

Introducción a las GUI – Java 2D básico

PROGRAMACIÓN INTERACTIVA

ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Qué debes saber al finalizar la clase!!



- ✓ *Principios básicos de Java 2D*
 - ✓ *Objeto Graphics, manejo de color, pintado de líneas, rectángulos, óvalos, texto e imágenes.*
 - ✓ *Funciones paintComponent(...), repaint().*
- ✓ *Uso de eventos del teclado*
- ✓ *Uso de clases adaptadoras para Manejo de Eventos*

Java 2D

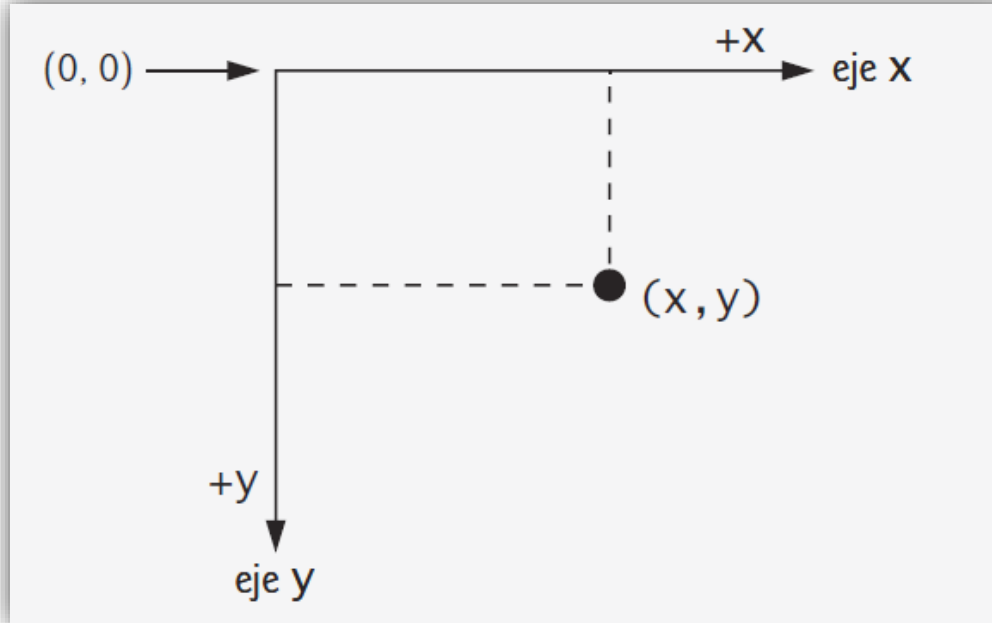


- ✓ *Es el API de Java que extiende AWT para dibujar bidimensional.*
- ✓ **Clase *Graphics*:**
 - ✓ *Clase base* para todos los contextos en los que pueda dibujarse cualquier cosa.
 - ✓ *Graphics encapsula la información* de estado necesaria para las operaciones de representación básicas que admite Java.
 - ✓ *JVM crea una instancia de Graphics* y la suministra al programa en ejecución.

