

Caso Práctico

PhotoTDS

Este curso seguiremos en la línea de abordar un caso práctico inspirado en alguna aplicación bien conocida por los alumnos, como Spotify, Youtube, Whatsapp o Blablacar en cursos anteriores, ahora una aplicación para compartir fotos. Existen muchas aplicaciones web y móviles destinadas a compartir fotos, algunas de ellas muy populares como *Instagram*, *Google Photos*, o *iCloud Photos*. El caso práctico de la asignatura “Tecnologías de Desarrollo de Software” consistirá en desarrollar una aplicación de escritorio (*desktop app*) llamada PhotoTDS, cuyos requisitos e interfaz de usuario están inspirados en aquellos de las mencionadas aplicaciones. Claro está, por cuestiones de la carga lectiva de la asignatura, sólo se considerará el desarrollo de una funcionalidad básica como *login* y registro de usuarios, publicar fotos, crear álbumes de fotos, búsqueda de usuarios y fotos/álbumes, y seguir a usuarios. No obstante, cada grupo de alumnos puede añadir funcionalidad tras consultar con su tutor. En este documento, primero se detallarán los requisitos de PhotoTDS, esto es, la lista de funciones que debe ofrecer a los usuarios. Luego se mostrarán y comentarán las ventanas que componen la interfaz gráfica de usuario. Después se indicará la arquitectura software de la aplicación. Finalmente se proporcionarán detalles sobre la gestión del proyecto, documentación y código a entregar, y finalmente sobre la evaluación.

1. Requisitos

Login y Registro de usuarios

Para entrar a la aplicación, los usuarios deben estar registrados y realizar un *login* con su nombre de usuario o email y su contraseña. Para registrarse un usuario debe indicar la siguiente información: email, nombre completo (se recomienda escribir nombre y apellidos de la persona), nombre de usuario y contraseña, fecha de nacimiento, y opcionalmente se puede agregar un texto de presentación (máximo 200 caracteres) y una foto. Esta información constituye el *perfil del usuario*. La presentación, contraseña y foto pueden ser cambiados en cualquier momento. Como se explica abajo los usuarios pueden compartir fotos con otros usuarios y seguir a otros usuarios. Los usuarios estarán almacenados en una base de datos.

Seguidores

Cuando un usuario “seguidor” sigue la cuenta de otro, el seguidor recibe una notificación cuando el usuario seguido agrega una nueva foto. Se considerará que todas las cuentas son públicas y no es necesario que el propietario conceda un permiso al usuario que desea seguirla. El perfil de usuario registrará el número de seguidores y el número de usuarios seguidos.

Fotos

Los usuarios pueden agregar fotos que son compartidas con otros usuarios. Cuando una foto es agregada se le pueden asociar etiquetas denominadas *hashtags*. Un *hashtag* está formado por el símbolo ‘#’ seguido de una palabra que tiene un máximo de 15 letras, por ejemplo #lunallena o #murcia. Una foto puede tener un máximo de 4 hashtags. Los usuarios pueden escribir comentarios sobre las fotos y/o marcar que le gustan. Los hashtags son introducidos por el usuario que publica la foto como parte del texto de un comentario sobre la foto. La información requerida sobre las fotos será registrada en la base de datos, tal como la ruta del archivo, un contador de “me gusta” y una lista de comentarios. Un usuario podrá conocer sus últimas 10 fotos que han recibido un “me gusta” (opcional). En cualquier momento un usuario puede eliminar una foto.

Álbumes de fotos

Los usuarios pueden agrupar fotos relacionadas por una temática, por ejemplo, un lugar visitado, un viaje o un objeto fotografiado. Estas colecciones de fotos serán denominadas “álbumes”. Un álbum no puede contener más de 16 fotos. Al igual que las fotos, un álbum puede tener hashtags y recibir “me gusta” que implicará un “me gusta” para cada foto de la colección. La información sobre los álbumes es también almacenada en la base de datos. Por simplicidad, no se controlará que un usuario otorgue más de un “me gusta” a la misma foto. Un álbum puede ser eliminado en cualquier momento y esto implica eliminar todas las fotos que contiene

Búsquedas

Un usuario puede buscar a otros usuarios indicando una secuencia de caracteres del nombre, nombre de usuario o email y como resultado recibirá una lista con la foto del perfil y el nombre de cada usuario encontrado. Seleccionando a un elemento de la lista se accederá al perfil del usuario. También puede buscar fotos o álbumes indicando hashtags, en este caso se retorna una lista de álbumes y fotos con los hashtags y el número de publicaciones (fotos más álbumes). Si se introduce más de un hashtag, se retornarán las publicaciones etiquetadas con todos ellos.

