# Explicación de la Clase EscenarioDistritoSombrio

La clase `EscenarioDistritoSombrio` representa un escenario específico dentro del juego. Extiende `BaseEscenario` y define un entorno con una imagen de fondo y desplazamiento. Se encarga de cargar la imagen del escenario y actualizar su posición en pantalla cuando el personaje se mueve.

## 1. Definición de la Clase

Ubicación: `juegoprog.escenarios.EscenarioDistritoSombrio`

Extiende: `BaseEscenario` (hereda funcionalidad común para los escenarios)

## 2. Explicación Detallada del Código

### Atributos

```java  
private BufferedImage imagenFondo;  
private int offsetX = 0;  
private int offsetY = 0;  
```

✔️ `imagenFondo`: Almacena la imagen del escenario que se dibujará en pantalla.

✔️ `offsetX` y `offsetY`: Representan el desplazamiento del fondo cuando el jugador se mueve.

### Constructor

```java  
public EscenarioDistritoSombrio() {  
 super(3192, 4096);  
 setSize(3192, 4096);  
 setPreferredSize(new Dimension(3192, 4096));  
 setVisible(true);  
 inicializar();  
}  
```

✔️ `super(3192, 4096);` llama al constructor de `BaseEscenario` y define el tamaño del escenario.

✔️ `setSize()` y `setPreferredSize()` aseguran que el panel tenga las dimensiones correctas.

✔️ `setVisible(true);` permite que el escenario se muestre en pantalla.

✔️ `inicializar();` llama a un método que carga la imagen del escenario.

### Método paintComponent()

```java  
@Override  
protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 if (imagenFondo != null) {  
 g.drawImage(imagenFondo, -offsetX, -offsetY, imagenFondo.getWidth(), imagenFondo.getHeight(), null);  
 }  
}  
```

✔️ `paintComponent(Graphics g)` se encarga de dibujar la imagen de fondo en pantalla.

✔️ `g.drawImage(imagenFondo, -offsetX, -offsetY, ...)` mueve el fondo en función del desplazamiento.

### Método inicializar()

```java  
private void inicializar() {  
 cargarImagen();  
}  
```

Este método simplemente llama a `cargarImagen()` para cargar la imagen del escenario.

### Método cargarImagen()

```java  
private void cargarImagen() { ... }  
```

✔️ Busca la imagen del escenario en la carpeta de recursos usando `getResource()`.

✔️ Si la encuentra, la carga en `imagenFondo` usando `ImageIO.read()`.

✔️ Si no se encuentra la imagen, muestra un mensaje de error.

### Método actualizarOffset()

```java  
public void actualizarOffset(int x, int y) {  
 this.offsetX = x;  
 this.offsetY = y;  
 System.out.println("Escenario Offset: X=" + offsetX + ", Y=" + offsetY);  
 repaint();  
}  
```

✔️ `offsetX` y `offsetY` se actualizan para reflejar el nuevo desplazamiento del fondo.

✔️ `repaint();` se llama para actualizar la pantalla y redibujar el escenario.

### Métodos getAncho() y getAlto()

```java  
public int getAncho() { return 3192; }  
public int getAlto() { return 4096; }  
```

Estos métodos devuelven el tamaño del escenario.

## 3. Ejemplo de Uso

En el código principal, podemos instanciar el escenario y actualizar su desplazamiento con:

```java  
EscenarioDistritoSombrio escenario = new EscenarioDistritoSombrio();  
escenario.actualizarOffset(100, 200);  
```

Esto movería el escenario 100 píxeles a la derecha y 200 píxeles hacia abajo.

## 4. Conclusión

La clase `EscenarioDistritoSombrio` define un escenario con desplazamiento y una imagen de fondo. Es un ejemplo de cómo se pueden crear múltiples escenarios en el juego heredando de `BaseEscenario`.