() {

return index;

}

public void setIndex(int index) {

this.index = index;

}

public String getTitulo() {

return titulo;

}

public void setTitulo(String titulo) {

this.titulo = titulo;

}

public String getIcoPath() {

return icoPath;

}

public void setIcoPath(String icoPath) {

this.icoPath = icoPath;

}

public ArrayList<JFormDisplay> getGrupo() {

return grupo;

}

public void setGrupo(ArrayList<JFormDisplay> grupo) {

this.grupo = grupo;

}

}

### JNavLabel

**package** interfaz.componentes;

**import** javax.swing.JLabel;

**public** **class** JNavLabel **extends** JLabel {

/\*\*

\*

\*/

**private** **static** **final** **long** serialVersionUID = 1L;

**private** String ico\_path;

**private** String ico\_hover\_path;

**private** String function;

**private** String grupo;

**private** **int** actualIndex = 1;

**public** JNavLabel(String ico\_path, String ico\_hover\_path, String grupo, String function) {

**super**();

**this**.ico\_path = ico\_path;

**this**.ico\_hover\_path = ico\_hover\_path;

**this**.grupo = grupo;

**this**.function = function;

}

**public** String getIco\_path() {

**return** ico\_path;

}

**public** **void** setIco\_path(String ico\_path) {

**this**.ico\_path = ico\_path;

}

**public** String getIco\_hover\_path() {

**return** ico\_hover\_path;

}

**public** **void** setIco\_hover\_path(String ico\_jover\_path) {

**this**.ico\_hover\_path = ico\_jover\_path;

}

**public** String getFunction() {

**return** function;

}

**public** **void** setFunction(String function) {

**this**.function = function;

}

**public** **int** getActualIndex() {

**return** actualIndex;

}

**public** **void** setActualIndex(**int** actualIndex) {

**this**.actualIndex = actualIndex;

}

**public** String getGrupo() {

**return** grupo;

}

**public** **void** setGrupo(String grupo) {

**this**.grupo = grupo;

}

}

### JOption

package interfaz.componentes;

import java.awt.Color;

import javax.swing.JPanel;

import interfaz.grupos.OptionGroup;

public class JOption extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private boolean selected;

private JDisplay display;

private String text;

private String ico\_path;

private OptionGroup grupo;

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private final Color COLOR\_SELECTED = new Color(85, 65, 118);

private final Color COLOR\_BACKGROUND = new Color(54,33,89);

public JOption(JDisplay display, String text, String ico\_path, OptionGroup grupo) {

super();

this.display = display;

this.text = text;

this.ico\_path = ico\_path;

this.grupo = grupo;

grupo.addOption(this);

}

public boolean isSelected() {

return selected;

}

public void setSelected(boolean selected) {

this.selected = selected;

if (selected){

this.setBackground(COLOR\_SELECTED);

}else {

this.setBackground(COLOR\_BACKGROUND);

}

}

public JDisplay getDisplay() {

return display;

}

public void setDisplay(JDisplay display) {

this.display = display;

}

public String getText() {

return text;

}

public void setText(String text) {

this.text = text;

}

public String getIco\_path() {

return ico\_path;

}

public void setIco\_path(String ico\_path) {

this.ico\_path = ico\_path;

}

public Color getCOLOR\_HOVER() {

return COLOR\_HOVER;

}

public Color getCOLOR\_SELECTED() {

return COLOR\_SELECTED;

}

public Color getCOLOR\_BACKGROUND() {

return COLOR\_BACKGROUND;

}

public OptionGroup getGrupo() {

return grupo;

}

public void setGrupo(OptionGroup grupo) {

this.grupo = grupo;

}

}

### JPopUp

package interfaz.componentes;

import java.awt.Color;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import javax.swing.JPanel;

import interfaz.Ventana;

public class JPopUp extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JLabel ico, text;

private JControlLabel control;

private final Color COLOR\_ERROR = new Color(219, 186, 70);

private final Color COLOR\_CHECK = new Color(50, 205, 50);

public void showPanel(String text, String ico\_path) {

this.setVisible(true);

this.text.setText(text);

this.ico.setIcon(new ImageIcon(Ventana.class.getResource(ico\_path)));