Usuario Premium

Una vez registrado un usuario podrá, en cualquier momento, activar su cuenta como “Premium” pagando una cantidad de dinero. Se aplican descuentos dependiendo de la edad y del número de “me gusta” recibidos. La implementación de esta funcionalidad deberá tener en cuenta la probable definición de nuevos descuentos en el futuro. La funcionalidad de cargar el pago no será parte de la aplicación PhotoTDS. Un usuario premium dispone de funcionalidad extra como la posibilidad de (i) generar un archivo Excel con la lista de sus seguidores: filas con 3 columnas para registrar su nombre, email, y presentación; (ii) generar un archivo PDF con la misma información, pero en forma de tabla con tres columnas; y (iii) visualizar la lista de las 10 fotos del usuario que más “me gusta” han recibido.

Cada grupo tiene libertad para considerar funcionalidad adicional o realizar cambios sobre la funcionalidad aquí explicada, **pero deberá consultarlo con su tutor de prácticas.**

2. Interfaces gráficas de usuario

A continuación, se detallan las ventanas que formarán parte de la aplicación. Las capturas mostradas son orientativas y **los alumnos tienen libertad para realizar los cambios que consideren oportunos**. No obstante, cambios significativos (por ejemplo, la estructura de una ventana o añadir/eliminar nuevas ventanas) deberían ser consultados con su tutor.

Ventana de Entrada

La Figura 1 muestra la ventana de entrada a la aplicación. El botón “Iniciar sesión” está destinado a abrir la ventana principal de PhotoTDS mostrada en la Figura 6. Se mostrará un mensaje de error cuando el nombre y contraseña no correspondan a los de un usuario registrado. Este mensaje se puede mostrar por medio de una de las ventanas de diálogo proporcionadas por Swing para mostrar todo tipo de mensajes. La ventana de registro se abrirá al hacer clic sobre en el botón “Crear Cuenta”.



Figura 1: Ventana de Entrada



Figura 2: Ventana de Registro de usuario

Ventana de Registro

La Figura 2 muestra la ventana de registro de un usuario. Como puede apreciarse el usuario debe introducir la información indicada en los requisitos. Para la fecha de nacimiento se puede usar el componente *JCalendar*, como es mostrado en la Figura 3. Más adelante se ofrecen detalles sobre el uso de este componente. Para subir la foto puede utilizarse la ventana de diálogo *JFileChooser* de Swing que permite navegar por la estructura de directorios hasta seleccionar un archivo, como mostrado en Figura 4.



Figura 3: Introducir fecha con JCalendar

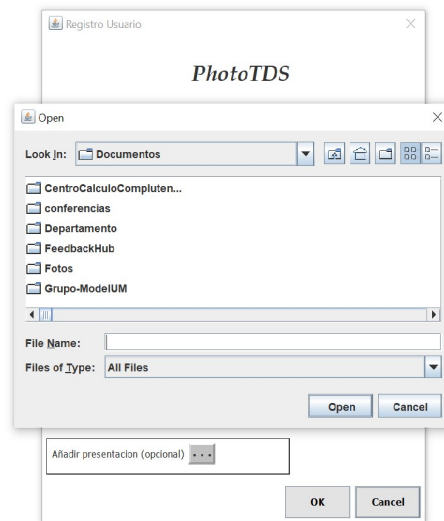


Figura 4: Seleccionar archivo con JFileChooser

Finalmente, el texto de la presentación se puede subir en una ventana que incluya un área de texto (*JTextArea*) como se muestra en ventana 5.

Registro Usuario

Texto presentación

Escribe tu presentación (máximo 200 caracteres)

Carlos Alcaraz
Age: 19 Birth Place: El Palmar, Murcia, Spain
ATP Titles: 5
Ranking 2022: 4

Fecha de Nacimiento: 04/09/2022

Añadir foto del usuario (opcional)

Añadir presentación (opcional)

OK Cancel

Figura 5: Introducir texto de presentación

Ventana Principal

La Figura 6 muestra la ventana principal de PhotoTDS. Muestra una lista de las publicaciones (fotos e historias) *más recientemente* publicadas por el usuario de la sesión o los usuarios a los que sigue. Las fotos son ordenadas por fecha de publicación de más a menos reciente y la lista contendrá un número máximo de 20. Una barra vertical de desplazamiento permitirá desplazarse por la lista, como muestra la Figura 6. Al clicar sobre una foto de la lista, se mostrará la foto en su tamaño original en una nueva ventana de diálogo.

