

# SISTEMAS MULTIMEDIA

## *Práctica de Evaluación*

Documentación sobre la práctica final de la asignatura, consistente en un sistema de dibujo similar al que ofrece el sistema operativo Windows, incluyendo además algunas operaciones para aplicar sobre imágenes.

*Spaint – Draw it*

Aarón Rosas Rodríguez

Ingeniería Técnica en Informática de Sistemas

Curso 2010-2011



# SISTEMAS MULTIMEDIA

## *Práctica de Evaluación*

### INTRODUCCIÓN

Utilizando como lenguaje de programación *Java*, con la potencia ofrecida por la librería *Swing* y basándose en la filosofía de la *PDO* (Programación Dirigida a Objetos) fue posible el desarrollo de la aplicación *Spaint*.

La aplicación posibilita la elaboración de dibujos a partir de formas prediseñadas a las cuales se les puede establecer diferentes atributos permitiendo así un gran número de combinaciones y variedad. También se invita al usuario a transformar la imagen y el lienzo sobre el que está trabajando a través de una serie de potentes operaciones. Todo esto accesible mediante una interfaz gráfica simple, amena e intuitiva.

Los detalles sobre los requisitos a cumplir, análisis y diseño serán tratados a continuación, con algunas explicaciones y comentarios esclarecedores.

### REQUISITOS

Veremos qué acciones son posibles llevar a cabo con *Spaint* a la hora de trabajar sobre un lienzo de dibujo. Muchos de los requisitos han sido exigidos para cumplir con la práctica de la asignatura pero cabe destacar también la presencia de otro grupo de requerimientos que han sido desarrollados como extra y dan mayor libertad y comodidad al usuario de *Spaint*.

Disponemos de diversas ventanas internas a nuestra aplicación principal:

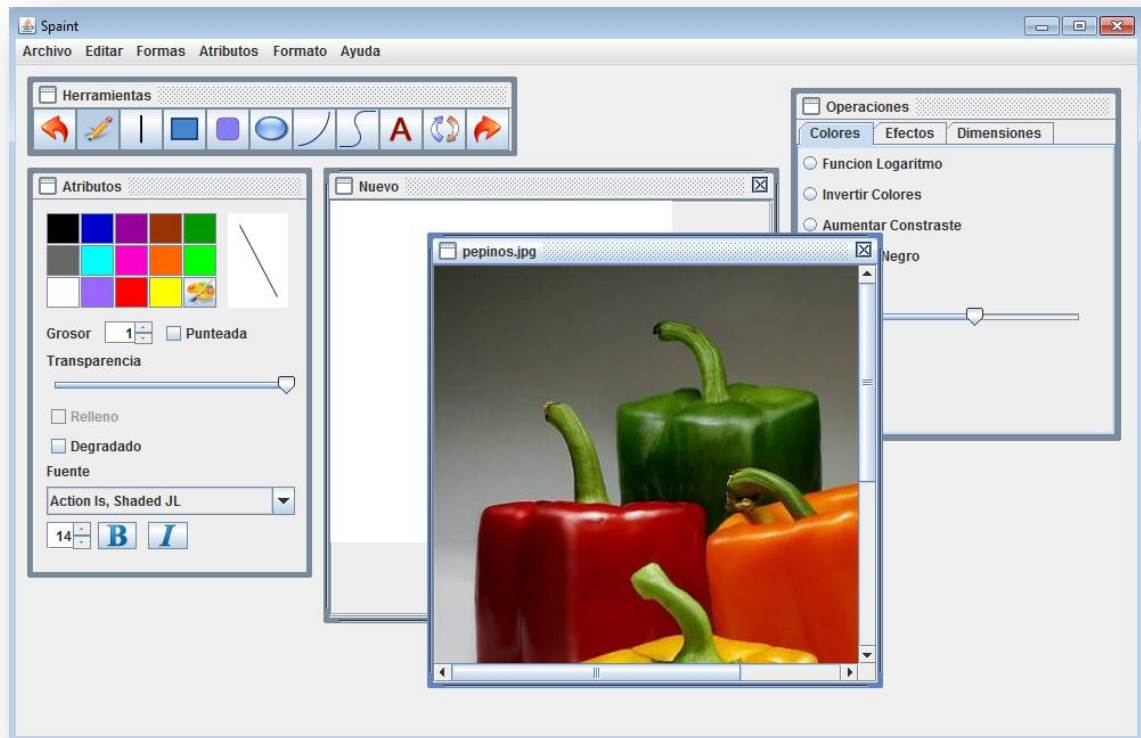
- Herramienta de dibujo
- Atributos de dibujo
- Imágenes
- Operaciones sobre imágenes