}

public void errorStyle() {

text.setForeground(Color.BLACK);

this.setBackground(COLOR\_ERROR);

}

public void infoStyle() {

text.setForeground(Color.WHITE);

this.setBackground(COLOR\_CHECK);

}

public JLabel getIco() {

return ico;

}

public void setIco(JLabel ico) {

this.ico = ico;

}

public JLabel getText() {

return text;

}

public void setText(JLabel text) {

this.text = text;

}

public JControlLabel getControl() {

return control;

}

public void setControl(JControlLabel control) {

this.control = control;

}

}

### JRespuestaBtn

**package** interfaz.componentes;

**import** javax.swing.JPanel;

**public** **class** JRespuestaBtn **extends** JPanel {

/\*\*

\*

\*/

**private** **static** **final** **long** serialVersionUID = 1L;

**private** **char** indice;

**public** JRespuestaBtn(**char** indice) {

**super**();

**this**.indice = indice;

}

**public** **char** getIndice() {

**return** indice;

}

**public** **void** setIndice(**char** indice) {

**this**.indice = indice;

}

}

### JRowBtn

package interfaz.componentes;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JLabel;

import javax.swing.JPanel;

import controladores.JRowBtnListener;

import interfaz.Ventana;

public class JRowBtn extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private final Color COLOR\_CHECK = new Color(50, 205, 50);

private final Color COLOR\_HOVER = new Color(110, 89, 222);

private Ventana ventana;

private JLabel text;

private int pk;

private String tipo;

public JRowBtn(int pk\_usuario, Ventana ventana, String tipo) {

this.ventana = ventana;

this.pk = pk\_usuario;

this.tipo = tipo;

this.setBackground(COLOR\_CHECK);

text = new JLabel("Inspeccionar");

text.setForeground(Color.WHITE);

text.setFont(new Font("Tahoma", Font.PLAIN, 18));

this.add(text);

this.addMouseListener(new JRowBtnListener(this, this.ventana));

}

public Color getCOLOR\_CHECK() {

return COLOR\_CHECK;

}

public Color getCOLOR\_HOVER() {

return COLOR\_HOVER;

}

public int getPk() {

return pk;

}

public void setPk(int pk) {

this.pk = pk;

}

public String getTipo() {

return tipo;

}

public void setTipo(String tipo) {

this.tipo = tipo;

}

}

### JRowPanel

package interfaz.componentes;

import java.awt.Color;

import java.awt.Font;

import java.awt.GridLayout;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JSeparator;

import javax.swing.SwingConstants;

import clases.Consulta;

import clases.Usuario;

import interfaz.Ventana;

public class JRowPanel extends JPanel {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JLabel col1, col2, col3, col4, col5;

private JRowBtn editar\_btn;

public JRowPanel(String col1t, String col2t, String col3t, String col4t, String col5t) {

super();

this.setLayout(new GridLayout(1, 5, 10, 2));

this.setOpaque(false);

this.setBounds(0, 0, 780, 50);

col1 = new JLabel(col1t);

this.add(col1);

col1.setForeground(Color.WHITE);

col1.setFont(new Font("Tahoma", Font.BOLD, 18));

col2 = new JLabel(col2t);

this.add(col2);

col2.setForeground(Color.WHITE);

col2.setFont(new Font("Tahoma", Font.BOLD, 18));

col3 = new JLabel(col3t);

this.add(col3);

col3.setBounds(0, 0, 70, 50);

col3.setForeground(Color.WHITE);

col3.setFont(new Font("Tahoma", Font.BOLD, 18));

col4 = new JLabel(col4t);

this.add(col4);

col4.setBounds(0, 0, 70, 50);

col4.setForeground(Color.WHITE);

col4.setFont(new Font("Tahoma", Font.BOLD, 18));

//col4.setHorizontalAlignment(SwingConstants.CENTER);

col4.setVerticalAlignment(SwingConstants.CENTER);

col5 = new JLabel(col5t);

this.add(col5);

col5.setBounds(0, 0, 70, 50);

col5.setForeground(Color.WHITE);

col5.setFont(new Font("Tahoma", Font.BOLD, 18));

col5.setHorizontalAlignment(SwingConstants.CENTER);

col5.setVerticalAlignment(SwingConstants.CENTER);