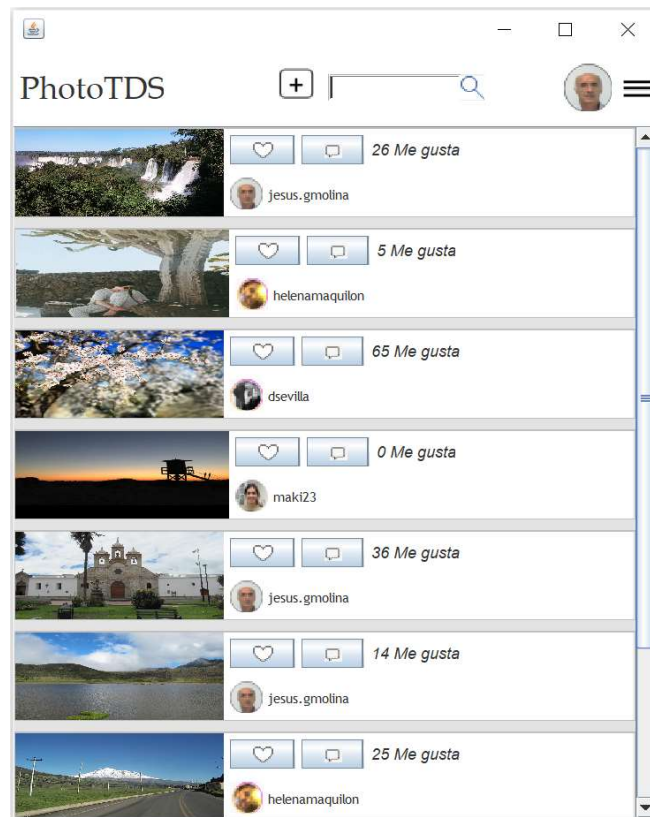


Figura 6. Ventana principal de la aplicación PhotoTDS

Junto a cada publicación, algunos controles y texto son organizados en dos filas de un panel. En la fila superior hay 3 elementos: (i) un botón para permitir indicar “me gusta” (icono forma de corazón), (ii) un botón para escribir un comentario y un texto mostrando el número de seguidores de la foto o historia. Al hacer clic sobre el botón “me gusta” se incrementará en 1 el número de “me gusta” de la foto asociada. Al hacer clic sobre el comentario se abrirá una ventana de diálogo, similar a la ventana de presentación, para incluir un comentario de un máximo de 120 caracteres. En la fila inferior, se muestra la foto del perfil del usuario que publicó la foto o historia y su nombre de usuario.

En la parte superior de la ventana principal hay una zona que incluye de izquierda a derecha: el logo de la aplicación PhotoTDS, un botón para añadir fotos, un campo de entrada para realizar búsquedas de usuarios y publicaciones, un botón mostrando la foto del perfil y un botón de menú. A continuación se describen las ventanas utilizadas para cada una de estas tareas.

Ventana Añadir Foto

Las fotos se podrán añadir usando *drag-and-drop* o bien usando una ventana JFileChooser. En el primer caso, las fotos se arrastran desde el explorador de archivos sobre el panel de “Agregar Foto”, como muestra la Figura 7. En un apartado posterior de este documento se mostrará cómo implementar un drag and drop en Swing. Como alternativa, se puede elegir el archivo de la foto clicando sobre el botón “Seleccionar de tu ordenador” que se puede ver en la ventana de la Figura 7.

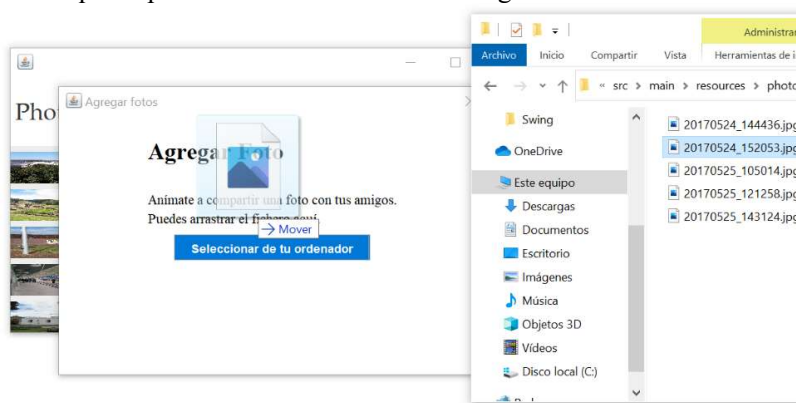


Figura 7. Añadir una publicación usando drag and drop.

Tras seleccionar la foto se abrirá un panel mostrando la foto y un área para escribir un comentario, como se muestra en la Figura 8. Nótese que el comentario incluye dos hashtags.



Figura 8. Panel con fotografía subida.

Búsqueda de usuarios y fotos

En el campo de búsqueda se podrá escribir el nombre o email de un usuario o bien un hashtag. En el primer caso se mostrará una lista de usuarios (foto y nombre completo) como muestra la figura 9. Al hacer clic sobre un elemento de la lista se mostrará la ventana del perfil del correspondiente usuario.