Como se enunciaba, la interfaz gráfica ha sido elaborada a partir de la biblioteca *Swing* de *Java*. Gran cantidad de componentes o clases de esta colección han sido empleados, son de mención importante los siguientes:

- |                       |                      |                        |
|-----------------------|----------------------|------------------------|
| • Panel               | • Botón 2 posiciones | • Spinner              |
| • Panel con pestañas  | • Casilla activación | • Barra de Menú        |
| • Panel desplazamie.  | • Botón de opción    | • Elemento de Menú     |
| • Panel de escritorio | • Grupo de Botones   | • Selector de Archivos |
| • Ventana interna     | • Lista desplegable  | • Selector de Color    |
| • Etiqueta            | • Campo de Texto     | • Panel de Opciones    |
| • Botón               | • Deslizador         | • Diálogo              |



La interfaz final con la cual trabajará el usuario resulta así:



## Ventana Herramienta de Dibujo

Es aquí donde el usuario tendrá constancia de la herramienta que está utilizando para pintar, además de poder cambiarla en todo momento. Dispone también de tres botones añadidos para manejar formas ya pintadas, son deshacer, rehacer y actualizar.

## Formas

Se presenta una herramienta para poder pintar:

- Trazo Libre
- Línea
- Rectángulo
- Rectángulo redondeado
- Elipse
- Curva con 1 punto de control
- Curva con 2 puntos de control
- Texto formateado

## Deshacer y Rehacer

Botones responsables de la anulación de forma más nueva o repintado de la forma anulada más reciente.

## Actualizar

Herramienta un tanto novedosa, al seleccionarla vamos a poder cambiar los atributos de la ultima forma pintada. Además, podemos pinchar en otra que no tiene ser necesariamente la última, y los cambios se producirán sobre ella.

## Ventana Atributos de Dibujo

En esta ventana interna encontramos todas las posibles características que podemos variar para nuestras formas. Sus elementos son:

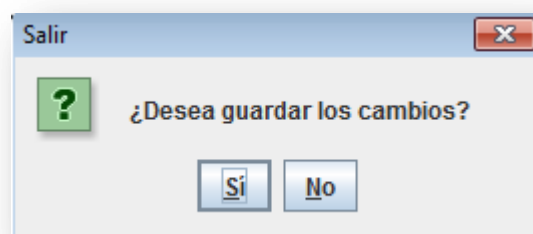
- Paleta de Colores: para elegir el color principal y secundario.
  - Pinchando con el botón izquierdo estaremos seleccionando el color principal, si lo hacemos con el derecho definiremos el secundario.
- Vista Previa: una previsualización de los atributos fijados.
- Selector de grosor: define el grosor del trazo.
- Tipo de trazo: punteado o liso.
- Variador de Transparencia: cambiar la transparencia de los colores.
- Tipo de relleno: determina si rellenar la forma.
- Degradado: ofrece cuatro tipos de degradados para el color.
- Selector de fuente: lista de fuentes disponibles para escribir texto.
- Tamaño de la fuente: tamaño en píxeles de la fuente con la que pintar caracteres.
- Estilo de la fuente: negrita y/o cursiva.

## Ventana Imagen

La base de nuestra aplicación. Podemos tener abiertas tantas ventanas de este tipo como imágenes queramos tratar, Cada una de ellas tendrá como título el nombre de la imagen o “Nuevo” si se trata de una creación a partir de un lienzo en blanco.

Un **Lienzo** interno a cada ventana será el encargado de gestionar todas las operaciones y dibujos que deseemos llevar a cabo. En la sección de Diseño, se estudia en profundidad el anterior nombrado Lienzo, sus posibilidades y manera de lograrlo.

Si cerramos alguna de estas ventanas, se lanzará un diálogo que nos ofrecerá guardar los cambios.



## Ventana Operaciones sobre imágenes

Un completo panel de control para seleccionar la operación que deseamos aplicar y de qué manera. Su diseño es compacto y va cambiando conforme a lo que deseemos hacer sobre nuestro lienzo actual.