}

public JRowPanel(Usuario usuario, Ventana ventana) {

this.setLayout(new GridLayout(1, 4, 10, 2));

this.setOpaque(false);

this.setBounds(0, 0, 780, 50);

col1 = new JLabel("<html>"+usuario.getNick());

this.add(col1);

col1.setForeground(Color.WHITE);

col1.setFont(new Font("Tahoma", Font.PLAIN, 18));

col2 = new JLabel(usuario.getDni());

this.add(col2);

col2.setForeground(Color.WHITE);

col2.setFont(new Font("Tahoma", Font.PLAIN, 18));

String accesoText = "";

switch(usuario.getAcceso()) {

case "alumno":

accesoText = "Alumno";

break;

case "profesor":

accesoText = "Profesor";

break;

case "administrador":

accesoText = "Administrador";

break;

case "desarrollador":

accesoText = "Desarrollador";

break;

}

col3 = new JLabel(accesoText);

this.add(col3);

col3.setBounds(0, 0, 70, 50);

col3.setForeground(Color.WHITE);

col3.setFont(new Font("Tahoma", Font.PLAIN, 18));

col4 = new JLabel(usuario.getEstado());

this.add(col4);

col4.setBounds(0, 0, 70, 50);

col4.setForeground(Color.WHITE);

col4.setFont(new Font("Tahoma", Font.PLAIN, 18));

//col4.setHorizontalAlignment(SwingConstants.CENTER);

col4.setVerticalAlignment(SwingConstants.CENTER);

editar\_btn = new JRowBtn(usuario.getPk\_usuario(), ventana, "usuario");

this.add(editar\_btn);

}

public JRowPanel(Consulta consulta, Ventana ventana) {

// TODO Auto-generated constructor stub

this.setLayout(new GridLayout(1, 5, 10, 2));

this.setOpaque(false);

this.setBounds(0, 0, 780, 50);

col1 = new JLabel("<html>"+consulta.getUsuario().getNick());

this.add(col1);

col1.setForeground(Color.WHITE);

col1.setFont(new Font("Tahoma", Font.PLAIN, 18));

String accesoText = "";

switch(consulta.getUsuario().getAcceso()) {

case "alumno":

accesoText = "Alumno";

break;

case "profesor":

accesoText = "Profesor";

break;

case "administrador":

accesoText = "Administrador";

break;

case "desarrollador":

accesoText = "Desarrollador";

break;

default:

accesoText = "Invitado";

}

col2 = new JLabel(accesoText);

this.add(col2);

col2.setForeground(Color.WHITE);

col2.setFont(new Font("Tahoma", Font.PLAIN, 18));

col3 = new JLabel(consulta.getAsunto().length()<16?consulta.getAsunto():consulta.getAsunto().substring(0, 16));

this.add(col3);

col3.setBounds(0, 0, 70, 50);

col3.setForeground(Color.WHITE);

col3.setFont(new Font("Tahoma", Font.PLAIN, 14));

col4 = new JLabel(consulta.getFecha().toString().substring(0, 16));

this.add(col4);

col4.setBounds(0, 0, 70, 50);

col4.setForeground(Color.WHITE);

col4.setFont(new Font("Tahoma", Font.PLAIN, 14));

//col4.setHorizontalAlignment(SwingConstants.CENTER);

col4.setVerticalAlignment(SwingConstants.CENTER);

editar\_btn = new JRowBtn(consulta.getPk\_consulta(), ventana, "consulta");

this.add(editar\_btn);

}

}

## Paquete interfaz.grupos

### DisplayGroup

package interfaz.grupos;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import interfaz.Ventana;

import interfaz.componentes.JDisplay;

public class DisplayGroup {

private ArrayList<JDisplay> grupo;

private JLabel titulo\_ico, titulo\_text;

private Ventana ventana;

public DisplayGroup(JLabel titulo\_ico, JLabel titulo\_text, Ventana ventana) {

super();

grupo = new ArrayList<JDisplay>();

this.titulo\_ico = titulo\_ico;

this.titulo\_text = titulo\_text;

this.ventana = ventana;