Java 2D – Class Graphics



✓ *Coordenadas de dibujo*



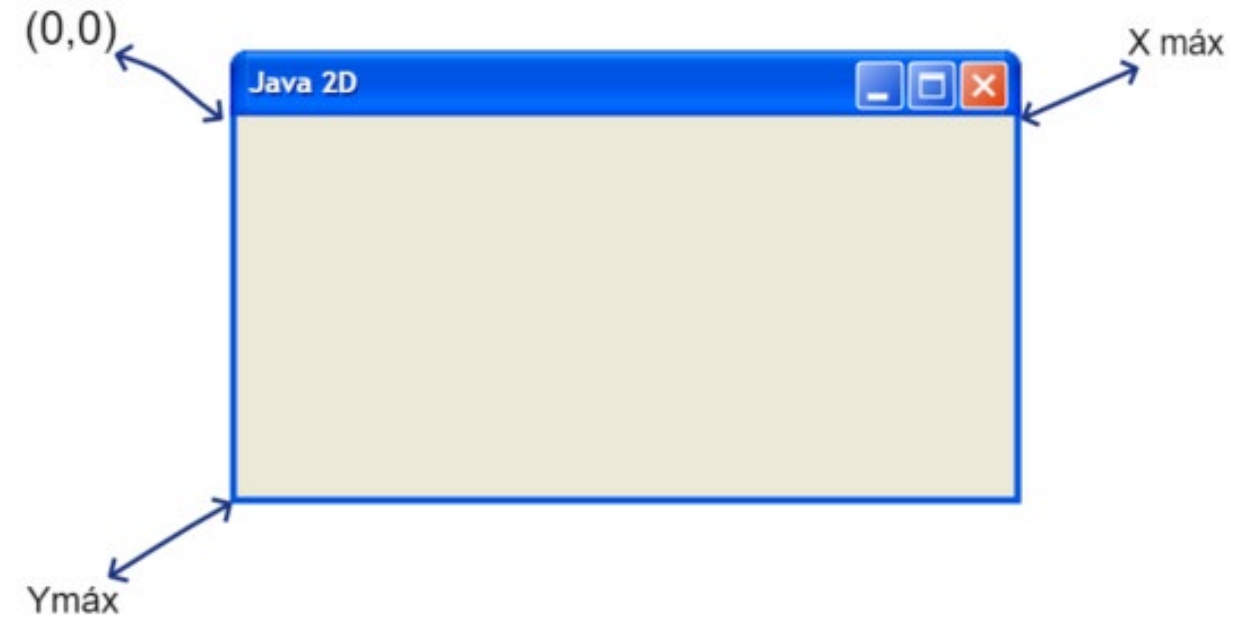
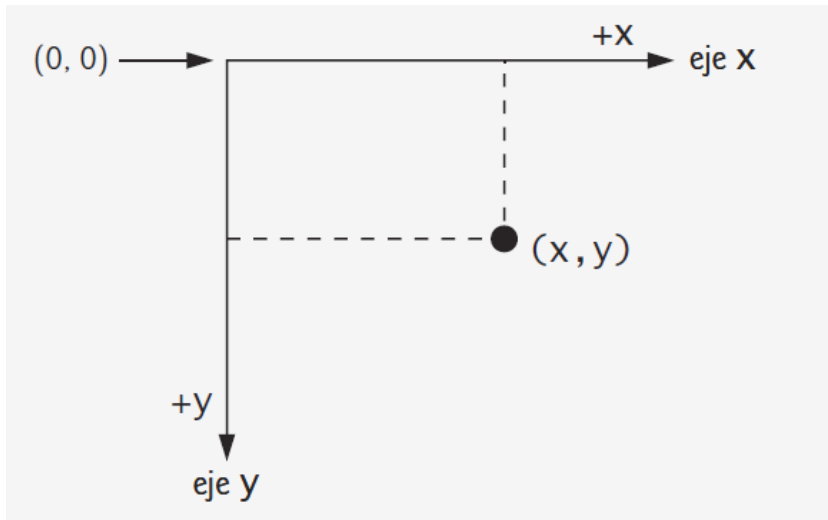
***Sistema de coordenadas en Java
medido en Pixeles.***



Java 2D – Class Graphics



✓ Coordenadas de dibujo



Java 2D – Class Graphics



- ✓ **Qué se puede dibujar y sobre qué se dibuja**
- ✓ **Dibujo sobre un JComponent – JPanel**
 - ✓ *public void paintComponent(Graphics g)*
Aquí escribe el código que representa el dibujo.
 - ✓ *public void repaint()*
Usa este método para invocar el dibujado.
- ✓ **Puedo dibujar:**
 - ✓ **Texto**
 - ✓ **Líneas**
 - ✓ **Rectángulos**
 - ✓ **Círculos**

Java 2D – Class Graphics



✓ Clases Auxiliares

✓ Clase **Color**:

Declara los métodos y constantes para manipular colores.