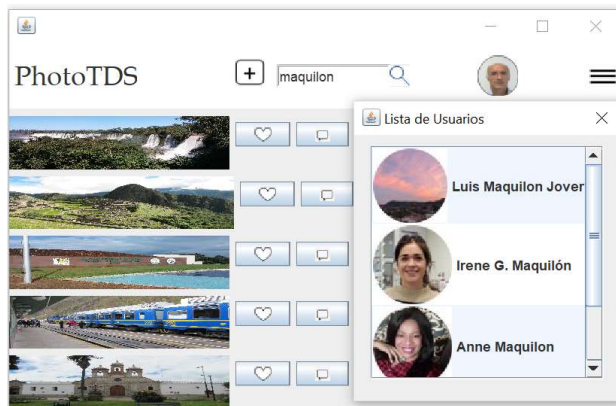


Figura 9. Lista de usuarios retornados para el patrón de búsqueda “maquilón”

El resultado de la búsqueda de un hashtag será una lista de todos los hashtags encontrados para el patrón de búsqueda introducido: por ejemplo, para #iguazu podrían aparecer #iguazu, #iguazucataratas, #puertoiguazu, #iguazufalls, o #turismoiguazu. Cada item de la lista consistirá de un hashtag seguido por el número de seguidores del usuario que público la foto, como muestra la Figura 10. Será igual en el caso de introducir varios hashtags.

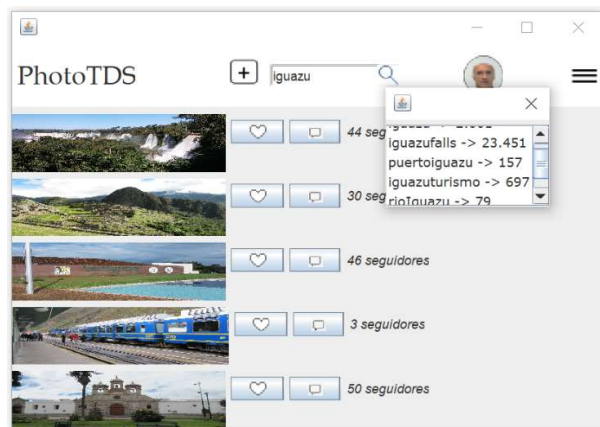


Figura 10. Lista de tags retornados para el patrón de búsqueda “iguazu”

Menú de opciones Premium

Al clicar sobre el menú (botón con icono de 3 líneas), se muestran las opciones para convertirse en usuario premium y acceder a la funcionalidad premium, como muestra la Figura 11.

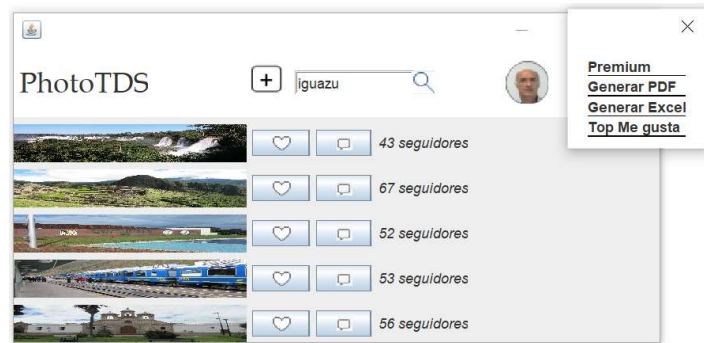


Figura 11. Opciones del menú “Premium”

Ventana de Perfil de un usuario

Haciendo clic sobre la foto del usuario en la ventana principal, la ventana cambia su aspecto para mostrar la ventana del perfil de usuario. Para ello, se realizan varios cambios sobre la estructura de la ventana principal, como muestra la Figura 12. En su zona norte se añade un botón para agregar álbumes de fotos, situado tras el botón de agregar publicación; y el panel de abajo se reemplaza por el panel del perfil que está formado por tres partes: En la superior aparece la foto del usuario a una resolución mayor y a su derecha aparecen ordenados en vertical (i) el email del usuario seguido de un botón para editar su perfil, (ii) el número de publicaciones, de seguidores y de usuarios seguidos, y (iii) el nombre del usuario. Debajo aparecen dos botones centrados que permiten seleccionar si en el panel inferior se muestra una matriz de iconos de fotos o de álbumes.

Al hacer clic sobre una foto se mostrará a su tamaño original y un área para añadir un comentario (no se consideran hashtags) en una forma similar a la ventana de la Figura 8, pero con un sólo botón OK para cerrar. Al clicar sobre el botón “Editar Perfil” se abre la ventana de Registro (Figura 2) que permitirá cambiar los campos que no son fijos: foto, presentación y contraseña, los demás estarán deshabilitados.