}

public void mostarDisplay(JDisplay display) {

Iterator<JDisplay> it = grupo.iterator();

while (it.hasNext()) {

JDisplay d = it.next();

ventana.getFocusFixer().requestFocus();

d.setVisible(false);

}

display.setVisible(true);

titulo\_ico.setIcon(new ImageIcon(Ventana.class.getResource(display.getIcoPath())));

titulo\_text.setText(display.getTitulo());

}

public void addDisplay(JDisplay display) {

grupo.add(display);

}

public ArrayList<JDisplay> getGrupo() {

return grupo;

}

public void setGrupo(ArrayList<JDisplay> grupo) {

this.grupo = grupo;

}

public Ventana getVentana() {

return ventana;

}

}

### OptionGroup

package interfaz.grupos;

import java.util.ArrayList;

import javax.swing.JPanel;

import interfaz.componentes.JOption;

public class OptionGroup {

private JPanel nav\_panel;

private ArrayList<JOption> grupo;

public OptionGroup() {

super();

this.grupo = new ArrayList<JOption>();

}

public void mostarNavPanel() {

nav\_panel.setVisible(true);

}

public void ocultarNavPanel() {

nav\_panel.setVisible(false);

}

public void addOption(JOption option) {

grupo.add(option);

}

public JPanel getNav\_panel() {

return nav\_panel;

}

public void setNav\_panel(JPanel nav\_panel) {

this.nav\_panel = nav\_panel;

}

public ArrayList<JOption> getGrupo() {

return grupo;

}

public void setGrupo(ArrayList<JOption> grupo) {

this.grupo = grupo;

}

}

### SesionSteup

**package** interfaz.grupos;

**public** **class** SesionSteup {

**private** OptionGroup opciones;

**private** DisplayGroup displays;

**public** SesionSteup(OptionGroup opciones, DisplayGroup displays) {

**super**();

**this**.opciones = opciones;

**this**.displays = displays;

}

**public** **void** abrirVistaDeSesion() {

opciones.getNav\_panel().setVisible(**true**);

}

**public** **void** cerrarVistaDeSesion() {

opciones.getNav\_panel().setVisible(**false**);

}

}

### TextFieldGroup

package interfaz.grupos;

import java.util.ArrayList;

import javax.swing.ImageIcon;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JSeparator;

import javax.swing.JTextField;

import acceso\_a\_datos.InfoMsg;

public class TextFieldGroup {

private JPanel content\_panel;

private JPanel info\_panel;

private JLabel textF\_icon, info\_ico, info\_text;

private JTextField textF;

private JSeparator separator;

private String tipo, defaultText;

private boolean check;

private boolean connected;

private ArrayList<TextFieldGroup> grupo\_logico;

public TextFieldGroup(JPanel content\_panel, JPanel info\_panel, JLabel textF\_icon, JLabel info\_ico, JLabel info\_text,

JTextField textF, JSeparator separator, String tipo, ArrayList<TextFieldGroup> grupo\_logico) {

super();

this.content\_panel = content\_panel;

this.info\_panel = info\_panel;

this.textF\_icon = textF\_icon;

this.info\_ico = info\_ico;

this.info\_text = info\_text;

this.textF = textF;

this.separator = separator;

this.tipo = tipo;

check = false;

connected = true;

defaultText = textF.getText();

grupo\_logico.add(this);

}

public TextFieldGroup(JPanel content\_panel, JPanel info\_panel, JLabel textF\_icon, JLabel info\_ico, JLabel info\_text,

JTextField textF, JSeparator separator, String tipo) {

super();

this.content\_panel = content\_panel;

this.info\_panel = info\_panel;

this.textF\_icon = textF\_icon;

this.info\_ico = info\_ico;

this.info\_text = info\_text;

this.textF = textF;

this.separator = separator;

this.tipo = tipo;

check = false;

connected = true;

defaultText = textF.getText();