Constante de Color	Valor RGB
<code>public final static Color RED</code>	255, 0, 0
<code>public final static Color GREEN</code>	0, 255, 0
<code>public final static Color BLUE</code>	0, 0, 255
<code>public final static Color ORANGE</code>	255, 200, 0
<code>public final static Color PINK</code>	255, 175, 175
<code>public final static Color CYAN</code>	0, 255, 255
<code>public final static Color MAGENTA</code>	255, 0, 255
<code>public final static Color YELLOW</code>	255, 255, 0
<code>public final static Color BLACK</code>	0, 0, 0
<code>public final static Color WHITE</code>	255, 255, 255
<code>public final static Color GRAY</code>	128, 128, 128
<code>public final static Color LIGHT_GRAY</code>	192, 192, 192
<code>public final static Color DARK_GRAY</code>	64, 64, 64

Java 2D – Class Graphics



✓ Clases Auxiliares

✓ Clase **Color**:

Método	Descripción
<i>Constructores y métodos de Color</i>	
<code>public Color(int r, int g, int b)</code>	Crea un color basado en los componentes rojo, verde y azul, expresados como enteros de 0 a 255.
<code>public int getRed()</code>	Devuelve un valor entre 0 y 255, el cual representa el contenido rojo.
<code>public int getGreen()</code>	Devuelve un valor entre 0 y 255, el cual representa el contenido verde.
<code>public int getBlue()</code>	Devuelve un valor entre 0 y 255, el cual representa el contenido azul.
<i>Métodos de Graphics para manipular objetos Color</i>	
<code>public Color getColor()</code>	Devuelve un objeto Color que representa el color actual para el contexto de gráficos.
<code>public void setColor(Color c)</code>	Establece el color actual para dibujar con el contexto de gráficos.

Java 2D – Class Graphics



✓ Clases Auxiliares

✓ **Clase Font:** *Declara los métodos y constantes para manipular el tipo de fuente.*

Método o constante	Descripción
<i>Constantes, constructores y métodos de Font</i>	
<code>public final static int PLAIN</code>	Constante que representa un estilo de tipo de letra simple.
<code>public final static int BOLD</code>	Constante que representa un estilo de tipo de letra en negritas.
<code>public final static int ITALIC</code>	Constante que representa un estilo de tipo de letra en cursivas.
<code>public Font(String nombre, int estilo, int tamaño)</code>	Crea un objeto Font con el nombre de tipo de letra, estilo y tamaño especificados.
<code>public int getStyle()</code>	Devuelve un valor entero que indica el estilo actual de tipo de letra.
<code>public int getSize()</code>	Devuelve un valor entero que indica el tamaño actual del tipo de letra.

El tamaño del tipo de letra se mide en puntos. Un **punto** es 1/72 de una pulgada

Java 2D – Class Graphics



✓ Clases Auxiliares

✓ **Clase Font:** Declara los métodos y constantes para manipular el tipo de fuente.

Método o constante	Descripción
<i>Constantes, constructores y métodos de Font</i>	
<code>public String getFamily()</code>	Devuelve el nombre de la familia del tipo de letra, como una cadena.
<code>public boolean isPlain()</code>	Devuelve true si el tipo de letra es simple; false en caso contrario.
<code>public boolean isBold()</code>	Devuelve true si el tipo de letra está en negritas; false en caso contrario.
<code>public boolean isItalic()</code>	Devuelve true si el tipo de letra está en cursivas; false en caso contrario.
<i>Métodos de Graphics para manipular objetos Font</i>	
<code>public Font getFont()</code>	Devuelve la referencia a un objeto Font que representa el tipo de letra actual.
<code>public void setFont(Font f)</code>	Establece el tipo de letra actual al tipo de letra, estilo y tamaño especificados por la referencia f al objeto Font.