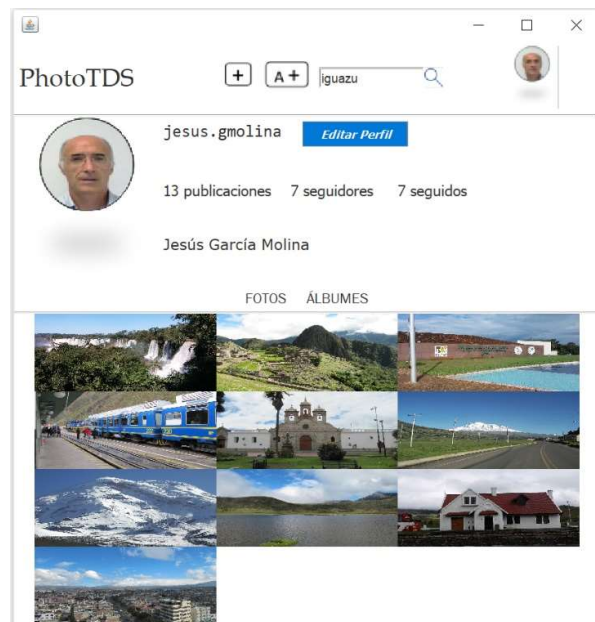


Figura 12. Ventana Perfil de Usuario

Álbumes de fotos

Al clicar sobre el botón “A+” de la ventana de perfil, aparecerá una ventana de diálogo que pedirá el nombre de un nuevo álbum. Se comprobará que el usuario no ha creado ya un álbum con el mismo nombre. Entonces se abrirán las ventanas para seleccionar y agregar una primera foto (Figuras 7 y 8).

Al clicar sobre el botón “ÁLBUMES”, aparecerá la colección de álbumes del usuario en forma de matriz 4x4, y para cada álbum se muestra el icono de la primera foto. Al hacer clic sobre una de ellas aparecerá la colección de fotos y se podrá clicar sobre el botón “+” para agregar nuevas fotos, o bien eliminar fotos del álbum. En el caso de ver una foto de un álbum, se muestra el nombre del álbum al que pertenece, como muestra la Figura 13.



Figura 13. Ver una foto del álbum “Subida al Chimborazo”

Seguir a un usuario

Como se ha indicado arriba, cuando un usuario busca a otro usuario obtiene una lista de usuarios cuyo nombre contiene el texto de búsqueda. Al seleccionar un elemento de la lista se mostrará la ventana de perfil para ese elemento, de la misma forma que se ha comentado arriba. En este caso, el botón “Editar Perfil” se reemplaza por un botón “Seguir” para indicar que el usuario desea seguir al usuario cuyo perfil se muestra.

Ventana para aplicar descuentos y ver la lista de fotos con más “me gusta” de usuario premium

Se deben aplicar, al menos, dos tipos de descuentos: por edad y por el número de “me gusta” recibidos. En el primer caso se considerará un porcentaje para jóvenes comprendidos en un rango de edad y otro para mayores de 65 años. En el segundo caso, el descuento se aplicará si tiene un número de fotos que superan una cantidad de “me gusta”. Al clicar la opción “Premium” del menú de la ventana principal, se mostrará una lista con las opciones de descuento y una vez se elija una aparecerá una ventana de diálogo de notificación que indicará la cantidad pagada y el usuario podrá aceptar o no. En cuanto a la lista de fotos top en “me gusta” recibidos, se mostrará una ventana de diálogo con una lista cuyos items son iconos de fotos y al lado el número de “me gusta” recibidos.

Eliminar fotos

Cuando se muestra una matriz de fotos o álbumes en el perfil de usuario se podrá eliminar una publicación mediante un menú contextual como el que muestra la Figura 14. En ella aparece el menú para la primera foto de la matriz. En ese menú también podría aparecer la opción “Ver foto”. El menú aparecerá pulsando el botón derecho del ratón.