}

public void hideInfo() {

info\_panel.setVisible(false);

}

public void showInfo(InfoMsg info) {

info\_panel.setVisible(true);

info\_panel.setBackground(info.getColor());

info\_text.setText(info.getText());

info\_ico.setIcon(new ImageIcon(this.getClass().getResource(info.getIco\_path())));

check = info.isCheck();

}

public JPanel getContent\_panel() {

return content\_panel;

}

public void setContent\_panel(JPanel content\_panel) {

this.content\_panel = content\_panel;

}

public JPanel getInfo\_panel() {

return info\_panel;

}

public void setInfo\_panel(JPanel info\_panel) {

this.info\_panel = info\_panel;

}

public JLabel getTextF\_icon() {

return textF\_icon;

}

public void setTextF\_icon(JLabel textF\_icon) {

this.textF\_icon = textF\_icon;

}

public JLabel getInfo\_ico() {

return info\_ico;

}

public void setInfo\_ico(JLabel info\_ico) {

this.info\_ico = info\_ico;

}

public JLabel getInfo\_text() {

return info\_text;

}

public void setInfo\_text(JLabel info\_text) {

this.info\_text = info\_text;

}

public JTextField getTextF() {

return textF;

}

public void setTextF(JTextField textF) {

this.textF = textF;

}

public JSeparator getSeparator() {

return separator;

}

public void setSeparator(JSeparator separator) {

this.separator = separator;

}

public String getTipo() {

return tipo;

}

public void setTipo(String tipo) {

this.tipo = tipo;

}

public boolean isCheck() {

return check;

}

public void setCheck(boolean check) {

this.check = check;

}

public ArrayList<TextFieldGroup> getGrupo\_logico() {

return grupo\_logico;

}

public void setGrupo\_logico(ArrayList<TextFieldGroup> grupo\_logico) {

this.grupo\_logico = grupo\_logico;

}

public void reset() {

textF.setText(defaultText);

info\_panel.setVisible(false);

check = false;

}

public String getDefaultText() {

return defaultText;

}

public void setDefaultText(String defaultText) {

this.defaultText = defaultText;

}

public boolean isConnected() {

return connected;

}

public void setConnected(boolean connected) {

this.connected = connected;

}

}

### TextFieldGroupRel

package interfaz.grupos;

import java.util.ArrayList;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JSeparator;

import javax.swing.JTextField;

public class TextFieldGroupRel extends TextFieldGroup {

TextFieldGroup rel;

String relType;

public TextFieldGroupRel(JPanel content\_panel, JPanel info\_panel, JLabel textF\_icon, JLabel info\_ico,

JLabel info\_text, JTextField textF, JSeparator separator, ArrayList<TextFieldGroup> grupo\_logico, TextFieldGroup rel, String relType) {

super(content\_panel, info\_panel, textF\_icon, info\_ico, info\_text, textF, separator, "rel", grupo\_logico);

// TODO Auto-generated constructor stub

this.rel = rel;

this.relType = relType;

}

public TextFieldGroup getRel() {

return rel;

}

public void setRel(TextFieldGroup rel) {

this.rel = rel;

}

public String getRelType() {

return relType;

}

public void setRelType(String relType) {

this.relType = relType;

}

}

## Paquete interfaz.listas

### JConsultaRowList

package interfaz.listas;

import java.awt.GridLayout;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.JPanel;

import clases.Consulta;

import clases.Usuario;

import interfaz.Ventana;

import interfaz.componentes.JRowPanel;

public class JConsultaRowList extends JRowList{

private ArrayList<Consulta> lista;

private JPanel contenedor;

private Ventana ventana;

public JConsultaRowList(ArrayList<Consulta> lista, JPanel contendor, Ventana ventana) {

super();

this.lista = lista;

this.contenedor = contendor;

this.ventana = ventana;

}

public void cargarLista() {

if(lista.size() < 7) {

contenedor.setLayout(new GridLayout(9, 1, 0, 20));

} else {

contenedor.setLayout(new GridLayout(0, 1, 0, 20));