Desaparece, en el momento que aplicamos alguna operación, para dar paso a una **Ventana de Confirmación** que pedirá al usuario a modo de diálogo, si realmente está seguro de aplicar dicha transformación. La ventana de la imagen sobre la que estemos trabajando, queda bloqueada y no se podrán realizar cambios hasta que se responda a la ventana de confirmación. ¿Por qué esta filosofía? En la parte de Diseño se explicará ampliamente.

### Operaciones Posibles

Aparecen agrupadas según su acción y pueden necesitar parámetros que serán solicitados al usuario de forma interactiva.

- Función logaritmo
- Invertir Colores
- Aumentar Contraste
- Blanco y Negro
- Gamma
- Brillo
- Efecto Dibujo
- Efecto Relieve
- Detección de Fronteras
- Enfocar
- Desenfocar (2 variantes)
- Rotar
- Zoom
- Estirar
- Trasladar
- Redimensionar

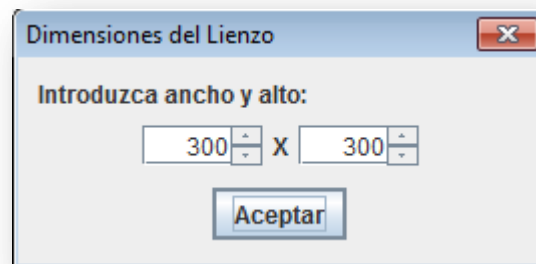
### Barra de Menú

Una completa barra de menú permite interactuar con el programa con el mismo efecto que si lo hiciéramos a través de las ventanas de herramientas y dibujo. Además los distintos elementos de menú están dotados de teclas aceleradoras para facilitar el uso al usuario.

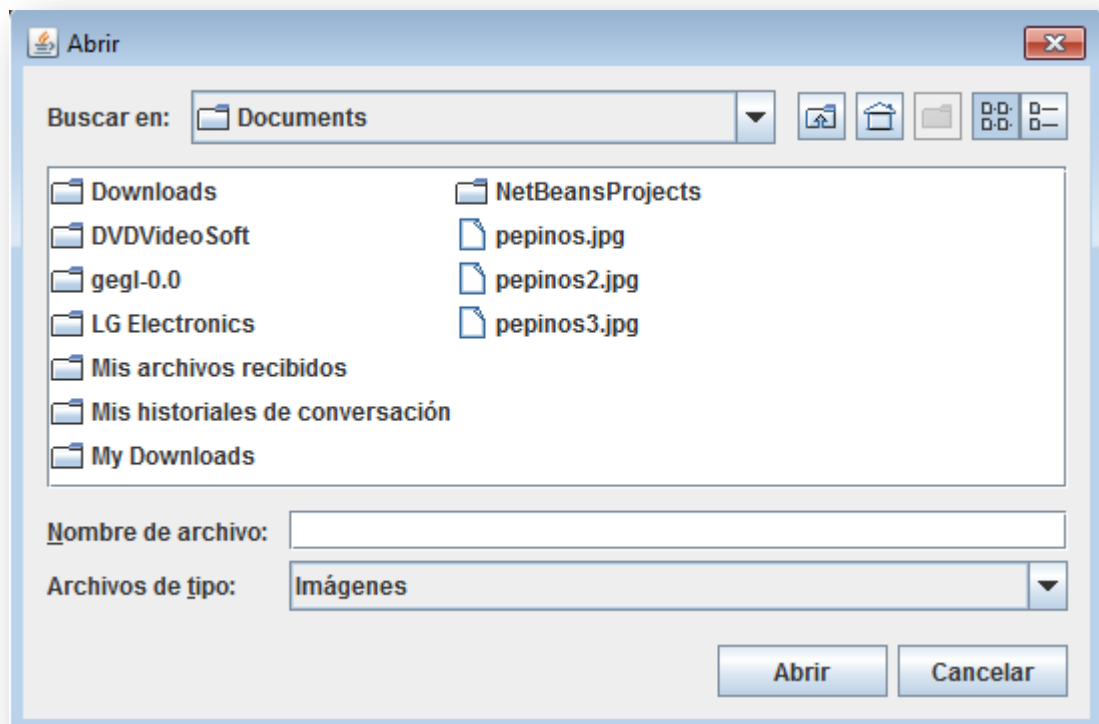
## Menú Archivo

Empleado en la mayoría de aplicaciones para gestionar el trabajo actual. Encontramos en esta sección los botones de:

- Nuevo: cargará una nueva ventana de imagen; esta será blanca con el tamaño que le especifiquemos en el diálogo que se lanzará al pinchar sobre este elemento de menú.