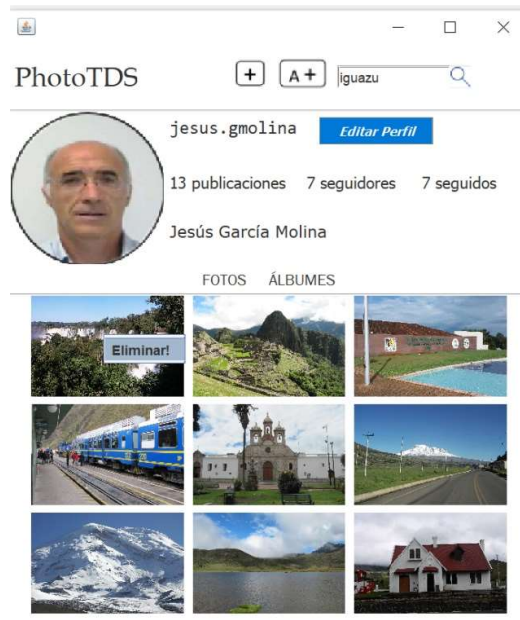


Figura 14. Menú contextual para eliminar una foto

3. Librerías Java y componentes Java Beans utilizados

Componente Java Bean para mostrar el calendario

Para introducir la fecha de nacimiento en el registro, como se mostró en la Figura 3, se puede usar el componente Java Bean JCalendar que puede ser descargado en <https://toedter.com/jcalendar/>. Abajo se muestra un ejemplo de su uso. Se ha elegido la clase JDateChooser proporcionada por JCalendar para mostrar el campo de entrada de la fecha con un botón que al ser clicado muestra un calendario que permite elegir año, mes y día. El método getDate() obtiene la fecha seleccionada como una instancia de Date.

```
// mostrar el componente en un panel llamado panelFechaNacim
JDateChooser dateChooser = new JDateChooser();
dateChooser.setFont(new Font("Tahoma", Font.PLAIN, 10));
dateChooser.setDateFormatString("dd/MM/yyyy");

//obtener la fecha introducida y mostrarla con una etiqueta
Date date = dateChooser.getDate();
DateFormat df = DateFormat.getDateInstance();
label.setText(df.format(date));
```

APIs para generar archivos PDF y Excel por usuarios Premium

Para crear una hoja Excel se puede usar la librería (API) *Apache POI* (<https://poi.apache.org/>). En la Web se pueden encontrar numerosos ejemplos de uso de esa librería para crear un archivo Excel, como <https://bit.ly/3w113L0> y <https://www.javatpoint.com/java-create-excel-file>.

También existen un buen número de APIs para crear archivos PDF en Java, siendo *iText* (<https://itextpdf.com/>) una de las más populares. Una descripción del uso de esta API se puede encontrar en <https://bit.ly/3B0bvA8>.

4. Utilidades de Swing

Drag-and-drop en Swing

A continuación se incluye código que muestra cómo realizar una operación de drag and drop en Swing, el cual ha sido escrito a partir del código dado en el post de Stackoverflow <https://bit.ly/3cS9aOE>.

```
JEditorPane editorPane = new JEditorPane();
contentPane.add(editorPane);
editorPane.setContentType("text/html");
editorPane.setText("<h1>Agregar Foto</h1><p>¡Anímate a compartir una foto con tus amigos.  
<br>Puedes arrastrar el fichero aquí. </p>");
editorPane.setEditable(false);
editorPane.setDropTarget(new DropTarget() {
    public synchronized void drop(DropTargetDropEvent evt) {
        try {
            evt.acceptDrop(DnDConstants.ACTION_COPY);
            List<File> droppedFiles = (List<File>)
                evt.getTransferable().getTransferData(DataFlavor.javaFileListFlavor);
            for (File file : droppedFiles) {
                System.out.println(file.getPath());
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
});
```

La clase JEditorPane es una de las que permite aplicar drag-and-drop en su panel. En este caso, el manejador del evento arrastrar (DropTargetDropEvent) se encarga de mostrar el nombre del archivo por consola.

Listas con ítems a medida

Una lista (instancia de JList<>) visualiza sus ítems utilizando un objeto ListCellRenderer que por defecto permite visualizar un texto y/o icono pudiéndose determinar cuál aparece primero. El renderer visualiza Object aplicando el método toString(). Si se desea una forma particular de visualización diferente al uso del toString() se debe crear una clase que implemente ListCellRenderer. En las clases prácticas de Swing se discutirá el ejemplo que se incluye en el Anexo 1.

Menú contextual

Abajo se muestra el código para crear un menú contextual (JPopupMenu) con los ítems “cut”, “copy” y “paste”.

```
JPopupMenu menu = new JPopupMenu();
JMenuItem cut = new JMenuItem("Cut");
JMenuItem copy = new JMenuItem("Copy");
JMenuItem paste = new JMenuItem("Paste");

menu.add(open);
menu.add(cut);
menu.add(copy);
menu.add(paste);
```

El menú contextual se asocia a un componente, en este caso un panel, agregando un *listener* de tipo “mouse” al componente.

```
panelTexto.addMouseListener(new PopMenuEdicionListener());
```

El listener podría implementarse mediante una clase anónima o una clase separada como se muestra abajo.

```

public class PopMenuEdicionListener extends MouseAdapter {
    public void mousePressed(MouseEvent e) {
        if (e.isPopupTrigger())
            mostrarMenu(e);
    }
    public void mouseReleased(MouseEvent e) {
        if (e.isPopupTrigger())
            mostrarMenu(e);
    }
    private void mostrarMenu(MouseEvent e) {
        PopUpMenuFoto menu = new PopUpMenuFoto();
        menu.show(e.getComponent(), e.getX(), e.getY());
    }
}

```

Directorio de las fotos

Las fotos serán almacenadas en una carpeta del proyecto Maven y deben ser cargadas mediante un acceso relativo.