}

Iterator<Consulta> it = lista.iterator();

contenedor.add(new JRowPanel("NICK", "ACESSO", "ASUNTO", "FECHA", " "));

while(it.hasNext()) {

contenedor.add(new JRowPanel(it.next(), ventana));

}

}

public void limpiarLista() {

contenedor.removeAll();

lista = null;

contenedor.setVisible(false);

contenedor.setVisible(true);

}

public void setLista(ArrayList<Consulta> lista) {

this.lista = lista;

}

}

### JRowList

package interfaz.listas;

import java.awt.GridLayout;

import java.util.Iterator;

import clases.Usuario;

import interfaz.componentes.JRowPanel;

public abstract class JRowList {

public abstract void cargarLista();

public abstract void limpiarLista();

}

### JUserRowList

package interfaz.listas;

import java.awt.GridLayout;

import java.util.ArrayList;

import java.util.Iterator;

import javax.swing.JPanel;

import clases.Usuario;

import interfaz.Ventana;

import interfaz.componentes.JRowPanel;

public class JUserRowList extends JRowList {

private ArrayList<Usuario> lista;

private JPanel contenedor;

private Ventana ventana;

public JUserRowList(ArrayList<Usuario> lista, JPanel contendor, Ventana ventana) {

super();

this.lista = lista;

this.contenedor = contendor;

this.ventana = ventana;

}

public void cargarLista() {

if(lista.size() < 7) {

contenedor.setLayout(new GridLayout(9, 1, 0, 20));

} else {

contenedor.setLayout(new GridLayout(0, 1, 0, 20));

}

Iterator<Usuario> it = lista.iterator();

contenedor.add(new JRowPanel("NICK", "DNI", "ACCESO", "ESTADO", " "));

while(it.hasNext()) {

contenedor.add(new JRowPanel(it.next(), ventana));

}

}

public void limpiarLista() {

contenedor.removeAll();

lista = null;

contenedor.setVisible(false);

contenedor.setVisible(true);

}

public void setLista(ArrayList<Usuario> lista) {

this.lista = lista;

}

}

## Paquete principal

### Launcher

package principal;

import java.awt.EventQueue;

import interfaz.Ventana;

public class Launcher {

/\*

\* Autor: Adan Jarillo Merida

\* Version: ALPHA 1.0.0

\* test build a100

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run(){

try {

Ventana window = new Ventana();