- Abrir: lanza un diálogo selector de archivos con filtros de archivos personalizados, con el objeto de que el usuario seleccione la imagen que desea cargar para su tratamiento.



- Guardar: igual que en caso anterior, el usuario percibe un diálogo en el que puede especificar la manera de guardar su creación, ya sea sobre una imagen nueva o alguna ya existente.
- Salir: cierra la aplicación

## ANÁLISIS

Para construir una aplicación robusta y fácil de mantener se propuso una modulación bien diferenciada, encapsulada y protegida. Podemos distinguir tres componentes principales para nuestro programa:

- Jerarquía de Formas

Un ejemplo claro de polimorfismo que gracias al lenguaje *Java* pudo ser desarrollado e hizo que la construcción del resto de la aplicación fuese más clara.

- Lienzo

Se trata del núcleo, el componente fundamental de *Spaint*. Tiene registrados una serie de atributos que determinarán la forma de pintar sobre la imagen de fondo que se halle.

Básicamente, manejando los eventos de pinchar, arrastrar y soltar el ratón, será posible dibujar e incluso mover y editar formas ya dibujadas.

Una serie de mensajes pueden ser enviados hacia este componente para operar la imagen actual, manejar el historial de figuras o texto dibujados, establecer atributos de dibujo...

- Interfaz

Su necesidad radica en el hecho de aportar una aplicación de escritorio, que sea interactiva y amigable con el usuario, controlando totalmente el lienzo con el que se esté trabajando.

Las características de esta se detallan conforme se comentan los requerimientos de la aplicación, en la parte de Requisitos.

La jerarquía será utilizada por el Lienzo para pintar sobre él las formas enunciadas en el apartado de Requisitos. La interfaz hará uso de Lienzo para lograr el sistema de dibujo, facilitando al usuario final la manejabilidad.

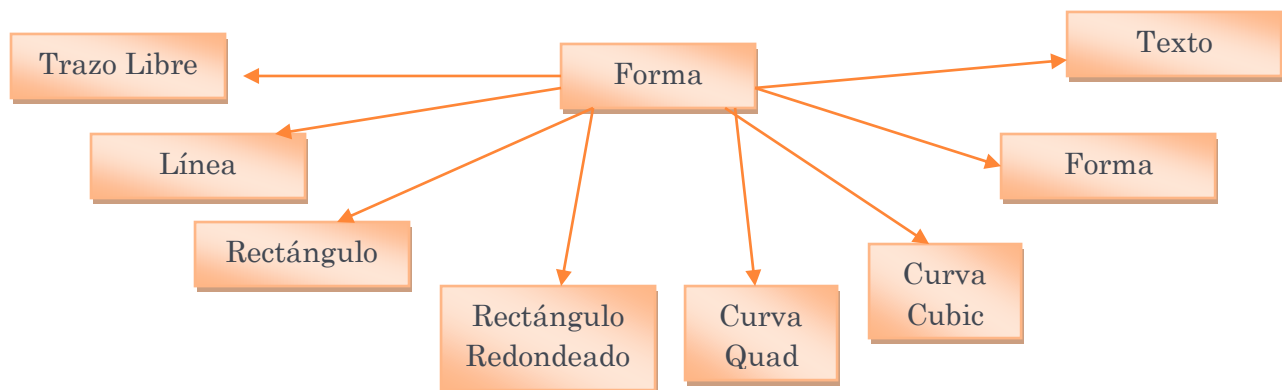
## DISEÑO

En esta sección se trata de dar una idea de alguna manera detallada sobre los aspectos tenidos en cuenta y el diseño de cada uno de los componentes diferenciados en la parte de Análisis.

Aquellas facultades de la aplicación cuya cumplimentación fue más laboriosa, serán debidamente ilustradas.

## Jerarquía de Formas

Para entender rápida y fácilmente el diseño de esta jerarquía de objetos, se propone la siguiente ilustración.