Arquitectura de la aplicación

Se organizará la aplicación de acuerdo con un modelo de tres capas (*Presentación, Lógica de Negocio y Almacenamiento*) que será descrito en un seminario de prácticas, aunque los aspectos básicos ya se discutirán en el primer tema de teoría. Para ilustrar cómo aplicar esta arquitectura, se entregará a los alumnos un proyecto con un caso práctico que sirva de ejemplo. Se utilizará la tecnología Java Swing para la implementación de la interfaz de usuario y un servicio de persistencia propio (implementado por el profesor Javier Bermúdez) para el almacenamiento de los datos de la aplicación. Se entregará documentación sobre el uso de estas tecnologías y se explicarán además en seminarios de prácticas.

Se utilizará el patrón DAO para desacoplar la capa de almacenamiento del resto de la aplicación, de acuerdo con las explicaciones proporcionadas en las clases de teoría y prácticas. Cuando se entregue a los alumnos el diagrama de clases de la aplicación se indicarán cuáles son las clases persistentes.

Más adelante se entregará a los alumnos la especificación de un sencillo componente Java Bean que deberán implementar para ser usado en PhotoTDS.

5. Otros aspectos

Grupos de prácticas

Los alumnos deberán formar **grupos de dos**. Los dos miembros de un grupo deberán pertenecer **al mismo grupo de teoría**. Se debería organizar el trabajo de forma que los dos alumnos **contribuyan por igual y que cada alumno tenga que implementar clases de cada capa**, de modo que cada uno de ellos tenga los conocimientos exigidos de la práctica (esto es, debería ser capaz de realizar la implementación en solitario). El tamaño de la práctica está pensado suponiendo que los alumnos se reparten por igual el esfuerzo.

Gestión del proyecto

Cada grupo deberá crear un proyecto **Maven** que debe ser almacenado en un repositorio Git (en concreto **GitHub**). Siempre que sea posible, las dependencias a las librerías externas se manejarán con Maven. Los alumnos deben invitar a su profesor de prácticas al repositorio creado para que pueda acceder a su código. Se dedicará un seminario de prácticas a explicar el uso de Maven y otro al de Git.

Se recomienda que cada grupo aplique una estrategia de desarrollo iterativo completando de forma incremental la funcionalidad de PhotoTDS y escribiendo el código necesario para cada capa: a) diseño e implementación de la interfaz de usuario (el diseño de ventanas expuesto arriba pretende facilitar el trabajo de los alumnos); b)

diseño e implementación del controlador; c) implementación de la lógica del negocio; d) implementación de la persistencia. El desarrollo del componente se debería dejar para el final, también a la implementación de las pruebas unitarias Junit (al menos, escribir cinco pruebas para una o varias clases del modelo).

Patrones de Diseño

Los conocimientos sobre patrones se valoran principalmente en el examen de teoría. No obstante, la realización de esta práctica conlleva el uso de algunos patrones aplicados directamente por los alumnos o de forma indirecta al utilizar algún componente de Swing o del software proporcionado por los profesores. El alumno deberá indicar en la documentación qué patrones ha usado y comentar muy brevemente su uso.

Código

El alumno debe esforzarse en escribir código simple y fácil de leer y comprender. Los mecanismos de Java 8 discutidos en clase (**expresiones lambda**, **streams** y **Optional**) deben ser empleados en aquellos puntos del código en los que se sea conveniente. El alumno no debe cometer errores relacionados con los patrones GRASP y principio separación modelo-vista.