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

}

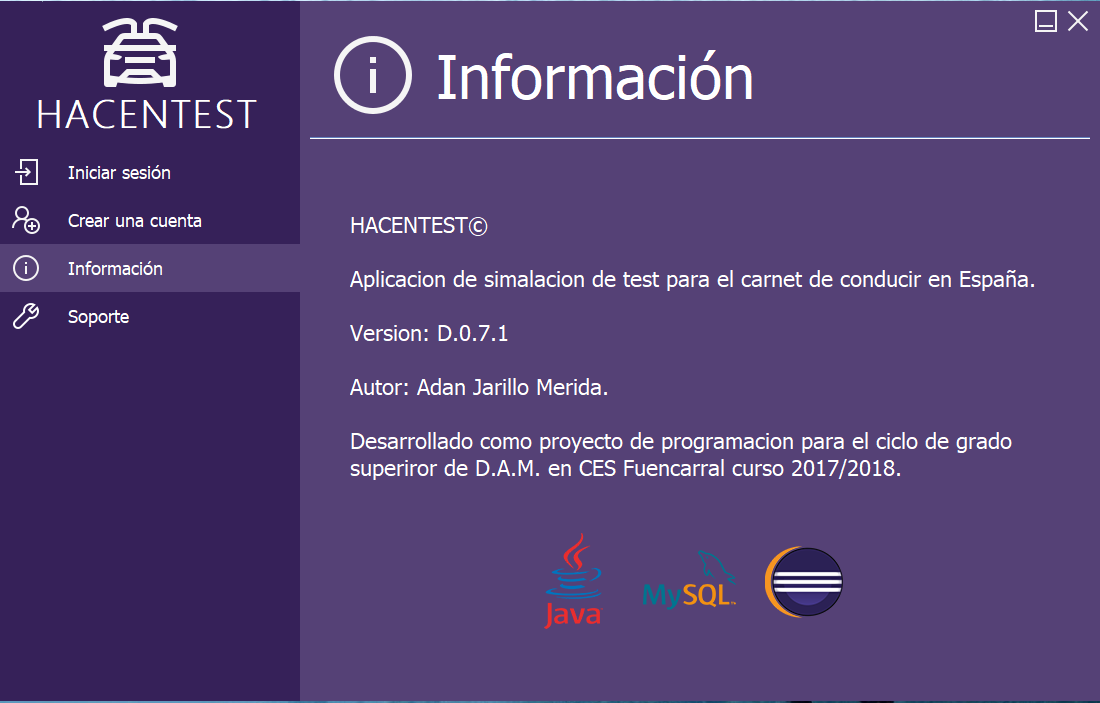
# Manual de usuario

## Introducccion

Hacentest es una plataforma de simulación de test teórico del carnet conducción. Este manual tiene el fin de aclarar cualquier duda sobre el correcto uso de la misma.

## Navegacion

La aplicación cuenta con un sistema de navegación basado en una barra lateral de opciones que cambiara según el punto en el que nos encontremos. La opción actual se mostrara de diferente color para saber en qué panel nos encontramos.



## Registro

Para acceder a la plataforma es necesario tener una cuenta registrada, para ello debemos

# IMPLEMENTACION

## Preparación del sistema

Sera necesaria que el sistema donde se desea implementar la plataforma cumpla con los requisitos anteriormente mencionados, así como un servidor web o local donde desplegar la base de datos ligada a la plataforma.

En el caso de que la base de datos no estuviera alojada en un servidor local los datos de acceso a la misma (URL, USUARIO, y CONTRASEÑA) será completamente necesario comunicarle los nuevos datos al desarrollador encargado de la implementación para modificar el código fuente de la aplicación.

## Instalación de la aplicación

### Despliegue base de datos

La plataforma está directamente relacionada con una base de datos de nombre “hacentest” durante su desarrollo, dicha base de datos se encuentra en formato .SQL en la carpeta del proyecto. Antes de iniciar la aplicación será necesario importar el backup adjuntado en el servidor que se fuera a usar (por defecto servidor local puerto 3306, relacionado con MySQL).

### Entorno de desarrollo

Si la aplicación se desea explorar en su versión de desarrollo será necesario de entorno capaz de realizar la compilación de la misma. Esta forma de ejecución está dirigida exclusivamente a la muestra y testeo de la aplicación y en ningún caso se debe optar por este método en la implementación para el cliente final.

### Versión ejecutable

### 

Destinada al cliente final tanto en su versión comercial como de prueba. Un archivo ejecutable instalado en el sistema del cliente que despliegue la propia aplicación.

### Futuras versiones

En futuras versiones se ha de tener en cuenta la necesidad de actualizar tanto la propia aplicación como la base de datos, por lo que será necesario implementar un sistema de descarga de actualizaciones para evitar la continua implementación de futuras versiones del producto.

# MANTENIMIENTO

## Servicio interno de consultas

La aplicación cuenta con un servicio interno de consultas destinado a poner en conocimiento de los desarrolladores cualquier incidencia acaecida durante el uso de la plataforma así como posibles dudas de los distintos usuarios con el fin de recopilarlos y poder implementar parches que corrijan estos problemas.

La intención de este sistema es la de ofrecer al cliente una continua mejora del producto y facilitar el trabajo de los desarrolladores pudiendo recopilar de manera más efectiva los fallos que se desean resolver.

## Actualizaciones

Puesto que la aplicación a sedo desarrollada en su primera versión para un único modelo de negocio y bajo los criterios de un único tipo de test se conoce la necesidad de trabajar en un modelo escalable de la misma, añadiendo soporte a nuevos idiomas, test y contenido (preguntas, funcionalidades, ampliación de gestión de usuarios).

Dichas adiciones a la plataforma están planeadas en forma de actualizaciones, la cuales serán necesarias documentar e implementar. Las propias actualizaciones podrán constar de nuevo contenido, optimización de recursos o facilidades de cara a la implementación.