Se presenta la **superclase Forma** de la cual heredan el resto. Un objeto de este tipo puede guardar información sobre posición, dimensiones, color, trazo, estilo de relleno y tipo de forma.

El resto de **subclases** que heredan de Forma, dependiendo del tipo, algunos sobrecargan métodos de la superclase en caso de que queden incompletos. También podemos encontrar tipos de formas que añadan nuevos métodos a los ofrecidos por la superclase, por ser de cierta especialidad.

Los beneficios de este diseño, son apreciados a la hora de manejar todos los tipos de formas. Conseguimos volcar en un objeto de tipo Forma, cualquier tipo de figura o texto, pudiendo modificar sus campos con mensajes únicos al objeto. La claridad del código es enormemente mejorada aprovechando este polimorfismo que se propone.

## Lienzo

A continuación serán destacados los aspectos más importantes sobre el diseño del componente troncal de la aplicación.

- Tres imágenes

Para permitir al usuario deshacer formas, operar sobre la imagen y previsualizar operaciones fueron necesarias tres imágenes:

- Original: considerada como la inicial
- Original + Formas: donde se va pintando
- Original + Formas + Operación: previsualización de operación.

- Bloqueo del Lienzo

Se restringe la edición de la imagen. Esto ocurre cuando establecemos una operación sobre la imagen, el lienzo en este caso muestra una previsualización y



hasta que no reciba el mensaje correspondiente a ***aplicar operación*** no volverá a ser desbloqueado.

El hecho de aplicar una operación sobre la imagen (ya previsualizada) conlleva perder la posibilidad de deshacerla, se establecerá como imagen original, la imagen nueva ya transformada.

Esta regla se tomó por considerar muy costoso en cuanto a memoria, mantener una colección de imágenes correspondientes al historial de operaciones realizadas sobre la original.

- Enumerados y Variables estáticas

Una serie de variables correspondientes a matrices para aplicar diversas operaciones sobre imágenes, fueron declaradas como estáticas para permitir su compartición por todos los lienzos que se instancien. Al igual ocurre con los enumerados que se desarrollaron para hacer referencia a distintos tipos de operaciones, degradados,...

- Mover y Redimensionar Polígonos

Cuando se dibujo un polígono, ya sea un rectángulo, rectángulo redondeado o elipse, se ofrece al usuario la posibilidad de moverlo y redimensionarlo. La manera de hacerlo es fácil e intuitiva: se muestra un marco con tres puntos resaltados (centro, esquina superior izquierda y esquina inferior derecha), donde con tan solo pinchar en alguno de ellos y arrastrándolo podemos conseguir el resultado deseado.

- Mensajes para cambiar atributos

Se disponen de los mensajes necesarios para cambiar los atributos de dibujo que nuestro lienzo concierne.

- Mejoras sobre el ***Rendering***

El tipo de trazo se construye mejorando el *Rendering* para percibir formas con bordes más suaves y naturales.

- Editar formas

Podemos poner el lienzo en modo ***Editar Forma***. En esta situación podemos cambiar los atributos de la última forma dibujada con los mensajes que anteriormente se enuncian, además de poder seleccionar que forma editar pinchando sobre ella.

Para permitir esta selección nos apoyamos de la función ***contains*** de la clase ***Shape*** de *Java*.

- Mensajes para operar con la imagen

Se suele seguir el estándar: nombre del mensaje relacionado con el tipo de operación. En caso de precisar parámetros, puede ser una variable del correspondiente tipo enumerado y específicos de cada caso.

## Interfaz

Sobre esta parte del proyecto cabe destacar la característica esconder componentes o ventanas que no son necesarios manejar hasta cierto momento, consiguiendo así dotar a la aplicación de cierta interactividad y ofrecer una vista más sencilla y menos abrumadora. Ese ha sido el principal criterio a la hora de su desarrollo.

Una clase interna ***Manejador de escritorio*** será la encargada de gestionar todas las ventanas internas para poder trabajar correctamente con todas las ventanas de imágenes que deseemos abrir.

La estructura puede ser conocida en la parte de Requisitos, donde se relaciona con los requerimientos a cumplir.

## CODIFICACIÓN

Se halla guardada en el proyecto, podemos explorarla yendo a la carpeta ***dist*** dentro de ***javadoc*** abriendo el documento ***html index***.