Java proporciona cinco nombres de tipos de letras (Serif, Monospaced, SansSerif, Dialog y DialogInput) que pueden usarse en todas las plataformas de Java

Java 2D – Class Graphics



✓ **Clases Auxiliares**

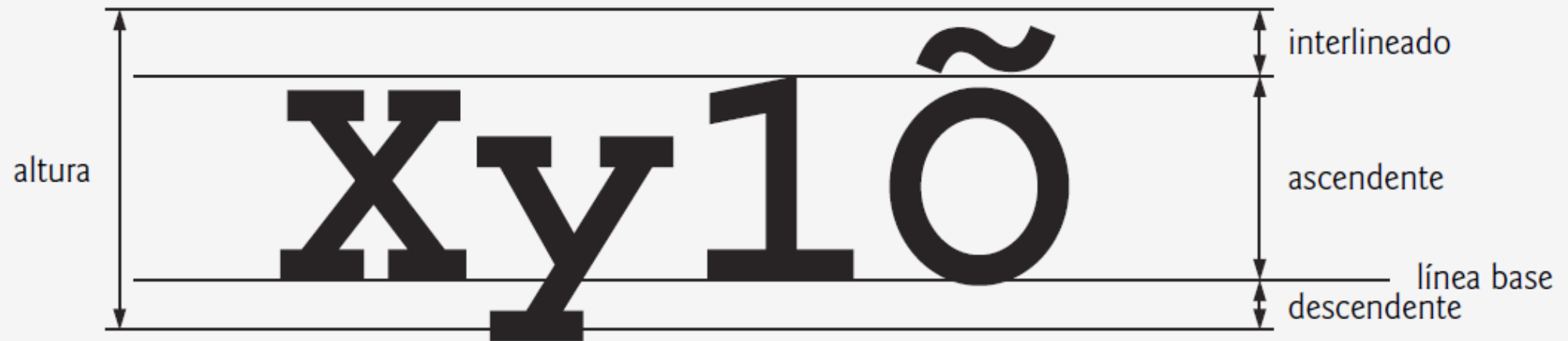
✓ **Clase *Color* y *Font*:**

*Los objetos instancia de Color y Font son **inmutables**.
Estas clases no tienen métodos para modificar las
características del tipo de Color o letra actual.*

Java 2D – Class Graphics



- ✓ *Clases Auxiliares*
- ✓ *Clase **FontMetrics***



Java 2D – Class Graphics



✓ Clases Auxiliares

✓ Clase **FontMetrics**

Método	Descripción
<i>Métodos de FontMetrics</i>	
<code>public int getAscent()</code>	Devuelve un valor que representa el ascendente de un tipo de letra, en puntos.
<code>public int getDescent()</code>	Devuelve un valor que representa el descendente de un tipo de letra, en puntos.
<code>public int getLeading()</code>	Devuelve un valor que representa el interlineado de un tipo de letra, en puntos.
<code>public int getHeight()</code>	Devuelve un valor que representa la altura de un tipo de letra, en puntos.
<i>Métodos de Graphics para obtener la métrica de un tipo de letra</i>	
<code>public FontMetrics getFontMetrics()</code>	Devuelve el objeto <code>FontMetrics</code> para el objeto <code>Font</code> actual para dibujar.
<code>public FontMetrics getFontMetrics(Font f)</code>	Devuelve el objeto <code>FontMetrics</code> para el argumento <code>Font</code> especificado.

Java 2D – Class Graphics / Graphics 2D



✓ Dibujando...

- **Texto:**
 - **drawString**(String s, int x, int y)
- **Línea:**
 - **drawLine**(startX, startY, endX, endY)
- **Rectángulos:**
 - **drawRect**(int x, int y, int ancho, int altura)
 - **fillRect**(int x, int y, int ancho, int altura)
- **Rectángulos con efectos:**
 - **drawRoundRect**(int x, int y, int anchura, int altura, int anchuraArco, int alturaArco)
 - **fillRoundRect**(int x, int y, int anchura, int altura, int anchuraArco, int alturaArco)
 - **draw3DRect**(int x, int y, int anchura, int altura, boolean b)
 - **fill3DRect**(int x, int y, int anchura, int altura, boolean b)
- **Círculos:**
 - **drawOval**(int x, int y, int ancho, int alto)
 - **fillOval**(int x, int y, int ancho, int alto)

Aplicación de Java2D



- ✓ **Caso HangMan**
 - ✓ *Mostrar la clave*
 - ✓ *Mostrar letras erróneas*
 - ✓ *Dibujar el hangman (paso a paso)*
 - ✓ *Debe identificar la letra oprimida por el usuario – Key listener*

***Construya la GUIHagman
aplicando Java 2D***