6. Hitos

1. Realización del **diagrama de clases inicial**. La fecha tope de entrega será el sábado **1 de octubre**. Se creará una tarea en el Aula Virtual para que cada grupo pueda subir su diagrama de clases junto a una explicación de los detalles que considere oportuno (**un único envío por grupo**). El archivo PDF entregado debe incluir los nombres de los alumnos y del profesor de prácticas.
2. En los seminarios de prácticas o clase de teoría de la semana que comienza el lunes 3 de octubre se discutirá el diagrama de clases entregado por los profesores y que los alumnos deberían utilizar en la implementación del caso práctico.
3. Los grupos que tengan previsto entregar en la convocatoria de Enero pueden realizar una entrevista de seguimiento con el profesor tutor en la **última semana de noviembre o primera de diciembre**. Para realizar esta entrevista el grupo debería haber trabajado en cada capa.
4. La fecha tope de entrega será el **miércoles 18 de enero** (el examen de teoría está previsto para el lunes 16 de enero). Se creará una **tarea** en el Aula Virtual para la entrega de un archivo rar conteniendo un archivo PDF con la **documentación** comentada abajo y los **dos proyectos Eclipse** con la aplicación PhotoTDS y el componente creado, exportados como ficheros zip.

Documentación

Constará de las siguientes partes:

1. Índice de contenidos
2. Diagrama de clases del dominio.
3. Un diagrama de interacción UML (secuencia o comunicación) para la operación añadir una foto a un álbum del usuario.
4. Una breve explicación de la arquitectura de la aplicación y decisiones de diseño que se consideren de interés para la comprensión del trabajo. Se deberían comentar los cambios significativos realizados sobre la funcionalidad propuesta.
5. Breve explicación de los patrones de diseño utilizados (directamente por haberlos implementado o indirectamente por formar parte de alguna librería Java usada como Swing o el servicio de persistencia proporcionado por los profesores de la asignatura).
6. Breve explicación sobre los componentes utilizados.
7. Tests unitarios implementados.
8. Un breve manual de usuario que explique cómo usar la aplicación.
9. Observaciones finales (deben incluir una **estimación** del tiempo dedicado a la práctica)

Evaluación

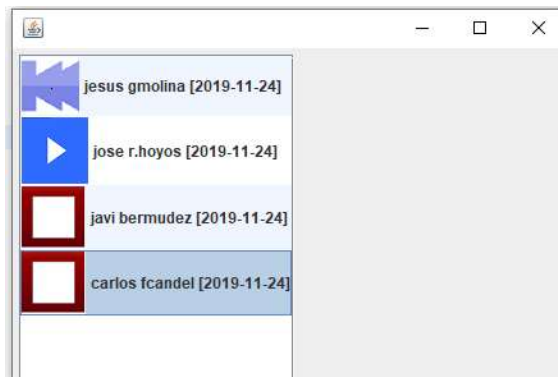
Cada parte se valorará con el siguiente porcentaje:

- Modelo del dominio de clases (5%)
- Diseño e Implementación de la interfaz de usuario (25%)
- Calidad del código (45%).
 - Legibilidad, ausencia de lógica condicional compleja, ausencia de métodos largos
 - Aplicar principios básicos de programación OO
 - Utilización de patrones GRASP,
 - Buen uso de expresiones lambda, Stream y Optional
 - Aplicar patrones de diseño cuando sea necesario
- Creación del componente Java bean (10%)
- Creación de test JUnit (5%),
- Documentación entregada (10%).

Es obligatorio el uso de Git y Maven.

ANEXO 1. Ejemplo de ListCellRenderer

El ejemplo de abajo ilustra un ejemplo de renderer para una lista que visualiza ítems con información sobre un objeto Usuario (una imagen y su nombre) y la fecha actual.



```
JList<Usuario> listaUsuarios;  
listaUsuarios.setCellRenderer(createListRenderer());  
JScrollPane pane = new JScrollPane(listaUsuarios);  
panel.add(pane);  
  
private static ListCellRenderer<? super Usuario> createListRenderer() {  
    return new DefaultListCellRenderer() {  
        private Color background = new Color(0, 100, 255, 15);  
        private Color defaultBackground = (Color) UIManager.get("List.background");  
        @Override  
        public Component getListCellRendererComponent(JList<?> list, Object value,  
                                                       int index, boolean isSelected, boolean cellHasFocus)  
        {  
            Component c = super.getListCellRendererComponent(  
                list, value, index, isSelected, cellHasFocus);  
            if (c instanceof JLabel) {  
                JLabel label = (JLabel) c;  
                Usuario usuario = (Usuario) value;  
                label.setIcon(new ImageIcon(  
                    VistaPrincipal.class.getResource(usuario.getImagen())));  
                label.setText(String.format("%s [%s]", usuario.getNombre(),  
                    usuario.getFecha()));  
            }  
        }  
    };  
}
```

```
                LocalDate.now().toString())
        if (!isSelected) {
            label.setBackground(
                index % 2 == 0 ? background : defaultBackground);
        }
    }
    return c;
}
};
```

Más información en los links:

<https://www.logicbig.com/tutorials/java-swing/list.html>

<https://www.codejava.net/java-se/swing/jlist-custom-renderer-example>

<https://docs.oracle.com/javase/tutorial/uiswing/components/list.html#renderer>